Cross-validation in PCA models with the element-wise *k*-fold (*ekf*) algorithm: Practical Aspects.

José Camacho [a] Alberto Ferrer [b]

[a] *Departamento de Teoría de la Señal, Telemática y Comunicaciones, Universidad de Granada, 18071, Granada, Spain*

[b] *Departamento de Estadística e Investigación Operativa Aplicadas y Calidad Universidad Politécnica de Valencia, 46022, Valencia, Spain*

Abstract

This is the second paper of a series devoted to provide theoretical and practical results and new algorithms for the selection of the number of Principal Components (PCs) in Principal Component Analysis (PCA) using cross-validation. The study is especially focused on the element-wise *k*-fold (*ekf*), which is among the most used algorithms for that purpose. In this paper, a taxonomy of PCA applications is proposed and it is argued that cross-validatory algorithms computing the prediction error in observable variables, like *ekf*, are only suited for a class of applications. A number of cross-validation methods, several of which are original, are compared in two applications of this class: missing data imputation and compression. The results show that the *ekf* is especially suited for missing data applications while other traditional cross-validation methods, those by Wold and Eastment and Krzanowski, are not found to provide useful outcomes in any of the two application. These results are of special value considering that the methods investigated are computed in the main commercial software packets for chemometrics. Finally, the choice of the missing data algorithm within *ekf* is also investigated.

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

1 Introduction

Principal Component Analysis (PCA) is a multivariate tool with application to different objectives: data understanding, anomalies detection, missing data estimation, compression, and others. To build a PCA model from a data set, one needs to select the number of Principal Components (PCs). There are plenty of methods to select this number, see the first paper of this series [1] for a review. Wold firstly proposed cross-validation for that purpose [2].

In the forest of methods to select the number of PCs, an important issue is forgotten: what is the PCA model going to be used for? PCA is a very versatile tool and depending on the application at hand, the determination of the appropriate number of PCs may be addressed in a different way. In a more theoretical perspective, the number of PCs is selected to maximize a given optimization function. That way, the chosen number can be said to be optimum for the application at hand. Different applications imply different optimization functions. Thus, using the same calibration data set, different number of PCs may be optimal for process monitoring and compression, for instance. Therefore, seeking a single approach to select the number of PCs in PCA for a general perspective [3] is an ill-defined goal.

Email address: `josecamacho@ugr.es` (José Camacho).

In this paper, a taxonomy of PCA applications is proposed. It is argued that the cross-validation approaches defined in the literature are only suited to determine the number of PCs in a specific category of applications: those where the focus is on the observable variables. Two important applications within that category are missing data estimation and compression. In this paper, the performance of several cross-validation approaches is assessed in these two application.

The paper is organized as follows: Section 2 introduces the state of the art in PCA cross-validation. In Section 3, a taxonomy of PCA applications is provided. Section 4 presents the simulation strategy for the generation of the data used in the comparisons and a real data set. Section 5 focuses on missing data applications whereas Section 6 is focused on compression. Section 7 illustrates the behavior of the cross-validation approaches in the real data set. Finally, Section 8 draws some concluding remarks.

2 Cross-validation in PCA

PCA follows the expression:

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

where $\mathbf{X}$ is a $N \times M$ matrix of data, $\mathbf{T}^A$ is the $N \times A$ score matrix containing the projection of the objects onto the A principal components (PCs) subspace, $\mathbf{P}^A$ is the $M \times A$ loading matrix containing the linear combination of the variables represented in each of the PCs, and $\mathbf{E}^A$ is the $N \times M$ matrix of residuals.

The simplest cross-validation method is the row-wise k-fold cross-validation or *rkf* ([4], through [5]). A detailed description of the algorithms can be found in [1]. In the *rkf*, the groups are arranged object-wise. Each time, a model is calibrated from the whole data-set but a group of objects. Using the model, the scores of the objects from that group are computed and their data are re-estimated using scores and loadings. Subtracting actual data from the estimates, the sum-of-squares of prediction error (PRESS) is computed. The PRESS associated to a variable m for A PCs is computed according to the following expressions:

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

$$r_{n,m}^A = x_{n,m} - \hat{x}_{n,m}, \tag{3}$$

where N is the number of objects used to compute the PRESS, $\hat{x}_{n,m}$ is the estimate of $x_{n,m}$ and $r_{n,m}^A$ is the reconstruction error.

The *rkf* method yields strictly decreasing curves of PRESS in terms of A, since the error computed within the algorithm is the reconstruction error [1]. To determine the number of PCs, a threshold can be applied. Thus, if the decrease of PRESS when adding the a-th PC is lower than the threshold, the PC is discarded and the model selected contains $a - 1$ PCs. Also, the curve can be corrected with the degrees of freedom consumed [3].

The *rkf* has been criticized because the PCA estimates are computed using the actual values as input [3]. Since there is not independence between actual values and estimates, the modelling error computed in *rkf* cannot be considered purely prediction error. Although it is not clear whether this issue has something to do with the ability of determining A, a number of cross-validatory approaches that satisfy–to a certain degree–the independence between estimates and actual values have been derived. The corresponding PRESS is obtained from the error computed between estimates and actual values following the equivalent expression to (2). Among the several cross-validatory algorithms proposed [6,3], those by Wold [2] and Eastment and Krzanowski [7] are the most cited and influent ones.

The proposal by Wold is based on the iterative computation of PCs in the NIPALS procedure [8]. Wold suggested to include PCs to the model until the following index exceeds a value of 1:

$$R^A = \frac{PRESS^A}{SSE^{A-1}} \tag{4}$$

where $PRESS^A$ is the PRESS computed for A PCs, and SSE^{A-1} is the Sum of Square Residuals after $A-1$ PCs have been extracted. The difference between PRESS and SSE is that the former is computed by cross-validation whereas the latter is computed at once from the entire data set.

Eastment and Krzanowski [7] proposed an alternative scheme based on the singular value decomposition (SVD) algorithm. To select the number of significative PCs, they propose the addition of the PCs up to the last one for which the following index exceeds the unity:

$$W^A = \frac{(PRESS^{A-1} - PRESS^A)/DF^A}{PRESS^A/DF^A_{rem}} \tag{5}$$

where DF^A is the number of degrees-of-freedom (DFs) used to fit the A-th PC and DF^A_{rem} is the remaining DFs after the A-th PC has been added to the model.

Wold [2] also suggested a possible alternative for those for which the NIPALS procedure is not available–something very unlikely nowadays. Wold did not pay very much attention to this alternative. Nonetheless, it presents an attractive feature: the PRESS curve obtained typically shows a valley-like shape [1], where the minimum of the curve signals the optimum number of PCs. This is, in principle, a logical behavior for the prediction error: decrease as the addition of PCs improves the prediction performance of the model and increase when this addition is noisy. It is also conveniently similar to the PRESS curve obtained when cross-validating regression models -e.g. for Partial Least Squares (PLS) models. This method is referred here as the the element-wise k-fold (*ekf*) method.

Bro et al. [3] compared most of the cross-validation methods which are currently used with spectral-type data. They concluded that the *ekf* generally outperforms the other methods studied. Because of this result, the first paper of this series [1] performed a detailed theoretical study entirely focused on this method. The original *ekf* proposal by Wold was the cross-validation algorithm in the first releases of the PLS_Toolbox [9]. This algorithm was based on the simplest missing data imputation method: the trimmed score imputation (TRI). The algorithm studied by Bro et al. [3] 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) [10]. In the present paper, several *ekf* variants with different missing data methods, including TRI and PMP, will be studied.

3 PCA applications

There are at least three categories of applications of PCA which should be distinguished:

a) **When the interest is in the observable variables**–that is the original variables in matrix **X** in eq. (1). One would like to select the number of PCs, A, so that the estimation of the variables with PCA is the most accurate. The objective is by itself a definition of how the number of PCs should be determined: A is selected so that the error computed by subtracting the actual value of the data from that estimated with PCA is minimum. Two types of applications within this category are possible, depending on whether the actual values that are estimated are available or not. For instance, when the objective is data compression or dimensionality reduction [11–14], probably the most frequent PCA application, the actual values are available and their estimates by PCA are employed as compressed data. On the other hand, when the data of the object is incomplete due to any problem during data collection, the actual values to estimate are not available and PCA is used to infer them [15–19]. In the following, this sort of applications will be termed Missing Data (MD) applications. Take for instance a number of temperature sensors in a chemical process. Typically, the readings of these sensors are fairly correlated and PCA can be used to develop a soft-sensor including all physical sensors. This soft-sensor may be robust to sensor failures. Thus, if one sensor breaks down during actual application, the missing reading can be recovered from the other readings. Obviously, the missing element is not used in its own estimation as it happens in compression. Notice that although the model may be built to be applied to incoming (future) data, the number of PCs is decided during model calibration with the data set at hand. During calibration, actual values are available to compute the estimation error in both compression and MD applications.

b) **When the interest is in the latent variables**. This would be the case when the objective is to interpret the model to gain data understanding [20–23]. For this purpose, it is necessary to have in mind the limitations of PCA itself. The PCA model in its traditional form can only model linear relationships[1]. Besides, systematic information does not necessarily have larger variance than the noise. In many cases, data understanding is gained by assessing the PCA models for several numbers of PCs instead of analyzing a single model. This was the approach of [21], where cross-validation by *rkf* was combined with other sources of information in the Structural and Variance Information (SVI) plots. On the other hand, there are other 2-way (Multivariate Curve Resolution) methods which are often aimed at obtaining pure components, which reflect the real underlying relationships in certain types of data. In these cases, the number of components is often known a-priori.

c) **When the interest is in the distributions in latent variables and residuals**. For most applications, matrix $\mathbf{P}^A$ in (1) is understood as the PCA model itself. This is because it is the only matrix applied to incoming (future) objects. Nonetheless, in statistical monitoring [24,25], matrices $\mathbf{T}^A$ and $\mathbf{E}^A$ in (1) are used to develop control limits for incoming data. There-

[1] Non-linearity may be an important issue also for a) and c), as well, but it is particularly problematic when meaningful interpretations are sought.

fore, $\mathbf{T}^A$, $\mathbf{E}^A$ and the limits themselves are also part of the model of the data. This should be taken into account to select A. In this type of applications, the goal is to select A so that the statistical distributions of $\mathbf{T}^A$ and $\mathbf{E}^A$ defined from the calibration data are representative of the distributions in incoming data, provided the process under analysis remains in control. Some guidelines to select A for monitoring by assessing the stability of $\mathbf{P}^A$ with *rkf* (and therefore indirectly the stability of $\mathbf{T}^A$ and $\mathbf{E}^A$) were suggested in [21], but it remains as an open issue. On the other hand, the PCA model may be developed for variability reduction (phase I) in a process using Multivariate Statistical Process Control (MSPC) techniques, which is a different objective to actual monitoring (phase II in MSPC) [26,25].

Cross-validation approaches, exception made on the *rkf*, are based on the prediction error in the sense that a specific piece of data is not used to compute its prediction. The prediction error computed by these methods is suited to select the number of PCs in the first category of applications (a)). Nonetheless, it should not be used for categories b) and c). This prediction error measures to what extent the model is able to recover missing elements, a completely different goal to data exploration, where we want to learn from the data, or monitoring, where stable loadings and residuals distributions are desired. Furthermore, as it was shown in the first paper of this series [1], the prediction based on the estimation of missing values present some features, for instance the directional dependence, which may be a problem for these applications. Reference [21] discusses some examples in which the number of PCs selected using the PRESS by cross-validation (*ekf*) is not the adequate one for data understanding (b)) and process monitoring (c)). In the remainder, the paper will focus on applications in category a) to study the performance of several cross-validatory approaches.

Category a) applications typically–although not always–follow two stages: model building and model exploitation. In model building, a set of calibration data is used to fit the parameters of the PCA model, with special emphasis on the matrix of loadings $\mathbf{P}^A$. This matrix is applied during the model exploitation to incoming objects, independent to the calibration data.

4 Data sets

Two different experiments have been performed for the study presented in this paper. Firstly, a set of data matrices obtained by simulation are analyzed. The optimum number of PCs are known a priori and the data matrices are contaminated with measurement noise of different magnitudes. Secondly, a real data set used in both the original papers by Wold [2] and Eastment and Krzanowski [7] is analyzed. The data set corresponds to the gas chromatography retention index matrix published by McReynolds [27].

4.1 Data generation for the simulation study

Four simulated data matrices will be used for comparison in this paper. Differently to the approach of [3], here it was preferred to handle a limited number of data sets in order to be able to interpret the results from their specific features. In all the cases, a number of latent variables (LVs) are simulated and from them a set of observable variables (OVs) are computed. These OVs represent the registered variables in a practical application. Thus, only the OVs can be used to determine the number of PCs.

The four data sets have a very different nature, with different number of LVs and OVs. The LVs are generated independently at random following a normal distribution with zero mean and unit variance. Each OV is obtained from one LV or as a linear combination of several LVs. For simplicity, all OVs are computed so that they have zero mean and unit variance.

The noise-free variables in the four data sets are generated according to Table 1. All data sets contain 100 objects. The four data set are examples of different correlation structures. Thus, for instance, the first data set may be similar to typical data sets composed of process variables (such as temperatures, pressures, etc.) while the fourth data set resembles spectral-like data sets.

Measurement noise is generated independently for each OV and at random, following a normal distribution with zero mean. The standard deviation used depends on the percentage of noise chosen to be added to the data sets:

$$x'_i = (x_i + (\sqrt{\sigma_n}) \cdot n)/(\sqrt{1 + \sigma_n})$$

where x'_i is the contaminated OV, x_i the noise-free OV, σ_n the standard deviation of the noise and n the noise generated. The data sets are corrupted with noise for $5\%, 10\%, 15\%, 20\%$ and 25% noise percentages, where percentages are computed so that the lowest standard deviation of a LV is the 100%. Thus, σ_n equals $0.05, 0.1, 0.15, 0.2$ and 0.25, respectively.

A simple initial analysis can be performed to assess how the PCA subspace is affected by the addition of noise in the simulated data. In the first data set, a 4 PCs model is computed for each of the data sets corrupted with noise from 5% to 25%, obtaining loadings matrices $\{P_5 ... P_{25}\}$. A 4 PCs model is also

First Data Set
$x_i = (\sqrt{i/5}) \cdot lv_1 + (\sqrt{1-i/5}) \cdot lv_2, i\epsilon\{1,..,5\}$
$x_i = (\sqrt{0.5}) \cdot lv_1 + (\sqrt{i/10-0.5}) \cdot lv_2 + (\sqrt{1-i/10}) \cdot lv_3, i\epsilon\{6,..,9\}$
$x_{10} = ((\sqrt{0.01}) \cdot lv_1 + (\sqrt{0.01}) \cdot lv_2 + (\sqrt{0.01}) \cdot lv_3 + lv_4)/\sqrt{1.03}$
Second Data Set
$x_i = (\sqrt{0.5}) \cdot lv_j + (\sqrt{0.5}) \cdot lv_k, i\epsilon\{1,..,6\}, j \neq k\epsilon\{1,..,4\}$
$x_i = (\sqrt{0.5}) \cdot lv_j + (\sqrt{0.5}) \cdot lv_k, i\epsilon\{7,8,9\}, j \neq k\epsilon\{5,6,7\}$
$x_{10} = lv_8$
Third Data Set
$x_i = lv_i, i\epsilon\{1,..,12\}$
$x_i = (\sqrt{0.5}) \cdot lv_j + (\sqrt{0.5}) \cdot lv_k, i\epsilon\{13,..,27\}, j \neq k\epsilon\{1,..,6\}$
Fourth Data Set
$x_i = (\sqrt{0.5}) \cdot lv_j + (\sqrt{0.5}) \cdot lv_k, i\epsilon\{1,..,45\}, j \neq k\epsilon\{1,..,10\}$
$x_{46} = lv_{11}, x_{47} = lv_{12}$
$x_{48} = (\sqrt{0.5}) \cdot lv_{11} + (\sqrt{0.5}) \cdot lv_{13}$
$x_{49} = (\sqrt{0.5}) \cdot lv_{12} + (\sqrt{0.5}) \cdot lv_{14}$
$x_{50} = lv_{15}$

Table 1

Generation of the observable variables in the data sets. x_i stands for the i-th observable variable and lv_j stands for the j-th latent variable.

computed for the noise-free original data representing the true LVs. Then, each of the eigenvectors corresponding to the LVs are projected onto matrices $\{P_5 ... P_{25}\}$ and the percentage of sum-of-squares captured by the subspace is computed, so that 100% means perfect matching, that is the LV is within the subspace of the PCA model from corrupted data, and 0% means non-correlation at all, i.e. the LV is orthogonal to the subspace. The results are shown in Table 2. The PCs subspace remains almost unaltered for all noise percentages, since the biggest amount of variance lost in a LV is lower than a 7%. This experiment was repeated for the other data sets. In the second and the third data sets (numerical results not shown), the amount of lost variance per LV is also low (less than a 4% and a 10%, respectively). Nonetheless, as shown in Table 3, in the fourth data set the amount of variance lost in some LVs is high and it is specially important for the last LV.

Table 2

Percentage of sum-of-squares of the eigenvectors corresponding to the true 4 latent variables captured by the first 4 PCs for different percentages of noise in the first simulated data set.

	Eigenvalues	5%	10%	15%	20%	25%
lv_1	732.7	99.9383	99.9447	99.8751	99.6825	99.8124
lv_2	101.7	99.9635	99.8324	98.3473	98.9402	98.5995
lv_3	64.8	99.7517	99.6387	98.0639	93.2534	94.4690
lv_4	41.9	99.0927	97.6381	95.0962	94.8524	97.1282

4.2 McReynolds Data Set

The data set used here is the same of [7]. It contains 225 objects with 10 variables each. One of the 226 original objects presented in [27] was elimi-

Table 3

Percentage of sum-of-squares of the eigenvectors corresponding to the true 15 latent variables captured by the first 15 PCs for different percentages of noise in the fourth simulated data set. Cases with more than a 10% of lost variance in bold.

	Eigenvalues	5%	10%	15%	20%	25%
lv_1	932.7	99.8655	99.4436	99.3449	99.3505	98.8665
lv_2	686.9	99.7152	99.5208	99.0184	98.9959	99.0173
lv_3	566.4	99.8139	99.3512	99.2420	98.7584	97.6257
lv_4	537.1	99.5989	99.4411	98.7374	98.7159	97.2403
lv_5	384.0	99.5886	98.9522	99.0641	97.9361	96.9172
lv_6	324.4	99.6424	98.8730	98.3711	98.2257	97.3564
lv_7	302.3	99.1420	98.2568	97.6809	98.3862	97.5041
lv_8	280.8	99.4415	98.7593	98.0924	97.5675	96.7161
lv_9	259.8	99.4100	98.7776	97.4315	96.9258	94.9566
lv_{10}	208.0	99.1831	98.3904	98.0796	95.9656	96.8327
lv_{11}	132.9	98.9694	97.2807	94.5677	92.8286	92.3113
lv_{12}	106.0	98.0947	96.0115	95.2737	92.0951	93.6385
lv_{13}	77.0	97.2661	95.4972	90.0593	**86.3426**	93.2747
lv_{14}	27.8	94.3743	**81.9770**	**77.0285**	**69.3470**	**72.3900**
lv_{15}	18.2	90.9541	**78.5135**	**82.0575**	**29.4430**	**47.3982**

nated from the data due to incompleteness. As in [2] and [7], the data was analyzed with and without outliers (a total of 13 outliers are found by Wold and Andersson [29]).

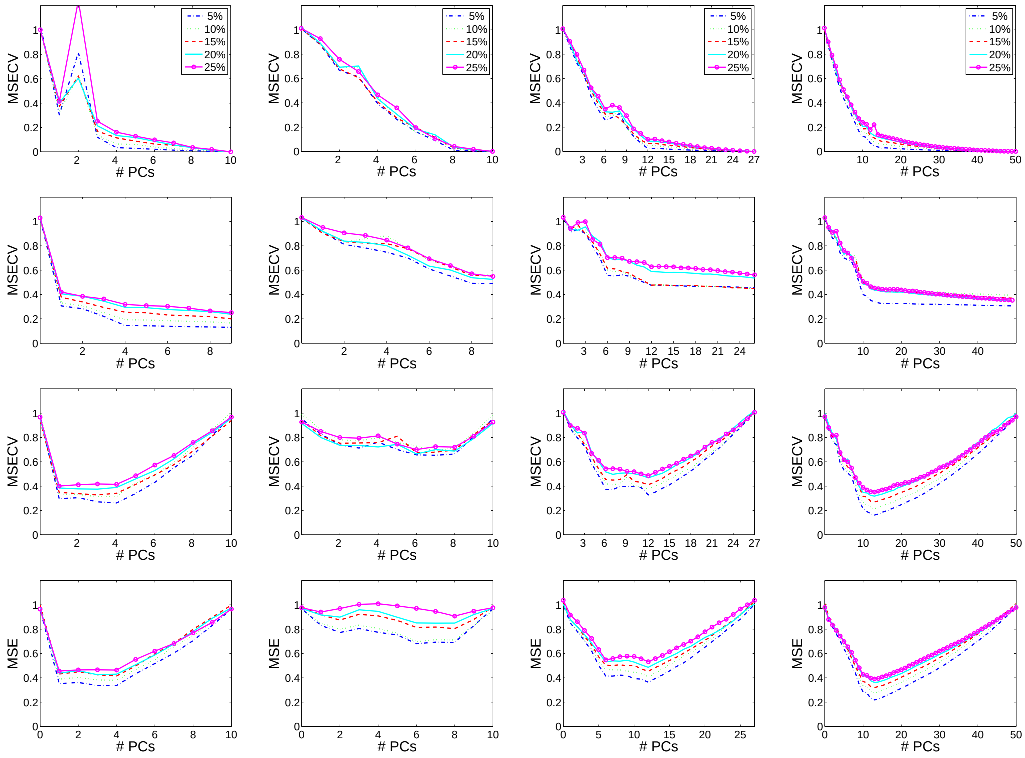
5 Missing data (MD) applications

When the PCA model is going to be used for the estimation of missing values in incoming data, cross-validation provides an estimate of the optimum number of components A and of the expected estimation error. As already commented, there are several cross-validation algorithms, among which the *ekf* and the approaches by Wold [2] and Eastment and Krzanowski [7] were previously highlighted. Figure 1 compares the estimation error computed by these three methods with the true estimation error at model exploitation in the simulated data. The latter is computed for each of the four simulated data sets as follows. Firstly, a PCA model is built using the calibration data, with 100 observations. Then, a new data set with 1000 observations, representing data during model exploitation, is generated according to Table 1. Finally, a 10% of missing values is introduced and recovered using the PCA models. From the actual and estimated values, the estimation error is computed. All the plots in Figure 1 show the PRESS normalized by the number of missing elements estimated, specified as Mean Squared Error (MSE):

$$MSE^A = \frac{1}{N_m} \cdot \sum_{n=1}^{N_m} (e_n^A)^2, \tag{6}$$

where N_m is the number of missing elements considered and e_n^A is the estimation error.

Figure 1 shows that the *ekf* provides a very accurate prediction of the estimation error at model exploitation in a MD application, clearly outperforming the other two approaches (Wold [2] and Eastment and Krzanowski [7]). This result is expectable since the *ekf* algorithm, detailed in the first paper of this series [1], perfectly resembles the two steps procedure of model building and model



(a) First data set (4/10)

(b) Second data set (8/10)

(c) Third data set (12/27)

(d) Four data set (15/50)

Fig. 1. Mean Squared Error by Cross-validation (MSECV) and for incoming data (MSE) in the data sets of Table 1. Data are corrupted with 5% (dashdot line), 10% (dotted line), 15% (dashed line), 20% (solid line) and 25% (solid line with circles) of measurement noise. The first three rows show the MSECV computed with the cross-validation by Wold [2] (first row), Eastment and Krzanowski [7] (second row) and *ekf* (third row). The fourth row presents the MSE at model exploitation.

exploitation. The algorithm presents three nested loops to iterate through the PCs, the observations and the variables. On the one hand, the second nested loop performing row-wise cross-validation resembles the fact that the missing data estimation will be carried out for objects during model exploitation, different to those used in model calibration. On the other hand, the inner loop performing variable-wise cross-validation resembles the fact that the objects during model exploitation will be incomplete and some variables will need to be estimated from the others. The approaches by Wold [2] and Eastment and Krzanowski [7] do not resemble the logical procedure in MD applications and because of that they do not provide good estimations of the error in model exploitation. As a consequence, they are not suited to select the number of PCs in MD applications.

5.1 MD methods within the ekf

The *ekf* method makes use of the missing data method referred here as direct estimation (see the first paper of this series [1]). This was also the method to estimate the missing elements employed at model exploitation in the results shown in Figure 1. Direct estimation is well known to provide a sub-optimal estimation solution, as shown in [18] where it is referred as trimmed score imputation (TRI). Since other missing data estimation methods provide more accurate estimates than direct estimation, it is sensible to use these approaches instead of direct estimation in MD applications. As it was previously shown, the cross-validation method should resemble the procedure followed in MD applications to provide adequate estimations of the error during model exploitation. Therefore, if a different estimation method is used during model exploitation, the *ekf* needs to be modified accordingly. In this paper, the extension of *ekf* to three MD methods is considered: iterative imputation [28], Projection to Model Plane (PMP) [15] and Trimmed Score Regression (TSR) [18]. The last method is representative of regression-based imputation methods [19], some of which were also presented in [15].

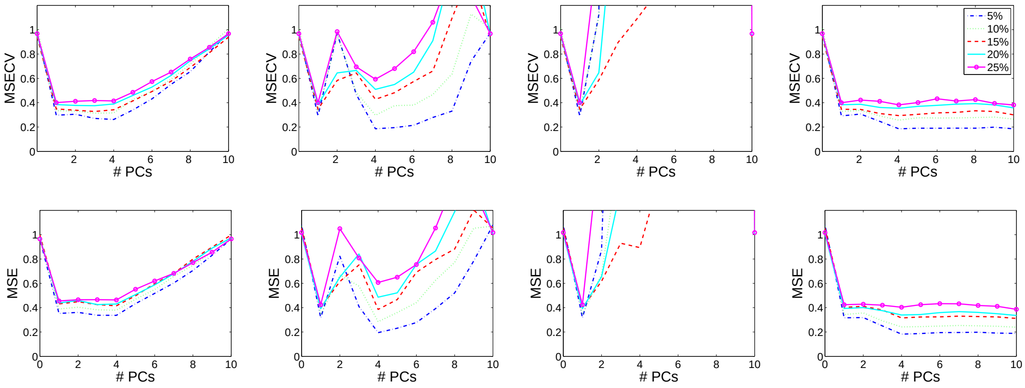
The extension to incorporate iterative imputation to the inner loop of the *ekf* algorithm is presented in Algorithm 1. The basic idea is to iterate the

```
For each PC ($A = 1...A_{max}$)
    For each group of objects ($g = 1...G$)
        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 ($h = 1...H$)
            Set $\mathbf{X}_{\#,h} = 0$
            Repeat until $\mathbf{X}_{\#,h}$ converges
                $\mathbf{T}_\#^A = \mathbf{X}_\# \cdot \mathbf{P}_*^A$
                $\hat{\mathbf{X}}_\# = \mathbf{T}_\#^A \cdot (\mathbf{P}_*^A)^t$
                $\mathbf{X}_{\#,h} = \hat{\mathbf{X}}_{\#,h}$
            end
            Restore its actual value to $\mathbf{X}_{\#,h}$
            $\mathbf{E}_{g,h}^A = \mathbf{X}_{\#,h} - \hat{\mathbf{X}}_{\#,h}$
        end
    end
    Combine matrices $\mathbf{E}_{g,h}^A$ in $\mathbf{E}^A$
    $PRESS^A = \sum_{n=1}^{N} \sum_{m=1}^{M} (e_{n,m}^A)^2$
end
```

Algorithm 1

Element-wise k-fold (*ekf*) algorithm based on iterative imputation.

estimation until convergence in the core of the algorithm. A discussion on the differences between direct and iterative estimation, geometric interpretation, and the proof of convergence is presented in the Appendix. It is interesting to note that the incorporation of the iterative estimation in *ekf* solves the



(a) Direct (b) Iterative [28] (c) PMP [15] (d) TSR [18]

Fig. 2. Mean Squared Error by Cross-validation (MSECV) and for incoming data (MSE) in data set 1 of Table 1. Data are corrupted with 5% (dashdot line), 10% (dotted line), 15% (dashed line), 20% (solid line) and 25% (solid line with circles) of measurement noise.

problem of inconsistency reported in the first paper of this series [1]. This is also further developed in the Appendix. Similarly as in Algorithm 1, the PMP and TSR estimation methods were introduced in the core of *ekf* algorithm (not shown).

In Figure 2, the MSE obtained in the estimation of missing elements by direct estimation, iterative estimation, PMP and TSR during model exploitation in the first simulated data set is compared to the MSE by *ekf* cross-validation using the corresponding missing data method in the core. The figures shows that the *ekf* yields good estimations for different missing data methods provided the corresponding method is employed in the core of the algorithm. Furthermore, the *ekf* algorithm may be useful to select, during model calibration, the best missing data method for the given correlation structure in a data set. Thus, according to the results of the *ekf* in Figure 2, it may be concluded that TSR is the preferred method for the first simulation data set. In TSR, if the number of PCs is overestimated, the imputation of missing data is not worsened in an important degree. Therefore, this overestimation is not an issue when TSR is used. This is a desirable feature since the use of an inadequate (too high) number of PCs may not affect dramatically the estimation in MD applications. The other approaches do not present such a nice feature, and the PRESS tends to increase fast for too high number of PCs. Thus, the model with 1 PC would be appropriate for a MD application based on any of the estimation methods, attaining a similar estimation error. The model with 4 PCs is only appropriate if direct estimation or TSR are used. If higher numbers of PCs are used in the model, only TSR is appropriate. The fact that direct estimation and especially TSR can be used for higher numbers of PCs shows that these estimation methods are less affected by the presence of noise than the other considered.

Let us further investigate the performance of the missing data methods. In Table 4, the minimum PRESS attained by *ekf* with the different imputation methods (direct, iterative, PMP and TSR) is shown. The best approach (lowest PRESS) for each case is highlighted in bold numbers. According to the results, the iterative estimation does not generally outperform direct estimation in the simulation examples, but rather tends to inflate the error. Iterative estimation and PMP are numerically equivalent except in specific cases (see the Appendix), as happens in the example in Figure 2. In those cases the error by PMP is highly inflated. TSR yields the best outcomes in Table 4, and also the best suited PRESS curve as shown before (Figure 2(d)).

The PRESS curves by *ekf* using direct estimation, iterative estimation, PMP and TSR are compared for the real data set in Figure 3 and the minimum PRESS values are listed in Table 5. The data set with and without outliers gives very similar results in all the approaches. In this case, the iterative esti-

Table 4

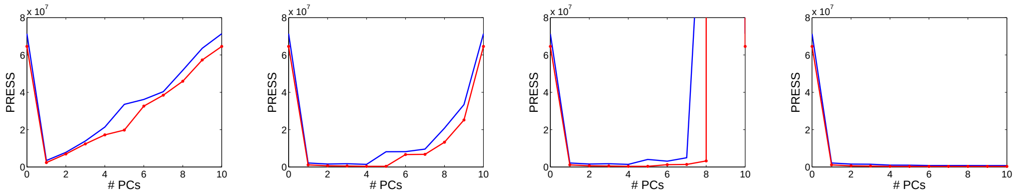
Minimum PRESS in the imputation of missing data for different approaches. The number of true latent and observed variables of the simulated data sets in parenthesis.

Noise	First data set (4/10)				Second data set (8/10)			
	Direct	Iterative	PMP	TSR	Direct	Iterative	PMP	TSR
5%	262	186	291	**185**	654	492	793	**440**
10%	313	297	329	**256**	688	710	829	**537**
15%	328	346	346	**294**	675	810	810	**564**
20%	375	383	383	**355**	665	794	794	**566**
25%	401	400	400	**382**	701	867	867	**645**

Noise	Third data set (12/27)				Fourth data set (15/50)			
	Direct	Iterative	PMP	TSR	Direct	Iterative	PMP	TSR
5%	884	878	878	**873**	814	784	784	**692**
10%	**1.017**	1.037	1.037	1.030	1.083	1.076	1.076	**981**
15%	**1.111**	1.147	1.147	1.138	1.351	1.356	1.356	**1.297**
20%	**1.264**	1.355	1.355	1.310	1.590	1.653	1.653	**1.574**
25%	**1.314**	1.461	1.461	1.408	**1.756**	1.835	1.835	1.760

mation and PMP yield a better missing data estimation performance than the direct estimation. PMP shows instability at the ninth PC, but the iterative estimation does not. In this data set, TSR shows the same good performance than in the first simulated data set: the overestimation of the number of PCs has no impact on the estimation performance and the minimum PRESS attained is lower than that of the other approaches.

Concluding, the *ekf* algorithm is suggested for the choice of the MD algorithm,



(a) Direct (b) Iterative [28] (c) PMP [15] (d) TSR [18]

Fig. 3. PRESS curves in the McReynolds data set (1970): complete data set (solid line) and reduced data set (solid line with circles).

Table 5

Minimum PRESS in the imputation of missing data for different approaches.

Data set	Results			
	Direct	Iterative	PMP	TSR
Complete	3.47×10^6	1.39×10^6	1.40×10^6	$\mathbf{0.77 \times 10^6}$
Without outliers	2.38×10^6	0.37×10^5	0.37×10^6	$\mathbf{0.30 \times 10^6}$

the number of PCs, and to estimate the MSE error of data during model exploitation. Iterative estimation and PMP are numerically equivalent except in specific cases where PMP tends to become unstable. Therefore, the PMP imputation is not suggested for MD applications. TSR gave very good results in both simulated and real data.

6 Compression

Outside the chemometrics field, PCA has been mainly employed in data compression [11–14]. Typically, PCA is applied as a preprocessing step for dimension reduction prior to other costly computations. The aim in PCA is to maximize the amount of useful information captured by a reduced number of PCs, and again this number needs to be selected. Also, if possible, any type of noise should be left in the residuals and discarded. As it was already commented, compression is different to MD applications in the fact that the objects are, in principle, complete at both model building and model exploitation.

In the first paper of this series [1], the valley shape of the PRESS curve by *ekf* was rationalized from the content of redundant, i.e. shared, information and non-redundant information in the variables. The influence of non-redundant information in the PRESS curve is three-fold: a) the total amount of non-redundant information establishes a minimum for the PRESS, b) non-redundant information captured by the model from variables with a (previously captured) portion of redundant information make the PRESS to increase and c) non-redundant information in variables with no content of redundant information barely affects the PRESS curve. As discussed in [1], effect b) has the nice consequence that the independent measurement noise tends to cause an increment of PRESS. This is because measurement noise usually presents less variance than structured information and so the latter is incorporated first into the model. Unfortunately, due to effect c), the PRESS curve by *ekf* is barely influenced by variables solely composed of non-redundant information. Thus if *ekf* is used to determine the number of PCs in the model for compression, there is a potential risk of losing the information in non-redundant variables. Notice that one of these variables may be relevant for the final application for which data is compressed (for instance, classification). This limitation of *ekf* is shared by all cross-validation methods based on the estimation of prediction error, including all the modified versions of *ekf* studied in the previous section and the approaches of [2] and [7].

The described particular behavior of the PRESS by *ekf* is related to the fact that the estimation error depends on structural parameters of the model. Since the error depends on these parameters, the same distribution of the scores may lead to different PRESS curves for different directions of the PCs. This was referred to as the directional dependence problem of *ekf* in [1]. It was also shown that the directional dependence problem can be solved by using *rkf* instead of *ekf* and this may be a valid alternative for compression. Nonetheless, as discussed in the previous paragraph, the directional dependence has the consequence that the independent measurement noise does not affect the PRESS, which results in a better selection ability of the number of PCs in the presence of noise. Therefore, it would be interesting to define a modification of the *ekf* algorithm that considers variables composed of non-redundant information, avoiding effect c) in the previous paragraph, while maintaining the capability of measurement noise detection, effect b). Although this seems to be contradictory, it is possible to a certain degree by augmenting the matrix of data X with redundant information. However, the direct duplication of the data in matrix in $\mathbf{X}_{aug} = [\mathbf{X}, \mathbf{X}]$ also duplicates the measurement noise, reducing the ability of *ekf* to filter it out in the PRESS curve. An alternative and very promising approach is to use the information of the PCA model itself (the scores) in the duplication. This information has been filtered with the PCA model, so that most of the noise has been subtracted from the data when the model has only significant PCs. For A PCs, the matrix of data can be augmented in two similar ways:

$$\mathbf{X}_{aug}^{A} = [\mathbf{X}, \mathbf{T}^{A}] \tag{7}$$

$$\mathbf{X}_{aug}^{A} = [\mathbf{X}, \mathbf{T}^{A} \cdot (\mathbf{P}^{A})^{t}] \tag{8}$$

The first approach adds A columns to the data matrix to obtain the PRESS for A PCs. Thus, an additional column is added to the augmented matrix every time a new PC is added to the model. The second approach doubles the size of the data matrix. Although not exactly equal, these are very similar approaches, being the first one preferred because the total number of variables in $\mathbf{X}_{aug}$ is lower and so it is the processing time. It should be taken into account that if $\mathbf{X}$ is scaled inside the cross-validation procedure and (7) is used, $\mathbf{T}^{A}$ must not be scaled. This is because the scale in $\mathbf{T}^{A}$ is proportional to the relevance of the PCs in terms of variance and so this information must be preserved.

The idea of using an augmented matrix of data and the *ekf* cross-validation procedure has been implemented in the corrected element-wise k-fold (*cekf*) algorithm in Algorithm 2. Following the same procedure explained in [1], an efficient version of the *cekf* algorithm can be derived. Due to data duplication, there is only redundant information in the data for the *cekf* algorithm when considering a PCA model with full rank. This has relevant consequences with respect to the properties of *ekf* introduced in the first paper of this series [1]. First, Property 2 states that the estimation error in *ekf* of a variable not in the span of the other variables for a PCA model with full rank is equal to the error in the initial estimation. This property does not hold for *cekf*, because there is no variable out of the span of the rest in the augmented matrix. Also, Property 3 states that the PRESS of a variable according to *ekf* is lower bounded by the sum of squares of the non-redundant information in that variable. Thus, there is always a minimum attainable for the PRESS by *ekf*. On the contrary, the minimum PRESS attainable by *cekf* is 0, since all non-redundant information is transformed to redundant. In Figure 4, a hypothetical example of PRESS curve by *ekf* is compared with its counterpart by *cekf*. The non-redundant significative information in the PCA model can be effectively detected by

```
For each PC ($A = 1...A_{max}$)
    For each group of objects ($g = 1...G$)
        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 $\mathbf{T}_*^A$
        $\mathbf{T}_\#^A = \mathbf{X}_\# \cdot \mathbf{P}_*^A$
        $\mathbf{X}_{*,aug}^A = [\mathbf{X}_*, \mathbf{T}_*^A]$, remember not to scale $\mathbf{T}_*^A$
        Fit a PCA model from $\mathbf{X}_{*,aug}^A$, obtaining $\mathbf{P}_{*,aug}^A$ and $\mathbf{T}_{*,aug}^A$
        For ēach group of variables ($h = 1...H$)
            Set $\mathbf{X}_{\#,h} = 0$
                $\mathbf{X}_{\#,aug}^A = [\mathbf{X}_\#, \mathbf{T}_\#^A]$
                $\mathbf{T}_{\#,aug}^A = \mathbf{X}_{\#,aug}^A \cdot \mathbf{P}_{*,aug}^A$
                $\hat{\mathbf{X}}_{\#,aug}^A = \mathbf{T}_{\#,aug}^A \cdot (\mathbf{P}_{*,aug}^A)^t$
            Restore its actual value to $\mathbf{X}_{\#,h}$
            $\mathbf{E}_{g,h}^A = \mathbf{X}_{\#,h} - \hat{\mathbf{X}}_{\#,h}$
        end
    end
    Combine matrices $\mathbf{E}_{g,h}^A$ in $\mathbf{E}^A$
    $PRESS^A = \sum_{n=1}^N \sum_{m=1}^M (e_{n,m}^A)^2$
end
```

Algorithm 2

Corrected element-wise k-fold (*cekf*) algorithm.

cekf. The adequate number of PCs is underestimated by *ekf* because the non-redundant variables are not taken into account. It should be noted that *cekf*

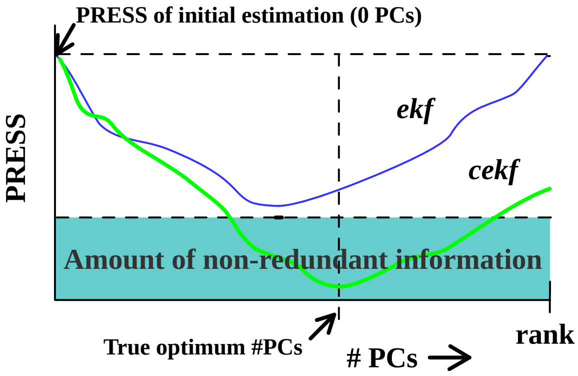


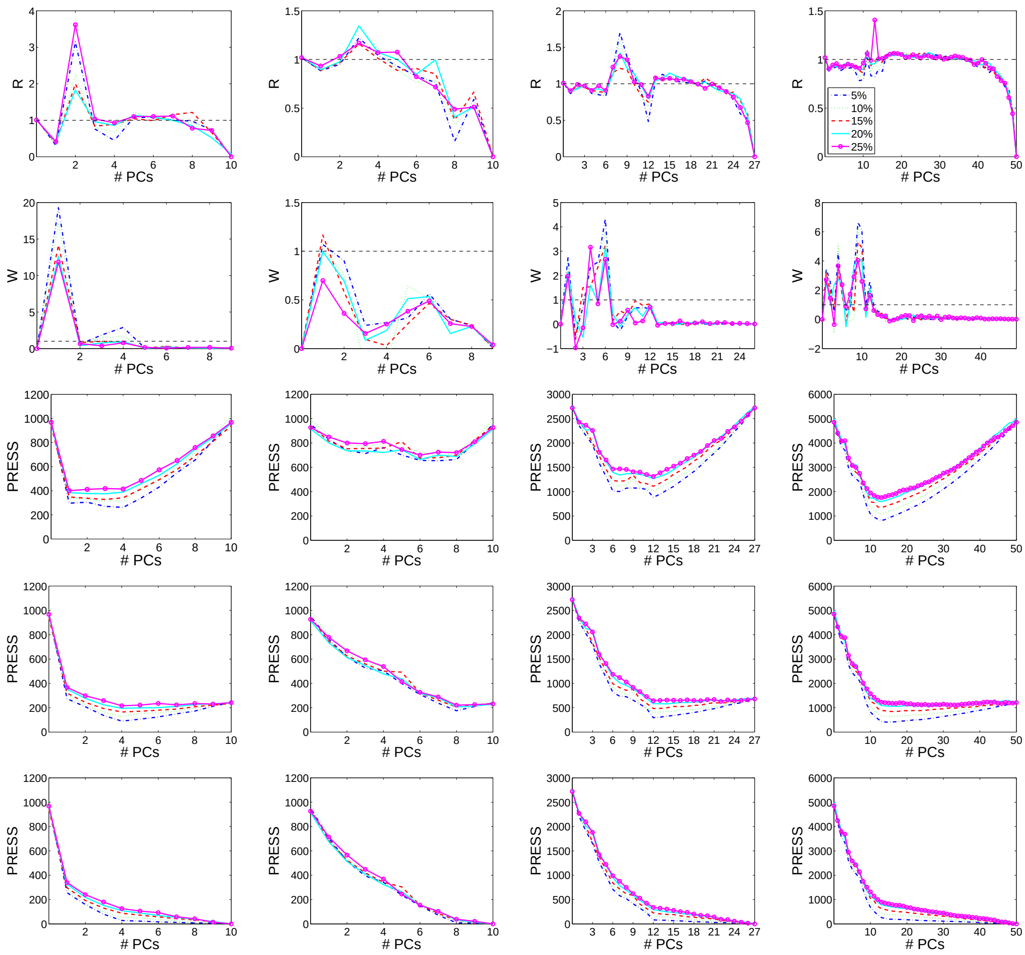
Fig. 4. Example of PRESS computed by *ekf* and *cekf*. The $\#PCs$ of minimum PRESS change since *cekf* takes into account the non-redundant information.

is suited for compression but not for MD applications.

6.1 Simulated data sets

Now let us compare some of the cross-validation methods in the determination of the number of PCs for compression. The simulation data sets generated according to Table 1 will be used in the comparison. The goal is to capture the main part of the structural data, leaving as much noise as possible in the residuals. Therefore, it will be assumed that the optimum number of PCs for compression is the number of LVs used in the data generation of the data sets (in Table 1).

In Figure 5, the outcomes of the different approaches considered, namely the R-statistic by Wold [2], the W-statistic by Eastment and Krzanowski [7] and the PRESS by *ekf*, *cekf* and *rkf*, are shown. The R-statistic and W-statistic are very irregular. This irregularity often causes the methods to arrive to the stopping rule for a number of PCs lower than the appropriate one. The PRESS by *ekf* reflects a steady behavior when the sources of variability captured in the PCs are not shared by several variables (i.e. they do not contain redundant



(a) First data set (4/10) (b) Second data set (8/10) (c) Third data set (12/27) (d) Four data set (15/50)

Fig. 5. Different approaches for the selection of the number of PCs in the data sets of Table 1. Data are corrupted with 5% (dashdot line), 10% (dotted line), 15% (dashed line), 20% (solid line) and 25% (solid line with circles) of measurement noise. The different methods are the R-statistic by Wold [2] (first row), the W-statistic by Eastment and Krzanowski [7] (second row) and the PRESS by *ekf* (third row), *cekf* (fourth row) and *rkf* (fifth row).

information). This is mainly observed in the PRESS curves for the first three data sets. This is solved in *cekf* at the expense of reducing the increasing tendency of the PRESS in the lasts PCs. The minimum of the PRESS by *cekf* detects the true number of latent variables in most of the cases. Finally, the PRESS curves by *rkf* are also reflecting the true number of PCs. This is visually apparent at least for the first and the third data sets.

Table 5

Number of PCs detected by different approaches. The number of true latent and observed variables of the simulated data sets in parenthesis.

Noise	First data set (4/10)				Second data set (8/10)			
	R	W	*ekf*	*cekf*	R	W	*ekf*	*cekf*
5%	1	**4**	**4**	**4**	2	1	7	**8**
10%	1	**4**	3	**4**	2	1	7	**8**
15%	1	1	3	**4**	2	1	6	**8**
20%	1	1	3	**4**	2	0	6	**8**
25%	1	1	1	**4**	1	0	6	**8**

Noise	Third data set (12/27)				Fourth data set (15/50)			
	R	W	*ekf*	*cekf*	R	W	*ekf*	*cekf*
5%	6	6	**12**	**12**	10	12	13	**15**
10%	6	6	**12**	**12**	10	12	13	**15**
15%	6	6	**12**	**12**	10	12	13	16
20%	6	6	**12**	**12**	10	12	13	16
25%	6	6	**12**	**12**	10	12	13	17

In Table 5, the number of PCs estimated by the different approaches considered, except for *rkf* [2], is shown. The correct estimations are highlighted in bold numbers. The proposed method *cekf* outperforms the other approaches, esti-

[2] The number of PCs selected by *rkf* is not included in Table 5 since a decision rule (a threshold) needs to be defined and this would dramatically affect the performance of the method and compromise the objectivity of the comparison.

mating the true number of components even for high noise percentages. The other approaches tend to underestimate the number of PCs. For the fourth data set, *cekf* overestimates the number of PCs for noise percentages larger than 10%. Nonetheless, Table 3 shows that for those noise percentages, there are latent variables with a high portion of their variance in the residuals. It is therefore difficult to state whether the true number of PCs should be 15 or else more components are necessary.

It should be noted that all the simulated data sets present a favorable situation, in which the percentage of measurement noise in each variable is lower than a 25% of the minimum amount of variability of a latent variable. Nonetheless, in real data sets, true latent variables may be masked by a higher level of noise or, more often, variables may not be related in a perfectly linear way. Therefore, we can always find situations in which PCA may not separate so nicely structural information from noise, and *ekf* or even *cekf* do not provide an adequate number of PCs. Still, in these situations, the number of PCs may be selected according to practical considerations. For instance, imagine we apply PCA for dimension reduction prior to a non-linear modelling task, e.g. using a support vector machine (SVM). The number of PCs may be selected as a compromise solution between the reduction of PRESS and the added complexity to the SVM when including an additional PC. For such an application, the PRESS curves by *ekf* or *cekf* may be misleading, since they are the result of a complex combination of structural information and variance captured by the model. Thus, a simpler PRESS curve, like that provided by the *rkf* method, is suggested. The PRESS by *rkf* is easier to understand than the PRESS by the other methods, and *rkf* is not directional dependent [1]. In *rkf*, the PRESS is only reflecting the amount of variance the model will capture in future objects and this idea can be easily combined with other considerations. For instance, the PRESS by *ekf* may be misleading if a practical consideration is to capture a certain percentage of information of each variable, whereas this is straightforward with *rkf*. Eventually, the number of PCs in compression should also be decided upon the final goal of the application. For instance, if the SVM is trained as a classifier, the classification figures of merit, computed in a *rkf* cross-validatory fashion, are used to determine the number of PCs.

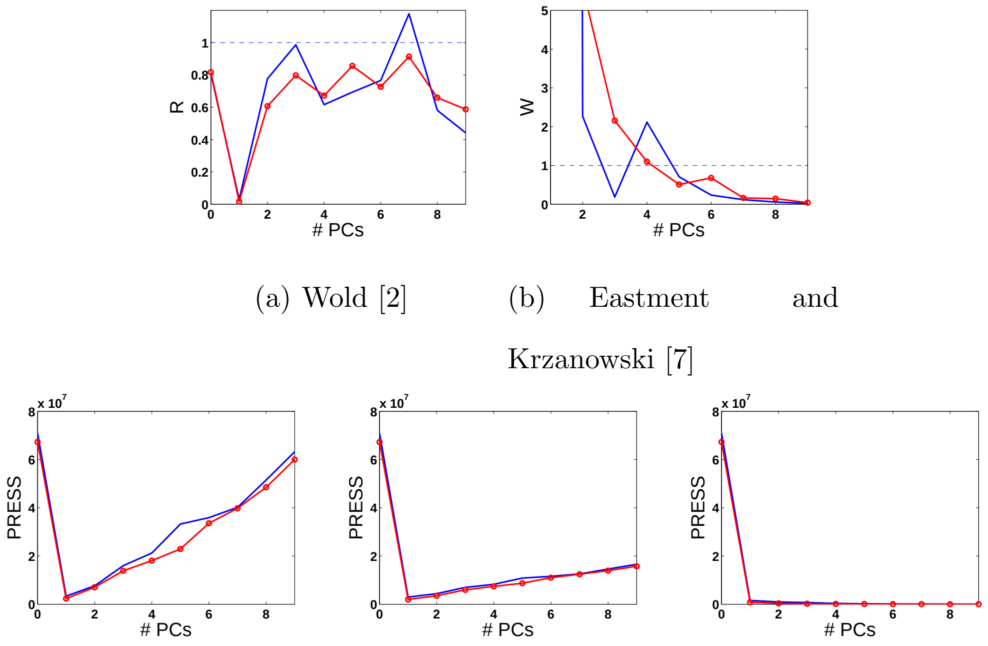
Let us return to the McReynolds data set. The results regarding the selection of the number of PCs are presented in Figure 6 and Table 6. Some small differences in the R-statistics presented here with those published by Wold are observed, probably caused by differences in the selection of the groups of the cross-validation and in the treatment of the data used. Nonetheless, note that the differences are negligible and that the number of PCs in Table 6 are the published ones.

Table 6

Number of PCs detected by different approaches in the McReynolds data set (1970).

	R	W	*ekf*	*cekf*
Full data	2	2	1	1
Reduced data	5	3	1	1

Figure 6 shows that the R and W statistics are affected by the presence of the 13 outliers. The shape of both statistics is different when computed with and without outliers. This was commented in the original papers. Whereas Wold stated that the optimum number of PCs was masked because of the outliers, Eastment and Krzanowski stated that in both cases the same number of PCs would be determined with their approach.



(a) Wold [2] (b) Eastment and Krzanowski [7]

(c) *ekf* (d) *cekf* (e) *rkf*

Fig. 6. Different approaches for the selection of the number of PCs in the McReynolds data set (1970): complete data set (solid line) and reduced data set (solid line with circles).

In the case of the *ekf*, *cekf* and *rkf* the elimination of the outliers did not lead to a significative change in the shape of PRESS. At least not for the only one significant PC found. The minimum value of W also points out one PC. Although 1 PC seems to be an adequate choice, care should be taken since real data do not necessary have to meet PCA modelling assumptions -i.e., that the information is hidden in the form of latent, linear combinations of variables- as simulated data was imposed to.

7 Conclusion

This is the second paper of a series devoted to provide theoretical results and new algorithms for the selection of the number of Principal Components (PCs) in Principal Component Analysis (PCA) using cross-validation. The first paper of this series [1] was focused on the theoretical study of the element-wise k-fold (*ekf*) cross-validation, which is among the most used algorithms to select the number of PCs in PCA and is included in a widely used commercial software packet: the PLS_Toolbox. Theoretical results showed that the use of *ekf* is a bad practice from a general perspective.

In the present paper, it is argued that the appropriate number of PCs for the same calibration data should be selected differently depending on the application the PCA is used for. A taxonomy with three categories of applications of PCA is proposed: those focused a) on the observable variables, b) on the latent variables and c) on the distributions of latent variables and residuals. Cross-validatory algorithms computing the prediction error in observable variables, like *ekf*, are only suited for the first category. Two applications within category a), missing data estimation and compression, are considered in this paper. A number of cross-validation methods, several of which are original, are compared using simulated data.

The results show that the *ekf* is suited for missing data applications. The original *ekf* proposal 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. In this paper, the *ekf* algorithm is also extended to other missing data methods, namely iterative estimation, Projection to Model Plane (PMP) and Trimmed Score Regression, the latter being preferred. The algorithm included in the new releases of the PLS_Toolbox is based on PMP. Practical and theoretical results presented here show that this choice is not adequate.

Regarding data compression, a new proposed algorithm (corrected *ekf*) is introduced to improve the *ekf* performance. The row-wise k-fold (*rkf*) method, the simplest PCA cross-validation method, shows up as an appropriate tool for compression despite of its numerous detractors.

Traditional cross-validation methods, such as the approaches by Wold [2] and Eastment and Krzanowski [7] were not found to be useful in the PCA applications considered.

A side but relevant contribution of this series of papers is the theoretical study of three types of error in PCA models: the reconstruction error, the error of direct estimation and the error of iterative estimation. Although this study was necessary for the understanding of cross-validation in general and for the design the new cross-validation algorithms, the theoretical findings may have a broader applicability, since these types of errors are used in many chemometric applications and contexts. The reconstruction error and the error of direct estimation were studied in the first paper of the series. The error of iterative estimation is studied in the Appendix of the present paper.

Acknowledgements

Research in this area was partially supported by the Spanish Ministry of Science and Innovation and FEDER funds from the European Union through grants DPI2008-06880-C03-01 and DPI2008-06880-C03-03 and the Juan de la Cierva program, Ministry of Science and Innovation. This study was carried out when José Camacho was at the Universidad Politécnica de Valencia and at the Universitat de Girona, Spain. The reviewers are gratefully acknowledged for their useful comments in both papers of the series.

A Iterative estimation

A.1 Notation

Scalars are specified with lower case letters, column (by default) 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.

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: $\sum_{v \in h}$

A sum including all variables in a group h except variable m is represented by: $\sum_{\substack{v \neq m \\ v \in h}}$

A.2 Direct estimation vs iterative estimation

The estimation with PCA of a data element follows [1]:

$$\hat{x}^A_{n,m} = x_{n,m} \cdot \alpha^A_m + \sum_{v \neq m} x_{n,v} \cdot \beta^A_{v,m}. \tag{A.1}$$

In practice, $x_{n,m}$ takes part in its own estimation with weight α^A_m and the rest of values $x_{n,v}$ with weight $\beta^A_{v,m}$. Now, consider the following definition:

$$\mathbf{Q}^A = \mathbf{P}^A \cdot (\mathbf{P}^A)^t. \tag{A.2}$$

Matrix $\mathbf{Q}_A$ is a $M \times M$ symmetric matrix where α^A_m is the element in the diagonal for row (or column) m and $\beta^A_{v,m}$ is the element out of the diagonal for row v and column m. $\mathbf{Q}_A$ has A eigenvalues equal to 1 and $M - A$ eigenvalues equal to 0 [21].

The reconstruction of $x_{n,m}$ from a PCA model can be expressed as:

$$x_{n,m} = x_{n,m} \cdot \alpha^A_m + \sum_{v \neq m} x_{n,v} \cdot \beta^A_{v,m} + r^A_{n,m}, \tag{A.3}$$

being $r^A_{n,m} = x_{n,m} - \hat{x}^A_{n,m}$ the reconstruction error of $x_{n,m}$ with A PCs and $\hat{x}^A_{n,m}$ its estimation from (A.1). Let us imagine the actual value $x_{n,m}$ cannot be used in its own estimation. Then, $x_{n,m}$ can be estimated substituting its value in equation (A.1) by a certain value $\hat{x}^{(0)}_{n,m}$. The estimation follows:

$$\hat{x}^{(1)}_{n,m} = \hat{x}^{(0)}_{n,m} \cdot \alpha^A_m + \sum_{v \neq m} x_{n,v} \cdot \beta^A_{v,m}. \tag{A.4}$$

This will be termed here as direct imputation. For $\hat{x}^{(0)}_{n,m} = 0$, it has been referred by [18] as trimmed score (TRI) imputation. The direct imputation can be extended to the more general case when the values of several variables are missing at the same time:

$$\hat{x}^{(1)}_{n,m} = \hat{x}^{(0)}_{n,m} \cdot \alpha^A_m + \sum_{\substack{v \neq m \\ v \in h_m}} \hat{x}^{(0)}_{n,v} \cdot \beta^A_{v,m} + \sum_{v \notin h_m} x_{n,v} \cdot \beta^A_{v,m}, \tag{A.5}$$

where h_m is a group of variables which are estimated at the same time than variable m. Equation (A.5) can be iteratively evaluated until the estimation converges:

$$\hat{x}^{(i)}_{n,m} = \hat{x}^{(i-1)}_{n,m} \cdot \alpha^A_m + \sum_{\substack{v \neq m \\ v \in h_m}} \hat{x}^{(i-1)}_{n,v} \cdot \beta^A_{v,m} + \sum_{v \notin h_m} x_{n,v} \cdot \beta^A_{v,m}, \tag{A.6}$$

where $\hat{x}^{(i)}_{n,m}$ is the estimate obtained when i iterations have been computed. This has been referred by [18] as iterative imputation. Notice in (A.6) the estimate of the value of a variable $\hat{x}^{(i)}_{n,m}$ is obtained according to the PCA model (loadings in α^A_m and $\beta^A_{v,m}$), and from: a) the estimate of the same variable in the previous iteration $\hat{x}^{(i-1)}_{n,m}$, b) the estimates in the previous iteration of the others variables in the group h_m, $\hat{x}^{(i-1)}_{n,v}$ for $v \in h_m$, and c) the actual values of the variables out of the group h_m, $x_{n,v}$ for $v \notin h_m$.

Figure A.1 illustrates the geometry of both direct imputation (TRI) and iterative imputation. The case for 2 original variables and 1 PC is presented. In the example, two observations with the same value in variable 1 but very different values in variable 2 are shown. The value corresponding to variable 2 is eventually missing in the observations. It is interesting to note that the values of variable 2 for the two original observations will be equally estimated by

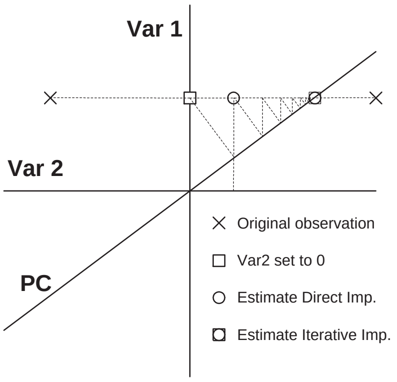


Fig. A.1. Geometric illustration of TRI and iterative imputation with 1 PC of two original samples in a 2-dimensional space.

the imputation methods, since these estimates start form the common value of variable 1. Assume the variable 2 is initially set to zero, i.e. $\hat{x}^{(0)}_{n,Var2} = 0$. 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) is projected on 'Var2'. The TRI estimate of the original observations is represented by the circle. If this operation is repeated successively till convergence, the iterative estimate is found. The iterative estimate of the original observations is represented by the circle inside the square.

The quality of estimation of a variable with PCA can be assessed with the sum of squares of estimation errors (SSE). The SSE associated to a variable m for A PCs is computed according to the following expressions:

$$SSE^A_m = \sum_{n=1}^{N_t} (e^A_{n,m})^2, \tag{A.7}$$

$$e^A_{n,m} = x_{n,m} - \hat{x}^{(i)}_{n,m}, \tag{A.8}$$

where N_t is the number of objects used to compute the SSE, $\hat{x}^{(i)}_{n,m}$ is the estimate of $x_{n,m}$ and $e^A_{n,m}$ is the estimation error. The difference between the reconstruction error $r^A_{n,m}$ and the estimation error $e^A_{n,m}$ in (A.8) is that in the latter, the estimate $\hat{x}^{(i)}_{n,m}$ is computed without using the actual value $x_{n,m}$. On the contrary, to obtain $r^{A}_{n,m}$, $x_{n,m}$ is used. Notice that the meaning of the terms reconstruction error and estimation error may be different in other documents of the literature. Also, when the estimation errors correspond to objects which were not used in the calibration of the PCA model, the SSE is commonly termed prediction error sum-of-squares (PRESS).

Theorem *The estimation of the iterative algorithm converges.*

Proof: The convergence in the iterative algorithm can be studied as the one of the multivariate discrete series from equation (A.6). The estimation at the i-th loop can be arranged in matrix form as follows:

$$\hat{\mathbf{x}}^{(i)}_{n,h_m} = \boldsymbol{\Omega}_{h_m} \cdot \hat{\mathbf{x}}^{(i-1)}_{n,h_m} + \boldsymbol{\Theta}_{h_m,\not{h}_m} \cdot \mathbf{x}_{n,\not{h}_m}, \tag{A.9}$$

where $\mathbf{x}_{n,h_m}$ and $\mathbf{x}_{n,\not{h}_m}$ are column vectors containing the values of variables in h_m and out of h_m, respectively, for object n; and $\boldsymbol{\Omega}_{h_m}$ and $\boldsymbol{\Theta}_{h_m,\not{h}_m}$ are sub-matrices of $\mathbf{Q}_A$ (A.2) so that:

$$\boldsymbol{\Omega}_{h_m} = \mathbf{P}_{A,h_m} \cdot \mathbf{P}^{t}_{A,h_m}, \tag{A.10}$$

$$\boldsymbol{\Theta}_{h_m,\not{h}_m} = \mathbf{P}_{A,h_m} \cdot \mathbf{P}^{t}_{A,\not{h}_m}, \tag{A.11}$$

where $\mathbf{P}_{A,h_m}$ is the sub-matrix of $\mathbf{P}_A$ containing the rows corresponding to the variables in h_m, and $\mathbf{P}_{A,\not{h}_m}$ is the sub-matrix of $\mathbf{P}_A$ with the rows corresponding to variables out of h_m.

It is well known that the discrete series (A.9) converges if and only if the eigenvalues of $\boldsymbol{\Omega}_{h_m}$ are inside the complex unit circle. Since $\boldsymbol{\Omega}_{h_m}$ is symmetric its eigenvalues are real. 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 [30]. That is, each eigenvalue of the resulting matrix will be between two eigenvalues of the original matrix. According to this, the eigenvalues of $\mathbf{\Omega}_{h_m}$ interlace those of $\mathbf{Q}_A$ and so they lie in the interval [0,1]. Therefore, the series (A.9) converges.

A.3 Characterization of the error by iterative estimation

Let us particularize (A.6) for the imputation of one variable at a time:

$$\hat{x}^{(i)}_{n,m} = \hat{x}^{(i-1)}_{n,m} \cdot \alpha^A_m + \sum_{v \neq m} x_{n,v} \cdot \beta^A_{v,m}. \tag{A.12}$$

This estimation can be re-expressed as a function of the initial estimation $\hat{x}^{(0)}_{n,m}$:

$$\hat{x}^{(i)}_{n,m} = (\alpha^A_m)^i \cdot \hat{x}^{(0)}_{n,m} + \left(\sum_{j=0}^{i-1} (\alpha^A_m)^j \right) \cdot \sum_{v \neq m} x_{n,v} \cdot \beta^A_{v,m}. \tag{A.13}$$

For $\alpha^A_m < 1$, the value to which $\hat{x}^{(i)}_{n,m}$ converges does not depend on the initial estimation since $\lim_{i\to\infty}(\alpha^A_m)^i = 0$. Furthermore, we know that the geometric series $\sum_{j=0}^{i-1}(\alpha^A_m)^j$ converges to $\frac{1}{1-\alpha^A_m}$ for $0 < \alpha^A_m < 1$. Therefore:

$$\lim_{i\to\infty} \hat{x}^{(i)}_{n,m} = \frac{1}{1-\alpha^A_m} \cdot \sum_{v \neq m} x_{n,v} \cdot \beta^A_{v,m}, \qquad \alpha^A_m < 1. \tag{A.14}$$

This equation provides an explicit formulae to compute the value to which the iterative algorithm converges when one variable at a time is being imputed. This can be used except for the specific case $\alpha^A_m = 1$ (i.e., m not in the span of the other variables and $A = Rank(X)$). In that case, according to [21] it holds that $\beta^A_{v,m} = 0$, $\forall v \neq m \in \{1, ..., M\}$. Thus, from (A.6) $\hat{x}^{(i)}_{n,m} = \hat{x}^{(0)}_{n,m}$, $\forall i = \{0, ..., \infty\}$. If $x_{n,m}$ is available (for instance, this is the case in cross-validation), (A.14) can be computed in an efficient way using (A.3):

$$\lim_{i\to\infty} \hat{x}^{(i)}_{n,m} = x_{n,m} - \frac{r^A_{n,m}}{1-\alpha^A_m}, \qquad \alpha^A_m < 1. \tag{A.15}$$

From (A.8), (A.3) and (A.14), the error of estimation in convergence is:

$$e^A_{n,m} = x_{n,m} \cdot \alpha^A_m + r^A_{n,m} - \frac{\alpha^A_m}{1-\alpha^A_m} \cdot \sum_{v\neq m} x_{n,v} \cdot \beta^A_{v,m}, \qquad \alpha^A_m < 1. \tag{A.16}$$

From (A.15) a computationally efficient form of (A.16) is:

$$e^A_{n,m} = \frac{r^A_{n,m}}{1-\alpha^A_m}, \qquad \alpha^A_m < 1 \tag{A.17}$$

,

Again, this holds except for the case $\alpha^A_m = 1$. In that case, (A.17) presents indeterminate form and $\hat{x}^{(i)}_{n,m} = \hat{x}^{(0)}_{n,m}$, $\forall i = \{0, ..., \infty\}$, as already discussed. Therefore, $e^A_{n,m} = \epsilon^{(0)}_{n,m}$ as it happens in direct imputation [1]. From (A.17) the SSE can be computed:

$$SSE^A_m = \frac{1}{(1-\alpha^A_m)^2} \cdot \sum_{n=1}^{N_t} (r^A_{n,m})^2, \qquad \alpha^A_m < 1, \tag{A.18}$$

which can be used except for the case $\alpha^A_m = 1$. As it happens with the direct imputation [1], we can assume by convention that $SSE^0_m = \sum_{n=1}^{N_t} (\epsilon^{(0)}_{n,m})^2$. For full rank, (A.18) applies except for $\alpha^A_m = 1$, and $SSE^{Rank(\mathbf{X})}_m = 0$. For $\alpha^A_m = 1$ (i.e., full rank and m not in the span of the other variables) there is no prediction power for m and: $SSE^{Rank(\mathbf{X})}_m = \sum_{n=1}^{N_t} (\epsilon^{(0)}_{n,m})^2 = SSE^0_m$.

According to the previous discussion, the error of estimation in the iterative algorithm of a variable which belongs to the span of other variables in a PCA model with $A = Rank(\mathbf{X})$ is equal to 0. This does not happen when using the direct imputation (see Property 1 in [1]). To extend this property to the general case where several variables are imputed at the same time, h_m should be carefully chosen so that $\mathbf{\Omega}_{h_m}$ in (A.9) does not have any eigenvalue equal to 1. To see this, let us follow the same procedure of the preceding demonstration. Let $\hat{\mathbf{x}}^{(i)}_{n,h_m}$ in (A.9) be re-expressed as a function of the initial estimation $\hat{\mathbf{x}}^{(0)}_{n,h_m}$:

$$\hat{\mathbf{x}}^{(i)}_{n,h_m} = \mathbf{\Omega}^{i}_{h_m} \cdot \hat{\mathbf{x}}^{(0)}_{n,h_m} + \left(\sum_{j=0}^{i-1} \mathbf{\Omega}^{j}_{h_m} \right) \cdot \mathbf{\Theta}_{h_m,\not{h}_m} \cdot \mathbf{x}_{n,\not{h}_m}, \tag{A.19}$$

where the power of a matrix $\mathbf{M}$ is defined as $\mathbf{M}^i = \Pi^{i}_{j=1}\mathbf{M}$. Since $\mathbf{\Omega}_{h_m}$ is a symmetric matrix, we know that:

$$\mathbf{\Omega}_{h_m} = \mathbf{O} \cdot \mathbf{D} \cdot \mathbf{O}^t, \tag{A.20}$$

where $\mathbf{O}$ is an orthonormal matrix and $\mathbf{D}$ is a diagonal matrix containing the eigenvalues of $\mathbf{\Omega}_{h_m}$. Then, the following holds:

$$\mathbf{\Omega}^2_{h_m} = \mathbf{\Omega}_{h_m} \cdot \mathbf{\Omega}_{h_m} = \mathbf{O} \cdot \mathbf{D} \cdot \mathbf{O}^t \cdot \mathbf{O} \cdot \mathbf{D} \cdot \mathbf{O}^t = \mathbf{O} \cdot \mathbf{D} \cdot \mathbf{D} \cdot \mathbf{O}^t, \tag{A.21}$$

and in general:

$$\mathbf{\Omega}^i_{h_m} = \mathbf{O} \cdot \mathbf{D}^i \cdot \mathbf{O}^t. \tag{A.22}$$

For the algorithm to converge to a $\hat{\mathbf{x}}^{(i)}_{n,h_m}$ value which is independent of the initial estimation:

$$\lim_{i \to \infty} \mathbf{\Omega}^i_{h_m} = 0, \tag{A.23}$$

which from (A.22) is equivalent to:

$$\lim_{i\to\infty} \mathbf{D}^i = 0. \tag{A.24}$$

As it was proved before, the eigenvalues of $\mathbf{\Omega}_{h_m}$ in the diagonal of $\mathbf{D}$ are real values which lie in the interval [0,1]. In particular, (A.24) will be true if all the eigenvalues of Ω_{h_m} are lower than 1. For the eigenvalues of $\mathbf{\Omega}_{h_m}$ lower than 1 it also holds:

$$\sum_{j=0}^{\infty} \mathbf{\Omega}_{h_m}^j = (I - \mathbf{\Omega}_{h_m})^{-1}. \tag{A.25}$$

Therefore, from (A.19):

$$\lim_{i\to\infty} \hat{\mathbf{x}}_{n,h_m}^{(i)} = (\mathbf{I} - \mathbf{\Omega}_{h_m})^{-1} \cdot \mathbf{\Theta}_{h_m,\not{h}_m} \cdot \mathbf{x}_{n,\not{h}_m}, \qquad eig(\mathbf{\Omega}_{h_m}) < 1. \tag{A.26}$$

Equation (A.26) provides an explicit formulae to compute the value to which the iterative imputation of multiple variables converges. This can be used if all the eigenvalues of Ω_{h_m} are lower than 1. This, in turn, can be assured by properly selecting the groups of variables left out at the same time except for the specific case $\alpha_m^A = 1$ (i.e., m not in the span of the other variables and full rank).

On the other hand, from (A.3) we know that:

$$(1 - \alpha_m^A) \cdot x_{n,m} - \sum_{\substack{v \neq m \\ v \in h_m}} x_{n,v} \cdot \beta_{v,m}^A = \sum_{v \notin h_m} x_{n,v} \cdot \beta_{v,m}^A + r_{n,m}^A, \tag{A.27}$$

re-arranged in matrix form:

$$(\mathbf{I} - \mathbf{\Omega}_{h_m}) \cdot \mathbf{x}_{n,h_m} = \mathbf{\Theta}_{h_m,\not{h}_m} \cdot \mathbf{x}_{n,\not{h}_m} + \mathbf{r}_{n,h_m}^A, \tag{A.28}$$

then:

$$\mathbf{x}_{n,h_m} = (\mathbf{I} - \mathbf{\Omega}_{h_m})^{-1} \cdot \mathbf{\Theta}_{h_m,\cancel{h}_m} \cdot \mathbf{x}_{n,\cancel{h}_m} + (\mathbf{I} - \mathbf{\Omega}_{h_m})^{-1} \cdot \mathbf{r}^{A}_{n,h_m}, \tag{A.29}$$

thus, from (A.26) and (A.29), the error of estimation for each group of variables h_m:

$$\mathbf{e}^{A}_{n,h_m} = (\mathbf{I} - \mathbf{\Omega}_{h_m})^{-1} \cdot \mathbf{r}^{A}_{n,h_m}, \qquad eig(\mathbf{\Omega}_{h_m}) < 1, \tag{A.30}$$

so that in particular for full rank ($\mathbf{r}^{A}_{n,h_m} = 0$):

$$\lim_{i\to\infty} \hat{\mathbf{x}}^{(i)}_{n,h_m} = \mathbf{x}_{n,h_m}, \qquad A = Rank(\mathbf{X}),\ eig(\mathbf{\Omega}_{h_m}) < 1. \tag{A.31}$$

Therefore, the error of estimation in the iterative algorithm of a variable which can be expressed as a linear combination of the other variables in a PCA model with $A = Rank(\mathbf{X})$ is equal to 0, provided the groups of variables left out at the same time are selected so that all the eigenvalues of $\mathbf{\Omega}_{h_m}$ are lower than 1. This result means that the inconsistency problem in direct imputation found in the first paper of this series [1] is solved by iteration. This will be further studied in the next section.

A.4 Inconsistency and directional dependence

Let us return to the same example used in [1] to show the inconsistency and directional dependence problems in direct estimation. The example will be repeated to illustrate the absence of the inconsistency problem but the presence of the directional dependence in iterative estimation.

Consider the hypothetical examples shown in Figure A.2. Let us imagine we have already calibrated the PCA model and that we are interested in geometrically characterizing the points in the space in which the estimation error of each coordinate from the other would be reduced by adding PCs to the model. For this, a grid on the square area spanned by the coordinates $\{x, y\}$, for $x \in [-10, 10]$ and $y \in [-10, 10]$, was performed. The sum-of-squares of estimation error (SSE) corresponding to each point $\{x, y\}$ in the grid is computed as $(e_x^A)^2 + (e_y^A)^2$, where e_x^A is the estimation error of coordinate x when this information is missing and e_y^A is the estimation error of y when it is missing.

In each of the two rows of figures in Figure A.2, the two original variables are represented by a different pair of PCs. In the first column (Figure A.2(a)), the directions of the 2 PCs are shown. In the rest of the figures, the SSE ratio of the missing coordinates in every point is compared for 1 and 2 PCs:

$$\frac{(e_x^2)^2 + (e_y^2)^2}{(e_x^1)^2 + (e_y^1)^2}, \tag{A.32}$$

This ratio is represented as a color map. The color code shows where there is an increase or reduction in SSE when the second PC is added to the model. In the second column of figures, the ratio of reduction of SSE according to direct estimation is represented. In the third column, the same for iterative estimation is represented.

In direct imputation, the points with the maximum improvement in prediction error when adding the second PC coincide with the bisectrices of the quadrants instead of with the second PC. This is referred to as the inconsistency problem of direct imputation in [1]. On the contrary, in iterative estimation the distribution of the error depends on the specific direction of the PCs. If the direction of the PC is rotated, the area where there is an improvement of

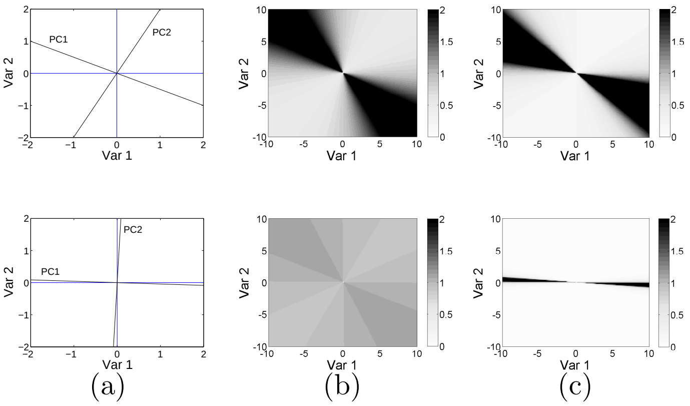


Fig. A.2. Geometrical illustration of the ratio of estimation error by TRI (b) and iterative estimation (c) when a second PC is added to the PCA model. Two examples for different directions of the PCs (a) are shown.

estimation is rotated accordingly. Therefore, the iterative imputation is consistent. However, the area where the SSE is improved is determined by the original variable which is closest to the PC. Thus, for the second example, where the first PC is very close to Var 1, the area where the estimation is improved is very narrow. The fact that the SSE will be determined by the relationship between original space and latent subspace means that the SSE by iterative estimation suffers from directional dependence, like happens for direct estimation. For more details on this example and its discussion, refer to [1].

A.5 Differences between the iterative imputation and PMP

Strictly speaking, the iterative estimation and PMP are different only when Ω_{h_m} in (A.9) has eigenvalues equal to 1 [18]. Numerically, differences can be observed when the eigenvalues approach 1. Notice that, for the estimation of one-variable-at-a-time, the eigenvalues in Ω_{h_m} coincide with the parameters α_m^A. If a variable m only composed of non-redundant information is almost completely captured by the PCA model, then α_m^A will approach 1 and PMP tends to inflate the estimation error, as shown in the examples in Figure 2. At the same time, the PRESS curve by iterative estimation and PMP will be different and so the number of PCs suggested may be also different.

References

[1] J. Camacho, A. Ferrer, Cross-validation in pca models with the element-wise k-fold (ekf) algorithm: theoretical aspects, Journal of Chemometrics (26) (2012) 361–373.

[2] S. Wold, Cross-validatory estimation of the number of components in factor and principal components, Technometrics 20 (4) (1978) 397–405.

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

[4] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.

[5] P. Zhang, Model selection via multifold crossvalidation, The Annals of Statistics 21 (1993) 299–313.

[6] D. Giancarlo, C. Tommasi, Cross-validation methods in principal component analysis: a comparison, Statistical Methods and Applications 11 (2002) 71–82.

[7] H. Eastment, W. Krzanowski, Cross-validatory choice of the number of components from a principal component analysis, Technometrics 24 (1) (1982) 73–77.

[8] H. Wold, E. Lyttkens, Nonlinear iterative partial least squares (nipals) estimation procedures, in: Bull. Intern. Statist. Inst. Proc., 37th session, London, 1969, pp. 1–15.

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

[10] P. Nelson, J. MacGregor, P. Taylor, The impact of missing measurements on pca and pls prediction and monitoring applications, Chemometrics and Intelligent Laboratory Systems 80 (2006) 1–12.

[11] Y. Song, F. Nie, C. Zhang, S. Xiang, A unified framework for semi-supervised dimensionality reduction, Pattern Recognition 41 (2008) 2789–2799.

[12] X. Zhao, Y. Liu, Generative tracking of 3d human motion by hierarchical annealed genetic algorithm, Pattern Recognition 41 (2008) 2470–2483.

[13] E. Alaa, D. Hasan, Face recognition system based on pca and feedforward neural networks, in: Lecture notes in computer science, ISSN 0302-9743, 2005, pp. 935–942.

[14] H. He, X. Yu, A comparison of pca/ica for data preprocessing in remote sensing imagery classification, in: D. Li, H. Ma (Eds.), MIPPR 2005: Image Analysis Techniques, Proceedings of the SPIE, Volume 6044, 2005, pp. 60–65.

[15] P. Nelson, P. Taylor, J. MacGregor, Missing data methods in pca and pls: score calculations with incomplete observations, Chemometrics and Intelligent Laboratory Systems 35 (1996) 45–65.

[16] D. Andrews, P. Wentzell, Applications of maximum likelihood principal component analysis: incomplete data sets and calibration transfer, Analytica Chimica Acta 350 (1997) 341–352.

[17] B. Walczak, D. Massart, Dealing with missing data: Part i, Chemometrics and Intelligent Laboratoy Systems 58 (2001) 15–27.

[18] F. Arteaga, A. Ferrer, Dealing with missing data in mspc: several methods, different interpretations, some examples, Journal of Chemometrics 16 (2002) 408–418.

[19] F. Arteaga, A. Ferrer, Framework for regression-based missing data imputation methods in on-line mspc, Journal of Chemometrics 19 (2005) 439–447.

[20] K. Kosanovich, K. Dahl, M. Piovoso, Improved process understanding using multiway principal component analysis, Engineering Chemical Research 35 (1996) 138–146.

[21] J. Camacho, J. Picó, A. Ferrer, Data understanding with pca: Structural and variance information plots, Chemometrics and Intelligent Laboratory Systems 100 (1) (2010) 48–56.

[22] J. Camacho, Missing-data theory in the context of exploratory data analysis, Chemometrics and Intelligent Laboratory Systems 103 (2010) 8–18.

[23] J. Camacho, Meda to unveil the conection between observations and variables in latent subspace models, Submitted to Journal of Chemometrics.

[24] T. Kourti, J. MacGregor, Multivariate spc methods for process and product monitoring, Journal of Quality Technology 28 (4).

[25] A. Ferrer, Multivariate statistical process control based on principal component analysis (mspc-pca): Some reflections and a case study in an autobody assembly process, Quality Engineering 19 (4) (2007) 311–325.

[26] N. Tracy, J. Young, R. Mason, Multivariate control charts for individual observations, Journal of Quality Technology 24 (2) (1992) 88–95.

[27] W. McReynolds, Characterization of some liquid phases, Journal of Chromatography Science 8 (1970) 685–691.

[28] B. Walczak, D. Massart, Dealing with missing data. part 2, Chemometrics and Intelligent Laboratory Systems 58 (2001) 29–42.

[29] S. Wold, K. Andersson, Major components influencing retention indices in gas chromatography, Journal of Chromatography 80 (1973) 43–59.

[30] M. Mercer, R. Mercer, Cauchy's interlace theorem and lower bounds for the spectral radius, International Journal of Mathematics and Mathematical Sciences 23 (1998) 563–566.