Camacho Paez, J.; Ferrer Riquelme, AJ. (2012). Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: theoretical aspects. Journal of Chemometrics. 26(1):361-373. doi:10.1002/cem.2440.

Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Theoretical aspects.

April 16, 2012

Abstract

Cross-validation has become one of the principal methods to adjust the meta-parameters in predictive models. Extensions of the cross-validation idea have been proposed to select the number of components in Principal Components Analysis (PCA). The element-wise k-fold (ekf) cross-validation is among the most used algorithms for PCA cross-validation. This is the method programmed in the PLS_Toolbox, and it has been stated to outperform other methods under most circumstances by Bro *et al.* (2008) in a numerical experiment. The *ekf* algorithm is based on missing data imputation and it can be programmed using any method for this purpose. In this paper, the *ekf* algorithm with the simplest missing data imputation method, trimmed score imputation (TRI), is analyzed. A theoretical study is driven to identify in which situations the application of *ekf* is adequate and, more important, in which situations it is not. The results presented show that the *ekf* method may be unable to assess the extent to which a model represents a test set and may lead to discard Principal Components (PCs) with important information. On a second paper of this series, other imputation methods are studied within the *ekf* algorithm.

Keywords: Principal Component Analysis, number of components, cross-validation, missing data, compression.

1 Introduction

Much work has been devoted to find an *'optimum'* -in some sense- or at least appropriate number of PCs in a PCA model, especially considering a calibration data set so limited in size so that external validation is not possible. A good survey on the matter can be found in the book by Jackson [1].

Wold [2] proposed the use of cross-validation for the determination of the number of PCs. In cross-validation, data are divided in G groups. Each time, a model is calibrated from the whole data-set but a group. Afterwards, the data from that group are predicted using the model and a Criterium of Goodness of Fit (CGF) is computed. This is repeated for each of the G groups and a total CGF for a model is obtained. In PCA, the CGF is computed for the models with 1 PC, 2 PCs, 3 PCs, and so on. From the shape of the CGF, the optimum number of PCs is estimated. Eastment and Krzanowski [3] and Nomikos and MacGregor [4] suggested the use of cross-validation when the PCA model is going to be used for future observations, which are independent of the calibration data. This is because cross-validation allows the estimation of the prediction error expected for incoming data.

Recently, Bro *et al.* [5] compared most of the methods which are currently used with "spectral-type" data. They concluded that the one implemented in the PLS_Toolbox [6] generally outperforms the other methods studied. Although cross-validation methods applying the Expectation-Maximization (EM) algorithm gave similar results, this was at the expense of being computationally intensive. The cross-validation approach in the PLS_Toolbox is referred here as the element-wise k-fold (*ekf*) algorithm. It was originally suggested by Wold as an alternative method to the one he also proposed in reference [2].

The *ekf* algorithm is based on the capability of missing data recovery of the PCA model [7, 8, 9]. In each cross-validation iteration, some elements of the matrix of data are artificially discarded and recovered with a missing data method; from the actual and the estimated values of the discarded data, an estimation error is computed. The sum-of-squares of estimation errors

(typically referred as the PRediction Error Sum-of-Squares or PRESS) is used as CGF to select the number of PCs. The original *ekf* proposal by Wold and the cross-validation in the first releases of the PLS_Toolbox were based on the simplest missing data imputation method: the trimmed score regression (TRI). The algorithm studied by Bro *et al.* [5] and the one found in new releases of the PLS_Toolbox are based on a slightly more complex imputation method: projection to the model plane (PMP) [7].

Unlike other cross-validation approaches for PCA models, the *ekf* algorithm provides a PRESS curve which may present a valley shape, with a minimum value. In principle, the lowest value of PRESS is signaling the optimum number of PCs in terms of estimation error. This curve may be easy to interpret for the practitioner due to its similarity to those obtained when cross-validating regression models -e.g. for Partial Least Squares (PLS) models.

Due to the promising results of the *ekf* found in [5], there is a clear interest in this method. This series of papers is devoted to characterize the PRESS curve provided by the *ekf* algorithm. This study is useful to understand the performance of the algorithm, to determine potential shortcomings and to identify in which situations the *ekf* is adequate to select the number of PCs and in which situations it is not. In this paper, the focus is on the TRI version of the algorithm. The PMP version, among other imputation methods, is studied in the companion paper. It should be noted that this study shows that the original method based on TRI presents better properties in front of noise, and therefore should be preferred.

The paper is organized as follows. In Section 2 the notation used throughout the paper is presented. In Section 3 the *ekf* algorithm is introduced. The TRI missing data method is treated in detail in Section 4. Using the results of this section, an efficient version of *ekf* is developed in Section 5. Section 6 is devoted to characterize and understand the PRESS in *ekf*. Section 7 presents the inconsistency and directional dependence problems in *ekf*. Section 8 discusses the results and Section 9 proposes some concluding remarks.

2 Notation

Scalars are specified with lower case letters, column vectors with bold lower case letters and matrices with bold upper case letters. Constants are specified with upper case letters.

Equations presenting matrix and vectorial products and sums of scalars are used indistinctly throughout the paper for the sake of easy understanding. Without loss of generality, an explicit ordering of the variables $m \in \{1, ..., M\}$, the observations $n \in \{1, ..., N\}$ and the loading vectors of the PCs $a \in \{1, ..., A\}$ is assumed in the sums. The number of PCs in a model is specified with A, whereas the maximum number of PCs in cross-validation is A_{max}. Groups of variables or observations are specified in capital regular, using G for a group of observations and H for a group of variables. The number of groups of observations and variables in cross-validation is specified as G_{tot} and H_{tot}, respectively.

A sum including all variables but m is represented by $\sum_{v \neq m}$

A sum including all variables in a group H is represented by $\sum_{v \in \mathrm{H}}$

A sum including all variables in a group H except variable m is represented by $\sum_{v \in \mathrm{H} \setminus m}$

3 Element-wise *k*-fold (*ekf*) Cross-validation

Let us define matrix $\mathbf{X}$ as a $N \times M$ matrix of data with N observations or objects on M variables. The PCA of matrix $\mathbf{X}$ follows the expression:

$$\mathbf{X} = \mathbf{T}^A \cdot (\mathbf{P}^A)^t + \mathbf{E}^A, \tag{1}$$

where $\mathbf{T}^A$ is the $N \times A$ score matrix, $\mathbf{P}^A$ is the $M \times A$ loading matrix and $\mathbf{E}^A$ is the $N \times M$ matrix of residuals.

For a $1 \times M$ object $\mathbf{x}_n^t$ (n-th row of $\mathbf{X}$) to be modelled, the corresponding $1 \times A$ score vector $(\boldsymbol{\tau}_n^A)^t$ (n-th row of $\mathbf{T}^A$) is obtained as follows:

$$(\boldsymbol{\tau}_n^A)^t = \mathbf{x}_n^t \cdot \mathbf{P}^A. \tag{2}$$

For each PC ($A = 1...A_{max}$)

For each group of objects (G = $1...G_{tot}$)

Form $\mathbf{X}_*$ with data from all groups but G

Form $\mathbf{X}_\#$ with data from G

Fit a PCA model from $\mathbf{X}_*$, obtaining $\mathbf{P}_*^A$ and $\mathbf{T}_*^A$

$\mathbf{T}_\#^A = \mathbf{X}_\# \cdot \mathbf{P}_*^A$

$\hat{\mathbf{X}}_\# = \mathbf{T}_\#^A \cdot (\mathbf{P}_*^A)^t$

$\mathbf{R}_\mathrm{G}^A = \mathbf{X}_\# - \hat{\mathbf{X}}_\#$

end

Combine matrices $\mathbf{R}_\mathrm{G}^A$ in $\mathbf{R}^A$

$PRESS^A = \sum_{n=1}^{N} \sum_{m=1}^{M} (r_{n,m}^A)^2$

end

Algorithm 1: Row-wise k-fold (*rkf*) algorithm.

From the scores and the PCA model, the object can be reconstructed according to:

$$\hat{\mathbf{x}}_n^A = \mathbf{P}^A \cdot \boldsymbol{\tau}_n^A, \tag{3}$$

the reconstruction error being:

$$\mathbf{r}_n^A = \mathbf{x}_n - \hat{\mathbf{x}}_n^A. \tag{4}$$

The simplest cross-validation procedure is the so-called row-wise k-fold cross-validation or *rkf* ([10], through [11]). In each iteration of the *rkf* algorithm, a model is calibrated from the whole data-set but a group of objects. These objects are afterwards passed through the PCA model and the reconstruction error (4) is computed.

The *rkf* algorithm is presented in Algorithm 1. For the sake of easy understanding, the algorithm is shown with two nested loops. The output of the algorithm is the matrix of reconstruction

errors $\mathbf{R}^A$ (with elements $r^A_{n,m}$ in the n-th row and m-th column) and the PRESS computed for $A = 1...A_{max}$ PCs. Some authors argue that $\mathbf{R}^A$ does not contain true prediction errors since the estimation is not independent of the actual values in the objects–$\hat{\mathbf{x}}^A_n$ is obtained from $\mathbf{x}_n$ through (2) and (3). Nonetheless, for the sake of homogeneity with the *ekf* algorithm introduced below, the term PRESS is used.

The *ekf* method is an extension of the *rkf* which is grounded in the following idea: since the PCA model establishes relationship structures among the variables, its prediction power should be measured by predicting the value of a variable from the rest taking into account these structures -i.e., the PCA model. This idea is incorporated by adding a third nested loop in the *ekf* algorithm which iterates through the variables.

The *ekf* method is specified in Algorithm 2. The inner loop, which iterates through the variables, is highlighted in dark gray color and the core of the algorithm is in light gray color. This core performs the missing values method, which is the direct estimation. This method will be treated in detail in the following section. In the algorithm, the initial value for left out (missing) variables is 0 ($\mathbf{X}_{\#,h} = 0$). Assuming data have been mean centered, this is an unconditional mean replacement which is equivalent to trimmed score imputation (TRI) [8]. The output of the algorithm is the matrix of prediction errors $\mathbf{E}^A$ (with elements $e^A_{n,m}$ in the n-th row and m-th column) and the PRESS computed for $A = 1...A_{max}$ PCs.

In the *rkf* and *ekf* algorithms, one controversial point is to decide whether the preprocessing information, i.e. the average and weight of the variables, should be estimated either from the entire calibration data $\mathbf{X}$ or else from $\mathbf{X}_*$ and then applied to $\mathbf{X}_\#$ within Algorithms 1 and 2. A discussion regarding this matter can be found in several papers [2, 12, 5]. Here, under the assumption that the model will be applied to future observations, the second option is preferred. For the sake of easy understanding, parameters related to the preprocessing are omitted throughout the paper.

As stated in the introduction, one of the advantages thought for *ekf* PRESS curves is their resemblance to PRESS curves from regression models. In regression models, PRESS curves tend to present a valley shape, where the minimum represents the optimum model in terms of prediction

For each PC ($A = 1...A_{max}$)

For each group of objects (G = $1...G_{tot}$)

Form $\mathbf{X}_*$ with data from all groups but G

Form $\mathbf{X}_\#$ with data from G

Fit a PCA model from $\mathbf{X}_*$, obtaining $\mathbf{P}_*^A$ and $\mathbf{T}_*^A$

For each group of variables (H = $1...H_{tot}$)

Set $\mathbf{X}_{\#,\mathrm{H}} = 0$

$$\mathbf{T}_\#^A = \mathbf{X}_\# \cdot \mathbf{P}_*^A$$

$$\hat{\mathbf{X}}_\# = \mathbf{T}_\#^A \cdot (\mathbf{P}_*^A)^t$$

Restore its actual value to $\mathbf{X}_{\#,\mathrm{H}}$

$$\mathbf{E}_{\mathrm{G,H}}^A = \mathbf{X}_{\#,\mathrm{H}} - \hat{\mathbf{X}}_{\#,\mathrm{H}}$$

end

end

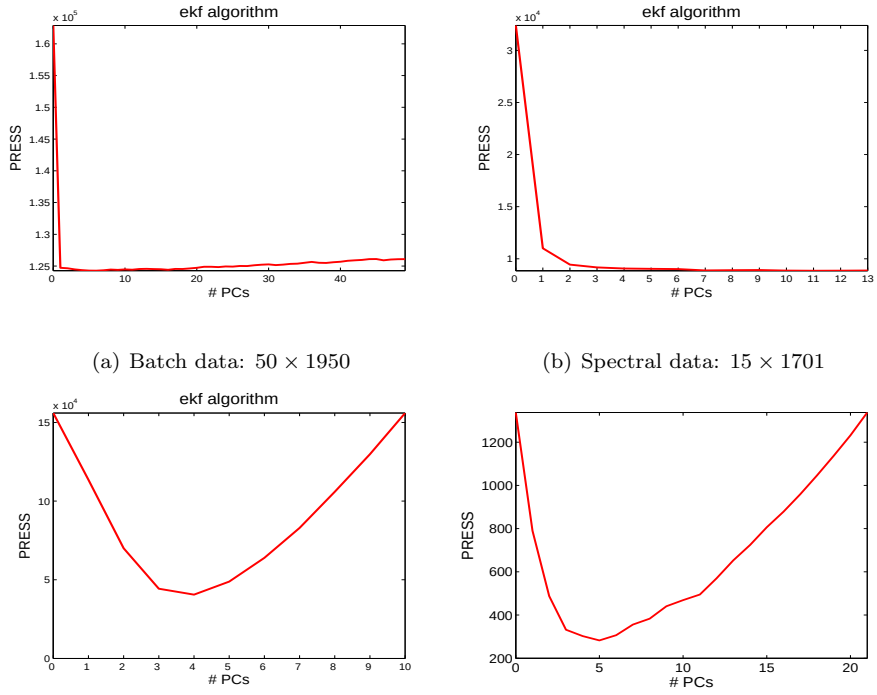
Combine matrices $\mathbf{E}_{\mathrm{G,H}}^A$ in $\mathbf{E}^A$

$PRESS^A = \sum_{n=1}^{N} \sum_{m=1}^{M} (e_{n,m}^A)^2$

end

Algorithm 2: Element-wise k-fold (*ekf*) algorithm based on the TRI imputation.

performance or otherwise stated the best trade-off between bias and variance. Nevertheless, it should be noted that the *ekf* does not always yield a PRESS curve with a clear valley shape or minimum. To illustrate this, in Figure 1 four PRESS curves obtained by *ekf* cross-validation are shown. The four correspond to typical chemometric data sets: batch process data, spectral data and data for multivariate image analysis. The top examples are fat matrices (i.e. $N << M$), and the valley shape and minimum is either not clear or nonexistent. The bottom examples in the figure are thin matrices (i.e. $N >> M$) which do present a convenient valley shape for the selection of the number of PCs. The remaining of this paper will be devoted to explain this behavior in the PRESS curve of *ekf*.



(a) Batch data: 50×1950

(b) Spectral data: 15×1701

(c) Batch data: 9750×10

(d) Image data: 60×21

Figure 1: Four examples of PRESS curves by *ekf* with typical chemometric data. The examples at the top correspond to fat matrices (more columns than rows) and the examples at the bottom correspond to thin matrices (more rows than columns). Figures (a) and (c) correspond to simulated batch data [13] batch-wise and variable-wise unfolded, respectively. Figure (b) corresponds to spectral data and Figure (d) corresponds to a data set for multivariate image analysis [14].

4 Overview of trimmed score imputation

To understand how the TRI method in the core of the *ekf* algorithm works, it is useful to characterize the way PCA captures the relationships among variables. A detailed theoretical study on this subject is performed in reference [15].

The reconstruction of an object $\mathbf{x}_n^t$ using PCA, presented in equation (3), can be reexpressed for each of the M elements $x_{n,m}$ of an object:

$$\hat{x}_{n,m}^A = (\boldsymbol{\tau}_n^A)^t \cdot \boldsymbol{\pi}_m^A, \tag{5}$$

where $(\boldsymbol{\pi}_m^A)^t$ is the $1 \times A$ vector with the loadings of variable m on the PCs, i.e. the m-th row of $\mathbf{P}^A$. Combining equations (2) and (5) yields:

$$\hat{x}_{n,m}^A = x_{n,m} \cdot \alpha_m^A + \sum_{v \neq m} x_{n,v} \cdot \beta_{v,m}^A, \tag{6}$$

where:

$$\alpha_m^A = \sum_{a=1}^{A} p_{m,a}^2 = (\boldsymbol{\pi}_m^A)^t \cdot \boldsymbol{\pi}_m^A, \tag{7}$$

$$\beta_{v,m}^A = \sum_{a=1}^{A} p_{v,a} \cdot p_{m,a} = (\boldsymbol{\pi}_v^A)^t \cdot \boldsymbol{\pi}_m^A. \tag{8}$$

Equation (6) is a regression model showing that $x_{n,m}$ takes part in its own estimation with weight α_m^A and the values of the rest of variables $x_{n,v}$ with weight $\beta_{v,m}^A$. Also, consider the following definition:

$$\mathbf{Q}^A = \mathbf{P}^A \cdot (\mathbf{P}^A)^t. \tag{9}$$

Matrix $\mathbf{Q}_A$ is a $M \times M$ symmetric matrix (projection matrix) where α_m^A is the element in the diagonal for row (or column) m and $\beta_{v,m}^A$ is the off-diagonal element on row v and column m. $\mathbf{Q}_A$ has A eigenvalues equal to 1 and $M - A$ eigenvalues equal to 0 [15].

The properties of α_m^A and $\beta_{v,m}^A$ are of utmost interest in this paper (see [15] for a detailed justification of these properties). Parameters $\beta_{v,m}^A$ take values in the interval [-0.5, 0.5]. Parameters α_m^A take values in the interval [0, 1] and they strictly increase with the number of PCs. $\alpha_m^A = 0$ is assumed for 0 PCs. As α_m^A increases, the relevance of other variables in the estimation of variable m according to (6) is reduced. $\alpha_m^A = 1$ will only happen when all the variability in m is captured by A PCs -strictly speaking, for full rank- and variable m is not in the span of the other variables.

The reconstruction error for $x_{n,m}$ can be expressed as:

$$r_{n,m}^A = x_{n,m} - (x_{n,m} \cdot \alpha_m^A + \sum_{v \neq m} x_{n,v} \cdot \beta_{v,m}^A). \tag{10}$$

Let us imagine that the actual value $x_{n,m}$ cannot be used in its own estimation in equation (6). This happens in the *ekf* algorithm, since values $x_{n,m}$ are treated as missing values. In this situation, $x_{n,m}$ can be estimated by substituting its value in equation (6) by a certain value $\hat{x}_{n,m}^{(0)}$. The estimation follows:

$$\hat{x}_{n,m}^{(1)} = \hat{x}_{n,m}^{(0)} \cdot \alpha_m^A + \sum_{v \neq m} x_{n,v} \cdot \beta_{v,m}^A. \tag{11}$$

This is termed here as the direct estimation. In particular, for $\hat{x}_{n,m}^{(0)} = 0$, this yields the TRI method [8].

The estimation error is computed according to the following expression:

$$e_{n,m}^A = x_{n,m} - \hat{x}_{n,m}^{(1)}. \tag{12}$$

The difference between the reconstruction error $r_{n,m}^A$ in (10) and the estimation error $e_{n,m}^A$ in (12) is that in the latter, the estimate $\hat{x}_{n,m}^{(1)}$ is computed without using the actual value $x_{n,m}$. Recall that $r_{n,m}^A$ is computed in the *rkf* algorithm and $e_{n,m}^A$ in the *ekf* algorithm. From the understanding of the relationship between $r_{n,m}^A$ and $e_{n,m}^A$, a more efficient version of the *ekf* algorithm is proposed in Appendix A.

Figure 2 illustrates the geometry of TRI. The case for 2 original variables and 1 PC is presented.

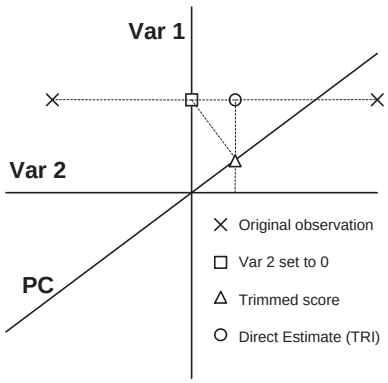


Figure 2: Geometric illustration of TRI with 1 PC of two original samples in a 2-dimensional space.

In the example, two observations represented by a cross have the same value in variable 1 but very different values in variable 2. Assume the value corresponding to variable 2 is missing in both observations. TRI starts setting those missing values to zero. Then, the two original observations are transformed into the point represented by the square. This point is projected on the PC and the resulting point (the trimmed score represented by a triangle) is projected on variable 2. The TRI estimate of the original observation is represented by the circle. Note that the estimate of both original observations is the same because they are computed from the common value of variable 1.

5 Characterization of the estimation error in *ekf*

The results presented in this section and Appendix B hold in general for the error by TRI, making no difference if the object was not used to calibrate the PCA model (as in cross-validation) or it was in fact part of the calibration data. Therefore, instead of the term PRESS, we will use the more general sum of squares of estimation errors (SSE). Notice the SSE includes the PRESS as a special case.

The SSE associated to a variable m for A PCs is computed according to the following expression:

$$SSE_m^A = \sum_{n=1}^{N_t} (e_{n,m}^A)^2, \tag{13}$$

where N_t is the number of objects used to compute the SSE and $e_{n,m}^A$ is the estimation error by TRI. The SSE of the complete data set, SSE_T^A, is equal to:

$$SSE_T^A = \sum_{m=1}^{M} SSE_m^A. \tag{14}$$

5.1 Theoretical constraints on the ekf curve

In PCA, an observable variable can be seen as the sum of: **redundant information** or shared variance, which can be found in another observable variable, and **non-redundant information** or unique variance, which is not found in any other observable variable. A variable with any content of non-redundant information is not in the span of the rest of variables, and so it cannot be expressed as a linear combination of the others. A variable solely composed of redundant information may or may not be in the span of the rest. For instance, take the simple case of two variables X_1 and X_2, so that $X_2 = X_1 + E$, with E a measurement error, and X_1 and E are independently generated. Although X_1 is completely composed of redundant information, since its variability is repeated in X_2, it is not in the span of X_2.

The estimation error of a single variable computed with *ekf* presents a number of properties - the mathematical proofs can be found in Appendix B:

Property 1 *The estimation error in ekf of a variable in the span of the other variables for a PCA model with* $A = Rank(\mathbf{X})$[1] *components is not null and depends on the error in the initial estimation.*

Property 2 *The estimation error in ekf of a variable not in the span of the other variables for a PCA model with* $A = Rank(\mathbf{X})$ *components is equal to the error in the initial estimation.*

[1] In the *ekf* cross-validation, the rank of interest is that of matrix $\mathbf{X}_*$.

Property 3 *The SSE of a variable (SSE_m^A) according to ekf attainable by any PCA model is lower bounded by the sum of squares of the non-redundant information in that variable.*

There are interesting comments on these properties. It is known that a variable which belongs to the span of the rest can be perfectly recovered from a linear combination of the others. Still, according to Property 1, the estimation error with *ekf* for a full rank model is not null as it would be expected. This is a straightforward consequence of using TRI as the missing data method and also applies to any direct estimation. Property 2 reflects the opposite case. For full rank, not even a portion of information of a variable out of the span of the rest is recovered. In this situation, the PCA model is useless for that purpose. Notice that even a slight portion of measurement noise or numerical error may cause a variable to be out of the span of the others, provided the number of observations is high enough. For instance, recall the previous example with variables X_1 and $X_2 = X_1 + E$ measured in a data set. If a value of X_1 is missing, it cannot be recovered by using TRI with a PCA for full rank (2 PCs). This is true even for very low variance in E. Property 3 reflects a limitation of *ekf*: the non-redundant information, although may be properly captured by the PCA model, is always included as part of the predictive error in *ekf*.

From the previous three properties, two additional properties for the SSE_T^A curve can be presented[2]. Recall that the SSE_T^A corresponds to the sum of SSE_m^A terms for all variables in the data set.

Property 4 *The SSE of a data set (SSE_T^A) according to ekf for $A = Rank(\mathbf{X})$ is lower or equal to the SSE of the initial estimation, being equal when the number of variables equals the rank.*

Property 5 *The SSE of a data set (SSE_T^A) according to ekf attainable by any PCA model is lower bounded by the sum of squares of the non-redundant information in the data.*

[2]Property 5 is a corollary of Property 3, but property 4 needs further derivation in Appendix B.

In Figure 3, two typical SSE_T^A curves computed by *ekf* are shown. When the number of variables is equal to the rank of the matrix of data ($\#var = rank$), all variables satisfy Property 2 and the SSE_m^A for both $A = 0$ and $A = Rank(\mathbf{X})$ match. A straightforward consequence is that the SSE_T^A for both $A = 0$ and $A = Rank(\mathbf{X})$ also match (Property 4). This is coherent with the PRESS curves found for the thin matrices in Figures 1(c) and 1(d). On the other hand, if the number of variables is higher than the rank, some variables satisfy Property 1 instead of Property 2. When the number of variables gets much higher than the rank ($\#var >> rank$), then most likely all variables satisfy Property 1. In that situation, the SSE_m^A for $A = Rank(\mathbf{X})$ is different to that for $A = 0$ and the resulting $SSE_T^{Rank(\mathbf{X})}$ is in fact lower than SSE_T^0, as illustrated in Figure 3. This is found in the PRESS curves of the fat matrices in Figures 1(a) and 1(b).

In real data, some content of non-redundant information is likely to be present in all variables since a certain amount of -linearly independent- measurement noise is always expected. Linearly dependent variables are only expected as a result of mathematic computations (artificial variables) or due to an insufficient number of objects in the data -so that the rank is determined by the number of objects N instead of the number of variables M. Therefore, $\#var >> rank$ is hardly found except when $N << M$ (i.e. fat matrices). The latter is the most challenging situation from the chemometrics point of view, since $N << M$ is often found in batch process data, spectroscopy, system biology, etc.

In both situations when $\#var = rank$ and when $\#var >> rank$, according to Properties 3 and 5, the amount of non-redundant information imposes a minimum value for each SSE_m^A and for SSE_T^A. This is also depicted in Figure 3. The MIA data set [14] (Figure 1(d)) will be further employed to illustrate this effect of the non-redundant information in the PRESS curve. The PRESS is computed for the data set corrupted with different levels of white measurement noise, i.i.d. in the 21 variables. As shown in Figure 4, the more the noise introduced the higher the minimum in the PRESS curve. This effect is expected since the white noise introduces only non-redundant information to the data, increasing the minimum attainable by the PRESS curve. Note that as measurement noise increases the minimum of the PRESS curve is attained at a

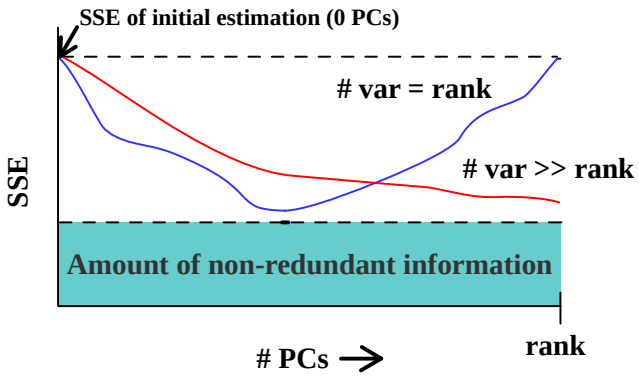


Figure 3: Typical examples of SSE_T^A by *ekf*. For $\#var = rank$, all variables satisfy Property 2 and the SSE_T^A for both $\#PCs = 0$ (the initial est.) and $\#PCs = rank$ coincide. For $\#var >> rank$, most likely all variables satisfy Property 1 instead of Property 2. In any case, the curves remain above the amount of non-redundant information (Property 3).

lower number of PCs. Therefore, a high content of non-redundant information may lead to an underestimation of the number of PCs.

5.2 Rationalization of the valley-shape of the *ekf* curve

Properties 1, 2 and 4 defined in previous section constrain the value of SSE_m^A and SSE_T^A for full rank. This constraint favors the valley shape of SSE_m^A and SSE_T^A curves. Let us focus on the leave-one-variable-out case for a deeper discussion on this valley shape. From (18) and (14), the SSE_m^A of TRI of leave-one-variable-out follows:

$$SSE_m^A = (\alpha_m^A)^2 \cdot \sum_{n=1}^{N_t} (x_{n,m})^2 + 2 \cdot \alpha_m^A \cdot \sum_{n=1}^{N_t} x_{n,m} \cdot r_{n,m}^A + \sum_{n=1}^{N_t} (r_{n,m}^A)^2. \tag{15}$$

Parameter α_m^A is a sum of squares (7) and so it is monotonically increasing with A. Therefore, the first factor in (15) is also monotonically increasing with A since $\sum_{n=1}^{N_t}(x_{n,m})^2$ remains unaltered as A varies. On the other hand, each PC added to a PCA model reduces the sum of squares of reconstruction error. This makes the third factor of (15) show decreasing tendency with A, although it does not need to be strictly decreasing. The way the second factor of (15) will evolve

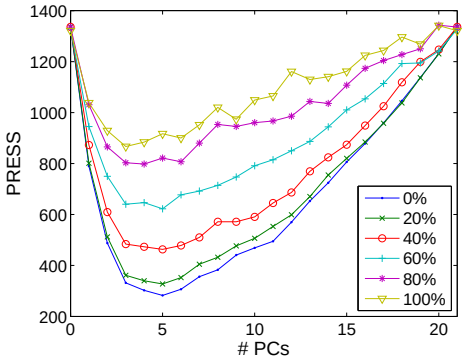


Figure 4: PRESS curves by *ekf* for the auto-scaled Firenze data set [14] with different levels of white noise introduced: from a 0% (noise-free data) to a 100% of noise (equal variance of noise and data).

as A increases is undetermined *a-priori*. It combines α_m^A, which increases with A, with $\sum_{n=1}^{N_t} x_{n,m} \cdot r_{n,m}^A$. The latter is a measure of the correlation between original data and residuals, correlation that clearly decreases with A.

For 0 PCs there is no information about the relationships among variables and no variance is captured by the model, so that $SSE_m^0 = \sum_{n=1}^{N_t}(x_{n,m})^2$. On the other hand, for full rank it holds that $r_{n,m}^A = 0 \ \forall n$, so that $SSE_m^{rank(\mathbf{X})} = (\alpha_m^A)^2 \cdot \sum_{n=1}^{N_t}(x_{n,m})^2$, fulfilling Property 1. In particular, since $\alpha_m^A \leq 1$ [15], then $SSE_m^{rank(\mathbf{X})} \leq SSE_m^0$, like in the examples of Figure 3. $SSE_m^{rank(\mathbf{X})}$ is maximum when m does not belong to the span of the rest of variables (and so it cannot be expressed as a linear combination of the others), so that $\alpha_m^A = 1$ [15]. Then, $SSE_m^{Rank(\mathbf{X})} = \sum_{n=1}^{N_t}(x_{n,m})^2 = SSE_m^0$, satisfying Property 2.

Equation (15) shows a compromise between variance captured (reduction of $r_{n,m}^A$) and model structure (α_m^A). This can also be thought of as the trade-off between bias and variance in the prediction of a variable from the others. When the SSE_m^A curve decreases during the first PCs it means that the decrease of the third factor in (15) dominates the increase in the first and second factors. This happens for PCs of high variance (high reduction of $r_{n,m}^A$) with high loads of many variables (low α_m^A values reflecting redundant information captured). On the other hand, the value

of SSE_m^A for full rank is constrained as already discussed. In particular, for $\#var = rank$ (Figure 3), the SSE_m^A has to rise during the last PCs to the same height it fell during the first PCs. Thus, if the SSE previously decreased, the first factor in (15) will dominate the third factor in (15) for the remaining PCs. This evidences a high content of non-redundant information captured (e.g. measurement noise) in those PCs. This causes the valley shape in the PRESS, observed for instance in Figure 4, especially for low noise percentages. On the other hand, variables solely composed of non-redundant information do not present such valley shape in the SSE_m^A curve. This is coherent with the effect observed in Figure 4 when introducing a high percentage of i.i.d. noise in the MIA data set.

It has been shown that typical SSE (or PRESS) curves may be easy to interpret but, why are there situations in which the SSE_T^A becomes so irregular? The answer to that is straightforward. The SSE_T^A is a pool of expressions similar to equation (15) for the different variables. Therefore, whereas the first factor may be low for some variables, it may be high (compared to the third factor) for others -especially those with a high content of non-redundant information. In that situations, the SSE_T^A may be complex to interpret. A nice example of a data set with such a pool of different behaviors is presented in [15].

6 Inconsistency and directional dependence

From the derivation presented in this paper, it is clear that the prediction error of a PCA model, computed with TRI, depends on how the information in a variable can be recovered from the others. This implies a number of shortcomings for the *ekf* cross-validation reviewed in this section.

Consider the examples shown in Figure 5. In each of the two rows of figures, a different direction for the first PC is considered. The figures in the second column geometrically characterize the ratio between the SSE of TRI in the model with 1 PC and the total sum of squares. Thus, observations laying in the areas with value lower than 1 (light areas) yield a lower prediction error for a model with 1 PC than for 0 PCs, the latter being their sum of squares.

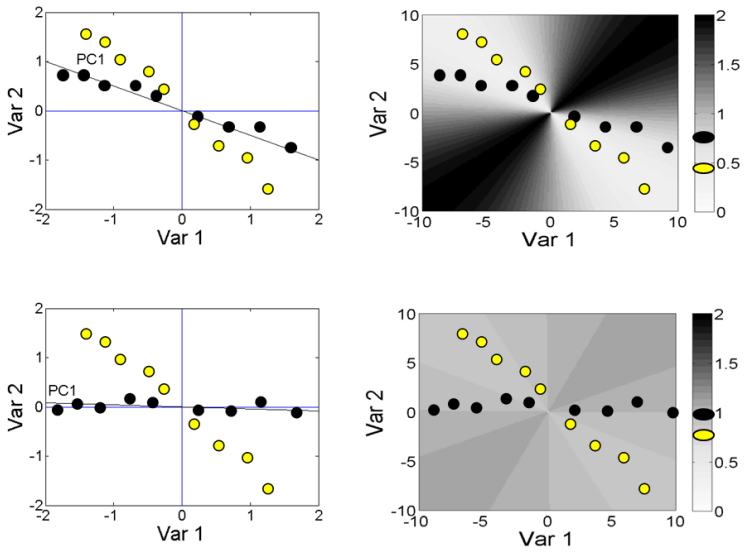
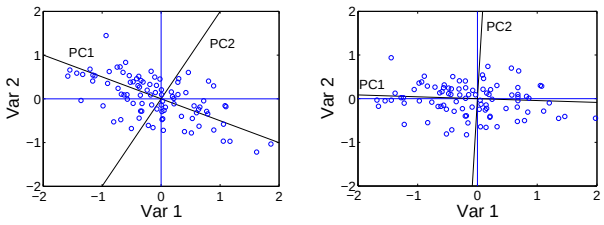


Figure 5: Geometrical characterization of the ratio between the SSE of TRI in the model with 1 PC and the total sum of squares. This allows to identify the areas where the first PC improves the prediction performance of missing elements. Two different directions for the first PC are shown in the rows. Also, two groups of observations are superimposed on the color maps and their estimation error by TRI is signaled.

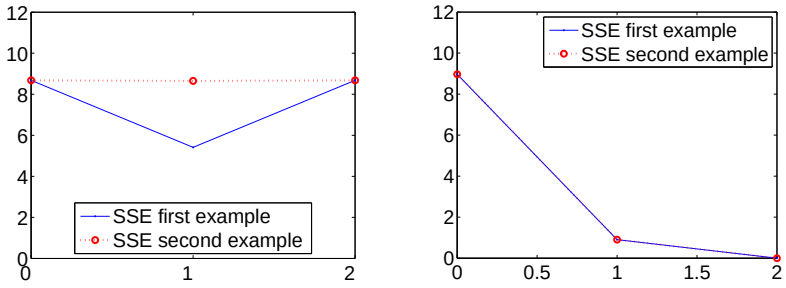


(a) (b)

Figure 6: Data obtained from simulated scores and different loading vectors. The scores for the first PC are simulated with zero mean and unit variance. The scores for the second PC are simulated with zero mean and 0.1 variance.

The areas in which there is an increase or reduction of SSE match for the two examples (top-left and bottom-right quadrants). In both cases, the best estimation performance is obtained for the points close to the line $-var1 = var2$, the bisector, independently of the specific direction of the first PC. Thus, the reduction of SSE is mainly determined by the direction of the original variables in the space, which establish the quadrants in the space. The direction of the PCs only determines in which quadrants there is a reduction of SSE. The implications of this are important. Take the two sets of observations in the plot. The observations in dark color are closer to the first PC that the observations in light color, which are closer to the bisector. Nevertheless, the SSE of the former is higher than the SSE of the latter. The conclusion is that the SSE by TRI (and so by the *ekf* algorithm) is inconsistent. The inconsistency problem can be found in general for any number of variables and eigenvalues distribution (see Appendix C), although the points where the SSE attains its minimum value do not necessary have to be the bisectors of the original variables. The inconsistency problem prevents from using the error by TRI to assess the extent to which a model represents a test set of observations, not used during the model calibration. This is because a set of observations with a variance-covariance structure not reflected by the PCA subspace may yield a lower estimation error than observations very much distributed according to the PCA subspace.

On the other hand, although the areas in which there is and increase or reduction in SSE match for the two examples in Figure 5, the amount of increase or reduction is clearly different.



(a) (b)

Figure 7: SSE_T^A curves for TRI (a) and reconstruction (b) errors using the data of Figure 6.

In the first example, the prediction error greatly varies for observations located at different points in the original space. In the second example, the difference between two points is much lower and the ratios are always close to 1. The magnitude of the ratio depends on the angle of rotation between both the original space and latent sub-space. As a conclusion, TRI (and so the *ekf* algorithm) suffers from directional dependence, since for rotated PCs the same true relationship yields different SSE values.

Again, directional dependence has important implications. Let us propose an illustrative example. The same distribution is assumed for a set of scores in the two examples of Figure 5. Imagine these two examples are the result of observing the same underlying phenomena with different measured variables[3]. For instance, the scores of the first PC are normally distributed with unit variance representing useful information. The scores of the second PC are also normally distributed with variance equal to 0.1 representing measurement noise. The distribution of the resulting observations is shown in Figure 6. If the prediction error is computed as a quadratic sum of the TRI errors, the SSE curve is completely different. This is shown in Figure 7(a). For the first case, the SSE presents a clear minimum for 1 PC, whereas in the second example the improvement is negligible. In this latter case, an analyst may arrive to the conclusion that the first PC is not representing useful information. This would be an incorrect conclusion. In both cases, the PC improves data understanding. For the first example, it tells us that a linear relationship among

[3]The use of different measured variables may also, but not necessarily, change the variance in the scores.

the two variables exists. For the second example, it tells us that the variability is concentrated in 'Var1' and that a linear relationship among the two variables is not found. This information is as important as the information given in the first example. Furthermore, if the model is used for monitoring, the first PC, which is representative of the underlying phenomena, should be added to the model in either of both examples. Therefore, in scenarios with high amount of non-redundant information the *ekf* algorithm may lead to underestimate the appropriate number of PCs (i.e. underfitting). This was already shown in Figure 4.

The directional dependence is a manifestation of the fact that the estimation error by *ekf* depends on the parameters α_m^A and $\beta_{v,m}^A$ (see equation (20)), i.e. on how the information is arranged through the observable variables. The non-redundant information in the data may affect the PRESS shape, but in a fairly complicated way, depending on how this information is combined with redundant information in the variables. If groups of variables are left out at the same time within the cross-validation algorithm, only variables outside one group affect the PRESS of variables within that group. Thus, the PRESS relies heavily on how the groups of variables are selected. Also, no matter the content of measurement noise in the variable, the PRESS is constrained to arrive to a certain value for full rank. Therefore, a small content of noise may cause a steep increase of the PRESS.

Let the same experiments be repeated using the reconstruction error. The results are presented in Figures 8 and 7(b). In this case, the distribution of the error does not depend on the original space, but only on the latent space. The areas where there is an improvement of estimation are rotated according to the PCs and without any additional transformation. Therefore, the reconstruction error (and in particular the *rkf* algorithm) does not suffer from inconsistency or directional dependence. In the literature it has been stated that this is an incorrect way to assess the prediction error of a PCA model, because the estimation of a value is not independently calculated from that value (6). Nonetheless, due to the inconsistency and directional dependence problems of the *ekf* algorithm, the use of *rkf* may be more appropriate than that of *ekf* in some applications. For instance, the *rkf* is applied to compute the goodness of prediction index Q^2 used

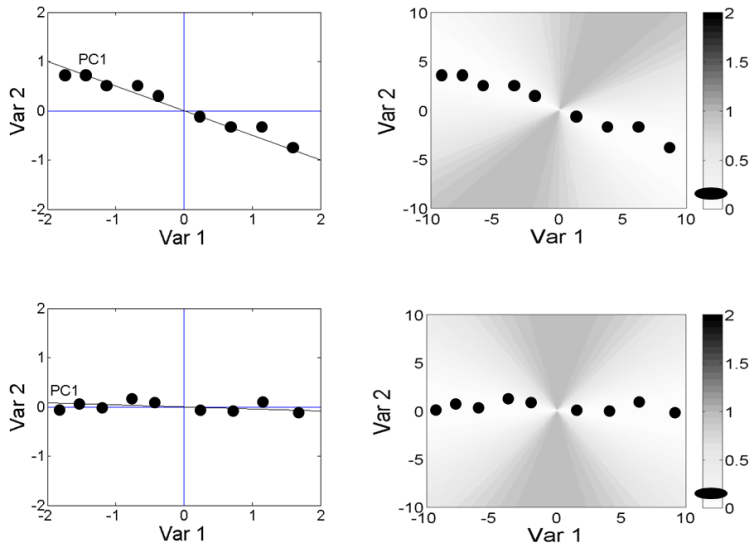


Figure 8: Geometrical characterization of the ratio between the reconstruction error in the model with 1 PC and the total sum of squares. Two different directions for the first PC are shown in the rows. Also, two groups of observations are superimposed on the color maps and their estimation error by TRI is signaled.

in the SVI plots proposed in [15] for data interpretation.

7 Discussion

There are several theoretical arguments against the convenience of the use of the *ekf* PRESS curve to determine the number of PCs in many situations. The PRESS by *ekf* measures the relevance of a piece of information by the amount of variance and the number of variables in which it is reflected. Therefore, this method is not suited to decide the number of PCs when the objective is the interpretation or the monitoring of the latent phenomena, or to compress the data. In none of these applications, the number of replications of the same piece of information should be a concern to decide the addition of a PC. For instance, relevant information for monitoring, interpretation or compression may be reflected in one single variable, and the PRESS curve by *ekf* would not

show this at all. At the same time, certain types of noise may be correlated in several variables.

The objective for which *ekf* was originally proposed [2] was to find the *'optimum'* number of PCs in the PCA model (1). This is not a well-defined objective unless we define the meaning of *'optimum'*. The error estimated by *ekf* may be defined as "*the error when trying to recover missing values in incoming data*". Therefore, the number of PCs selected according to *ekf* is the one which minimizes the sum-of-squares of this prediction error. The model with this number of PCs is expected to yield the lowest prediction error of missing data in future objects, provided these missing elements are recovered with the same estimation method used in *ekf*. Thus, the *ekf* is specifically suited when the objective of the PCA model is missing data recovery.

Considering the previous discussion, a reader may wonder why the *ekf* has been found to yield in general a good performance in numerical experiments [5], in particular in "spectral-like" data. First of all, it should be noted that the cited reference is restricted to the application referred to here as compression (nor monitoring or interpretation), where the aim is to distinguish between true structure and measurement noise. Although theoretically the *ekf* is not suited to determine the number of components in compression, it should be noted that it is in general a good heuristic for that provided the variables are correlated, like in spectra. In that case, the first PCs capture a high portion of variance shared by a lot of variables. These PCs make the PRESS by *ekf* reduce. On the other hand, non-correlated measurement noise of low variance will be found in the last PCs, which satisfy the aforementioned requirements to make the PRESS rise. Nevertheless, this good performance in compression may not generalize to all types of chemometrics data sets, specially when variables solely composed of non-redundant information are found. For instance, this may be the case of some industrial process data sets registering measurements from different instruments, process stages, and so on.

8 Conclusion

In applications in which Principal Components Analysis (PCA) is used, the results obtained can be very different depending on the number of principal components (PCs) selected. Therefore, the method to determine the appropriate number is critical. It is extremely important to be aware of the features of the method used, instead of applying it blindly. Unfortunately, this is not the current practice. For instance, most cross-validation approaches are applied without a conscientious knowledge of their features.

The aim of this paper is to provide a theoretical study on cross-validation. In particular, this paper is devoted to characterize the predictive error sum-of-squares (PRESS) curve provided by the element-wise k-fold (*ekf*) algorithm based on the trimmed score imputation (TRI) method. This is the algorithm originally programmed in the PLS_Toolbox. Also, an extension of this algorithm based on other imputation method was recently stated to outperform other methods under most circumstances for the determination of the number of components in PCA models [5], in particular in "spectral-like" data. This extension, among others, is studied in the second part of this series.

In the *ekf*, the minimum in the PRESS curve is used to determine the number of PCs to retain. The theoretical derivation performed in this paper is not only useful to understand the PRESS curve provided by the algorithm, but also to determine its shortcomings and to identify in which situations its use is adequate and, more important, in which situations it is not. The results presented show that the *ekf* method may be unable to assess the extent to which a model represents a test set and may lead to discard Principal Components with important information.

Another contribution of this paper is a computationally efficient version of the *ekf* algorithm based on TRI. Although this version is faster than the original one, it is specially profitable for a high number of variables in the data set. For instance, for 10.000 variables the computation time was reduced in one order of magnitude.

Acknowledgements

Research in this area is partially supported by the Spanish Ministry of Science and Innovation and FEDER funds from the European Union through grant DPI2008-06880-C03-03. José Camacho was funded by the Juan de la Cierva program, Ministry of Science and Innovation, Spain. This study was carried out when José Camacho was at the Universidad Politécnica de Valencia and at the Universitat de Girona, Spain.

A An efficient *ekf* algorithm

The direct estimation introduced in Section 4 for the leave-one-variable-out case, eq. 11, can be extended to the more general case when the values of several variables are missing at the same time:

$$\hat{x}^{(1)}_{n,m} = \hat{x}^{(0)}_{n,m} \cdot \alpha^A_m + \sum_{v \in \mathrm{H}_m \setminus m} \hat{x}^{(0)}_{n,v} \cdot \beta^A_{v,m} + \sum_{v \notin \mathrm{H}_m} x_{n,v} \cdot \beta^A_{v,m}, \tag{16}$$

where H_m is a group of variables which are estimated at the same time than variable m in the inner loop of *ekf*.

In this section, a computationally efficient version of the *ekf* algorithm is proposed. The standard formula for TRI used in the *ekf* algorithm is based on (16) and (12) for $\hat{x}^{(0)}_{n,v} = 0, \forall v \in \mathrm{H}_m$:

$$e^A_{n,m} = x_{n,m} - \sum_{v \notin \mathrm{H}_m} x_{n,v} \cdot \beta^A_{v,m}. \tag{17}$$

Alternatively, an efficient formula can be found by using the reconstruction error. Let us show the procedure for direct estimation and then particularize for TRI. From (10), (11) and (12), the estimation error associated to the direct estimation of the leave-one-variable-out case can be computed:

$$e^A_{n,m} = \epsilon^{(0)}_{n,m} \cdot \alpha^A_m + r^A_{n,m}, \tag{18}$$

where:

$$\epsilon^{(0)}_{n,m} = x_{n,m} - \hat{x}^{(0)}_{n,m}. \tag{19}$$

This can be straightforwardly extended to the case where multiple variables are estimated at the same time:

$$e^A_{n,m} = \epsilon^{(0)}_{n,m} \cdot \alpha^A_m + \sum_{v \in \mathrm{H}_m \setminus m} \epsilon^{(0)}_{n,v} \cdot \beta^A_{v,m} + r^A_{n,m}. \tag{20}$$

Since in TRI it holds that $\hat{x}_{n,v}^{(0)} = 0, \forall v \in \mathrm{H}_m$, the formula (20) becomes:

$$e_{n,m}^{A} = x_{n,m} \cdot \alpha_{m}^{A} + \sum_{v \in \mathrm{H}_m \setminus m} x_{n,v} \cdot \beta_{v,m}^{A} + r_{n,m}^{A}. \tag{21}$$

Although in (21) there are more operations than in (17), $r_{n,m}^{A}$ can be computed outside the inner loop of *ekf*. When the number of variables out of group H_m is higher than that in H_m, the use of (21) is more profitable than that of (17) in terms of computation time. To design the *ekf* algorithm using (21), the inner loop (in dark gray color in Algorithm 2) is replaced by:

$\mathbf{T}_{\#}^{A} = \mathbf{X}_{\#} \cdot \mathbf{P}_{*}^{A}$

$\hat{\mathbf{X}}_{\#} = \mathbf{T}_{\#}^{A} \cdot (\mathbf{P}_{*}^{A})^{t}$

$\mathbf{R}_{\mathrm{G}}^{A} = \mathbf{X}_{\#} - \hat{\mathbf{X}}_{\#}$

For each group of variables ($\mathrm{H} = 1...H_{tot}$)

$\mathbf{Q}_{*,\mathrm{H}}^{A} = \mathbf{P}_{*,\mathrm{H}}^{A} \cdot (\mathbf{P}_{*,\mathrm{H}}^{A})^{t}$

$\mathbf{E}_{\mathrm{G,H}}^{A} = \mathbf{X}_{\#,\mathrm{H}} \cdot \mathbf{Q}_{*,\mathrm{H}}^{A} + \mathbf{R}_{G,\mathrm{H}}^{A}$

end

where $\mathbf{P}_{*,\mathrm{H}}^{A}$ is the sub-matrix of $\mathbf{P}_{*}^{A}$ corresponding to the variables (rows) belonging to group H. Thus, $\mathbf{P}_{*,\mathrm{H}}^{A}$ has as many rows as the number of variables in H and A columns.

In particular, for $H_{tot} = M$ (one variable per group, i.e. $\mathrm{H}_m = \{m\}$), $\mathbf{Q}_{*,\mathrm{H}}^{A}$ becomes a scalar and the inner loop can be substituted by a matrix multiplication.

In Table 1, *rkf* and the two versions of *ekf* (named *ekf* and efficient *ekf* or *eekf*) are compared in terms of computation time for different matrix sizes and $G_{tot} = N$ and $H_{tot} = M$ in the MATLAB environment. The *eekf* version where the inner loop is replaced by a matrix multiplication, referred to as *eekf2*, is also considered. The algorithm *eekf* is faster than *ekf* in all the cases studied, but larger differences are found for high H_{tot} values. The differences between *eekf* and *eekf2* approaches are due to the different computation time in the MATLAB environment between a 'for' loop and a matrix operation performing the same computation. Therefore, these differences are expected to hold only in the MATLAB environment. The *eekf2* method is so fast

its computation time is similar to that of the *rkf* approach.

Table 1: Comparison of computation time (seconds) in the MATLAB execution of the traditional *ekf* version, the efficient version (*eekf*), the efficient version where the inner loop has been replaced by a matrix multiplication (*eekf2*) and *rkf*. For all the cases, $A = 10$, $G_{tot} = N$ and $H_{tot} = M$. Computations performed on Intel(R) Core(TM)2 Duo CPU T7500 2.20GHz and 1,99 GB RAM running XP Professional operative system.

$N \times M$	100×10	100×100	100×1000	100×10000
ekf	0.40	1.93	7.13	351.03
eekf	0.32	1.80	4.53	30.19
eekf2	0.31	1.72	3.76	21.96
rkf	0.31	1.72	3.73	21.88

B Properties of *ekf*

Property 1 *The estimation error in ekf (or in general from direct estimation) of a variable in the span of the other variables for a PCA model with $A = Rank(\mathbf{X})$ components is not null and depends on the error in the initial estimation.*

Proof:

The reconstruction of an observation from the PCA model is perfect for $A = Rank(\mathbf{X})$. Therefore, the reconstruction error is null:

$$x_{n,m} = x_{n,m} \cdot \alpha_m^A + \sum_{v \in \mathrm{H}_m \setminus m} x_{n,v} \cdot \beta_{v,m}^A + \sum_{v \notin \mathrm{H}_m} x_{n,v} \cdot \beta_{v,m}^A, \qquad A = Rank(\mathbf{X}). \tag{22}$$

Thus, the estimation error follows:

$$e^A_{n,m} = (x_{n,m} - \hat{x}^{(0)}_{n,m}) \cdot \alpha^A_m + \sum_{v \in \mathrm{H}_m \setminus m} (x_{n,v} - \hat{x}^{(0)}_{n,v}) \cdot \beta^A_{v,m}, \qquad A = Rank(\mathbf{X}). \tag{23}$$

Therefore, this error depends on the initial estimation. Notice this holds in general for every initial value of $\hat{x}^{(0)}_{n,m}$, not only for 0.

Property 2 *The estimation error in ekf (or in general from direct estimation) of a variable not in the span of the other variables for a PCA model with $A = Rank(\mathbf{X})$ components is equal to the error in the initial estimation.*

Proof:

Equation (22) can be re-expressed as:

$$x_{n,m} = \sum_{v \neq m} x_{n,v} \cdot \frac{\beta^A_{v,m}}{1 - \alpha^A_m}, \qquad A = Rank(\mathbf{X}). \tag{24}$$

Since variable m does not belong to the span of the others, it can be expressed as:

$$x_{n,m} = \sum_{v \neq m} x_{n,v} \cdot k_{v,m} + y_{n,m}, \qquad y_{n,m} \neq 0, \tag{25}$$

this is a contradiction of (24) except for the case $\alpha^A_m = 1$. In that case, it holds [15] that $\beta^A_{v,m} = 0,\ \forall v \neq m \in \{1, ..., M\}$ and (24) presents an indeterminate from. In that situation, from (16) we know that:

$$\hat{x}^{(1)}_{m,n} = \hat{x}^{(0)}_{m,n}. \tag{26}$$

Property 3 *The SSE of a variable (SSE^A_m) according to ekf (or in general from TRI) attainable by any PCA model is lower bounded by the sum of squares of the non-redundant information in that variable.*

Proof:

Any variable can be expressed as the sum of redundant information -which can be recovered from the other variables- and non-redundant information ($y_{n,m}$):

$$x_{n,m} = \sum_{v \neq m} x_{n,v} \cdot k_{v,m} + y_{n,m}. \tag{27}$$

For those variables which are a linear combination of the others, then $y_{n,m} = 0$. Rearranging equation (27):

$$y_{n,m} = x_{n,m} - \sum_{v \neq m} x_{n,v} \cdot k_{v,m}. \tag{28}$$

By convention, let us set parameters $k_{v,m}$ so that the sum of squares of $y_{n,m}$ for all observations is minimum, i.e. $\min_{k_{v,m}} \sum_{n=1}^{N} y_{n,m}^2$ - that is, the best fit in the quadratic sense. This means that we are choosing $k_{v,m}$ so that the sum of squares of the error of estimating m from the rest of the variables is minimum. With this definition we can assure that any q_n of the form:

$$q_n = x_{n,m} - \sum_{v \neq m} x_{n,v} \cdot r_v, \tag{29}$$

will satisfy that $\sum_{n=1}^{N} y_{n,m}^2 \leq \sum_{n=1}^{N} q_n^2$.

From (16), the prediction error of *ekf* in equation (12) can be expressed as:

$$e_{n,m}^A = x_{n,m} - (\hat{x}_{n,m}^{(0)} \cdot \alpha_m^A + \sum_{v \in \mathrm{H}_m \setminus m} \hat{x}_{n,v}^{(0)} \cdot \beta_{v,m}^A + \sum_{v \notin \mathrm{H}_m} x_{n,v} \cdot \beta_{v,m}^A), \tag{30}$$

which can be rearranged as:

$$e_{n,m}^A + \hat{x}_{n,m}^{(0)} \cdot \alpha_m^A + \sum_{v \in \mathrm{H}_m \setminus m} \hat{x}_{n,v}^{(0)} \cdot \beta_{v,m}^A = x_{n,m} - \sum_{v \notin \mathrm{H}_m} x_{n,v} \cdot \beta_{v,m}^A, \tag{31}$$

which follows the form in (29) for:

$$r_v = \begin{cases} 0, & for \quad v \in \mathrm{H}_m \\ \beta_{v,m}^A, & for \quad v \notin \mathrm{H}_m \end{cases}, \tag{32}$$

so that the following can be assured:

$$\sum_{n=1}^{N} y_{n,m}^2 \leq \sum_{n=1}^{N} (e_{n,m}^A + \hat{x}_{n,m}^{(0)} \cdot \alpha_m^A + \sum_{v \in \mathrm{H}_m \setminus m} \hat{x}_{n,v}^{(0)} \cdot \beta_{v,m}^A)^2. \tag{33}$$

In particular, for TRI (as in the definition of the *ekf* algorithm of this paper):

$$\sum_{n=1}^{N} y_{n,m}^2 \leq \sum_{n=1}^{N} (e_{n,m}^A)^2. \tag{34}$$

Property 4 *The SSE of a data set (SSE_T^A) according to ekf (or in general from direct estimation) for $A = Rank(\mathbf{X})$ is lower or equal to the SSE of the initial estimation, being equal when the number of variables equals the rank.*

Proof:

Let us name $\{m_1, ..., m_{\mathrm{H}}\}$ the variables in the group H_m, which are estimated at the same time. Let us define the following matrices:

$$\mathbf{E}_{\mathrm{H}_m}^A = \begin{bmatrix} e_{1,m_1}^A, & ..., & e_{1,m_{\mathrm{H}}}^A \\ ... & ... & ... \\ e_{N,m_1}^A, & ..., & e_{N,m_{\mathrm{H}}}^A \end{bmatrix}, \tag{35}$$

$$\mathbf{E}_{\mathrm{H}_m}^0 = \begin{bmatrix} \epsilon_{1,m_1}^0, & ..., & \epsilon_{1,m_{\mathrm{H}}}^0 \\ ... & ... & ... \\ \epsilon_{N,m_1}^0, & ..., & \epsilon_{N,m_{\mathrm{H}}}^0 \end{bmatrix}. \tag{36}$$

Let us finally define $\mathbf{\Omega}_{\mathrm{H}_m}$ as the sub-matrix of $\mathbf{Q}_A$ (9) taking the rows and columns corresponding to H_m. From (23) it holds:

$$\mathbf{E}_{\mathrm{H}_m}^A = \mathbf{E}_{\mathrm{H}_m}^0 \cdot \mathbf{\Omega}_{\mathrm{H}_m}, \quad A = Rank(\mathbf{X}). \tag{37}$$

The SSE corresponding to the group of variables follows:

$$SSE_{\mathrm{H}_m}^A = tr((\mathbf{E}_{\mathrm{H}_m}^A)' \cdot \mathbf{E}_{\mathrm{H}_m}^A), \tag{38}$$

and

$$SSE^0_{\mathrm{H}_m} = tr((\mathbf{E}^0_{\mathrm{H}_m})' \cdot \mathbf{E}^0_{\mathrm{H}_m}), \tag{39}$$

where tr stands for the trace of the matrix. From (37) and (38), using the properties of the trace follows:

$$SSE^A_{\mathrm{H}_m} = tr((\mathbf{E}^0_{\mathrm{H}_m})^t \cdot \mathbf{\Omega}^t_{\mathrm{H}_m} \cdot \mathbf{\Omega}_{\mathrm{H}_m} \cdot \mathbf{E}^0_{\mathrm{H}_m}), \quad A = Rank(\mathbf{X}). \tag{40}$$

According to the Cauchy's interlace theorem, if a row-column pair is deleted from a real symmetric matrix, then the eigenvalues of the resulting matrix interlace those of the original one [16]. That is, each eigenvalue of the resulting matrix will be between two eigenvalues of the original matrix. According to this, the eigenvalues of $\mathbf{\Omega}_{\mathrm{H}_m}$ interlace those of $\mathbf{Q}_A$ and so they lie in the interval [0,1]. Thus, the eigenvalues of $\mathbf{\Omega}'_{\mathrm{H}_m} \cdot \mathbf{\Omega}_{\mathrm{H}_m}$ also lie in the interval [0,1], so that:

$$tr((\mathbf{E}^0_{\mathrm{H}_m})' \cdot \mathbf{\Omega}'_{\mathrm{H}_m} \cdot \mathbf{\Omega}_{\mathrm{H}_m} \cdot \mathbf{E}^0_{\mathrm{H}_m}) \leq tr((\mathbf{E}^0_{\mathrm{H}_m})' \cdot \mathbf{E}^0_{\mathrm{H}_m}), \tag{41}$$

which is equivalent to $SSE^A_{\mathrm{H}_m} \leq SSE^0_{\mathrm{H}_m}$ for $A = Rank(\mathbf{X})$. The equality $SSE^A_{\mathrm{H}_m} = SSE^0_{\mathrm{H}_m}$ will hold for all variables in H_m satisfying Property 2, so that the number of variables need to be equal to the rank. Since $SSE^A_{\mathrm{H}_m} \leq SSE^0_{\mathrm{H}_m}$ for $A = Rank(\mathbf{X})$ holds for each group of variables, it also holds for the SSE^A_T, therefore proving the property.

Property 5 *The SSE of a data set (SSE^A_T) according to ekf (or in general from TRI) attainable by any PCA model is lower bounded by the sum of squares of the non-redundant information in the data.*

Proof:

Since Property 3 holds for each of the variables, it also holds for the SSE^A_T, therefore proving the property.

C Inconsistency of direct imputation

The inconsistency problem implies that the points of the space where the sum-of-squares of the error by direct estimation corresponding to observation $\mathbf{x}_n$, SSE_n^A, attains its minimum are not in the latent subspace of the first A PCs. To see this, the partial derivative of SSE_n^A for a point in that subspace is computed. For simplicity, the demonstration will be restricted to the leave-one-out case. From eq. (A3), the estimation error associated to direct estimation of the leave-one-variable-out case for the m-th element of $\mathbf{x}_n$, for $r_{n,m}^A = 0$ (i.e. the observation lies on the latent subspace), holds:

$$e_{n,m}^A = \epsilon_{n,m}^{(0)} \cdot \alpha_m^A, \tag{42}$$

then

$$\frac{\partial (e_{n,v}^A)^2}{\partial x_{n,m}} = \begin{cases} 2 \cdot \epsilon_{n,m}^{(0)} \cdot (\alpha_m^A)^2, & for \quad v = m \\ 0, & for \quad v \neq m \end{cases}, \tag{43}$$

and

$$\frac{\partial SSE_n^A}{\partial x_{n,m}} = 2 \cdot \epsilon_{n,m}^{(0)} \cdot (\alpha_m^A)^2. \tag{44}$$

For a given point to attain a minimum of SSE_n^A, the partial derivative with respect to the M variables should be 0. Therefore, a point $\mathbf{x}_n$ in the subspace of the first A PCs will not attain the minimum value of SSE_n^A exception made on two possibilities: a perfect initial estimation, i.e. $\mathbf{x}_n^{(0)} = \mathbf{x}_n$, or for null alpha values, i.e. $\alpha_m^A = 0, \forall m = 1, .., M$. Provided the first choice is possible because $\mathbf{x}_n$ is available, then the estimation error would become the reconstruction error and the *ekf* would become the *rkf*. The second choice is not possible for $A > 0$. Therefore, the direct estimation will be inconsistent for any number of variables in the data set and for any eigenvalues distribution.

The inconsistency problem can also be observed for the ratio $F_n^A = SSE_n^A / SSE_n^{A-1}$, used in

the illustration of this problem in Figure 5 of the paper for $A = 1$. The partial derivative of F_n^A for each element of $\mathbf{x}_n$ is:

$$\frac{\partial F_n^A}{\partial x_{n,m}} = \frac{(\partial SSE_n^A/\partial x_{n,m}) \cdot SSE_n^{A-1} - (\partial SSE_n^{A-1}/\partial x_{n,m}) \cdot SSE_n^A}{(SSE_n^{A-1})^2}, \tag{45}$$

which should be equal to 0 in the points where F_n^A attains its extreme values. Again for simplicity, $A = 1$ will be considered. Note that:

$$SSE_n^0 = \sum_{v=1}^{M} x_{n,v}^2 \tag{46}$$

and

$$\frac{\partial SSE_n^0}{\partial x_{n,m}} = 2 \cdot x_{n,m}. \tag{47}$$

Therefore, for those points in the first PC:

$$\frac{\partial F_n^1}{\partial x_{n,m}} = \frac{2 \cdot \epsilon_{n,m}^{(0)} \cdot (\alpha_m^1)^2 \cdot \sum_{v=1}^{M} x_{n,v}^2 - 2 \cdot x_{n,m} \cdot \sum_{v=1}^{M} (\epsilon_{n,v}^{(0)} \cdot \alpha_v^1)^2}{(\sum_{v=1}^{M} x_{n,v}^2)^2}. \tag{48}$$

Considering all partial derivatives of F_n^1 should be equal to 0 to attain a minimum value, then the following equality should hold:

$$\frac{\epsilon_{n,m}^{(0)} \cdot (\alpha_m^1)^2}{x_{n,m}} = \frac{\sum_{v=1}^{M} (\epsilon_{n,v}^{(0)} \cdot \alpha_v^1)^2}{\sum_{v=1}^{M} x_{n,v}^2} \quad \forall m \in \{1, ..., M\}. \tag{49}$$

This equality will not hold in general, except for very particular cases. For instance, if the initial estimates $x_{n,m}^{(0)}$ are set to 0 (TRI), then all alpha values should be equal to satisfy eq. (49). Similar theoretical derivations, but further more elaborated, can be performed to show that the inconsistency problem will affect the ratio F_n^A for $A > 1$.

References

[1] Jackson J.E.. *A User's Guide to Principal Components.* England: Wiley-Interscience 2003.

[2] Wold S.. Cross-Validatory Estimation of the Number of Components in Factor and Principal Components *Technometrics.* 1978;20:397–405.

[3] Eastment H.T., Krzanowski W.J.. Cross-Validatory Choice of the Number of Components From a Principal Component Analysis *Technometrics.* 1982;24:73–77.

[4] Nomikos P., MacGregor J.F.. Multivariate SPC Charts for Monitoring Batch Processes *Technometrics.* 1995;37:41–59.

[5] Bro R., Kjeldahl K., Smilde A.K., Kiers H.A.. Cross-validation of component models: a critical look at current methods. *Anal Bioanal Chem..* 2008;390:1241–1251.

[6] Wise B.M., Gallagher N.B., Bro R., Shaver J.M., Windig W., Koch R.S.. *PLSToolbox 3.5 for use with Matlab.* Eigenvector Research Inc. 2005.

[7] Nelson P.R.C., Taylor P.A., MacGregor J.F.. Missing data methods in PCA and PLS: score calculations with incomplete observations *Chemometrics and Intelligent Laboratory Systems.* 1996;35:45–65.

[8] Arteaga F., Ferrer A.. Dealing with missing data in MSPC: several methods, different interpretations, some examples *Journal of Chemometrics.* 2002;16:408–418.

[9] Arteaga F., Ferrer A.. Framework for regression-based missing data imputation methods in on-line MSPC *Journal of Chemometrics.* 2005;19:439–447.

[10] Breiman L., Friedman J.H., Olshen R.A., Stone C.. *Classification and Regression Trees.* Belmont, CA: Wadsworth 1984.

[11] Zhang P.. Model selection via multifold crossvalidation *The Annals of Statistics.* 1993;21:299-313.

[12] Louwerse D.J., Smilde A.K., Hiers H.A.L.. Cross-validation of Multiway Component Models *Journal of Chemometrics.* 1999;13:491-510.

[13] Lei F., Rotbøll M., Jørgensen S.B.. A biochemically structured model for Saccharomyces cerevisiae *Journal of Biotechnology.* 2001;88:205–221.

[14] López F., Valiente J.M., Prats-Montalbán J.M., Ferrer A.. Performance evaluation of soft color texture descriptors for surface grading using experimental design and logistic regression *Pattern Recognition.* 2008;41:1744–1755.

[15] Camacho J., Picó J., Ferrer A.. Data understanding with PCA: Structural and Variance Information plots *Chemometrics and Intelligent Laboratory Systems.* 2010;100:48–56.

[16] Mercer A.MCD., Mercer P.R.. Cauchy's Interlace Theorem and Lower Bounds for the Spectral Radius *International Journal of Mathematics and Mathematical Sciences.* 1998;23:563–566.