Revista Colombiana de Estadística
ISSN: 0120-1751
revcoles_fcbog@unal.edu.co
Universidad Nacional de Colombia
Colombia

González Rojas, Victor Manuel
Inter-Battery Factor Analysis via PLS: The Missing Data Case
Revista Colombiana de Estadística, vol. 39, núm. 2, julio, 2016, pp. 247-266
Universidad Nacional de Colombia
Bogotá, Colombia

Revista Colombiana de Estadística
July 2016, Volume 39, Issue 2, pp. 247 to 266
DOI: http://dx.doi.org/10.15446/rce.v39n2.52724

Inter-Battery Factor Analysis via PLS: The Missing Data Case

Análisis Factorial Interbaterías vía PLS: el caso de datos faltantes

Victor Manuel González Rojas[a]

Escuela de Estadística, Facultad de Ingeniería, Universidad del Valle, Cali, Colombia

Abstract

In this article we develop the Inter-battery Factor Analysis (IBA) by using PLS (Partial Least Squares) methods. As the PLS methods are algorithms that iterate until convergence, an adequate intervention in some of their stages provides a solution to problems such as missing data. Specifically, we take the iterative stage of the PLS regression and implement the "available data" principle from the NIPALS (Non-linear estimation by Iterative Partial Least Squares) algorithm to allow the algorithmic development of the IBA with missing data. We provide the basic elements to correctly analyse and interpret the results. This new algorithm for IBA, developed under the R programming environment, fundamentally executes iterative convergent sequences of orthogonal projections of vectors coupled with the available data, and works adequately in bases with or without missing data.

To present the basic concepts of the IBA and to cross-reference the results derived from the algorithmic application, we use the complete Linnerud database for the classical analysis; then we contaminate this database with a random sample that represents approximately 7% of the *non-available* (NA) data for the analysis with missing data. We ascertain that the results obtained from the algorithm running with complete data are exactly the same as those obtained from the classic method for IBA, and that the results with missing data are similar. However, this might not always be the case, as it depends on how much the 'original' factorial covariance structure is affected by the absence of information. As such, the interpretation is only valid in relation to the available data.

***Key words*:** Algorithm, Convergence, Missing data, Partial Least Squares Regression.

[a]Professor. E-mail: victor.m.gonzalez@correounivalle.edu.co

Resumen

En este artículo se desarrolla el Análisis Factorial Interbaterías (AIB) mediante el uso de métodos PLS (Partial Least Squares). Ya que los métodos PLS son algoritmos que iteran hasta la convergencia, permiten ser intervenidos adecuadamente en algunas de sus etapas para tratar problemas tales como datos faltantes. Específicamente se toma la fase iterativa de la regresión PLS y se implementa el principio de *"datos disponibles"* del algoritmo NIPALS (Non-linear estimation by Iterative Partial Least Squares) para permitir el desarrollo algorítmico del AIB con datos faltantes, proporcionando los elementos básicos para el análisis e interpretación de los resultados. Este nuevo algoritmo para AIB elaborado bajo el entorno de programación R, fundamentalmente realiza secuencias iterativas convergentes de proyecciones ortogonales de vectores emparejados con los datos disponibles y funciona adecuadamente en bases con y sin datos faltantes.

Para efectos de presentar los conceptos básicos del AIB y cotejar los resultados derivados de la aplicación algorítmica, se toma la base de datos completa de Linnerud para el análisis clásico; y luego esta base es contaminada con una muestra aleatoria que representa aproximadamente el 7% de los datos *no disponibles* (NA) para el análisis con datos faltantes. Se comprueba que con datos completos los resultados derivados del algoritmo son idénticos a los obtenidos mediante el desarrollo del método clásico para AIB, y que los resultados con datos faltantes son similares, aunque esto no siempre será así porque ello dependerá de que tanto se afecta la estructura de covarianza factorial 'original' ante la cantidad de información ausente; por tanto la interpretación será valida solo en relación con los datos disponibles.

***Palabras clave*:** algoritmo, convergencia, datos faltantes, regresión con mínimos cuadrados parciales.

1. Introduction

Among the PLS methods created by Wold (1985), the most important are NIPALS, PLS-Regression (PLS-R) and PLS-Path Modeling (PLS-PM), which were designed for the treatment of one, two and k quantitative data matrices, respectively. PLS-R studies the relationship between two groups of variables X and Y even in the presence of multicollinearity, and has been applied with great success in fields such as Chemometrics, Sensometrics, Genetics, Medical Imaging (Pérez & González 2013), among others.

These PLS methods are convergent algorithms, and, as such, they allow intervention in some of their stages or phases in order to optimally handle missing data problems, mixed data, etc. For this reason, the development of PLS algorithms that replace classical methods like IBA is important (that being the main focus of this article), as it happened with NIPALS (Wold 1966) for the Principal Component Analysis (PCA) or GNM-NIPALS (Aluja & González 2014) for the treatment of a mixed data matrix.

In recent literature (Tenenhaus & Tenenhaus 2011), IBA is considered as a special case of the Regularized Generalized Canonical Correlation Analysis (RGCCA)

for the optimization problem of two continuous data blocks that take advantage of the flexibility of PLS-PM. See the *rgcca()* function from the *RGCCA* package (Tenenhaus & Guillemot 2013).

When studying interrelations between two groups of variables $X_{n,p}$ and $Y_{n,q}$ via IBA (Tucker 1958, Tenenhaus 1998), it is frequent to find missing data. I such a case, it is not possible to apply the classical method without suppressing or estimating the individuals whose data is missing as IBA requires the spectral decomposition of the product between the inter-group covariance matrices $X'YY'X$, (see, for example the *interbat()* function from the *plsdepot* package (Sanchez, G. 2012)).

However, the PLS-R algorithmic regression methods (Tenenhaus 1998) with one (PLS1) or multiple (PLS2) Y variables provide a solution alternative as they are based on the regression concepts. In effect, this can be seen as an orthogonal projection between vectors of *available* data according to the basis of the NIPALS algorithm for missing data, without resorting to data imputation.

PLS2 investigates the t_h and u_h components in each group X and Y and for each stage $h = 1, 2, \ldots, s$[1], maximizing $cov(t_h, u_h)$. These $X_{h-1}a_h$ and $Y_{h-1}b_h$ components are a linear combination of the variables from the respective groups. The ideas behind the PLS2 algorithm are retaken during the convergence phase, as in the limit, and through successive replacements. Then, the stationarity equations associated with the first stage of IBA are verified in order to obtain the first λ_1 eigenvalue associated with the product between the covariance matrices for each h stage.

After obtaining convergence for orthonormal a_h, the $X_{h-1} - t_h p'_h$ matrices of the first group, and the $Y_{h-1}t_h b'_h$ matrices of the second group are deflated, both with respect to t_h, in order to proceed to the next iteration on stage $h + 1$ (see section 2.3.1).

However, these deflated matrices must be modified, taking the form $X_{h-1} - t_h a'_h$ in the first group and $Y_{h-1} - u_h b'_h$ in the second group. In this way, IBA and its properties are obtained via PLS with the previous orthonormalization of b_h (see section 3).

In this article, a PLS algorithm for the IBA method is developed under the R environment, breaking the rigidity of the classical method, and contributing to a solution to the missing data problem. This problem is solved by adequately intervening in certain phases of the algorithm and implementing the *available data* principle, according to the NIPALS method.

In section two, the methodologies inherent to the process are presented. Firstly a recapitulation of IBA is created, and then the NIPALS and PLS2 methods are described, including the pseudo-algorithms, which are useful in the algorithmic solution proposed for IBA.

Chapter 3 ties together the basic concepts of the aforementioned procedures, and proposes the basic structure of the algorithm, which executes classic IBA with complete data and an IBA with missing data (see IBA R code in Appendix).

[1] $s = range(X'Y)$

Section four describes the application: first, the *linnerud* database which is, taken from the *calibrate* package, Graffelman (2013). This database is used for the application of the IBA algorithm, both with the complete data, and the missing data, which is the result of randomly contaminating 7% of the data set (declaring them NA (not available) for the analysis). Subsequently, the results are presented, highlighting the equivalences with the classic IBA (complete data), and puttying an emphasis on the analysis performed an missing data, without forgetting that these results, regardless of their likeness, must be upheld solely from an available data starting point.

Finally, section five is dedicated to the main conclusions and recommendations derived from the study. We particularly highlight future investigations oriented towards IBA with missing data and mixed data that optimally quantify the qualitative variables from a k-dimensional function starting point, according to the GNM-NIPALS method (Aluja & González 2014).

2. Methodologies

2.1. Inter-Battery Factor Analysis

The Inter-battery Analysis (developed by Tucker 1958) starting points are two data sets X, and Y, containing n individuals and p and q variables (columns) respectively, in which the $t_h = Xa_h$ and $u_h = Yb_h$ components are investigated. Their own group is then explained and it is always as correlated as possible. It is imposed on $a_h \in R^p$ and $b_h \in R^q$ to be orthonormal.

The objective is then to *maximize the covariance* or simultaneously to maximize the product between their variances and correlation, which is:

$$\max[cov(Xa_h, Yb_h)] = \max[r(Xa_h, Yb_h)\sqrt{v(Xa_h)}\sqrt{v(Yb_h)}]$$

This method is, in itself, a compromise between the Canonical Analysis (CA) of X and Y that $\max[r(Xa_h, Yb_h)]$ and the Principal Component Analysis (PCA) of X that $\max[v(Xa_h)]$ and Y that $\max[v(Yb_h)]$.

The variables are supposed to be centered and reduced; hence, the covariance, or intra-X correlation matrix is $R_{11} = \frac{1}{n}X'X$, and the intergroup matrix corresponds to $R_{12} = \frac{1}{n}X'Y$; $R_{21} = R'_{12}$. Observe that if A contains every a_h then:

$$\sum_h^p v(t_h) = ||XA||^2 = trace(XAA'X') = trace(X'X) = p$$

equally:

$$\sum_h^q v(u_h) = q$$

2.1.1. Optimal Solution

From the covariance:

$$\gamma_h = cov(Xa_h, Yb_h) = cov(t_h, u_h) = \frac{1}{n} t_h' u_h$$
$$= a_h' R_{12} b_h = cos(a_h, R_{12} b_h) ||R_{12} b_h||$$

We can deduce that we have reached reach an optimal value when the cosine equals 1, i.e., when the vector a_h is collinear with $R_{12}b_h$, and so $\gamma_h = ||R_{12}b_h||$. In the same way, we reach an optimal value when the vector b_h is collinear with $a_h' R_{12}$, and with this taken into account, $\gamma_h = ||R_{21}a_h||$.

Applying the langrangian to $cov(Xa_h, Yb_h)$ under $a_h' a_h = 1$ and $b_h' b_h = 1$ we obtain the following system:

$$L = \frac{1}{n} a' X' Y b - \lambda(a'a - 1) - \mu(b'b - 1)$$

which derivatives $\frac{\delta L}{\delta a} = 0$ and $\frac{\delta L}{\delta b} = 0$ and leads to:

$$\frac{1}{n} X' Y b = 2\lambda a \text{ and } \frac{1}{n} Y' X a = 2\mu b$$

The previous system is relatively different to that found through CA. Premultiplying the two equations by a' and b' respectively we obtain $2\lambda = 2\mu = \gamma$, and, therefore,

$$a = \frac{1}{n\gamma} X' Y b; \;\; b = \frac{1}{n\gamma} Y' X a \tag{1}$$

verifying the previously noted collinearities. The stationarity equations are:

$$\frac{1}{n^2} X' Y Y' X a = \gamma^2 a; \;\; \frac{1}{n^2} Y' X X' Y b = \gamma^2 b \tag{2}$$

That is to say, a_h is a p order *eigenvector* of the symmetric matrix $R_{12}R_{21}$, associated with the largest γ_h^2 *eigenvalue*, guaranteeing maximum covariance. In this way, the a_h form an orthonormal base in R^p. Analogously, b_h is an eigenvector of $R_{21}R_{12}$ associated to the same biggest γ_h^2 eigenvalue and form an orthonormal base in R^q.

2.1.2. Properties of the t_h and u_h Components

- The t_h components of the same group are not orthogonal, because of from (4)

$$X' Y Y' X a_h = \gamma_h^2 a_h; \text{ and } a_l' X' Y Y' X a_h = t_l' Y Y' t_h = \gamma_h^2 a_l' a_h = 0$$

  and with this $t_h' t_l \neq 0$ (and analogously $u_h' u_l \neq 0$) must be taken into account when calculating the explained variances or redundancies.

- The t_h and u_l components of different groups and orders are orthogonal given that:

$$cov(t_h, u_l) = \frac{1}{n} t_h' u_l = \frac{1}{n} a_h' X' Y b_l = a_h' R_{12} b_l = a_h' a_l \gamma_l = 0$$

- The interpretation of the components starting from (3) is:

$$a_h = \frac{1}{n\gamma_h} X' u_h = \frac{1}{n r_h \sigma_{t_h}} \frac{X' u_h}{\sigma_{u_h}} = \frac{1}{r_h \sigma_{t_h}} \{r(x_j, u_h) ... \forall j)\} \tag{3}$$

with this, a_h is collinear to the correlations vector between X_j and u_h. In a similar fashion

$$b_h = \frac{1}{n\gamma_h} Y' t_h, \tag{4}$$

b_h is collinear to the correlations vector between Y_k and t_h.

There is coherence between the variable coefficients and the correlations between variables of one group and the components of the other group.

2.1.3. Decomposing the Correlation Matrix R_{12}

The PCA, like the reconstitution of R_{12} in (4), is given by $R_{12} = \sum_h^s \gamma_h a_h b_h'$ and, through (5) and (6), leads to:

$$r(x_j, y_k) = \sum_h^s \frac{1}{r_h} r(x_j, u_h) r(y_k, t_h) \tag{5}$$

The inter-group correlation matrix can be visualized using the correlations between the group variables and the other group's components.

In addition, the best approximation is obtained in the direction of the least squares of R_{12} through its simile with dimension p, q and range m:

$$R_{12_m} = \sum_h^m \gamma_h a_h b_h'; \text{ and, as } ||R_{12}||^2 = ||R_{12_m}||^2 + ||R_{12} - R_{12_m}||^2. \tag{6}$$

We can measure the quality of the approximation, defining the number of components to be retained. In addition, these norms are calculated in terms of the *eigenvalues*:

$$||R_{12}||^2 = trace(R_{12} R_{21}) = \sum_h^s \gamma_h^2; \quad ||R_{12_m}||^2 = \sum_{h=1}^m \gamma_h^2 \tag{7}$$

$$||R_{12} - R_{12_m}||^2 = \sum_{h=m+1}^s \gamma_h^2 \tag{8}$$

2.1.4. Relation Between the Two Variable Sets: Factorial Structure

In this section we describe the principal elements to be retained for the results analysis. Firstly, the γ_h^2 values and the eigenvectors a_h, b_h are associated with the matrix $R_{12}R_{21}$, and therefore, with the t_h and u_h components. This verifies that the sum of variances across all of h is p and q, respectively.

As we have the correlations between the components of both groups $r(t_h, u_h)$ and the factorial structure, we can reconstitute R_{12} starting from m components according to (7). The factorial structure refers to the correlations $r(x_j, t_h)$, $r(x_j, u_h)$, $r(y_k, u_h)$, and $r(y_k, t_h)$.

We are going to obtain the explained variance parts and the commonalities, and due to the correlation between the components, this calculation must be made with the help of regression.

- **Intra-group Communality**

  We can measure the *variance* part of each variable *explained* in its m canonical components to be retained, and these indexes are called *commonalities*, as in factorial analysis. The intra-X communality with m components is defined as:

  $$R^2(x_j, t_1, \ldots, t_m)$$

  We calculate the variance of X_j explained in $t_1; (t_1, t_2); \ldots; (t_1, t_2, \ldots, t_m)$. As in PCA, we have the reconstitution[2] $X = \sum_h^m t_h a_h'$, and so the variable $X_j = t_1 a_{1j} + \cdots + t_m a_{mj}$. As such, when performing the regression with m components we obtain:

  $$X_j = \hat{X}_j + e = \hat{\beta}_1 t_1 + \cdots + \hat{\beta}_m t_m + e$$

  When $m = 1$, this corresponds to a simple regression, in which $\hat{\beta}_1 = X_j' t_1$; with $m = s$, the estimation is exactly the same as that of the PCA because the coefficients $a_{1j} = \hat{\beta}_1, ., a_{sj} = \hat{\beta}_s$ match. In any regression, the determination coefficient $R^2 = \frac{v(\hat{X}_j)}{v(X_j)}$ is obtained, and it measures the variability percentage of X_j which is explained by the $\hat{X}_j$ regression. However, as $v(X_j) = 1$ then

  $$v(\hat{X}_j) = \hat{\beta}_1^2 v(t_1) + \cdots + \hat{\beta}_m^2 v(t_m) + 2 \sum_{i,k>i}^{m} \hat{\beta}_i \hat{\beta}_k cov(t_i, t_k) = R^2(X_j, t_1, \ldots, t_m)$$

  represents the intra-group communality of X_j in the m components. As such, we need to execute as many progressive regressions as the number of components we have, that is to say with $t_1; (t_1, t_2); \ldots; (t_1, t_2, \ldots, t_m)$. For $m = s$, $R^2 = 1$.

[2] $X a_\alpha a_\alpha' = t_\alpha a_\alpha' \Rightarrow X \sum_\alpha a_\alpha a_\alpha' = X = \sum_\alpha t_\alpha a_\alpha'$

Analogously, from $v(u) = 1$

$$R^2(y_k, u_1, \ldots, u_m) = \sum_h^m r^2(y_k, u_h) = \sum_h^m \frac{1}{r_h^2} r^2(y_k, t_h) \tag{9}$$

As before, these coefficients are obtained from as many progressive regressions as u_m components we have.

The variable with weak intra-group communality, does not participate much in the study as they are not particularly related with the active variables of the other group.

- **Inter-Group Communality**

  It's defined as the cross variance, that is to say, the variance of each variable explained in the m components of the other group:

$$R^2(x_j, u_1, \ldots, u_m) = \sum_h^m r^2(x_j, u_h)$$

$$R^2(y_k, t_1, \ldots, t_m) = \sum_h^m r^2(y_k, t_h)$$

  The variables with little inter-group communality are specific from their own group, they are not very related to the other group; these variables can be suppressed without perturbing the analysis.

2.2. NIPALS Algorithm

This algorithm is the base of the PLS regression (Wold 1966). It fundamentally executes the singular decomposition of a data matrix through the use of convergent iterative sequences for orthogonal projections (geometric concept of simple regression). With complete databases, the results are equivalent to those found using PCA; however, and this is probably its greatest virtue, it can execute PCA even with missing data and obtain its estimations starting from the reconstituted data matrix.

If $X_{n,p}$ is the data matrix of range $a \leq p$, columns $X_1, \ldots, X_p$ are supposed to be centered or standardized (under S_n). The reconstitution derived from the PCA leads to $X = \sum_h^a t_h p_h'$ where t is the *principal component (scores)*and p_h' the *eigenvector (loadings)* on the h axis.

$$[X_1, \ldots, X_p] = t_1 p_1' + \cdots + t_a p_a'$$

In this way, column $X_j = \sum_h^a p_{hj} t_h$ with $j = 1, \ldots, p$ and the *ith* row $x_i = \sum_h^a t_{hi} p_h$ with $i = 1, \ldots, n$.

It can be observed then that if $h = 1$, column j is expressed as $X_j = p_{1j}t_1$, that is $p_{hj} = X_j't_h$ acts like the coefficient (slope)[3] in the regression (without intercept) of X_j over t_h. In the space of rows, t_{hi} is the constant-less regression coefficient of the individual x_i over p_h.

If $h > 1$, p_{hj} is the regression coefficient of t_h in the simple regression of the deflated vector $X_j - \sum_l^{h-1} p_{lj}t_l$ over t_h, and t_{hi} is the coefficient of p_h in the regression of $x_i - \sum_l^{h-1} t_{li}p_l$ over p_h.

2.2.1. NIPALS Pseudo-Code

For $h = 1$, the algorithm starts by taking any column of the matrix X as the first *principal component* t_1, in order to immediately calculate a normalized p_1 and then recalculate t_1 in an iterative process until p_1 converges. The flow diagram associated with the convergence procedure is:

$$\begin{array}{ccccc} X = X_1 \to t_1 & \to & p_1^+ = X_1't_1/t_1't_1 & \to & p_1 = \dfrac{p_1^+}{||p_1^+||} \\ & \uparrow & & & \downarrow \\ & \leftarrow & t_1 = X_1p_1 & \leftarrow & \end{array}$$

After that, on each stage $h = 2, \ldots, a$, the deflated matrix $X_h = X_{h-1} - t_hp_h'$ will be built, and from it we will take t_h orthogonal to t_{h-1} in order to start the convergence process of p_h orthonormal to p_{h-1}, according to the previous flow diagram t_1, p_1 and X_1 will be replaced with t_h, p_h an X_h, respectively.

NIPALS' main characteristic is that it works in terms of a series of scalar products of the coupled elements. This allows the management of missing data, adding the available pairs in each operation. Geometrically the procedure 'takes' the omitted elements as if they fell over the regression line: they are not leverage points.

The NIPALS pseudo-algorithm associated with missing data provides the basic elements to develop the IBA with missing data in the sense of only executing the scalar products with the coupled available data. This is described in stage 2.2.1

- **NIPALS pseudo-code with missing data**

  $X_0 = X_h$ ▷ Stage 1
  for $h = 1, 2, \ldots, a$ **do** ▷ Stage 2
    $t_h =$ first column of X_{h-1} ▷ Stage 2.1
    repeat ▷ Stage 2.2
      for $j = 1, 2, \ldots, p$ **do** ▷ Stage2.2.1

$$p_{hj} = \frac{\sum_{\{i:x_{ji} \ \ and \ \ t_{hi} \ \ exist\}} x_{h-1,ji}t_{hi}}{\sum_{\{i:x_{ji} \ \ and \ \ t_{hi} exist\}} t_{hi}^2}$$

[3]From the simple regression $\hat{\beta}_1 = \hat{p}_{hj} = \frac{cov(t_h, X_j)}{S_{t_h}^2} = \frac{X_j't_h}{||t||^2} = X_j't_h = r$ if x and t are standardized.

end for
Normalize p_h to 1 ▷ Stage 2.2.2
for $i = 1, 2, \ldots, n$ **do** ▷ Stage 2.2.3

$$t_{hi} = \frac{\sum_{\{j:x_{ji} \ \ exists\}} x_{h-1,ji} p_{hj}}{\sum_{\{j:x \ \ exists\}} p_{hj}^2}$$

end for
until p_h converges
$X_h = X_{h-1} - t_h p_h'$ ▷ Stage 2.3
end for

In stages 2.2.1 and 2.2.3 we calculate the slopes of the least square lines that pass through the origin of the cloud of points over the *available* data. The p_{hj} and t_{hi} must preserve j and i in their positions as well as the missing data characteristic given by x_{ij}, which can be expressed as NA *(Not available)*. This allows an excellent management through R functions such as *na.omit()* at the moment of developing the corresponding script.

2.3. Multivariate Regression PLS2

We use the most important presentations of this algorithm in the books by Wold, Martens & Wold (1983), Martens & Nars (1989), Esbensen, Schönkopf & Midtgaard (1994), and the article by Vega & Guzmán (2011) as a starting point.

If Y is the matrix of dependent variables $y_1, \ldots, y_r$ and X is the matrix of independent variables $x_1, \ldots, x_p$ with rank a over n individuals, and all the variables are centered and reduced, then there is the possibility of multicollinearity in the interior of each block. Even of r and p are greater than n, there is also the possibility that there is some missing data.

For now, we have two sets of variables Y and X, for which we assume that a latent relation between the two blocks exists. This can be explained by $H \leq a$ latent orthogonal components t_h $(h = 1, 2, \ldots, H)$, which are obtained as a linear combination of the variables of the predictor set X. They are highly related with Y through their linear combination $u_h = Y c_h$

As such, the predictor and answer matrices are decomposed as follows:

$$X = T_H P_H' + X_H$$
$$Y = T_H C_H' + Y_H$$

where P_H and C_H are the weight matrices containing the parameters for the model, and X_H and Y_H the residual matrices representing the variability of the data unexplained by the parameter models.

2.3.1. PLS2 Pseudo-Algorithm

There are numerous versions of the PLS2 algorithm that differ in the level of normalization chosen. Here we describe the classical PLS2 regression algorithm,

taking into account the missing data management in accordance with the principles extracted from the NIPALS (Lindgren, Geladi & Wold 1993).

$X_0 = X$, $Y_0 = Y$
for $h = 1, 2, \ldots, a$ **do**
1. Initialize: u_h (u_1 : first col of $Y_{h-1}, ...$)
2.
repeat
$w_h = X'_{h-1}u_h/||u_h||$
$w_h = w_h/||w_h||$
$t_h = X_{h-1}w_h/(w'_h w_h)$
$c_h = Y'_{h-1}t_h/t'_h t_h$
$u_h = Y_{h-1}c_h/(c'_h c_h)$
until w_h converges
3. $p_h = X'_{h-1}t_h/(t'_h t_h)$
4. $X_h = X_{h-1} - t_h p'_h$
5. $Y_h = Y_{h-1} - t_h c'_h$
end for

When there is missing data, we apply the principles from the NIPALS algorithm: the coordinates of the vectors w_h, t_h, c_h, u_h and p_h are calculated as the slope of the least squares' lines passing through the origin (only over the available data).

2.3.2. Optimization Criteria

We can pin down the convergence on stage 2. The cyclical relationships of this stage show that, on the limit, the vectors w_h, t_h, c_h and u_h, through successive replacements, verify the following equations:

$$\left(\frac{1}{n-1}X'_{h-1}Y_{h-1}\right)\left(\frac{1}{n-1}Y'_{h-1}X_{h-1}\right)w_h = \lambda_h w_h$$

$$\left(\frac{1}{n-1}X_{h-1}X'_{h-1}\right)\left(\frac{1}{n-1}Y_{h-1}Y'_{h-1}\right)t_h = \lambda_h t_h$$

$$\left(\frac{1}{n-1}Y'_{h-1}X_{h-1}\right)\left(\frac{1}{n-1}X'_{h-1}Y_{h-1}\right)c_h = \lambda_h c_h$$

$$\left(\frac{1}{n-1}Y_{h-1}Y'_{h-1}\right)\left(\frac{1}{n-1}X_{h-1}X'_{h-1}\right)u_h = \lambda_h u_h$$

λ_h is the greatest common eigenvalue between these matrices, which have been divided by *n-1* to reclaim the eigenvalues. Therefore, Stage two corresponds to an application of the iterative power in order to calculate the eigenvector of a matrix, associated to the largest eigenvalue for each *h*.

We can obtain the t_h and u_h components starting which is the first stage of Tucker's IBA from the tables X_{h-1} and Y_{h-1}. On each stage h we investigate two normalized vectors w_h and c_h^*, maximizing the criteria $cov(X_{h-1}w_h, Y_{h-1}c_h^*)$, or globally maximizing the criteria:

$$\sum_{h=1}^{s} cov^2(X_{h-1}w_h, Y_{h-1}c_h^*)$$

the vector c_h is collinear with the vector $c_h^* = c_h/||c_h||$ and $s \leq a$.

We will now build the deflated matrices X_h and Y_h in stages 4 and 5 as residues of the regressions of X_{h-1} and Y_{h-1} over the t_h component. Observe that there deflations must be modified to obtain the orthonormality properties on both the a_h and the b_h, according to the IBA.

3. The IBA Algorithm Via PLS

This algorithm describes the relations between two data sets X and Y by maximizing the covariance between the latent components t_h and u_h of each set, respectively. Basically, we perform a spectral decomposition of $X'YY'X$ and $Y'XX'Y$ in order to obtain the respective $h = 1, \ldots, H$ components; $(H = s)$.

The algorithm is built over the structure of the PLS2 procedure, changing the calculation of the vectors w_h, t_h, c_h, and u_h for a_h, t_h, b_h and u_h respectively, with or without missing data (see *ej* cycle in section 3.1). The convergence of these vectors on each stage h is quickly secured, usually in no more than 20 iterations; nonetheless the *ej* cycle executes 100 iterations in order to leave some convenient room. We can set the threshold $\varepsilon = 0.0001$ so that if $||a_{h,j} - a_{h,j+1}|| < \varepsilon$ we can guarantee convergence of a_h in the jth iteration in order to continue with the next stage h.

Once the a_h, t_h, b_h and u_h vectors have converged, the initial matrices are deflated through the procedure $X_0 - t_h a_h'$ and $Y_0 - u_h b_h'$ in order to guarantee the orthonormality of the vectors a_h and b_h in the next stage h: this is the principal restriction of the IBA.

Observe in Appendix (IBA R Code) that in order to calculate these vectors we use the *na.omit()* function, which uses the coupled available data of the two vectors X_j and u_h. The scalar product between these two vectors allows us to obtain, for example, a_h.

The algorithm inherits the properties described in 2.1.2 for the Classic IBA. With missing data, the said properties are guaranteed through the *orthonormalization()* function of the *far* library (see Appendix). The same process is analogously applied in the calculation to obtain b_h.

Finally, through *list(aH,tH,bH,uH,lH,rH)* the algorithmic function named *fAIBna* returns these vectors along with the eigenvalues lH and the correlations rH between the components. The pseudo-code for the IBA with or without missing data is presented in section 3.1, and the R code is presented in Appendix.

3.1. IBA: Pseudo-Code Algorithm

The algorithm is based on the principles proposed by the NIPALS algorithm with missing data, as well as an the structure proposed by the PLS2 algorithm. The order exception is the deflation stages, which have been adequately modified in order to maintain the IBA properties.

$X_0 = X$, $Y_0 = Y$
for $h = 1, 2, \ldots, s$ **do**
  $u_h = 1^{st}$ col of Y_{h-1}
  repeat
    for $j = 1, \ldots, p$ **do**

$$a_{hj} = \frac{\Sigma_{\{i:x_{ij} \ \ and \ \ u_{hi} \ \ exist\}} x_{h-1,ij} u_{hi}}{\Sigma_{\{i:x_{ij} \ \ and \ \ u_{hi} \ \ exist\}} u_{hi}^2}$$

    end for
    orthonormalization$(a_1, \ldots, a_h)$
    for $i = 1, \ldots, n$ **do**

$$t_{hi} = \frac{\Sigma_{\{j:x_{ij} \ \ exists\}} x_{h-1,ij} a_{hj}}{\Sigma_{\{j:x_{ij} \ \ exists\}} a_{hj}^2}$$

    end for
    for $k = 1, \ldots, q$ **do**

$$b_{hk} = \frac{\Sigma_{\{i:y_{ik} \ \ and \ \ t_{hi} exist\}} y_{h-1,ik} t_{hi}}{\Sigma_{\{i:y_{ik} \ \ and \ \ t_{hi} exist\}} t_{hi}^2}$$

    end for
    orthonormalization$(b_1, \ldots, b_h)$
    for $i = 1, \ldots, n$ **do**

$$u_{hi} = \frac{\Sigma_{\{k:y_{ik} \ \ exists\}} y_{h-1,ik} b_{hk}}{\Sigma_{\{k:y_{ik} \ \ exists\}} b_{hk}^2}$$

    end for
  until a_h converges
  $X_h = X_{h-1} - t_h a_h'$
  $Y_h = Y_{h-1} - u_h b_h'$
end for

4. Application

The IBA via PLS algorithm (IBApls) is implemented and run using the *linnerud* and *linnerudNA* databases in order to study the relation between two matrices with or without missing data.

4.1. Linnerud Database

The *linnerud* database can be obtained from R's *calibrate* package. It contains the physical and exercise variables of 20 users of a gymnastics club. The database is conformed by two groups, i.e., the first matrix X contains the physical variables weight, height, and pulse (Poids, Tail, Pouls) that will be related to the exercise variables contained in the matrix Y: Traction, flection, jump (Tracti, Flexin and Sauts). Both matrices have a 20x3 dimension.

Table 1 represents the incomplete database (*linnerudNA*), approximately 7% of the data has been declared Not Available (NA).

Table 1: LinnerudNA database.

#	Poids	Tail	Pouls	Tracti	Flexin	Sauts
1	191	36	50	5	162	60
2	189	NA	52	2	110	60
3	193	38	58	12	NA	101
4	162	35	62	12	105	37
5	189	35	46	13	NA	58
6	182	36	NA	4	101	42
7	211	38	56	8	101	38
8	167	34	60	6	125	40
9	176	31	74	15	200	40
10	154	33	56	17	251	250
11	169	34	50	17	120	38
12	166	33	52	13	210	115
13	154	34	64	14	215	105
14	247	46	50	1	50	50
15	193	36	46	6	70	31
16	NA	37	62	12	NA	120
17	NA	37	54	4	60	NA
18	157	32	52	11	230	80
19	156	33	54	15	225	73
20	138	33	68	2	110	43

4.2. Results

The application of this algorithm (see Appendix) through the *fAIBna(Y,X)* function to the complete *linnerud* database, formed by the X and Y subgroups, leads to the same results as those obtained by applying the classical IBA method (Tenenhaus 1998). The results are the following:

The eigenvalues 1.27243, 0.00566 and 0.00111 correspond to the squared covariances γ_h^2 on stages $h = 1, 2, 3$. Tables 2 and 3 show the eigenvectors and the components associated with the classical IBA (complete data).

For the missing data case (NA) we use the database *linnerudNA* that is listed in section 4.1; the same as before, the first three columns make up X and the last three Y. By Applying the *fAIBna(Y,X)* function over this matrices, we get the following results:

TABLE 2: a_h and b_h eigenvectors, classical IBA.

	a1	a2	a3	b1	b2	b3
[1,]	−0.590	0.772	0.236	0.613	0.214	0.7603
[2,]	−0.771	−0.452	−0.448	0.747	0.156	−0.6464
[3,]	0.239	0.447	−0.862	0.257	−0.964	0.0644

TABLE 3: t_h and u_h components, classical IBA.

	t1	t2	t3	u1	u2	u3
[1,]	−0.643	−0.0747	0.76432	−0.3714	0.0544	−0.8229
[2,]	−0.770	−0.1546	0.36612	−1.3403	−0.1964	−0.7172
[3,]	−0.907	0.2008	−0.45300	−0.0823	−0.5849	0.8656
[4,]	0.688	−0.0973	−0.80858	−0.3550	0.6286	0.7438
[5,]	−0.487	−0.2437	1.36340	0.4631	0.3986	0.3975
[6,]	−0.229	0.0154	−0.03941	−1.3058	0.2007	−0.3591
[7,]	−1.404	0.6398	−0.04148	−0.8618	0.4379	0.2111
[8,]	0.744	0.0765	−0.38167	−0.7973	0.3790	−0.3220
[9,]	1.715	1.6485	−1.55022	1.1423	0.9300	0.1976
[10,]	1.163	−0.4365	0.11210	3.0344	−2.8115	0.2222
[11,]	0.365	−0.4802	0.83341	0.4092	0.8496	1.3092
[12,]	0.743	−0.3090	0.70536	1.4051	−0.5366	−0.0991
[13,]	1.187	−0.0824	−0.98450	1.5307	−0.2956	−0.0195
[14,]	−4.390	0.2642	−0.09814	−2.2227	−0.1981	−0.2537
[15,]	−0.823	−0.2599	1.26183	−1.4990	0.4115	0.2349
[16,]	−0.749	0.8711	−0.70534	1.3141	−0.6711	−0.2366
[17,]	−0.393	−0.4373	0.00248	−1.8804	0.4184	0.0431
[18,]	1.199	−0.4492	0.75905	1.2366	0.0904	−0.6373
[19,]	1.049	−0.4978	0.37044	1.6060	0.3716	−0.0192
[20,]	1.942	−0.1938	−1.47619	−1.4254	0.1233	−0.7385

The eigenvalues 1.17246, 0.00962 and 0.00138 are relatively similar to those obtained with classical IBA, and, in Tables 4 and 5, which contain the eigenvectors and the components associated with the missing data IBA, we can also see a similarity with the classical IBA results.

TABLE 4: a_h° and b_h° eigenvectors, missing data IBA.

	a1°	a2°	a3°	b1°	b2°	b3°
[1,]	−0.670	0.733	0.122	0.615	0.3408	0.711
[2,]	−0.707	−0.579	−0.405	0.745	0.0464	−0.666
[3,]	0.226	0.357	−0.906	0.260	−0.9390	0.225

It can be seen Table 6 that the correlations between components of different groups and dimensions are practically 0, despite the absence of data. These results are relatively similar to those obtained with the complete *linnerud* database; however, these factorial similarities will not always appear as they depend on how much the 'Original' matrix is affected due to the absence of some data and how this absence influences the correlation structure. The results must be interpreted as a function of the *available data*.

Note that the correlations of the t_h and t_h° components and the u_h and u_h° components are generally high, with or without missing data, given that:

Table 5: t_h° and u_h° components, missing data IBA.

	t1°	t2°	t3°	u1°	u2°	u3°
[1,]	−0.691	−0.0252	0.72679	−0.374	−0.0428	−0.8396
[2,]	−0.860	−0.3273	0.28022	−1.317	−0.2732	−0.7124
[3,]	−0.933	0.0634	−0.49023	0.986	−0.3182	−0.0221
[4,]	0.655	−0.1047	−0.75573	−0.326	0.7870	0.5834
[5,]	−0.544	−0.0984	1.33052	0.760	0.5238	0.0364
[6,]	−0.283	−0.0148	−0.00168	−1.277	0.1773	−0.4301
[7,]	−1.468	0.4854	−0.15959	−0.832	0.5081	0.0905
[8,]	0.679	0.1202	−0.36322	−0.781	0.3597	−0.4148
[9,]	1.519	1.5863	−1.66097	1.122	0.9933	0.0304
[10,]	1.115	−0.2697	0.18797	2.996	−2.6680	0.6965
[11,]	0.321	−0.3042	0.86964	0.431	1.1018	1.1073
[12,]	0.677	−0.1169	0.73461	1.382	−0.4951	−0.0130
[13,]	1.143	−0.0613	−0.91464	1.505	−0.2449	0.0268
[14,]	−4.331	−0.1854	−0.24537	−2.168	−0.1982	−0.2782
[15,]	−0.866	−0.1605	1.22567	−1.454	0.4846	0.1073
[16,]	−0.335	0.3233	−0.88698	1.201	−0.6577	−0.0457
[17,]	−0.778	−0.0401	0.11008	−1.706	0.0458	0.0136
[18,]	1.131	−0.1988	0.81559	1.201	0.0279	−0.6392
[19,]	1.001	−0.3085	0.44218	1.574	0.4098	−0.0806
[20,]	1.903	−0.1520	−1.35642	−1.402	0.0365	−0.7867

Table 6: Correlation between t_h° and u_h° components.

	t_1°	t_2°	t_3°	u_1°	u_2°	u_3°
t_1°	1	0.13478	−0.230058	0.5506	−0.015547	0.09351
t_2°		1	−0.568142	0.0998	0.290962	0.00483
t_3°			1	0.0155	−0.000473	0.08878
u_1°				1	−0.409760	0.39060
u_2°					1	0.00403
u_3°						1

`r(t1,t1°)=0.995; r(t2,t2°)=0.913 and r(t3,t3°)=0.995`

`r(u1,u1°)=0.985; r(u2,u2°)=0.985 and r(u3,u3°)=0.891`

Figure 1 displays the typology of the subject cloud, starting from the relations between the t_1 and u_1 components. Regarding Figure 2, subject 14 exhibits a poor performance in the exercises as a result of its low potential; meanwhile, subject 10 exhibits the best results from the whole group. Subject 20 has great potential, but lacks training: this is evidenced by its mediocre results. Subjects 9, 12, 13, 18 and 19, on the other, hand have good results that pertain to their potential while the rest of the subjects experience a medium level of potential and development.

The correlation chart of Figure 2 is constructed starting with an estimation of X through the PCA. This can be see in the following.

$$t_\alpha = Xa_\alpha \Rightarrow t_\alpha a'_\alpha = Xa_\alpha a'_\alpha \Rightarrow \hat{X} = \sum_\alpha t_\alpha a'_\alpha$$

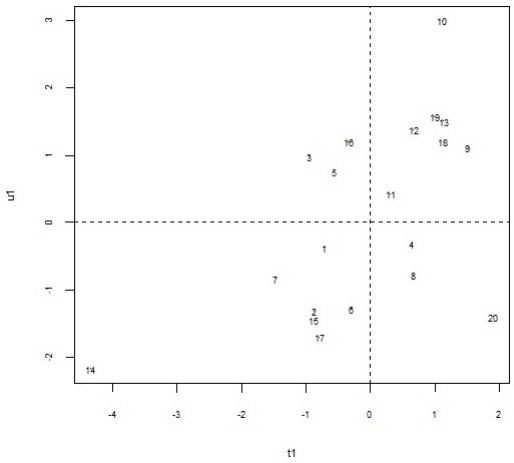

FIGURE 1: IBA with missing data, t_1 vs u_1 graph.

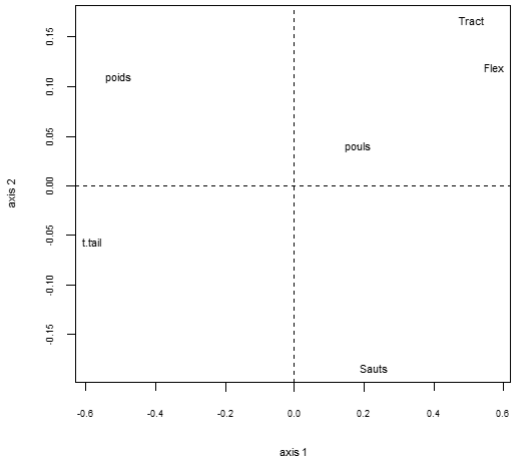

FIGURE 2: Correlations chart; variables vs components of the other group.

We then proceed to calculate the correlations between each of the physical measures $\hat{x}_j$ and the first two components u_1, u_2. Analogously, $\hat{Y} = \sum_\alpha u_\alpha b'_\alpha$, therefore we can calculate the correlations between the exercise variables $\hat{y}_k$ with the components t_1 and t_2. These correlations constitute the coordinates for axes 1 and 2.

Figure 2 portrays the inter-group correlation matrix R_{12}. Axis 1 corresponds to the physical potential fundamentally expressed through the weight and height (poids, tail) of the subjects, attenuated by the pulse (pouls); axis 2, on the other hand, grades the global performance on the exercises, opposing the pushups and pullups (flex, tract) with the jumps (sauts).

5. Conclusions and Recommendations

- The IBA via the PLS (IBApls) method was developed, preserving all of its properties and optimization characteristics, and providing an algorithmic procedure under R that leaves aside the rigidity of the classical method.
- The IBApls was run with databases that had missing data, proving its functionality. The analysis was done under the available data principle as in NIPALS, without data imputation.
- The linnerud database was used to apply the IBApls with or without missing data. With the complete data set, the results are equivalent to those found using the classical IBA, and with approximately 7% of the data missing, the results are relatively similar. However, the analysis must be made as a function of the available data.
- Starting with the flexibility of the IBApls, its possible to solve the mixed data (quanti-qualitative variables) problem through the optimal GNM-NIPALS quantification criteria.
- With these solutions, it is possible to find an optimal, joint solution for IBA with mixed and missing data.

[Received: August 2015 — Accepted: March 2016]

References

Aluja, T. & González, V. M. (2014), 'GNM-NIPALS: General Nonmetric - Nonlinear Estimation by Iterative Partial Least Squares', *Revista de Matemática: Teoría y Aplicaciones* **21**(1), 85–106.

Esbensen, K., Schönkopf, S. & Midtgaard, T. (1994), *Multivariate Analysis in Practice*, Olav Tryggvasons, Trondheim, Norway.

Graffelman, J. (2013), *calibrate*.
*https://cran.r-roject.org/web/packages/calibrate/calibrate.pdf

Lindgren, F., Geladi, P. & Wold, S. (1993), 'The kernel algorithm for PLS', *Journal of Chemometrics* **7**, 45–59.

Martens, H. & Nars, T. (1989), *Multivariate calibration*, John Wiley & Sons, New York.

Pérez, R. A. & González, G. (2013), 'Partial Least Squares Regression on Symmetric Positive Definite Matrices', *Revista Colombiana de Estadística* **36**, 177–192.

Sanchez, G. (2012), *plsdepot*.
*https://cran.r-project.org/web/packages/plsdepot/plsdepot.pdf

Tenenhaus, A. & Guillemot, V. (2013), *RGCCA and sparse GCCA for multi-block data analysis*.
*https://cran.r-roject.org/web/packages/RGCCA/index.html

Tenenhaus, A. & Tenenhaus, M. (2011), 'Regularized Generalized Canonical Correlation Analysis', *Psychometrika* **76**, 257–284.

Tenenhaus, M. (1998), *La régression PLS théorie et pratique*, Editions Technip, Paris.

Tucker, L. R. (1958), 'An inter-battery method of factor analysis', *Psychometrika* **23**(2), 111–136.

Vega, J. & Guzmán, J. (2011), 'Regresión PLS y PCA como solución al problema de multicolinealidad en Regresión Múltiple', *Revista de Matematica: Teoría y Aplicaciones* **18**(1), 9–20.

Wold, H. (1966), Estimation of principal component and related models by iterative least squares, *in* P. R. Krishnaiah, ed., 'Multivariate Analysis', Academic Press, New York.

Wold, H. (1985), 'Partial Least Squares', *Encyclopedia of Statistical Sciences* **6**, 581–591.

Wold, S., Martens, H. & Wold, H. (1983), The multivariate calibration problem in chemistry solved by the pls methods, *in* A. Ruhe & B. Kagstrom, eds, 'Lectures Notes in Mathematics', Proceedings of the Conference on Matrix Pencils, Springer, Heidelberg, New York.

Appendix. IBA R Code

```
fAIBna <- function(Y,X)
{
    library(far)
    Z <- as.matrix(cbind(X,Y))    #
    Yo <- scale(Y) ; Xo <- scale(X)    # omits NA when it scales
    p <- ncol(Xo); n <- nrow(Xo); q <- ncol(Yo)
    H <- qr(t(X)%*%Y)$rank  # H=s
    aH <- matrix(0,p,H); tH <- matrix(0,n,H)
    bH <- matrix(0,q,H); uH <- matrix(0,n,H)
    for(h in 1:H)       # H componentes t e u.
    {
        uh <- Yo[,1]   # numeric
        for(ej in 1:100)
        {
            for(j in 1: p)
            {
                aju <- na.omit(cbind(Xo[,j],uh))
```

```
            aH[j,h] <- sum(aju[,1]*aju[,2])/sum(aju[,2]^2)
        }
        if(any(!is.finite(Z))){
            ah. <- orthonormalization(aH[,1:h])
            ah <- ah.[,h]
        } else ah <- aH[,h]/sqrt(sum(aH[,h]^2))   # numeric
        for(i in 1:n)
        {
            tia <- na.omit(cbind(Xo[i,],ah))  # na.omit f(cols)
            tH[i,h] <- sum(tia[,1]*tia[,2])/sum(tia[,2]^2)
        }
        th <- tH[,h]
        for(k in 1:q)
        {
            bkt <- na.omit(cbind(Yo[,k],th))
            bH[k,h] <- sum(bkt[,1]*bkt[,2])/sum(bkt[,2]^2)
        }
        if(any(!is.finite(Z))){
            bh. <- orthonormalization(bH[,1:h])
            bh <- bh.[,h]
        } else bh <- bH[,h]/sqrt(sum(bH[,h]^2))

        for(i in 1:n)
        {
            uib <- na.omit(cbind(Yo[i,],bh))
            uH[i,h] <- sum(uib[,1]*uib[,2])/sum(uib[,2]^2)
        }
        uh <- uH[,h]
      }  # end ej
      X1 <- Xo - th%*%t(ah); Xo <- X1
      Y1 <- Yo - uh%*%t(bh); Yo <- Y1
      aH[,h]<-ah; tH[,h]<- th; bH[,h]<- bh; uH[,h]<-uh
  }   # end h
  Lh <- diag(t(tH)%*%uH); lH <- Lh^2/(n-1)^2         # val.p
  rH <- cor(cbind(tH,uH))
  r.AIBna <- list(aH,tH,bH,uH,lH,rH)
  return(r.AIBna)
} # end fAIBna with or without missing data
```