On the use of the observation-wise k-fold operation in PCA cross-validation

Edoardo Saccenti[1,*]

Wageningen University and Research Center

José Camacho[2,*]

University of Granada

Abstract

Cross-validation (CV) is a common approach for determining the optimal number of components in a principal component analysis model. To guarantee the independence between model testing and calibration, the observation-wise k-fold operation is commonly implemented in each cross-validation step. This operation renders the CV algorithm computationally intensive and it is the main limitation to apply CV on very large data sets. In this paper we carry out an empirical and theoretical investigation of the use of this operation in the element wise k-fold (*ekf*) algorithm, the state-of-the-art CV algorithm. We show that when very large data sets need to be cross-validated and the computational time is a matter of concern, the observation-wise k-fold operation can be skipped. The theoretical properties of the resulting modified algorithm, referred to as column wise k-fold (*ckf*) algorithm, are derived. Also, its performance is evaluated with several artificial and real data sets. We suggest the *ckf* algorithm to be a valid alternative to the standard *ekf* to reduce the computational time needed to cross-validate a data set.

Keywords: Cross-validation, Principal component analysis, dimensionality assessment

*Corresponding author

Email addresses: esaccenti@gmail.com (Edoardo Saccenti), josecamacho@ugr.es (José Camacho)

[1]Laboratory of Systems and Synthetic Biology, Wageningen University and Research Centre, Dreijenplein 10, 6703 HB Wageningen, The Netherlands.

[2]Network Engineering and Security Group, Signal Theory, Networking and Communications Department, University of Granada, C/ Periodista Daniel Saucedo Aranda s/n 18071, Granada, Spain

1. Introduction

In chemometrics and related disciplines, cross-validation (CV) is probably the most commonly used method to determine the optimal number of components (PC) to retain in a principal component analysis (PCA) model. Several cross-validatory algorithms have been proposed [1, 2] after the pioneering works of Wold [3] and Eastment and Krzanowski [4]. The simplest version of CV is the so-called row wise k-fold (*rkf*) method [5, 6], where groups of rows (observations) of the data set are left-out to build the model. The *rkf* has been criticized because the estimation of the left-out observations is not independent from the observations themselves [2].

An alternative, proposed by Wold himself (see page 401 in [3]: *An alternative scheme*), is the the so called element wise k-fold (*ekf*) cross-validation. This method is based on the capability of PCA to recover missing data [7, 8, 9, 10, 11, 12, 13]: some of the elements of the data matrix $\mathbf{X}$ are set to be missing and are subsequently recovered using a missing data imputation strategy and the model built on the remaining data. An estimation error is then derived by comparing the true values with their reconstruction, which is, in this case, independent from the left-out elements [2].

The *ekf* was found to outperform other CV methods (among them the *rkf*) in a comparative study [2] (where it was referred to as the Eigenvector algorithm, see also [14] for historical reasons), and recently it has benefited of an in-depth analysis at both the theoretical and practical levels [14, 15]. The *ekf* is especially suited to select the number of components when the PCA model is going to be used for future missing data recovery [14] or when a model is derived for Multivariate Exploratory Data Analysis (MEDA) with the goal of finding groups of related variables.

The biggest drawback of *ekf* is its computational cost, which prevents the use on very large data sets, like for instance *omics* data, and this limitation is common to other CV methods. For this task a number of alternative and faster methods have been proposed to determine the optimal number of components either considering numerical approximation of the cross-validation procedure [16] or statistical approaches [17, 18]. Despite this limitation *ekf* is one of the most used CV methods in chemometrics [2].

The most computationally intensive operation within *ekf*, if the efficient algorithm proposed in [15] is used, is the observation-wise k-fold operation. In this paper a thorough study on the convenience of this operation is performed. It is shown that the observation-wise k-fold operation does not provide the *ekf* algorithm with any relevant property. Thus, we claim that if computational time is a matter of concern, this operation can be skipped. The mathematical properties of the resulting algorithm, referred to as column wise k-fold (*ckf*) algorithm, are studied and its performance is compared with that of *ekf* in order to show its suitability to select the number of components in very large data sets. We suggest the *ckf* algorithm to be a valid alternative to the standard *ekf* to reduce the computational time needed to cross-validate a data set.

The paper is organized as follows. Section 2 presents the *rkf*, *ekf* and *ckf*

algorithms. Section 3 is dedicated to the mathematical characterization of the *ckf* algorithm. In Sections 4 and 5 the performance of the two algorithms is compared on simulated and real data, respectively. Section 6 offers some conclusions and recommendations for the use of the *ckf* algorithm.

2. Description of the *rkf*, *ekf* and *ckf* algorithms

Let the PCA model for a $N \times M$ data matrix $\mathbf{X}$ be defined as:

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

where $\mathbf{T}^A$ is the $N \times A$ score matrix, $\mathbf{P}^A$ is the $M \times A$ loading matrix, and $\mathbf{R}^A$ is the $N \times M$ matrix of residuals. Consistently with [14, 15], the principal components used in a model is indicated with A.

2.1. The rkf algorithm

The *rkf* algorithm is presented in Box 1. Here the outer loop iterates through the observations rather than trough the components as in the formulation in [15]: this change reduces the number of PCA models to be fitted. The observations (rows) are arranged in G disjoint groups. Thus, if a leave-one-out scheme is employed, each group contains a single observation. In each iteration, one group of observations is left out and a PCA model is fitted from the rest of the groups. This model is fitted with the maximum number of PCs considered in the cross-validation.

The inner loop iterates through the PCs. It starts with a model using the first PC, then using the first two PCs, and so on. In each iteration, the reconstruction error for the left-out group of observations is computed. The output of the algorithm is the matrix of reconstruction errors $\mathbf{R}^A$ from which the corresponding sum of squares of the prediction error (PRESS^A) can be computed as follows:

$$\text{PRESS}^A_{rkf} = ||\mathbf{R}^A||^2_F \tag{2}$$

where $||x||_F$ indicates the Frobenius norm of x.

The *rkf* method yields strictly decreasing PRESS curves since the error computed within the algorithm is the reconstruction error.

In the *rkf*, the estimation of the left-out observations is not independent from the observations themselves: the first equation of the inner loop shows that the scores of the left-out samples are computed from the actual observations. 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 $\mathbf{X}_*$ (data from all groups but G) and then applied to $\mathbf{X}_\#$ (data from G). Under the assumption that the model will be applied to future observations the second option is preferred. For the sake of simplicity, the parameters relating to preprocessing are here omitted.

For each group of observations (G = 1...G_{tot})
    Form $\mathbf{X}_*$ with data from all groups but G
    Form $\mathbf{X}_\#$ with data from G
    Fit the PCA model: $\mathbf{X}_* = \mathbf{T}_*^{A_{max}}(\mathbf{P}_*^{A_{max}})^t + \mathbf{R}_*^{A_{max}}$
    For each PC ($A = 1...A_{max}$)
        $\mathbf{T}_\#^A = \mathbf{X}_\# \cdot \mathbf{P}_*^A$
        $\hat{\mathbf{X}}_\# = \mathbf{T}_\#^A \cdot (\mathbf{P}_*^A)^t$
        $\mathbf{R}_\mathrm{G}^A = \mathbf{X}_\# - \hat{\mathbf{X}}_\#$
    end
    Combine matrices $\mathbf{R}_\mathrm{G}^A$ in $\mathbf{R}^A$
end

Algorithm box 1: Row-wise k-fold (*rkf*) algorithm. $\mathbf{X}$, $\mathbf{T}$, $\mathbf{P}$, $\mathbf{R}$ represent the data, scores, loading and reconstruction error matrices. The superscript A indicates the components used in the model. G indicates the disjoint groups in which the observations (rows) are arranged and the columns selected.

2.2. The ekf algorithm

In comparison to *rkf*, the *ekf* can be seen as a combination of a k-fold operation in both the rows and the columns of the data matrix. It should be noted, however, that the two k-fold operations are not identical: the k-fold operation in the observations discards complete observations to construct the model, while this is not the case in the k-fold operation in the variables. The *ekf* method is outlined in Algorithm box 2.

The outer and the intermediate loops reproduce the *rkf* algorithm. The difference with *rkf* is the inner loop that iterates through the variables, arranged in H disjoint groups. Again, a leave-one-out scheme can be used in the groups of variables if desired. The loop computes the CV error by using the missing data method referred to as trimmed score imputation (TRI) [19]. Very briefly, the scores of incomplete observations are estimated by imputing missing values with their unconditional means (*i.e.* zero value for mean-centered data). In this way a so called trimmed-score, which is a least square estimator, is obtained. See [15] and [20] for a thorough analysis on the use of this and other imputation methods.

The inner loop yields the corresponding error for the elements which belong to the rows in G and the columns in H. The output of the algorithm is the matrix of prediction errors $\mathbf{E}^A$ (with elements $e_{n,m}^A$ in the n-th row and m-th column) from which the corresponding PRESS^A can be computed as follows:

$$\mathrm{PRESS}_{ekf}^A = ||\mathbf{E}^A||_F^2 \tag{3}$$

The PRESS curves computed in this way typically show a U-valley shape where the minimum identifies the optimal number of components.

For each group of observations (G = 1...G_{tot})
  Form $\mathbf{X}_*$ with data from all groups but G
  Form $\mathbf{X}_\#$ with data from G
  Fit the PCA model: $\mathbf{X}_* = \mathbf{T}_*^{A_{max}}(\mathbf{P}_*^{A_{max}})^t + \mathbf{R}_*^{A_{max}}$
  For each PC ($A = 1...A_{max}$)
    $\mathbf{T}_\#^A = \mathbf{X}_\#\mathbf{P}_*^A$
    $\hat{\mathbf{X}}_\# = \mathbf{T}_\#^A(\mathbf{P}_*^A)^t$
    $\mathbf{R}_\#^A = \mathbf{X}_\# - \hat{\mathbf{X}}_\#$
    For each group of variables (H = 1...H_{tot})
      Select the rows of $\mathbf{P}_*^A$ in H yielding $\mathbf{P}_{*,\mathrm{H}}^A$
      Select the columns of $\mathbf{R}_\#^A$ in H yielding $\mathbf{R}_{\#,\mathrm{H}}^A$
      $\mathbf{Q}_{*,\mathrm{H}}^A = \mathbf{P}_{*,\mathrm{H}}^A(\mathbf{P}_{*,\mathrm{H}}^A)^t$
      $\mathbf{E}_{\mathrm{G,H}}^A = \mathbf{X}_{\#,\mathrm{H}}\mathbf{Q}_{*,\mathrm{H}}^A + \mathbf{R}_{\#,\mathrm{H}}^A$
    end
    Combine matrices $\mathbf{E}_{\mathrm{G,H}}^A$ in $\mathbf{E}^A$
  end
end

Algorithm box 2: Element-wise k-fold (*ekf*) algorithm. $\mathbf{X}$, $\mathbf{T}$, $\mathbf{P}$, $\mathbf{R}$, $\mathbf{E}$ represent the data, scores, loading and reconstruction and prediction error matrices. The superscript A indicates the components used in the model. G and H indicate the disjoint groups in which the observations (rows) are arranged and the columns selected.

The steps followed in the intermediate and inner loops to compute the *ekf* error were proposed in [15], providing a much more efficient way to iterate through the variables than in the traditional, more intuitive version of [2]. It should also be noted that [2] makes use of the Projection to Model Plane (PMP) imputation method [19] instead of TRI, a procedure that was demonstrated to introduce instability in the *ekf* algorithm. The reader is referred to [14] for more information on the application of different imputation algorithms to *ekf*.

The properties of *ekf* descend from the use of the Trimmed Score Imputation (TRI) [9] in the algorithm. Although the*ekf* may not be well suited to separate structure from noise (as the variance in one original variable is neither treated as structure nor as noise) it may be indicated to use *ekf* with typical chemometrics data sets, like spectroscopy data, where independent variables are uncommon [15].

2.3. The ckf algorithm

If the observation k-fold operation in *ekf* is skipped, the algorithm greatly simplifies since the outer loop disappears. The resulting algorithm is outlined in Algorithm box 3. As this algorithm only iterates through the columns, we propose the name of column-wise k-fold (*ckf*) algorithm. It should be noted that, following [15], if a leave-one-out (loo) operation is done in the columns

Fit the PCA model: $\mathbf{X} = \mathbf{T}^{A_{max}}(\mathbf{P}^{A_{max}})^t + \mathbf{R}^{A_{max}}$
For each PC ($A = 1...A_{max}$)
    $\mathbf{R}^A = \mathbf{X} - \mathbf{T}^A(\mathbf{P}^A)^t$
    For each group of variables (H = $1...H_{tot}$)
        Select the rows of $\mathbf{P}^A$ in H yielding $\mathbf{P}^A_{\mathrm{H}}$
        Select the columns of $\mathbf{R}^A$ in H yielding $\mathbf{R}^A_{\mathrm{H}}$
        $\mathbf{Q}^A_{\mathrm{H}} = \mathbf{P}^A_{\mathrm{H}}(\mathbf{P}^A_{\mathrm{H}})^t$
        $\mathbf{E}^A_{\mathrm{H}} = \mathbf{X}_{\mathrm{H}}\mathbf{Q}^A_{\mathrm{H}} + \mathbf{R}^A_{\mathrm{H}}$
    end
    Combine matrices $\mathbf{E}^A_{\mathrm{H}}$ in $\mathbf{E}^A$
end

Algorithm box 3: Column-wise k-fold (*ckf*) algorithm. $\mathbf{X}$, $\mathbf{T}$, $\mathbf{P}$, $\mathbf{R}$, $\mathbf{E}$ represent the data, scores, loading and reconstruction and prediction error matrices. The superscript A indicates the components used in the model. H indicates the columns selected.

within the *ckf*, the inner loop is transformed to a single matrix operation. This leads to an algorithm that is particularly fast in computational environments designed for matrix operations, like Matlab [21] or Octave [22, 23].

Likewise the *ekf*, *ckf* makes use of the TRI procedure but in one single step, so that only one PCA model is fitted. The idea of skipping the k-fold operation in the observations is also substantiated by the fact that the properties of PRESS curve of the *ekf* do not depend on the k-fold operation in the observations [15] but only in the properties of the error by TRI. Thus, the PRESS by *ckf* retains all properties highlighted in the introduction, including the convenient valley-shape where the minimum value identifies the number of PCs.

3. Mathematical characterization of the *ckf* algorithm

As shown in [24], very different data distributions can lead to the same covariance matrix. Thus, the function that links a data matrix with its corresponding covariance matrix is one to many; the covariance matrix can be seen as a summary of the data where the information about data distribution is lost.

We now set to demonstrate that the output of *ckf* depends on the covariance matrix, and we define a kernel version of the algorithm.

For any group H of variables (without any particular order) the corresponding reconstruction error $\mathbf{R}^A_{\mathrm{H}}$ is given by

$$\mathbf{R}^A_{\mathrm{H}} = \mathbf{X}_{\mathrm{H}} - \mathbf{X}\mathbf{P}^A(\mathbf{P}^A_{\mathrm{H}})^t \tag{4}$$

The loading vectors $\mathbf{P}^A$ of a PCA model for a $N \times M$ data matrix $\mathbf{X}$ can be also obtained using the eigendecomposition of the cross-product matrix:

$$\mathbf{\Psi} = \mathbf{X}^t\mathbf{X}. \tag{5}$$

The cross-product matrix for any group H of variables of $\mathbf{X}$ can be obtained as sub-matrix of the full cross-product matrix $\mathbf{\Psi}$. We set

$$\mathbf{\Psi}_{\mathrm{H,H}} = (\mathbf{X}_{\mathrm{H}})^t \mathbf{X}_{\mathrm{H}} \tag{6}$$

and

$$\mathbf{\Psi}_{\mathrm{M,H}} = \mathbf{X}^t \mathbf{X}_{\mathrm{H}} \tag{7}$$

Both *ekf* and *ckf* use the TRI procedure: it can be shown (see [15]) that the error by TRI $\mathbf{E}_{\mathrm{H}}^A$ for a group of variables H in a PCA model fitted with A components is given by

$$\mathbf{E}_{\mathrm{H}}^A = \mathbf{X}_{\mathrm{H}} \mathbf{Q}_{\mathrm{H}}^A + \mathbf{R}_{\mathrm{H}}^A \tag{8}$$

with:

$$\mathbf{Q}_{\mathrm{H}}^A = \mathbf{P}_{\mathrm{H}}^A (\mathbf{P}_{\mathrm{H}}^A)^t \tag{9}$$

The PRESS for the model computed with A components can be computed as:

$$\mathrm{PRESS}^A = \sum_{\mathrm{H}} \left(\mathbf{E}_{\mathrm{H}}^A\right)^2 = \sum_{\mathrm{H}} \mathrm{tr}\left\{(\mathbf{E}_{\mathrm{H}}^A)^t \mathbf{E}_{\mathrm{H}}^A\right\} \tag{10}$$

Substituting Equation (8) in the third member of Equation (10), with some basic matrix algebra manipulation and using the properties of the trace operator it is immediate to arrive at

$$\begin{aligned} \mathrm{tr}\left\{(\mathbf{E}_{\mathrm{H}}^A)^t \mathbf{E}_{\mathrm{H}}^A\right\} = \mathrm{tr}\left\{(\mathbf{Q}_{\mathrm{H}}^A)^t (\mathbf{X}_{\mathrm{H}})^t \mathbf{X}_{\mathrm{H}} \mathbf{Q}_{\mathrm{H}}^A)\right\} + \mathrm{tr}\left\{(\mathbf{R}_{\mathrm{H}}^A)^t \mathbf{R}_{\mathrm{H}}^A\right\} + \\ + 2\,\mathrm{tr}\left\{(\mathbf{R}_{\mathrm{H}}^A)^t \mathbf{X}_{\mathrm{H}} \mathbf{Q}_{\mathrm{H}}^A\right\} \end{aligned} \tag{11}$$

By plugging Equation (11) in (10) and re-arranging the sum over H, the PRESS is given by

$$\begin{aligned} \mathrm{PRESS}^A = \sum_{\mathrm{H}} \mathrm{tr}\left\{(\mathbf{R}_{\mathrm{H}}^A)^t \mathbf{R}_{\mathrm{H}}^A\right\} + \sum_{\mathrm{H}} \mathrm{tr}\left\{(\mathbf{Q}_{\mathrm{H}}^A)^t (\mathbf{X}_{\mathrm{H}})^t \mathbf{X}_{\mathrm{H}} \mathbf{Q}_{\mathrm{H}}^A)\right\} + \\ + \sum_{\mathrm{H}} 2\,\mathrm{tr}\left\{(\mathbf{R}_{\mathrm{H}}^A)^t \mathbf{X}_{\mathrm{H}} \mathbf{Q}_{\mathrm{H}}^A\right\} \end{aligned} \tag{12}$$

For the first term, it holds that:

$$\sum_{\mathrm{H}} \mathrm{tr}\left\{(\mathbf{R}_{\mathrm{H}}^A)^t \mathbf{R}_{\mathrm{H}}^A\right\} = \mathrm{tr}\left\{(\mathbf{R}^A)^t \mathbf{R}^A\right\} \tag{13}$$

with $\mathbf{Q}^A = \mathbf{P}^A (\mathbf{P}^A)^t$. However, the same simplification cannot be performed on the other two terms of Equation (12) due to the $\mathbf{Q}_{\mathrm{H}}^A$ terms.

It is now easy to show that the three terms of Equation (12) can be expressed in terms of the elements of the data cross-product matrix $\mathbf{\Psi}$. The first term

in Equation (12) is just the trace of the cross-product matrix of the residual matrix $\mathbf{R}^A$ and can be computed from the data cross product matrix $\mathbf{\Psi}$

$$\mathrm{tr}\left\{(\mathbf{R}^A)^t\mathbf{R}^A\right\} = \mathrm{tr}\left\{\mathbf{\Psi} - \mathbf{Q}^A\mathbf{\Psi}(\mathbf{Q}^A)^t\right\} \tag{14}$$

Using definition (6) the second term of Equation (12) is

$$\sum_{\mathrm{H}} \mathrm{tr}\left\{(\mathbf{Q}_{\mathrm{H}}^A)^t(\mathbf{X}_{\mathrm{H}})^t\mathbf{X}_{\mathrm{H}}\mathbf{Q}_{\mathrm{H}}^A\right\} = \sum_{\mathrm{H}} \mathrm{tr}\left\{(\mathbf{Q}_{\mathrm{H}}^A)^t\mathbf{\Psi}_{\mathrm{H,H}}\mathbf{Q}_{\mathrm{H}}^A\right\} \tag{15}$$

By making use of formula (4) and definitions (6) and (7) the last term of expression (12) becomes

$$\begin{aligned}\sum_{\mathrm{H}} 2\,\mathrm{tr}\left\{(\mathbf{R}_{\mathrm{H}}^A)^t\mathbf{X}_{\mathrm{H}}\mathbf{Q}_{\mathrm{H}}^A\right\} &= \sum_{\mathrm{H}} 2\,\mathrm{tr}\left\{(\mathbf{X}_{\mathrm{H}} - \mathbf{X}\mathbf{P}^A(\mathbf{P}_{\mathrm{H}}^A)^t)^t\mathbf{X}_{\mathrm{H}}\mathbf{Q}_{\mathrm{H}}^A\right\} = \\ &= \sum_{\mathrm{H}} 2\,\mathrm{tr}\left\{\mathbf{\Psi}_{\mathrm{H,H}}\mathbf{Q}_{\mathrm{H}}^A - \mathbf{P}_{\mathrm{H}}^A(\mathbf{P}^A)^t\mathbf{\Psi}_{\mathrm{M,H}}\mathbf{Q}_{\mathrm{H}}^A\right\}\end{aligned} \tag{16}$$

Thus PRESS^A can be re-written as

$$\begin{aligned}\mathrm{PRESS}^A = \mathrm{tr}\left\{\mathbf{\Psi} - \mathbf{Q}^A\mathbf{\Psi}(\mathbf{Q}^A)^t\right\} + \sum_{\mathrm{H}} \mathrm{tr}\left\{(\mathbf{Q}_{\mathrm{H}}^A)^t\mathbf{\Psi}_{\mathrm{H,H}}\mathbf{Q}_{\mathrm{H}}^A\right\} + \\ 2\sum_{\mathrm{H}} \mathrm{tr}\left\{\mathbf{\Psi}_{\mathrm{H,H}}\mathbf{Q}_{\mathrm{H}}^A - \mathbf{P}_{\mathrm{H}}^A(\mathbf{P}^A)^t\mathbf{\Psi}_{\mathrm{M,H}}\mathbf{Q}_{\mathrm{H}}^A\right\}\end{aligned} \tag{17}$$

This ends the proof. The resulting kernel algorithm is presented in Algorithm box 4. This kernel algorithm is expected to be more suitable than the one given in Algorithm box 3 when the number of observations is large enough so that matrix $\mathbf{\Psi}$ can only be iteratively computed [25]. It should be noted that different data sets can yield the same covariance matrix: this would yield different PRESS curves according to *ekf* but equal PRESS curves according to *ckf*. Thus, the *ckf* is much faster but less specific to the fitting data set than *ekf*. Therefore, by comparing both algorithms, we can conclude whether the k-fold operation in *ekf* is convenient from a practical point of view or not.

4. Comparison of *ekf* and *ckf* with simulated data

In this section we discuss the practical implication of using *ckf* instead of *ekf*. A first and obvious consequence of using *ckf* in place of *ekf* is the different computational time required. In Table 1 the computational times of both algorithms on matrices of different sizes in a specific setup are compared. The *ekf* is used in two modes: observation-wise leave-one-out (loo) and with a 7-fold split. When the number of observations is large, a common approach is to use a reduced number of folds to perform CV, instead of the loo operation. As expected, the *ckf* is much faster than any version of *ekf*. This result is especially relevant for very large matrices, in the observations and/or in the variables, since the

Fit a PCA model with A_{max} PCs from $\mathbf{XX}$, obtaining $\mathbf{P}^{A_{max}}$
For each PC ($A = 1...A_{max}$)
    $\mathbf{Q}^A = \mathbf{P}^A(\mathbf{P}^A)^t$
    $\mathbf{R}^A = \mathbf{\Psi} - \mathbf{Q}^A\mathbf{\Psi}(\mathbf{Q}^A)^t$
    $\text{PRESS}^A = \text{tr}(\mathbf{R}^A)$
    For each group of variables ($\text{H} = 1...H_{tot}$)
        Select the rows of $\mathbf{P}^A$ in H yielding $\mathbf{P}^A_{\text{H}}$
        $\mathbf{Q}^A_{\text{H}} = \mathbf{P}^A_{\text{H}}(\mathbf{P}^A_{\text{H}})^t$
        $\text{PRESS}^A_H = \text{PRESS}^A_H + \text{tr}((\mathbf{Q}^A_{\text{H}})^t\mathbf{\Psi}_{\text{H,H}}\mathbf{Q}^A_{\text{H}}) + 2\,\text{tr}(\mathbf{\Psi}_{\text{H,H}}\mathbf{Q}^A_{\text{H}})$
        $\text{PRESS}^A_H = \text{PRESS}^A_H - 2\,\text{tr}(\mathbf{P}^A_{\text{H}}(\mathbf{P}^A)^t\mathbf{\Psi}_{\text{M,H}}\mathbf{Q}^A_{\text{H}})$
    end
end

Algorithm box 4: Column-wise k-fold (*ckf*) kernel algorithm.

computational time of *ekf* is too large to be used in those circumstances. This makes *ckf* an interesting and attractive alternative to *ekf* for large data sets. Also, the kernel *ckf* is suitable for data sets in which the number of observations is much larger than that of variables.

Other aspects to be evaluated are the price to pay for this improvement in computational efficiency and the actual benefit of using the observation-wise k-fold operation within the cross-validation. As already discussed in Section 3, the *ekf* takes as input the original data matrix while *ckf* takes only the covariance matrix. When the covariance matrix is obtained from the original data matrix, the original distribution of the observations is lost. Indeed, as shown in [24], very different data distributions can lead to the same covariance matrix. For this reason, the *ekf* is expected to be more specific to the actual data set than the *ckf*: an expected advantage of the former could be that the PRESS curve depends, to some extent, on the data distribution. To check whether this is the case, the ADICOV algorithm [24] is used to create three different data sets for the same, exact, covariance matrix. These distributions are a multinormal distribution, a distribution with one outlier and a distribution with two multinormal clusters. This experiment is repeated twice for two different covariance matrices. The first one presents equally separated eigenvalues, so that each eigenvalue is one order of magnitude higher that the following. In the second one, this difference was only present between the first and second eigenvalues, while the remaining were of a similar to the second one. Results are shown in Figures 1 and 2, respectively.

From Figure 1 a clear conclusion can be derived: there are circumstances where the observation-wise k-fold operation does not provide any hint about data distribution. This can be seen from the leave-one out *ekf* cross-validation PRESS curves which are the indistinguishable although the three data sets show different variable distributions (random multinormal, with 1 outlier and a two clusters multinormal distribution). Moreover, the PRESS curves are the

same for the 7-fold *ekf* and, remarkably, for *ckf*. Thus, in this simulation, the observation wise k-fold operation does not carry information about the data distribution.

The results shown in Figure 2 lead to similar conclusions for what concern both loo *ekf* and 7-fold *ekf* which give similar PRESS curves for the three different data distributions. In the three cases, 1 principal component would be selected. Again, this shows that although *ekf* is expected to be more specific for the data set, this is not the case and it does not provide any insight on the data distribution.

Here the *ckf* is markedly different and indicates a different number of significant components. Let's recall that the second experiment was designed so that the first eigenvalue is one order of magnitude higher than the remaining. However, if the difference in variance between the first and the remaining eigenvalues grows of 2 orders of magnitude, *ekf* and *ckf* again provide the same PRESS and thus the same results (not shown).

Clearly, there are some specific cases where *ekf* and *ckf* provide different results and we further investigate these cases. In Figure 3, the PRESS curves shown in Figure 2 corresponding to loo *ekf* and *ckf* are decomposed in the three following terms of Equation (12):

$$\text{term1}^A = \sum_{\text{H}} \text{tr}\left\{(\mathbf{R}_{\text{H}}^A)^t \mathbf{R}_{\text{H}}^A\right\} \tag{18}$$

$$\text{term2}^A = \sum_{\text{H}} 2\,\text{tr}\left\{(\mathbf{R}_{\text{H}}^A)^t \mathbf{X}_{\text{H}} \mathbf{Q}_{\text{H}}^A\right\} \tag{19}$$

$$\text{term3}^A = \sum_{\text{H}} \text{tr}\left\{(\mathbf{Q}_{\text{H}}^A)^t (\mathbf{X}_{\text{H}})^t \mathbf{X}_{\text{H}} \mathbf{Q}_{\text{H}}^A)\right\} \tag{20}$$

According to [15], the first factor is monotonically increasing with A, the third factor is monotonically decreasing with A and the second one is a crossed factor between the other two with unpredictable tendency. Also, it should be noted that the first factor incorporates structural information in $\mathbf{Q}_{\text{H}}^A$ and that the third factor corresponds to the PRESS by *rkf*, criticized to violate the independence in the CV loop. This factor is only affected by the row-wise k-fold operation, and not the column-wise k-fold operation.

Figure 3 shows that the difference between *ekf* and *ckf* is found in terms 2 and 3. Since term 2 is a cross product of terms 1 and 3, it is term 3 which makes the difference. Therefore, we can conclude that the structural information in term 1 is mainly determined by the column-wise k-fold operation and not the row-wise k-fold operation. Furthermore, we can make the following equivalence: *ekf* has the same relationship with *ckf* that *rkf* has with a variance plot in terms of the number of components.

A last simulated experiment is performed in this section. We inherit the simulation approach of [14] which consists of four different data sets (D.1, D.2,

D.3 and D.4) where the number of latent variables (LVs) ranges from 4 to 15. The LVs are generated independently at random following a normal distribution with zero mean and unit variance. Observable variables (OVs) are computed from the LVs according to 4 generating rules, reproduced for convenience of the reader in the Appendix. These data sets are chemometrics-like data sets, similar to process variables (data set D.1) or spectral variables (data set D.4).

For each of the four rules, 100 noise-free data sets are generated containing 100 observations. At each data set $\mathbf{X}$ (dropping indexes for the sake of simplicity) random noise is added following a normal distribution with zero mean and given variance σ_n^2, $\mathcal{N}(0, \sigma_n^2)$. The noised data set are obtained as

$$\mathbf{X}_n = (\mathbf{X} + n\sqrt{\sigma_n}) / (\sqrt{1 + \sigma_n}) \tag{21}$$

where the subscript n indicates the level of noise generated, and σ_n equals 0.05, 0.1, 0.15, 0.2, 0.25 and 0.5 corresponding to 5%, 10%, 15%, 20%, 25% and 50% respectively. The noise percentages are computed such that the lowest standard deviation of a LV is 100%. For more details see [14].

Results of this simulated experiment are shown in Table 2. Clear differences are only found in the third data set and, to some smaller extent, in the forth one. The decomposition in the three terms (18)-(20) of the PRESS curve for data set D.3 is given in Figure 4. Similar conclusions do apply: differences are only found in the last two terms.

5. Comparison of *ekf* and *ckf* on real data

As stated by Jolliffe [26] and remarked by [27] *simulation of multivariate data sets can always be criticized as unrepresentative because they can never explore more than a tiny fraction of the vast range of possible correlations and covariance structure*. For this reason we compare the performance of the *ekf* and *ckf* algorithms on real data sets from different areas of research for which the dimensionality is not known *a priori* and for which also the error structure is mostly unknown. By making use of real data set we aim to establish how often and on what kind of data *ekf* and *ckf* would select a different (optimal) number of components.

Figure 5 allows a visual comparison of the PRESS curves of 4 different data sets. Two out of four, the second and third ones, show no difference in the PRESS curves of *ekf* and *ckf*. The PRESS for the first data set, a NIR spectroscopy data set, show different magnitudes but still the same shape of the PRESS, and thus the same number of significant components. It should be noted that in this case, the 7-fold *ekf* presented a very high variability, so that different runs of the CV led to very different numbers of components selected. This is a clear drawback of k-fold *ekf* that is only solved bu using a loo approach or *ckf*. Finally, clear differences appear in the PRESS obtained by *ekf* and *ckf* on the fourth example, a gene expression (microarray) data set.

Table 3 presents the results of the comparison of *ekf* and *ckf* on 15 more data sets in addition to the four previously described (Data sets 16 to 19).

The two CV-procedures provide the same results on 12 out 19 data sets. In three cases (data sets 5, 11 and 15) the difference between the estimates of the optimal number of components is 1, and this attributable to the rounding off of the PRESS minimum values. Only 4 out of 20 data sets (data sets 1, 4, 12 and 19) provided different results in the estimated number of components where *ckf* tends to overestimate the number of components in respect to *ekf*, although in the simulation data sets we observed the contrary.

6. Conclusions

Principal component analysis is one of the most commonly used multivariate tools to describe and summarize large data sets. Determining the optimal number of components that best fit the data is a fundamental task in the multivariate analysis of biochemical and biological data. A common approach for this is to use cross-validation. However, cross-validation is a costly operation, particularly when it includes the observation-wise k-fold operation.

In this paper, we have performed an in-depth study of the use of the observation-wise k-fold operation in the state-of-the-art PCA cross-validation element-wise k-fold (*ekf*) algorithm. As a result, we propose a variant of *ekf*, termed column wise k-fold (*ckf*) algorithm, in which the observation-wise k-fold operation is removed. We have theoretically shown that the *ckf* cross-validation procedure can be obtained from the data covariance matrix rather than from the data itself using a kernel algorithm; this is not possible for the *ekf*. This kernel algorithm is especially suited for very large data sets where the large number of observations renders *ekf* unfeasible either because of the large computational time required or because of allocation memory problems

By comparative investigation of *ekf* and *ckf* using both simulated and real data, we provide the following conclusion/suggestions

- The *ckf* is much faster than the *ekf*. Thus, *ckf* can be of practical use in data sets where *ekf* takes too long to compute.
- The observation wise k-fold operation does not provide *ekf* with the capability of highlighting different types of distribution. In particular, the observation wise k-fold operation is not robust against outliers. Considering that this operation makes the computation of *ekf* prohibitive in many cases, *ckf* seems to be a promising alternative.
- Algorithms *ckf* and *ekf* output the same estimation of the optimal number of components in most cases.
- Algorithms *ckf* and *ekf* treat PCA structural information in the same way. The difference between the two is determined by the way in which residual variance is computed. Thus, *ekf* keeps the same relationship with *ckf* that the simple row-wise k-fold (*rkf*) keeps with the residual variance plot in terms of the number of components.

- Algorithm *ekf* should be used with a leave-one-out operation in the observations to avoid variability of the results. This renders *ckf* the only valid alternative for fast computation.

As a main conclusion, *ckf* is a recommended algorithm to select the number of PCs in very large data sets: it performs similarly to *ekf* and it is computational efficient. We also suggest the use of *ckf* to perform a first cross-validation using 1 to $A_{max} = p - 1$ components to detect the presence of a global minimum in the *ckf* PRESS curve. The *ekf* could be then used to explore the behavior of PRESS in the surrounding of the minimum obtained by *ckf*. This will significantly reduce the computational time needed to cross-validate a data set.

Appendices

Computational and numerical methods

The Multivariate Exploratory Data Analysis Toolbox for Matlab (MEDA Toolbox) [28] was used to perform both *ekf* and *ckf* cross-validation and the (ADICOV) method [24]. The Toolbox is available at:
`github.com/josecamachop/MEDA-Toolbox/releases/tag/v1.0`.

In-house Matlab scripted routines were use for further analysis.

Smulated data sets

Generatin rules (R.1 to R.4) for the 4 data sets D.1 to D4 investigted in Section 4 as proposed in [14].

R.1 :

$$x_i = \sqrt{\frac{i}{0.5}} \cdot lv_1 + \sqrt{1 - \frac{i}{5}} \cdot lv_2 \quad \text{for } i \in 1, 2, \dots 5$$

$$x_i = \sqrt{0.5} \cdot lv_1 + \sqrt{\frac{i}{10} - 0.5} \cdot lv_2 + \sqrt{1 - \frac{i}{10}} \cdot lv_3 \quad \text{for } i \in 6, \dots 9$$

$$x_{10} = \left(\sqrt{0.01} \cdot lv_1 + \sqrt{0.01} \cdot lv_2 + \sqrt{0.01} \cdot lv_3 + lv_4\right) / \sqrt{1.03} \quad \text{for } i \in 6, \dots 9$$

R.2 :

$$x_i = \sqrt{0.5} \cdot lv_j + \sqrt{0.5} \cdot lv_k \quad \text{for } i \in 1, 2, \dots 6 \text{ and } \quad j \neq k \in 1, \dots 4$$

$$x_i = \sqrt{0.5} \cdot lv_j + \sqrt{0.5} \cdot lv_k \quad \text{for } i \in 7, 8, 9 \text{ and } \quad j \neq k \in 5, 6, 7$$

$$x_{10} = lv_8$$

R.3 :

$$x_i = lv_i \quad \text{for } i \in 1, 2, \dots 12$$

$$x_i = \sqrt{0.5} \cdot lv_j + \sqrt{0.5} \cdot lv_k \quad \text{for } i \in 13, 14 \dots 27 \text{ and } \quad j \neq k \in 1, 2, \dots 6$$

R.4 :

$$x_i = \sqrt{0.5} \cdot lv_j + \sqrt{0.5} \cdot lv_k \quad \text{for } i \in 1, 2 \dots 45 \text{ and } \quad j \neq k \in 1, 2, \dots 10$$

$$x_{46} = lv_{11}, \quad x_{47} = lv_{12}$$

$$x_{48} = \sqrt{0.5} \cdot lv_{11} + \sqrt{0.5} \cdot lv_{13}$$

$$x_{49} = \sqrt{0.5} \cdot lv_{12} + \sqrt{0.5} \cdot lv_{14}$$

$$x_{50} = lv_{15}$$

7. Acknowledgments

This work was partly supported by the European Commission-funded FP7 project INFECT (Contract No.305340) and the Spanish Ministry of Economy and Competitiveness and FEDER funds from the through grant TIN2014-60346-R.

References

[1] D. Giancarlo and C. Tommasi, "Cross-validation methods in principal component analysis: a comparison," *Statistical Methods and Applications*, vol. 11, pp. 71–82, 2002.

[2] R. Bro, K. Kjeldahl, A. Smilde, and H. Kiers, "Cross-validation of component models: a critical look at current methods," *Analytical and bioanalytical chemistry*, vol. 390, no. 5, pp. 1241–1251, 2008.

[3] S. Wold, "Cross-validatory estimation of the number of components in factor and principal components models," *Technometrics*, vol. 20, no. 4, pp. 397–405, 1978.

[4] H. Eastment and W. Krzanowski, "Cross-validatory choice of the number of components from a principal component analysis," *Technometrics*, vol. 24, no. 1, pp. 73–77, 1982.

[5] P. Zhang, "Model selection via multifold cross validation," *The Annals of Statistics*, pp. 299–313, 1993.

[6] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees, wadsworth international group, belmont, ca, 1984," *Case Description Feature Subset Correct Missed FA Misclass*, vol. 1, pp. 1–3, 1993.

[7] P. R. Nelson, P. A. Taylor, and J. F. MacGregor, "Missing data methods in pca and pls: Score calculations with incomplete observations," *Chemometrics and intelligent laboratory systems*, vol. 35, no. 1, pp. 45–65, 1996.

[8] P. R. Nelson, J. F. MacGregor, and P. A. Taylor, "The impact of missing measurements on pca and pls prediction and monitoring applications," *Chemometrics and intelligent laboratory systems*, vol. 80, no. 1, pp. 1–12, 2006.

[9] F. Arteaga and A. Ferrer, "Dealing with missing data in mspc: several methods, different interpretations, some examples," *Journal of chemometrics*, vol. 16, no. 8-10, pp. 408–418, 2002.

[10] F. Arteaga and A. Ferrer, "Framework for regression-based missing data imputation methods in on-line mspc," *Journal of Chemometrics*, vol. 19, pp. 439–447, 2005.

[11] B. Walczak and D. Massart, "Dealing with missing data. part 1," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, pp. 15–17, 2001.

[12] J. Josse, J. Pagès, and F. Husson, "Multiple imputation in principal component analysis," *Advances in data analysis and classification*, vol. 5, no. 3, pp. 231–246, 2011.

[13] V. Audigier, F. Husson, and J. Josse, "A principal component method to impute missing values for mixed data," *Advances in Data Analysis and Classification*, pp. 1–22, 2013.

[14] J. Camacho and A. Ferrer, "Cross-validation in pca models with the element-wise ekf algorithm: Practical aspects," *Chemometrics and Intelligent Laboratory Systems*, vol. 131, pp. 37–50, 2014.

[15] J. Camacho and A. Ferrer, "Cross-validation in pca models with the element-wise k-fold (ekf) algorithm: theoretical aspects," *Journal of Chemometrics*, vol. 26, no. 7, pp. 361–373, 2012.

[16] J. Josse and F. Husson, "Selecting the number of components in principal component analysis using cross-validation approximations," *Computational Statistics & Data Analysis*, vol. 56, no. 6, pp. 1869–1879, 2012.

[17] S. Kritchman and B. Nadler, "Determining the number of components in a factor model from limited noisy data," *Chemometrics and Intelligent Laboratory Systems*, vol. 94, no. 1, pp. 19–32, 2008.

[18] E. Saccenti and J. Camacho, "A comparison of methods for determining the number of components in principal components analysis based on random matrix theory, cross-validation and numerical approximation," *Submitted*, 2014.

[19] F. Arteaga and A. Ferrer, "Dealing with missing data in mspc: several methods, different interpretations, some examples," *Journal of Chemometrics*, vol. 16, pp. 408–418, 2002.

[20] J. Camacho, J. Picó, and A. Ferrer, "Data understanding with pca: structural and variance information plots," *Chemometrics and Intelligent Laboratory Systems*, vol. 100, no. 1, pp. 48–56, 2010.

[21] MATLAB, *version 8.10.0 (R2012b)*. Natick, Massachusetts: The MathWorks Inc., 2012.

[22] J. W. Eaton, D. Bateman, and S. Hauberg, *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2009. ISBN 1441413006.

[23] Octave community, "GNU Octave 3.8.1," 2014.

[24] J. Camacho, P. Padilla, J. Díaz-Verdejo, K. Smith, and D. Lovett, "Least-squares approximation of a space distribution for a given covariance and latent sub-space," *Chemometrics and Intelligent Laboratory Systems*, vol. 105, no. 2, pp. 171–180, 2011.

[25] J. Camacho Páez, "Visualizing Big data with Compressed Score Plots: Approach and Research Challenges.," *Chemometrics and Intelligent Laboratory Systems*, vol. 135, pp. 110–125, 2014.

[26] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[27] S. Dray, "On the number of principal components: A test of dimensionality based on measurements of similarity between matrices," *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 2228–2237, 2008.

[28] J. Camacho, A. Pérez-Villegas, R. A. Rodríguez-Gómez, and E. Jiménez-Mañas, "Multivariate exploratory data analysis (meda) toolbox for matlab," *Chemometrics and Intelligent Laboratory Systems*, vol. 143, pp. 49–57, 2015.

[29] E. Saccenti and L. Tenori, "Multivariate modeling of the collaboration between luigi illica and giuseppe giacosa for the librettos of three operas by giacomo puccini," *Literary and Linguistic Computing*, p. fqu006, 2014.

[30] E. Saccenti, M. Suarez-Diez, C. Luchinat, C. Santucci, and L. Tenori, "Probabilistic networks of blood metabolites in healthy subjects as indicators of latent cardiovascular risk," *Journal of proteome research*, 2014.

[31] L. Tenori, X. Hu, P. Pantaleo, B. Alterini, G. Castelli, I. Olivotto, I. Bertini, C. Luchinat, and G. F. Gensini, "Metabolomic fingerprint of heart failure in humans: a nuclear magnetic resonance spectroscopy analysis," *International journal of cardiology*, vol. 168, no. 4, pp. e113–e115, 2013.

[32] E. Saccenti, J. A. Westerhuis, A. K. Smilde, M. J. van der Werf, J. A. Hageman, and M. M. Hendriks, "Simplivariate models: uncovering the underlying biology in functional genomics data," *PloS one*, vol. 6, no. 6, p. e20747, 2011.

[33] P. Bernini, I. Bertini, C. Luchinat, S. Nepi, E. Saccenti, H. Schaefer, B. Schuetz, M. Spraul, and L. Tenori, "Individual human phenotypes in metabolic space and time," *Journal of proteome research*, vol. 8, no. 9, pp. 4264–4271, 2009.

[34] "Retrieved at wwww.metaboanalyst.ca,"

[35] "Matlab, the mathworks inc," *Natick, MA*, 2015.

[36] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[37] S. Aeberhard, D. Coomans, and O. De Vel, "Comparative analysis of statistical pattern recognition methods in high dimensional settings," *Pattern Recognition*, vol. 27, no. 8, pp. 1065–1077, 1994.

[38] C. Q. Fang Zhou and R. D. King, "Predicting the geographical origin of music," *ICDM*, 2014.

[39] J. Christensen, L. Nørgaard, H. Heimdal, J. G. Pedersen, and S. B. Engelsen, "Rapid spectroscopic analysis of marzipanâĂŤcomparative instrumentation," *Journal of near infrared spectroscopy*, vol. 12, no. 1, pp. 63–75, 2004.

[40] J. Christensen, E. M. Becker, and C. Frederiksen, "Fluorescence spectroscopy and parafac in the analysis of yogurt," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 2, pp. 201–208, 2005.

[41] C. M. Andersen and R. Bro, "Quantification and handling of sampling errors in instrumental measurements: a case study," *Chemometrics and intelligent laboratory systems*, vol. 72, no. 1, pp. 43–50, 2004.

[42] M. Hubert, P. J. Rousseeuw, and K. Vanden Branden, "Robpca: a new approach to robust principal component analysis," *Technometrics*, vol. 47, no. 1, pp. 64–79, 2005.

[43] W. McReynolds, "Characterization of some liquid phases," *Journal of Chromatographic Science*, vol. 8, no. 12, pp. 685–691, 1970.

[44] I. Bertini, A. Calabro, V. De Carli, C. Luchinat, S. Nepi, B. Porfirio, D. Renzi, E. Saccenti, and L. Tenori, "The metabonomic signature of celiac disease," *Journal of proteome research*, vol. 8, no. 1, pp. 170–177, 2008.

[45] M. Imielinski, R. N. Baldassano, A. Griffiths, R. K. Russell, V. Annese, M. Dubinsky, S. Kugathasan, J. P. Bradfield, T. D. Walters, P. Sleiman, *et al.*, "Common variants at five new loci associated with early-onset inflammatory bowel disease," *Nature genetics*, vol. 41, no. 12, pp. 1335–1340, 2009.

Figures

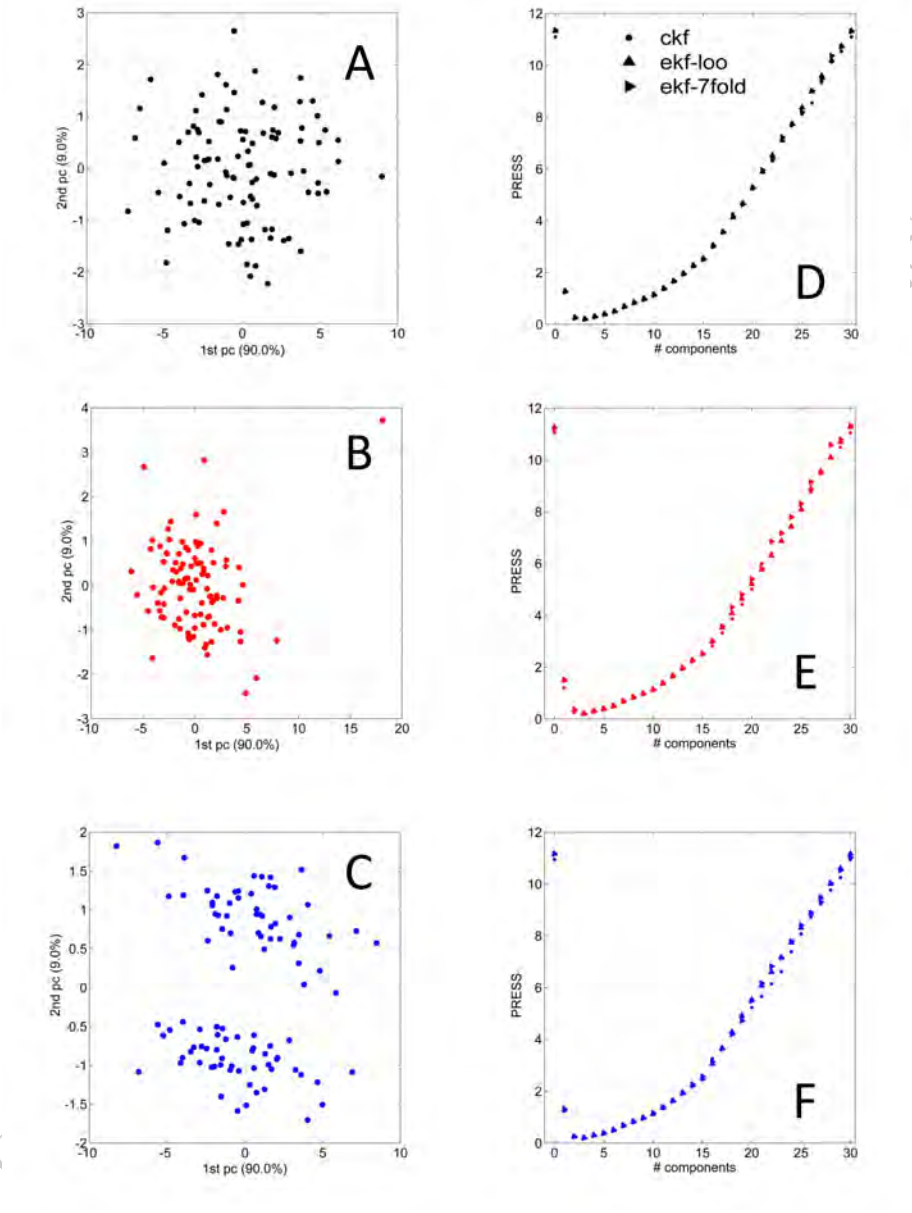

Figure 1: Score plots for of three different data distributions that provide the same PCA loadings, computed using ADICOV [24]: a multinormal distribution (A), a distribution with one outlier (B) and a distribution with two clusters (C). Panels D, E and F show the CV PRESS curves for the 3 data sets with observation-wise leave-one-out (*ekf*-loo) *ekf*, 7-fold *ekf* and *ckf*.

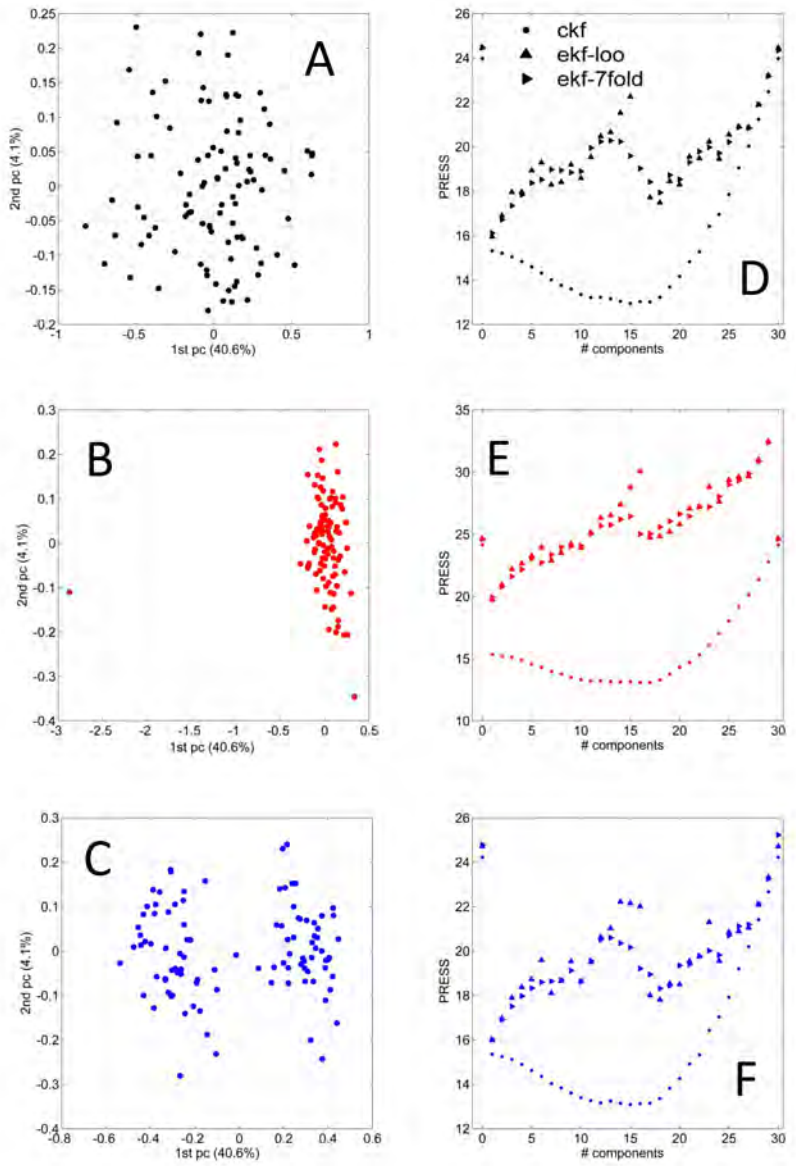

Figure 2: Score plots for of three different data distributions that provide the same PCA loadings, computed using ADICOV [24]: a multinormal distribution (A), a distribution with one outlier (B) and a distribution with two clusters (C). Panels D, E and F show the CV PRESS curves for the 3 data sets with observation-wise leave-one-out (*ekf*-loo) *ekf*, 7-fold *ekf* and *ckf*.

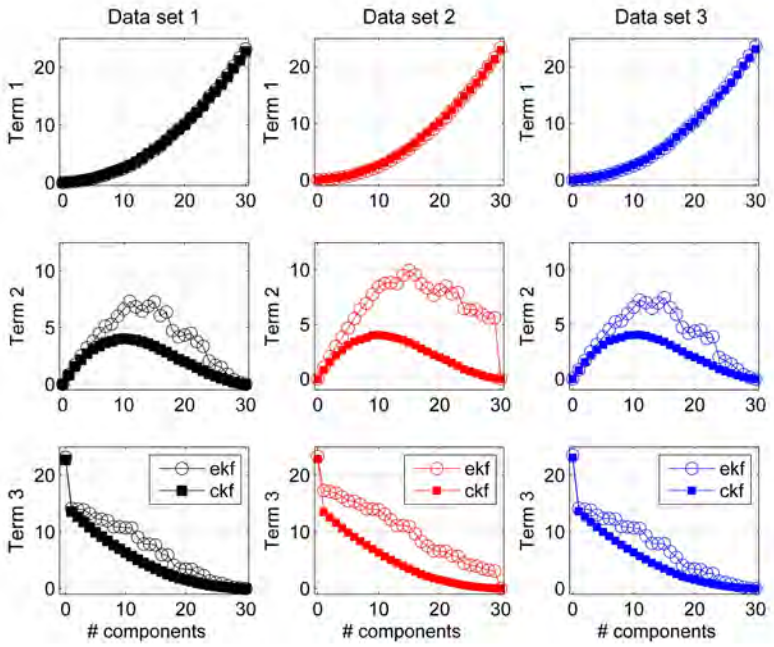

Figure 3: Decomposition in the three terms of PRESS (see Equation (17) and Equations (18)-(20)) of the PRESS curves corresponding to the multinormal distribution (Data set 1), the distribution with one outlier(Data set 2) and the distribution with two clusters (Data set 3) shown in Figure 2 (Panels A, B and C).

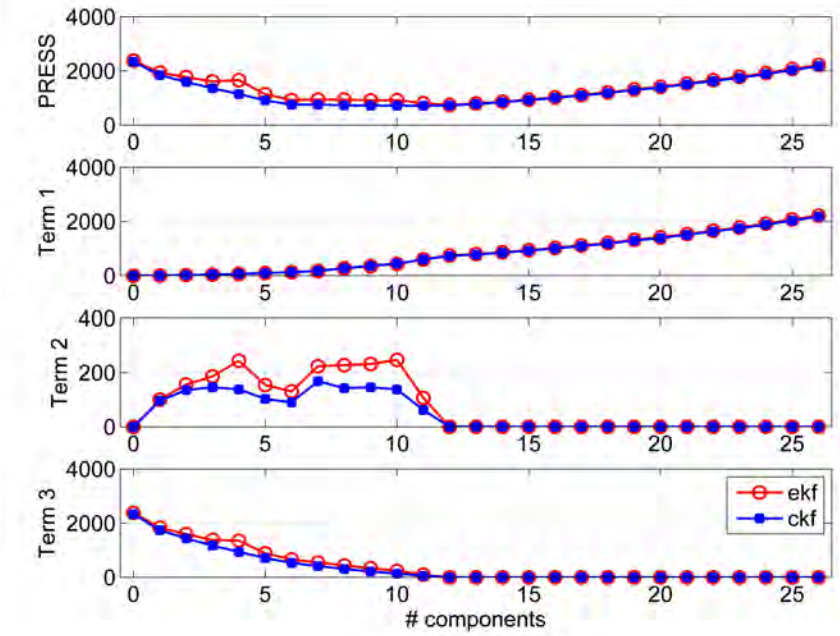

Figure 4: PRESS curve for the cross-validation of data set D.3 of the simulation study (panel A). Plot of the three terms (term1, term2 and term3, Equations (18), (19), and (20), respectively) summing up to make the PRESS.

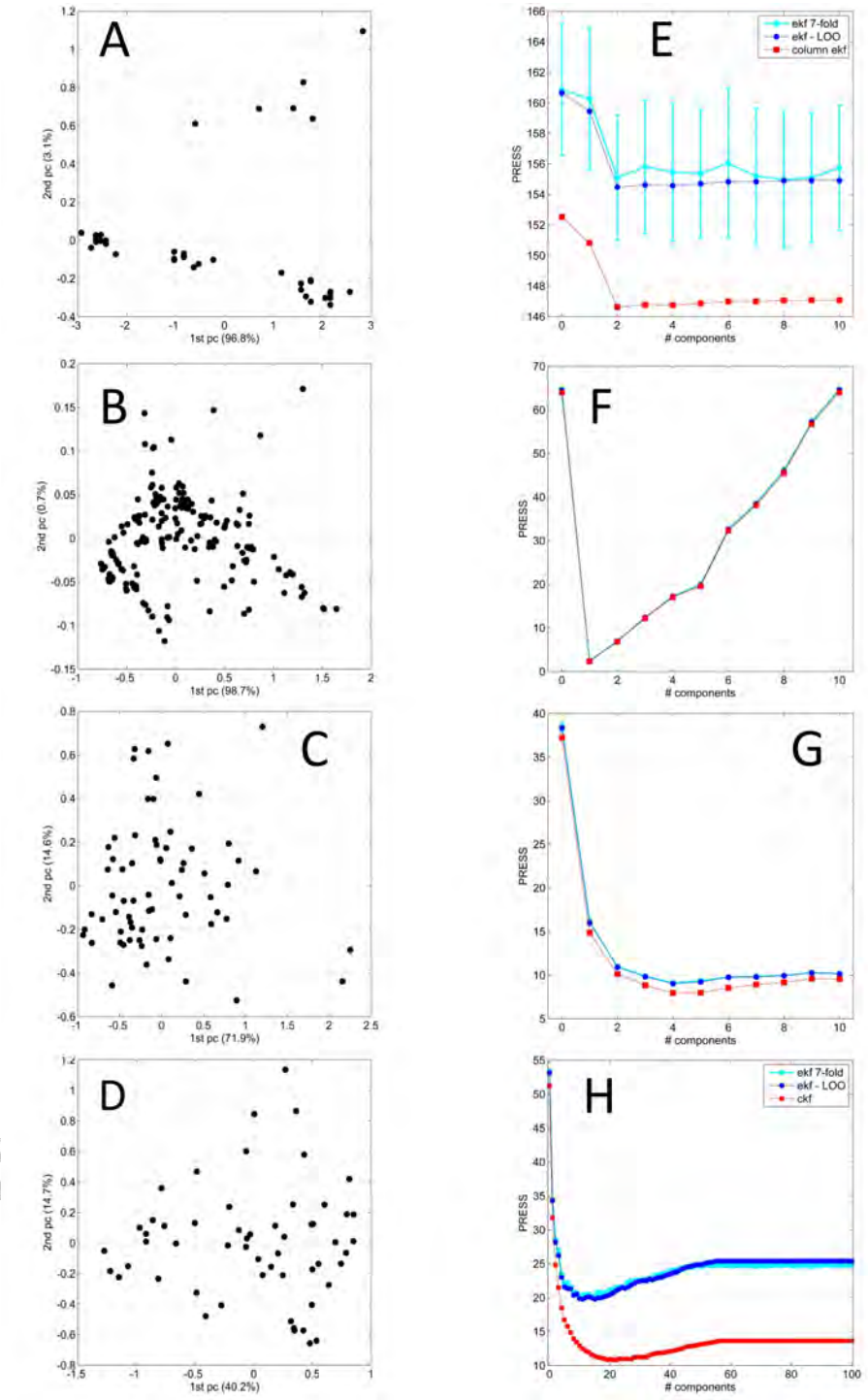

Figure 5: Score plot and PRESS cross-validation curves for four different data sets. From left to right: A-E) chemometrics (NIR data); B-F) chemometrics (McReynolds data), C-G) metabolomics data (NMR data) and D-H) genomics (microarray) MA. Cross=validation results for these data sets are reported also in Table 3 (data sets number 16 to 19)). Data sets have been cross-validated using the *ekf* algorithm with both leave-one (loo) out and 7-fold procedure for the observation-wise CV step and with the new *ckf* algorithm.

Tables

	Computational time			
Data set size	***ekf-loo***	***ekf-7fold***	***ckf***	***ckf-kernel***
10×10	<0.1 s	<0.1 s	<0.1 s	<0.1 s
100×10	<0.1 s	<0.1 s	<0.1 s	<0.1 s
100×100	2.0 s	0.2 s	0.1 s	0.4 s
100×1000	1.1 m	5.0 s	2.2 s	47.4 s
1000×100	46.0 s	0.7 s	0.8 s	0.4 s
1000×200	3.0 m	4.0 s	4.7 s	2.5 s
1000×500	1.0 h	59.2 s	37.7 s	58.9 s
1000×1000	7.4 h	7.4 m	3.6 m	3.7 h

Table 1: Computational time for the three different CV algorithms: *ekf* with leave-one out (loo), *ekf* with 7 fold CV (7fold) and *ckf*. The maximum number of components for which different models are fitted is set equal to the rank of the data matrix. Calculation has been performed on a Intel CPU Dual-Core Pentium E5200, running Windows 7 Enterprise equipped with 4 GB ram. PCA is performed via SVD.

	Data set D.1 (4/8)		
Noise level (%)	**ckf**	**ekf-loo**	**ekf-7fold**
0	3.8 (0.4)	3.8 (0.4)	3.9 (0.3)
5	3.7 (0.5)	4.0 (0.0)	4.0 (0.0)
10	3.7 (0.5)	3.9 (0.3)	3.9 (0.3)
15	3.6 (0.5)	3.7 (0.5)	3.8 (0.4)
20	3.6 (0.5)	4.0 (0.0)	4.0 (0.0)
25	3.8 (0.4)	3.9 (0.3)	3.9 (0.3)
50	3.6 (0.5)	3.8 (0.4)	3.8 (0.4)
	Data set D.2 (8/10)		
Noise level (%)	**ckf**	**ekf-loo**	**ekf-7fold**
0	5.9 (0.3)	5.9 (0.3)	6.0 (0.5)
5	5.9 (0.3)	6.0 (0.5)	6.0 (0.5)
10	5.9 (0.3)	6.1 (0.6)	6.3 (0.8)
15	6.0 (0.0)	6.0 (0.0)	6.1 (0.3)
20	6.0 (0.0)	6.1 (0.3)	6.1 (0.3)
25	5.9 (0.3)	6.1 (0.6)	6.2 (0.6)
50	5.9 (0.3)	6.1 (0.6)	6.1 (0.6)
	Data set D.3 (12/27)		
Noise level (%)	**ckf**	**ekf-loo**	**ekf-7fold**
0	9.1 (1.7)	12.0 (0.0)	12.0 (0.0)
5	9.5 (1.4)	11.9 (0.3)	12.0 (0.0)
10	10.0 (1.9)	12.0 (0.0)	11.9 (0.3)
15	9.9 (1.3)	12.0 (0.0)	12.0 (0.0)
20	8.9 (1.7)	12.0 (0.0)	12.0 (0.0)
25	9.6 (2.0)	12.0 (0.0)	11.9 (0.3)
50	9.2 (2.0)	12.0 (0.0)	12.0 (0.0)
	Data set D.4 (15/50)		
Noise level (%)	**ckf**	**ekf-loo**	**ekf-7fold**
0	12.4 (0.5)	13.0 (0.0)	13.0 (0.0)
5	12.3 (0.5)	13.0 (0.0)	13.0 (0.0)
10	12.4 (0.5)	13.0 (0.0)	13.0 (0.0)
15	12.5 (0.5)	13.0 (0.0)	13.0 (0.0)
20	12.6 (0.5)	13.0 (0.0)	13.0 (0.0)
25	12.5 (0.5)	13.0 (0.0)	13.0 (0.0)
50	12.5 (0.5)	13.0 (0.0)	13.0 (0.0)

Table 2: Estimation of the number of component for four data sets: D.1 (4/8 significant components), D.2 (8/10), D.3 (12/27) and D.4 (15/50) obtained using the three different CV algorithms: *ekf* with leave-one out (loo), *ekf* with 7 fold CV (7fold) and *ckf*.

Data set	Description	Size	Reference	Number of Component		
				ekf-loo	***ekf-7fold***	***ckf***
1	Computational linguistics	71×317	[29]	14	14	18
2	NMR metabolomics	978×29	[30]	2	2	2
3	NMR metabolomics	994×29	[31]	1	1	1
4	GC-MS proteomic	39×590	[32]	5	5	8
5	NMR metabolomics	705×373	[33]	15	16	15
6	NMR metabolomics	77×63	[34]	1	1	1
7	NIR gasoline	60×401	[35]	6	6	6
8	Fisher's Iris data	150×4	[36]	1	1	1
9	City rating	329×9	[35]	1	1	1
10	Wine type attributes	178×14	[37]	1	1	1
11	Music origins	1059×70	[38]	26	25	25
12	NIR on marzipan	32×1000	[39]	12	13	10
13	Florescence on yogurt	125×15	[40]	1	1	1
14	NMR low field	908×256	[41]	4	4	4
15	GS-MS	12×409	[34]	9	8	8
16	NIR	39×227	[42]	2	2	2
17	Chromatography	212×10	[43]	1	1	1
18	NMR metabolomics	68×416	[44]	4	4	4
19	Microarray	56×1000	[45]	11	11	22

Table 3: Estimation of the number of component for 19 different experimental from different disciplines obtained using the three different CV algorithms: *ekf* with leave-one out (loo), *ekf* with 7 fold CV (7fold) and *ckf*. PCA plots and PRESS curves for data set 16 to 19 are given in Figure 5.