Procrustes cross-validation of multivariate regression models

Kucheryavskiy, Sergey; Rodionova, Oxana; Pomerantsev, Alexey

Published in:
Analytica Chimica Acta

DOI (link to publication from Publisher):
10.1016/j.aca.2023.341096

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Analytica Chimica Acta 1255 (2023) 341096

Contents lists available at ScienceDirect

Analytica Chimica Acta

journal homepage: www.elsevier.com/locate/aca

Procrustes cross-validation of multivariate regression models

Sergey Kucheryavskiy [a,*], Oxana Rodionova [b], Alexey Pomerantsev [b]

[a] Department of Chemistry and Bioscience, Aalborg University, Esbjerg, Denmark
[b] Federal Research Center for Chemical Physics RAS, Moscow, Russia

HIGHLIGHTS

- A generalization for Procrustes cross-validation approach.
- Can be used for validation of most chemometric methods.
- Provides additional tools for exploring the data heterogeneity and validation quality.

ARTICLE INFO

Handling Editor: Prof. L. Buydens

ABSTRACT

A generalization of Procrustes Cross-Validation — recently introduced novel approach for validation of chemometric models — is proposed. The generalized approach is faster than its predecessor by several orders of magnitude and can be used for validation of a wider range of models. Furthermore, it provides new tools for exploring the heterogeneity of the dataset, quality of cross-validation splits, presence of outliers, etc. The paper describes methodological aspects of the generalized approach, based on using Procrustean rules, the mathematical background, as well as presents practical results obtained using real chemical datasets.

1. Introduction

Validation — testing of a model performance for exploration, optimization and developing — is one of the corner stones in chemometrics [1]. There are two main validation approaches currently in use [2]. The first is based on employing a dedicated set of measurements or observations — a validation set. Data for the validation set can be obtained independently, similar to the test set, or, if the original dataset is large enough, it can be split randomly to the training and the validation subsets.

Using the dedicated validation set is preferable for several reasons. First of all, it mimics the use of the model in real life — making prediction for a new data — and, if properly taken, it estimates the sampling error in the optimal way [1]. Second, it explicitly evaluates the performance of the model and provides all possible outcomes (e.g. scores, explained and residual variances, etc.) in a direct unambiguous form.

Alternative to the validation set is cross-validation, which is based on iterative resampling of the training set [3]. Cross-validation does not directly evaluate the performance of the model of interest (from now on referenced as the *global model*). Instead, it develops several *local models*, with the same parameters as the global model, and collects the performance results from these different models together. This limits the cross-validation outcomes to the performance statistics mainly. However, in most of cases, this is acceptable for optimization purposes, assuming that the performance of the final model will be later assessed using an independent test set.

Another drawback is that cross-validation is an iterative procedure, in which the local models are calibrated and validated at each cross-validation step. This can take a long time for large datasets as well as in case of complex optimization processes, which involve not only estimation of the optimal model complexity, but also the selection of the best combination of preprocessing methods and/or the most important variables.

Recently we proposed a new approach for validation of Principal Component Analysis (PCA) and Soft Independent Modeling of Class Analogies (SIMCA) models — Procrustes cross-validation (PCV), which can be considered as an alternative to the conventional cross-validation (CCV) [4]. The idea of the approach is to utilize cross-validation to obtain variation between the global and the local models. Then this variation is introduced into the training set, thus creating a new set of data — *pseudo-validation* (PV) set. Once created, the PV-set can be applied for validation of the global model similar to the dedicated

* Corresponding author.
E-mail address: svk@bio.aau.dk (S. Kucheryavskiy).

https://doi.org/10.1016/j.aca.2023.341096
Received 4 December 2022; Received in revised form 4 March 2023; Accepted 13 March 2023
Available online 22 March 2023

validation set. Since the first publication, the method has been applied by several researchers for solving different problems and has proven its efficiency [5–8], especially in the case of short datasets, when the number of objects is much smaller than the number of variables [9].

This paper reports recent developments of this approach. First of all, we present a new algorithm for PCV, which is in by several orders of magnitude faster than the original one proposed in Ref. [4]. Second, we generalize the PCV approach, so that it can be used for validation of regression models, in particular, principal component regression (PCR) and partial least squares regression (both PLS1 and PLS2). Finally we show that PCV also provides additional tools for exploration and quality control of both data and models.

All calculations and plots in this paper are done using R (v. 4.2.2) supplemented with package *mdatools* [10] (v. 0.13.1) and several additional scripts. All presented algorithms are implemented as an R package and a MATLAB Toolbox. Both can be installed from official repositories (CRAN for R and File Exchange for MATLAB). The source code for both as well as all technical details are freely available from GitHub (https://github.com/svkucheryavski/pcv). We also made an interactive web-application where anyone can upload data as a CSV file and create a PV-set: https://mda.tools/pcv/. The application works directly in a user's browser and does not send the uploaded data anywhere.

2. Methods and algorithms

2.1. Cross-validation

Cross-validation is an iterative resampling method, where at each iteration, k, $k = 1,...K$, the original (global) training set, represented as the $I \times J$ matrix $\mathbf{X}$, is split into local validation set, the $I_k \times J$ matrix $\mathbf{X}_k$, and local calibration set, the $(I - I_k) \times J$ matrix $\widetilde{\mathbf{X}}_k$ ($I_k = I/K$). The split can be done using random or systematic approach (e.g. venetian blinds); more details about the implementation of cross-validation can be found elsewhere [3].

For the response variables, which are represented by the $I \times L$ matrix $\mathbf{Y}$, we use a similar notation: the $(I - I_k) \times L$ matrix $\widetilde{\mathbf{Y}}_k$ for a local calibration set and the $I_k \times L$ matrix $\mathbf{Y}_k$ for a local validation set. This notation is used for both uni- and multivariate response, assuming that $L = 1$ in case of a single response.

After splitting the data, the local calibration set is used to train a local model, $\mathcal{M}_k$, which is then applied to the local validation set, resulting in a structure with outcomes, $\mathbf{R}_k$. This structure contains matrices/vectors, such as the predicted response values, the orthogonal and score distances, etc. The outcomes obtained for all K iterations are combined into performance characteristics (e.g. the total residual distance or the root mean squared error) or used as they are.

2.2. Procrustes cross-validation

Our original approach for Procrustes cross-validation was based on Procrustean rotation [4]. In this paper we propose a generalization of this approach, based on *Procrustean rules*, which are defined as *standards that are enforced uniformly without regard to individuality*. In this case the principles of Procrustes cross-validation can be defined as follows.

Let's consider a global training set $\mathbf{X}$ (or $\{\mathbf{X}, \mathbf{Y}\}$ in case of regression). For the sake of clarity we continue the description using $\mathbf{X}$ matrix only. The global training set is used to train a global model, $\mathcal{M}$, with A latent variables. The model is represented using a set of its parameters (e.g. loadings, weights, regression coefficients, etc.).

In the conventional cross-validation (CCV) procedure, the global training set, $\mathbf{X}$, is split into K segments. For a particular segment k we have a local calibration set, $\widetilde{\mathbf{X}}_k$, and a local validation set $\mathbf{X}_k$. The local calibration set, $\widetilde{\mathbf{X}}_k$, is used to develop a local model, $\mathcal{M}_k$, which is then applied to the local validation set, $\mathbf{X}_k$, producing a set of outcomes, $\mathbf{R}_k$, which has the same number of elements as the number of rows (objects) in $\mathbf{X}_k$ (I_k).

Let us suppose that we have developed a pseudo-validation set that is the $I \times J$ matrix $\mathbf{X}_{pv}$. Then for any segment, k, we take a subset of $\mathbf{X}_{pv}$, the $I_k \times J$ matrix $\mathbf{X}_{pv_k}$ that has the same rows as in $\mathbf{X}_k$, and apply the global model $\mathcal{M}$ to this subset. This results in a set of outcomes $\mathbf{R}_{pv_k}$. The goal is for all values in $\mathbf{R}_{pv_k}$ to be as close as possible to the corresponding values from $\mathbf{R}_k$:

$$\mathbf{R}_{pv_k} \approx \mathbf{R}_k \tag{1}$$

This should be held for any number of components, $a = 1,...,A$, and for all segments $k = 1,...K$. Or in a more general way:

$$\mathbf{R}_{pcv} \approx \mathbf{R}_{ccv} \tag{2}$$

In other words, when a pseudo-validation set is used to validate the global model, the outcomes should be as close as possible (ideally, identical) to the CCV outcomes for any level of complexity (the number of latent variables) up to the selected limit of A.

The choice of the outcomes to be included in $\mathbf{R}$ (and hence constrain the PV-set calculations) depends on a particular method and requirements. For example, it can include orthogonal and score distance in case of PCA or SIMCA, or y-residuals in case of regression models. In the next sections we present the implementation of this general approach for several different cases.

2.3. Procrustes cross-validation for PCA/SIMCA models

The PCA method decomposes a matrix $\mathbf{X}$ using A principal components as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} = \widehat{\mathbf{X}} + \mathbf{E} \tag{3}$$

Here we assume that the columns of $\mathbf{X}$ are mean centered or autoscaled.

The loadings matrix, $\mathbf{P}$, whose columns are orthonormal vectors, defines the direction of the principal components. They can be found as the eigenvectors of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ using, for example, the truncated Singular Values Decomposition (SVD) [11]. The score matrix, $\mathbf{T}$, is computed by projecting the matrix $\mathbf{X}$ using the loading matrix $\mathbf{P}$:

$$\mathbf{T} = \mathbf{X}\mathbf{P} \tag{4}$$

The amount of variation, captured by each component, can be estimated using the eigenvalues that are collected in vector $\boldsymbol{\lambda}$. Hence a PCA model is unambiguously represented by a set of three model parameters, $\mathcal{M} = \{A, \mathbf{P}, \boldsymbol{\lambda}\}$.

For any object $\mathbf{x}_i$, which can be new or part of a training set, the prediction is made by using the following procedure:

1. Compute vector with scores: $\mathbf{t}_i = \mathbf{x}_i\mathbf{P}$.
2. Compute vector with residuals: $\mathbf{e}_i = \mathbf{x}_i - \mathbf{t}_i\mathbf{P}^{\mathrm{T}}$.

The results are then summarized by computing two distances. These are the orthogonal distance, q_i, which is the squared Euclidean distance between the original data point and its projection:

$$q_i = \sum_{a=1}^{A} e_{ia}^2 \tag{5}$$

and the score distance, h_i, which is the squared Mahalanobis distance between the projection and the PC space origin:

$$h_i = \sum_{a=1}^{A} \frac{t_{ia}^2}{\lambda_a} \tag{6}$$

Here t_{ia} and e_{ia} are the elements of vectors $\mathbf{t}_i$ and $\mathbf{e}_i$, and λ_a are the elements of vector $\boldsymbol{\lambda}$.

Hence the relationship between a data point, $\mathbf{x}_i$, and any PCA model,

$\mathscr{M} = \{A, \mathbf{P}, \boldsymbol{\lambda}\}$, can be determined using two vectors of distances, $\mathbf{q} = \{q_i\}$ and $\mathbf{h} = \{h_i\}$. These two distances are the outcomes that are included into the set $\mathbf{R}$, introduced in the previous section.

Now we can define the requirements for the PV-set that is used to validate PCA and SIMCA models. For a given cross-validation segment k, the local PCA model, $\mathscr{M}_k = \{A, \mathbf{P}_k, \boldsymbol{\lambda}\}$, is as follows:

$$\widetilde{\mathbf{X}}_k = \widetilde{\mathbf{T}}_k \mathbf{P}_k^T + \widetilde{\mathbf{E}}_k \tag{7}$$

This model is applied to the local validation set:

$$\mathbf{T}_k = \mathbf{X}_k \mathbf{P}_k \tag{8}$$

$$\mathbf{E}_k = \mathbf{X}_k - \mathbf{T}_k \mathbf{P}_k^T \tag{9}$$

And the vectors of two distances are obtained:

$$\begin{aligned} \mathbf{q}_k &= \sum_{a=1}^{A} e_{ia}^2 \\ \mathbf{h}_k &= \sum_{a=1}^{A} \frac{t_{ia}^2}{\lambda_a} \end{aligned} \tag{10}$$

Here t_{ia} and e_{ia} are the elements of matrices $\mathbf{T}_k$ and $\mathbf{E}_k$. The two distance vectors are now included into the matrix of outcomes $\mathbf{R}_k$. Note that we do not compute vector $\boldsymbol{\lambda}$ for the local models but just reuse the one obtained from the global model.

Now we can create matrix $\mathbf{X}_{pv_k}$, such as:

$$\mathbf{T}_{pv_k} = \mathbf{X}_{pv_k} \mathbf{P} \tag{11}$$

$$\mathbf{E}_{pv_k} = \mathbf{X}_{pv_k} - \mathbf{T}_{pv_k} \mathbf{P}^T \tag{12}$$

The Procrustean rule is that the both distances, $\mathbf{q}_{pv_k}$ and $\mathbf{h}_{pv_k}$, should be equal to the local distances $\mathbf{q}_k$ and $\mathbf{h}_k$ for any number of PC, a, less than or equal to A ($a \leq A$).

If the full PCA decomposition is used (A equals to rank of $\mathbf{X}$), the $\mathbf{h}$ distances are equal if:

$$\mathbf{T}_{pv_k} \mathbf{P}^T = \mathbf{T}_k \mathbf{P}_k^T \tag{13}$$

Since the columns of both $\mathbf{P}$ and $\mathbf{P}_k$ are orthonormal ($\mathbf{P}^T\mathbf{P} = \mathbf{P}_k^T\mathbf{P}_k = \mathbf{I}$) the exact solution for $\mathbf{T}_{pv_k}$ is as follows:

$$\mathbf{T}_{pv_k} = \mathbf{T}_k \mathbf{P}_k^T \mathbf{P} \tag{14}$$

Geometrically this is equivalent to the rotation of the projected data points between the coordinate systems associated with the two models (local and global). The explained part of $\mathbf{X}_{pv_k}$ is computed as:

$$\widehat{\mathbf{X}}_{pv_k} = \mathbf{T}_{pv_k} \mathbf{P}^T = \mathbf{T}_k \mathbf{P}_k^T \mathbf{P} \mathbf{P}^T = \mathbf{X}_k \mathbf{P}_k \mathbf{P}_k^T \mathbf{P} \mathbf{P}^T \tag{15}$$

However, if the decomposition is truncated, in order to meet the second rule, $\mathbf{q}_{pv_k} = \mathbf{q}_k$, we should create matrix $\mathbf{E}_{pv_k}$, which has the same sum of squared row elements as $\mathbf{E}_k$. Assuming that the PV-set is created using large enough number of components, A, which ensures that no systematic variation is left in the residuals, this can be done using the following procedure:

3. Create a matrix $\mathbf{Z}$ of size $I_k \times I_k$ as a set of random values from any distribution with zero expectation.
4. Project the columns of $\mathbf{X}_k$: $\mathbf{E}_{pv_k}^{(1)} = \mathbf{Z}\mathbf{X}_k$ and normalize the columns of $\mathbf{E}_{pv_k}^{(1)}$ to the unit length.
5. Orthogonalize the columns relative to the global PC space: $\mathbf{E}_{pv_k}^{(2)} = \mathbf{E}_{pv_k}^{(1)}(\mathbf{I} - \mathbf{P}\mathbf{P}^T)$ and normalize the rows of $\mathbf{E}_{pv_k}^{(2)}$ to the unit length.
6. Scale the rows of $\mathbf{E}_{pv_k}^{(2)}$ to meet the requirement: $\mathbf{E}_{pv_k} = \mathbf{E}_{pv_k}^{(2)}\mathbf{Q}_k$, where $\mathbf{Q}_k$ is a diagonal matrix with elements equal to the square root of elements from $\mathbf{q}_k$.

The final $\mathbf{X}_{pv_k}$ is computed as:

$$\mathbf{X}_{pv_k} = \widehat{\mathbf{X}}_{pv_k} + \mathbf{E}_{pv_k} \tag{16}$$

It can be noticed that residuals, $\mathbf{E}_{pv}$, can be computed outside the cross-validation loop, for the entire PV-set, $\mathbf{X}_{pv}$. In this case we just need to collect all the $\mathbf{q}_k$ together and apply the procedure described above.

It is also important to notice that if we create two PV-sets, one using $A = A_1$ components and the second one using $A = A_2$ components and apply each for validation of a PCA/SIMCA model developed for $A = A_3$ components, then the results of the validation will be identical if $A_3 \leq A_2 \leq A_1$. This makes generation of PV-set independent of the selection of optimal number of components. One just should use the number of components, A, which is large enough to include the optimal number.

The original algorithm for PCV of PCA/SIMCA model, presented in Ref. [4] was based on rotation of the data points from $\mathbf{X}$ in the original variable space. This requires an additional internal loop over the principal components in order to build a rotation matrix. The new algorithm described above provides identical results (in terms of the two distances), but is faster and can be generalized for the regression models as it is shown in the next sections.

2.4. Procrustes cross-validation for PCR models

The regression problem is defined as:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}^{(y)} = \widehat{\mathbf{Y}} + \mathbf{E}^{(y)} \tag{17}$$

which is solved subject to minimization of the sum of squared elements of $\mathbf{E}^{(y)}$ (errors of prediction). Because the minimization of the sum is the main objective of optimization, it is reasonable to use $\mathbf{E}^{(y)}$ as the outcome object in $\mathbf{R}$, which can be further utilized for constraining of PV-set as:

$$\mathbf{E}_{pv_k}^{(y)} \approx \mathbf{E}_k^{(y)} \tag{18}$$

This rule should hold for any number of components $a = 1, \ldots, A$.

Here $\mathbf{E}_k^{(y)}$ is the matrix with the prediction residuals obtained at k-th segment of cross-validation by applying a local regression model $\mathscr{M}_k$ to the local validation set, $\{\mathbf{X}_k, \mathbf{Y}_k\}$. Matrix $\mathbf{E}_{pv_k}^{(y)}$ contains the prediction residuals, obtained by applying a global regression model, $\mathscr{M}$, to the corresponding PV-subset $\{\mathbf{X}_{pv_k}, \mathbf{Y}_k\}$. equation (18) can be rewritten as follows:

$$\mathbf{X}_{pv_k}\mathbf{B} \approx \mathbf{X}_k\mathbf{B}_k \tag{19}$$

In other words, in case of regression, the pseudo-validation set is created using only predictors, $\mathbf{X}_{pv}$. In this section we show how to implement this rule for Principal Component Regression (PCR).

PCR is a multiple linear regression method based on the PCA factorization of the original predictor matrix $\mathbf{X}$ and a response matrix $\mathbf{Y}$. It can be defined as follows:

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{E}^{(x)} \\ \mathbf{Y} &= \mathbf{T}\mathbf{C}^T + \mathbf{E}^{(y)} \end{aligned} \tag{20}$$

Here $\mathbf{T}$, $\mathbf{P}$ are the scores and loadings obtained from the PCA decomposition of $\mathbf{X}$. If $\mathbf{Y}$ has only one response, the matrix $\mathbf{C}$ has the size $1 \times A$ and is considered as a vector with regression coefficients, which may also be interpreted as the Y-loadings. It is computed as follows:

$$\mathbf{C}^T = \left(\mathbf{T}^T\mathbf{T}\right)^{-1}\mathbf{T}^T\mathbf{Y} \tag{21}$$

Therefore, we can rewrite the requirement from Equation (19) as follows:

$$\mathbf{T}_{pv_k}\mathbf{C}^T = \mathbf{T}_k\mathbf{C}_k^T \tag{22}$$

Or, for each principal component a:

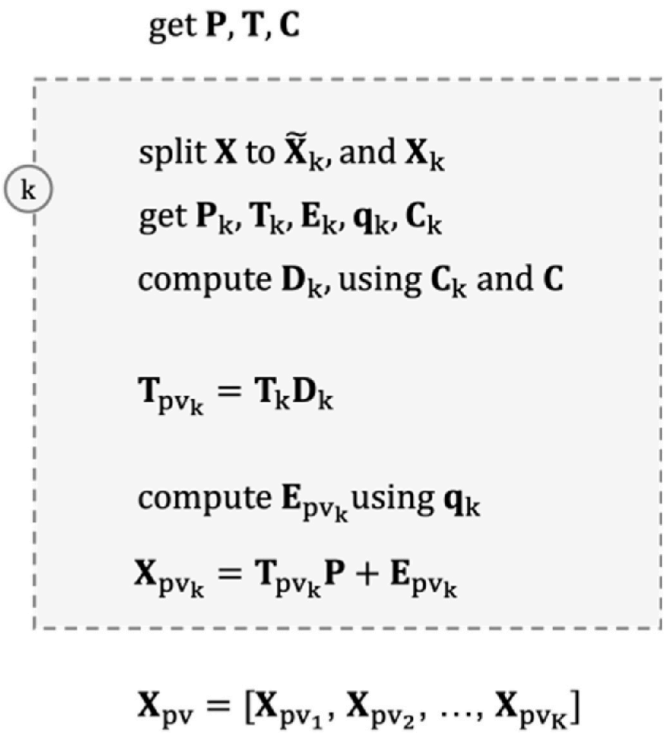

Fig. 1. Algorithm for creating a PV-set for Principal Component Regression.

$$\mathbf{t}_{\mathrm{pv_{ka}}} c_{\mathrm{a}} = \mathbf{t}_{\mathrm{ka}} c_{\mathrm{ka}} \tag{23}$$

Note that in this case both c_{a} and c_{ka} are scalars. This provides a simple solution for $\mathbf{T}_{\mathrm{pv_k}}$:

$$\mathbf{T}_{\mathrm{pv_k}} = \mathbf{T}_{\mathrm{k}} \mathbf{D}_{\mathrm{k}} \tag{24}$$

where $\mathbf{D}_{\mathrm{k}}$ is the $A \times A$ diagonal matrix containing the ratios $c_{\mathrm{ka}} / c_{\mathrm{a}}$ as its diagonal elements:

$$\mathbf{D}_{\mathrm{k}} = \begin{pmatrix} c_{\mathrm{k1}}/c_1 & 0 & \ldots & 0 \\ 0 & c_{\mathrm{k2}}/c_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & c_{\mathrm{ka}}/c_{\mathrm{a}} \end{pmatrix} \tag{25}$$

The explained part of the PV-set is calculated as follows:

$$\hat{\mathbf{X}}_{\mathrm{pv_k}} = \mathbf{T}_{\mathrm{pv_k}} \mathbf{P}^{\mathrm{T}} = \mathbf{X}_{\mathrm{k}} \mathbf{P}_{\mathrm{k}} \mathbf{D}_{\mathrm{k}} \mathbf{P}^{\mathrm{T}} \tag{26}$$

Assuming that the maximum number of components, A, used for calculation of the explained part, is large enough to capture all possible systematic variation, the residuals, $\mathbf{E}_{\mathrm{pv_k}}$ can be computed using the same procedure as described for PCA in the previous section.

By repeating this procedure for all $k = 1, \ldots, K$ and combining $\mathbf{X}_{\mathrm{pv_k}}$ together, the entire pseudo-validation set is obtained. Fig. 1 shows the algorithm schematically.

2.5. Procrustes cross-validation for PLS models

Partial least squares (PLS) regression can be presented in a form similar to PCR:

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E}^{(\mathrm{x})} \\ \mathbf{Y} &= \mathbf{T}\mathbf{C}^{\mathrm{T}} + \mathbf{E}^{(\mathrm{y})} \end{aligned} \tag{27}$$

However, in contrast to PCR, the scores $\mathbf{T}$ are calculated by taking into account the covariance between columns of $\mathbf{X}$ and $\mathbf{Y}$. The covariance pattern is captured by a matrix of weights, $\mathbf{W}$ which is computed iteratively for each latent variable. This can be done using NIPALS [12] or SIMPLS [13] algorithms.

Once the matrix of weights is obtained, one can calculate the score matrix $\mathbf{T}$ and complete the decomposition in different ways. Thus, originally, it has been proposed to compute both the weights $\mathbf{W}$ and the loadings $\mathbf{P}$, so the scores are found as:

$$\begin{aligned} \mathbf{T} &= \mathbf{X}\mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1} \\ \mathbf{X} &= \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E}^{(\mathrm{x})} \end{aligned} \tag{28}$$

This way is often referenced in literature as Wold's factorization. Martens [14] proposed a PLS factorization without using the loadings $\mathbf{P}$:

$$\begin{aligned} \mathbf{T} &= \mathbf{X}\mathbf{W} \\ \mathbf{X} &= \mathbf{T}\mathbf{W}^{\mathrm{T}} + \mathbf{E}^{(\mathrm{x})} \end{aligned} \tag{29}$$

In this case, the weight matrix, $\mathbf{W}$, has the orthonormal columns. There are also other factorizations, such as Biorthogonal PLS, proposed by Rolf Ergon [15].

In any case, once the score matrix is obtained, the matrix of Y-loadings, $\mathbf{C}$ can be found similar to PCR:

$$\mathbf{C}^{\mathrm{T}} = \left(\mathbf{T}^{\mathrm{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{Y} \tag{30}$$

This can be done for any number of responses (columns of $\mathbf{Y}$).

In case of single response (PLS1), the algorithm for calculation of PV-set for a PCR model, described in the previous section, can be applied as is. Note that in case of Marten's factorization, the $\mathbf{W}$ matrix should be used instead of the $\mathbf{P}$ matrix.

In case of multiple responses (PLS2), matrix $\mathbf{C}$ has more than one column and thus Equation (23) does not contain scalars anymore:

$$\mathbf{t}_{\mathrm{pv_{ka}}} \mathbf{c}_{\mathrm{a}}^{\mathrm{T}} = \mathbf{t}_{\mathrm{ka}} \mathbf{c}_{\mathrm{ka}}^{\mathrm{T}} \tag{31}$$

However, this equation can be solved as:

$$\mathbf{t}_{\mathrm{pv_{ka}}} = \mathbf{t}_{\mathrm{ka}} \mathbf{c}_{\mathrm{ka}}^{\mathrm{T}} \mathbf{c}_{\mathrm{a}} \left(\mathbf{c}_{\mathrm{a}}^{\mathrm{T}} \mathbf{c}_{\mathrm{a}}\right)^{-1} \tag{32}$$

Apparently Equation (23) is a special case of Equation (32) when both $\mathbf{c}_{\mathrm{a}}$ and $\mathbf{c}_{\mathrm{ka}}$ have one element. Here $\mathbf{c}_{\mathrm{ka}}^{\mathrm{T}} \mathbf{c}_{\mathrm{a}} (\mathbf{c}_{\mathrm{a}}^{\mathrm{T}} \mathbf{c}_{\mathrm{a}})^{-1}$ is a scalar as well, which can be used as a diagonal element of matrix $\mathbf{D}$:

$$\mathbf{D}_{\mathrm{k}} = \begin{pmatrix} \mathbf{c}_{\mathrm{k1}}^{\mathrm{T}} \mathbf{c}_1 \left(\mathbf{c}_1^{\mathrm{T}} \mathbf{c}_1\right)^{-1} & 0 & \ldots & 0 \\ 0 & \mathbf{c}_{\mathrm{k2}}^{\mathrm{T}} \mathbf{c}_2 \left(\mathbf{c}_2^{\mathrm{T}} \mathbf{c}_2\right)^{-1} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \mathbf{c}_{\mathrm{kA}}^{\mathrm{T}} \mathbf{c}_{\mathrm{A}} \left(\mathbf{c}_{\mathrm{A}}^{\mathrm{T}} \mathbf{c}_{\mathrm{A}}\right)^{-1} \end{pmatrix} \tag{33}$$

In the case of Wold's factorization the explained part of the PV-set can be found as follows:

$$\hat{\mathbf{X}}_{\mathrm{pv_k}} = \mathbf{T}_{\mathrm{pv_k}} \mathbf{P}^{\mathrm{T}} = \mathbf{X}_{\mathrm{k}} \mathbf{W}_{\mathrm{k}} \left(\mathbf{P}_{\mathrm{k}}^{\mathrm{T}} \mathbf{W}_{\mathrm{k}}\right)^{-1} \mathbf{D}_{\mathrm{k}} \mathbf{P}^{\mathrm{T}} \tag{34}$$

In the experimental part it is shown that in case of single response we obtain the exact equivalence of the prediction errors ($\mathbf{E}_{\mathrm{pv_k}}^{(\mathrm{y})} = \mathbf{E}_{\mathrm{k}}^{(\mathrm{y})}$) and in the case of multiple responses the equivalence is held approximately ($\mathbf{E}_{\mathrm{pv_k}}^{(\mathrm{y})} \approx \mathbf{E}_{\mathrm{k}}^{(\mathrm{y})}$).

2.6. Matrix D as a diagnostic tool in PCV

It should be noted, that the diagonal elements of matrix $\mathbf{D}_{\mathrm{k}}$ can also be used as an additional source of information about the quality of the dataset, the homogeneity of the cross-validation segments and the robustness of the model. Thus, if the global and the local models for a given segment k are identical, then $\mathbf{D}_{\mathrm{k}} = \mathbf{I}$ as $c_{\mathrm{ka}}/c_{\mathrm{a}} = 1$ (or $\mathbf{c}_{\mathrm{ka}}^{\mathrm{T}} \mathbf{c}_{\mathrm{a}} (\mathbf{c}_{\mathrm{a}}^{\mathrm{T}} \mathbf{c}_{\mathrm{a}})^{-1} = 1$ in case of multiple response). In practice, values c_{ka} vary around values c_{a}. This variation depends on the heterogeneity of the dataset, similarity of the subsamples from different splits, as well as on the importance of a particular latent variable for a prediction.

If the variation is relatively small, the values $c_{\mathrm{ka}}/c_{\mathrm{a}}$ are randomly

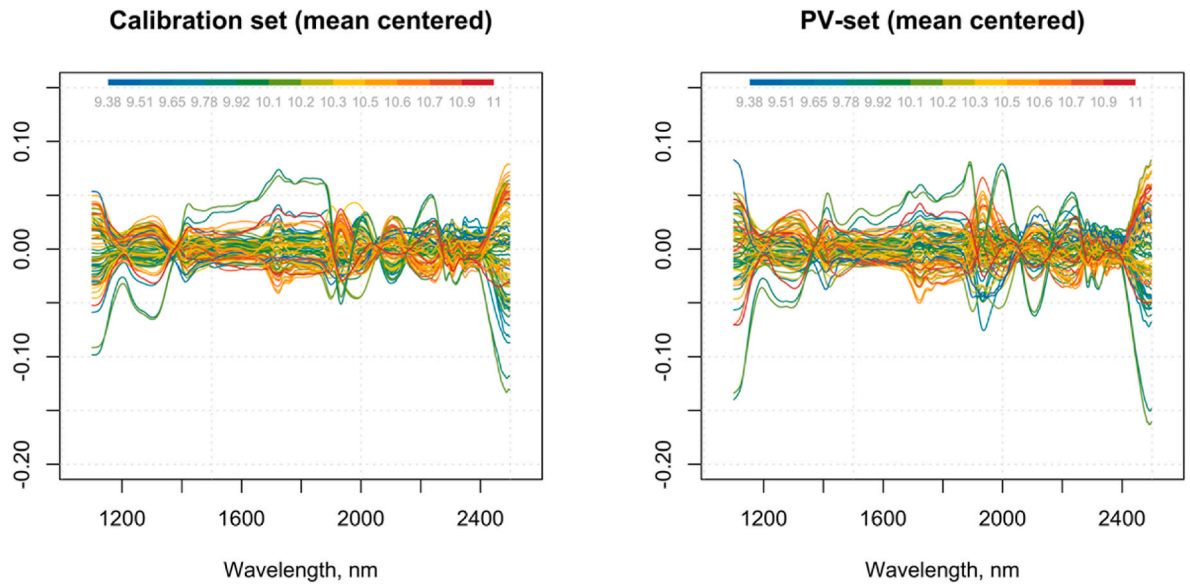

Fig. 2. Calibration set (left) and pseudo-validation set (right) for the Corn spectra. The PV-set was created using PCR based algorithm.

distributed around 1. Very large, small or negative values could indicate a large discrepancy between the local and global models. This difference can be explained by one of the following reasons:

7. Wrong splitting strategy, for example, all objects with large response values are located in one segment.
8. Presence of one or several extreme observations in one of the segments.
9. Current latent variable explains irrelevant systematic variation or large random variation.

In the experimental section it is shown that the exploration of the diagonal elements of $\mathbf{D}_k$ can identify such situations. We suppose that this approach adds a new useful tool to the chemometric toolbox.

3. Datasets

3.1. NIR spectra of corn samples

The *Corn* dataset consists of Near Infrared (NIR) spectra of 80 corn samples. Each spectrum contains the absorbance values for 750 wavelengths covering the range from 1100 to 2498 nm. The spectra were downloaded from Eigenvector Research, Inc. Website, where they are publicly available (https://eigenvector.com/resources/data-sets/).

The original data contains several subsets of spectra acquired using different instruments (*m5*, *mp5* and *mp6*). The *mp5* spectra are used in this paper. The spectra were corrected using Standard Normal Variate (SNV) normalization.

The dataset also contains the moisture, oil, protein and starch responses for each of the corn samples.

The *Corn* dataset is used in order to demonstrate that PCA, PCR and PLS1 algorithms, described in the previous section, work as intended.

3.2. Tecator

Another dataset, *Tecator*, is also well known (http://lib.stat.cmu.edu/datasets/tecator/). It consists of 215 Near Infrared Transmission (NIT) spectra recorded in the range of 850–1050 nm using Tecator Infratec Food and Feed Analyzer. The spectra were acquired for finely chopped pure meat samples with different moisture, fat and protein contents. The dataset is split into the training set (172 samples) and the test set (43 samples).

The spectra are affected by strong light scattering effect and hence require proper preprocessing in order to get a model with decent predictions of the response values.

This dataset is used for demonstration of how PV-set can be applied for the grid search of the best combination of preprocessing methods, and also to show how PCV works in case of multiple responses (PLS2).

4. Results

4.1. PCA analysis of corn data

This short section aims at confirming that the proposed modification of the PCV algorithm for PCA models provides results identical to the original algorithm, described in Ref. [4]. In this case two PV-sets are created using the two algorithms (the old and new) for $A = 20$, and Venetian blinds split with $K = 4$ segments. Both sets are used for validation of a PCA model developed for the mean centered *Corn* spectra.

Fig. 10 (available in supplementary materials) shows the distance plots created for both sets with $A = 2$ and $A = 20$ (data points for calibration set are shown in all plots as well). The obtained results are identical (also confirmed numerically), showing a clear overfitting pattern for $A = 20$. However, generation of the PV-set using the modified algorithm was more than 300 times faster (58.2 s vs. 0.16 s average time for repeating the generation procedure 30 times for each algorithm on the same computer).

The modified approach was also applied to reproduce all examples from Ref. [4] and resulted in the identical outcomes (not shown here).

4.2. PCR analysis of corn data

A PCR model with $A = 20$ PCs was created using the NIR spectra from the *Corn* dataset as predictors and the moisture content as the response. The CCV procedure based on Venetian blinds split with $K = 4$ segments was applied to the model. A PV-set was developed using the approach proposed in Section 2.4, with $A = 20$ and the same split as in CCV.

Fig. 2 shows spectral plots (after mean centering) both for the calibration set, $\mathbf{X}$, and the pseudo-validation set $\mathbf{X}_{pv}$ for visual inspection. The spectra are color coded based on the corresponding response value using the gradient from blue to red (the color map legend is shown in all plots). As one can notice, although the common trends are quite similar in the two datasets, the spectra do not look identical. The difference is

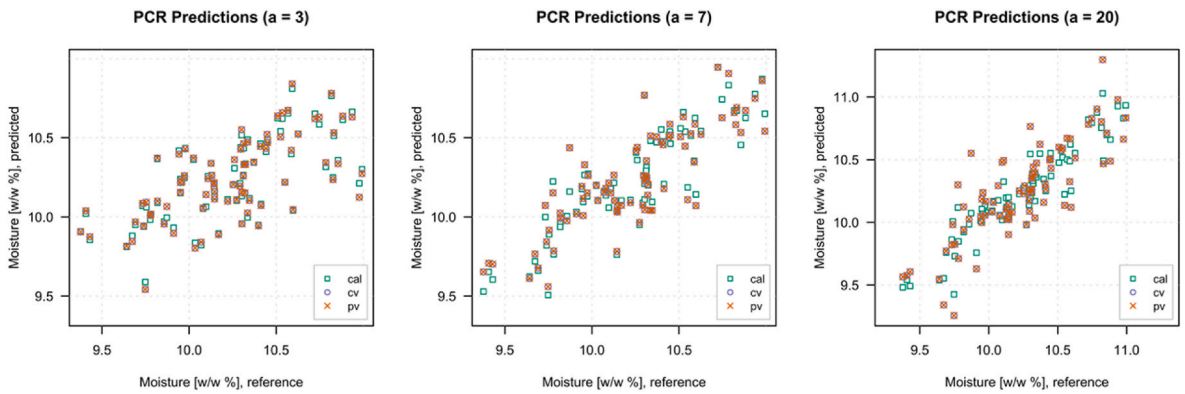

Fig. 3. Predicted vs. reference moisture content obtained for the PCR model of Corn data for different number of PCs.

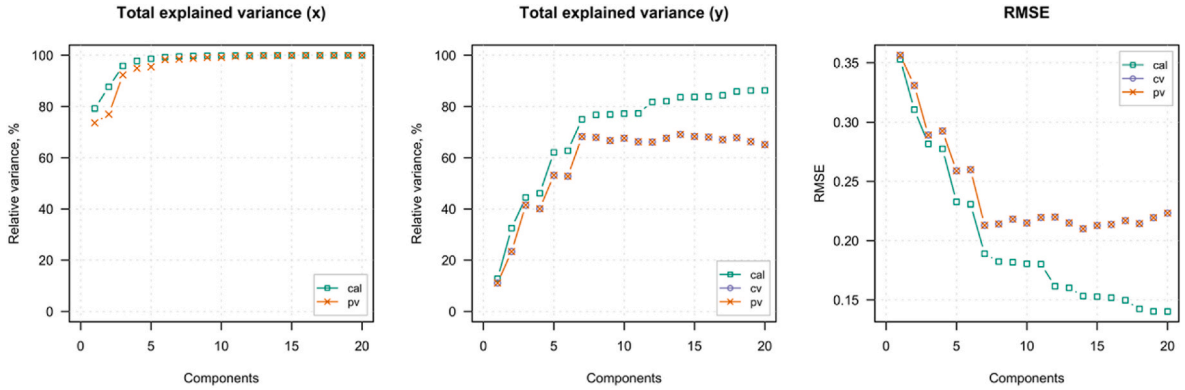

Fig. 4. Total explained variance for predictors and responses and the root mean square error for the PCR model of Corn data.

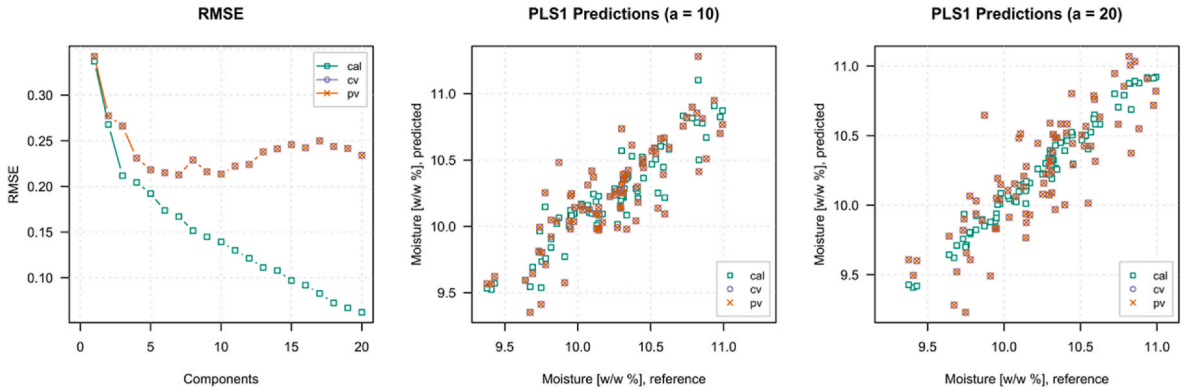

Fig. 5. Root mean squared error vs. number of components and two plots with predicted vs. measured response values (a = 10 and a = 20) for the PLS1 model of Corn data.

especially clear for two spectra, which look extreme in both sets.

The PV-set was used as a validation set to validate the global PCR model. Fig. 3 shows the predicted vs. measured plots for $a = 3$, $a = 7$ and $a = 20$. In all plots points shown as blue squares correspond to the prediction computed for the calibration set, orange circles stand for the CCV predictions and red crosses represent the PCV predictions. As one can notice, the last two results are identical regardless the number of components, although in case of PCV the prediction was obtained by applying the global model to the whole PV-set.

Fig. 4 shows the overall model performance using the total explained variance (TEV) plots for X and Y, as well as the root mean squared error (RMSE) values. Same color coding is used. In case of the explained X-variance, the CCV results are not shown as it is not possible to compute them correctly. As expected, the TEV and RMSE computed for CCV and PCV outcomes are identical. This proves that the algorithm works as intended.

4.3. PLS1 analysis of corn data

A PLS model with $A = 20$ was developed and the corresponding PV-set was created using the same data and rules as for PCR, described in the previous section. SIMPLS was used as the algorithm for PLS factorization (both for the global model and for the PV-set generation).

Fig. 5 contains three plots with main outcomes obtained for calibration, CCV and PCV. The first one shows RMSE values. The other two plots show predicted vs. reference response values obtained for $a = 10$

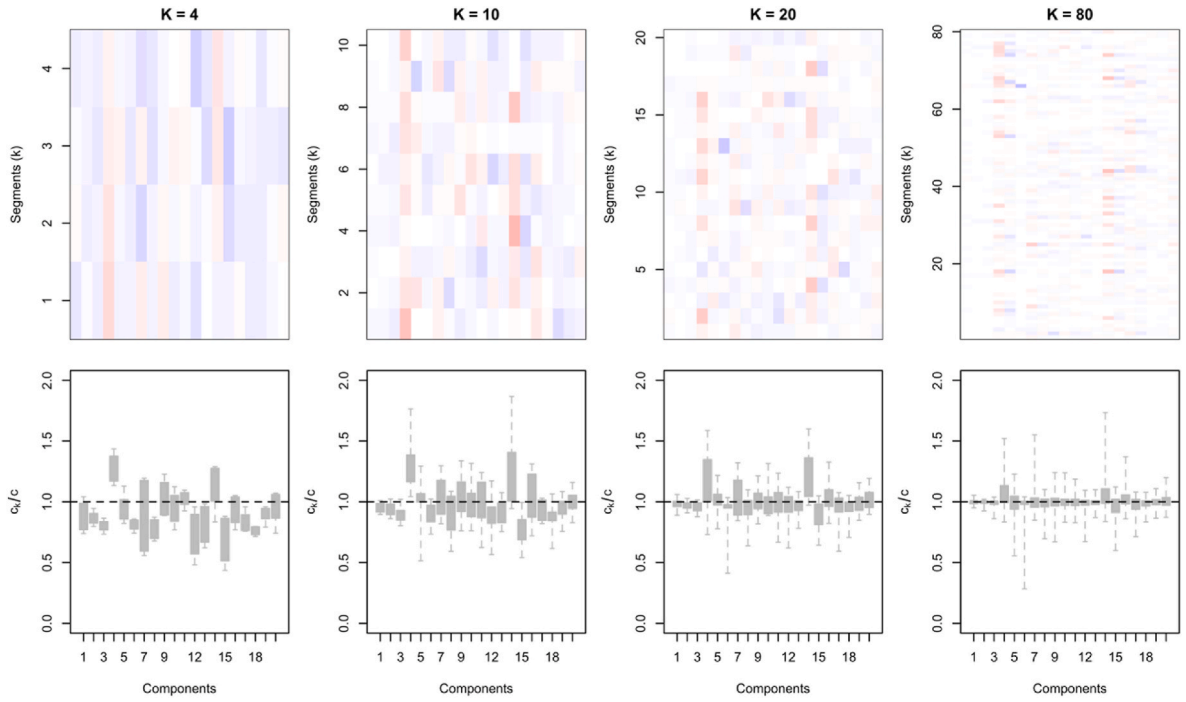

Fig. 6. Visualization of c_k/c ratios for different splits and individual components in case of K = 4, 10, 20 and 80 (Corn data).

and $a = 20$. In all cases the results obtained for PCV and CCV are identical.

Fig. 6 contains plots which illustrate the behaviour of the $c_k/\ c$ values for individual LV (abscissa axis in all plots). The plots are made using the PCV procedure with different number of segments $K = 4$, 10, 20 and 80. The top four plots are the heatmaps of the ratios, where the values $c_k/\ c < 1$ are shown as the shades of blue and the values $c_k/\ c > 1$ are shown as the shades of red color. Values $c_k/c = 1$ are shown in white. The bottom plots are the box-and-whisker plots developed using all $c_k/\ c$ values for each LV.

It can be seen that the ratio values vary randomly around 1, as expected. The variation depends on the number of segments. Thus in case of the leave-one-out splits ($K = 80$, the most right plots), the most of the values are very close to 1 with a few extremes. In case of fewer segments (e.g. for $K = 4$, the most left plots), the variation magnitude is larger. This effect is in line with our knowledge about the cross-validation, which underestimates the sampling error when too many segments are used.

Therefore, we can conclude that the c_k/c plots can provide an additional information about the quality and homogeneity of the dataset and the splits. For example, the presence of negative or large positive values (above 2) indicates the lack of stability for a particular LV. This means that exclusion of a small part of data leads to local Y-loadings that are significantly different from the Y-loadings of the global model obtained for the same LV. This effect has been observed earlier when PCV was applied for short datasets in SIMCA classification models [9].

Negative values would indicate that the corresponding Y-loading vector (or a value in case of a single response) swapped its direction. It means that an LV, which gives a positive contribution to the predicted response values in the global model, contributes negatively in the local model and vice versa. The large absolute values would indicate that the magnitude of the contribution in the local model is very much different from the contribution of this LV in the global model.

These effects can be caused by the presence of several extremes or even outliers in the same segment or by the case when an LV explains a large but non-relevant variation. None of these effects are observed for this dataset.

The additional exploration of the local model makes it clear that for $a = 4$ the ratio is persistently positive, even in the case of larger number of splits. This situation is unlikely for a random selection of samples in the segments. Investigation of the PLS weights obtained using the global and the local models for this particular LV confirms this effect (Fig. 11, available in supplementary materials) for both $K = 4$ and $K = 10$ cases (two top plots in the figure). As one can see, in the range of 1500–1650 nm the weights obtained for most of the local models are consistently smaller than the global weights. The bottom plots show the weights for $a = 5$ for comparison, in which the weights from the local models vary more randomly around the global model weights.

Thus, the Procrustes cross-validation of PLS models not only provides all capabilities of the conventional validation set but also gives the additional tools for deeper exploratory analysis of the datasets and the models.

4.4. PLS1 analysis of tecator data

This example demonstrates how PV-set can be used for a faster grid search of the best preprocessing method. As it was already mentioned, the spectra from Tecator dataset suffer from the scattering effect and therefore preprocessing is necessary. The selection of methods to tackle this problem is quite wide, from Standard Normal Variate (SNV) or Multiplicative Scatter Correction (MSC) to Savitzky-Golay (SG) filter for the first or second derivative. In case of SG two additional parameters have to be tuned — the filter width and the degree of polynomial.

When such a broad selection of possible preprocessing methods exists, the grid search application, in which all the methods and their combinations are tested in order to find the best one, is quite common. Thus, if 20 different combinations should be tested and the CCV with $K = 4$ is employed, then 100 PLS models are to be calibrated in total (1 global and 4 local models for each combination) during the grid search. Moreover, cross-validation itself gives the additional overhead related to the local predictions, combining all results together, etc.

Using PCV in this case makes possible to create a single model for each combination since the PV-set can be created only once, for the original, non-preprocessed dataset. However, in this case the prediction performance of the CCV and the PCV will be different as the first is based on already preprocessed data.

Table 1
Preprocessing methods used for grid search.

Method	Abbreviation
SNV	snv
SG filter (width = 3, polynomial degree = 1, first derivative)	sg311
SG filter (width = 5, polynomial degree = 1, first derivative)	sg511
SG filter (width = 7, polynomial degree = 1, first derivative)	sg711
SG filter (width = 9, polynomial degree = 1, first derivative)	sg911
Combination of sg311 and SNV	sg311 + snv
Combination of sg511 and SNV	sg511 + snv
Combination of sg711 and SNV	sg711 + snv
Combination of sg911 and SNV	sg911 + snv
SG filter (width = 3, polynomial degree = 2, second derivative)	sg322
SG filter (width = 5, polynomial degree = 2, second derivative)	sg522
SG filter (width = 7, polynomial degree = 2, second derivative)	sg722
SG filter (width = 9, polynomial degree = 2, second derivative)	sg922
Combination of sg322 and SNV	sg322 + snv
Combination of sg522 and SNV	sg522 + snv
Combination of sg722 and SNV	sg722 + snv
Combination of sg922 and SNV	sg922 + snv

In order to compare the two methods, the grid search with 17 combinations of SNV and SG filter with different settings was carried out. The full list of combinations with all details is shown in Table 1.

Fig. 7 shows the original training set (left) and the corresponding PV-set (right) that was generated using $K = 4$ and $A = 20$. Similar to the previous cases, the plots show mean centered spectra. The scattering effect is clearly visible in both sets.

Fig. 8 presents the bar plot developed for the RMSE values obtained using the CCV (orange) and the PCV (red) of the PLS-models based on the preprocessed spectra. In both cases the optimal number of LVs (shown as a number on the top of each bar) was identified as the first local minimum of the corresponding RMSE values. The highlighted bars correspond to the combination of preprocessing methods that resulted in the smallest RMSE (best two results for each method).

It is clear that the results obtained using the two validation approaches are very similar with small deviations of the individual RMSE values.

The combination of preprocessing methods that provide two best results is identical: in both cases it is a combination of SG filter for the second derivative (filter width 7 and 9) followed by SNV normalization (last two rows in Table 1). Both approaches also suggest the same optimal number of components ($a = 8$) although PCV is more pessimistic in the estimation of the RMSE values.

We suppose the result to be quite impressive taking into account that PV-set was created for the raw spectra of the original training set while the CCV is based on already preprocessed spectra. It should be also mentioned that the overall procedure was on average 7 times faster in case of PCV compared to CCV. The procedure was repeated 30 times for each approach on the same computer. In the case of CCV it took from 6 to 8 s to complete the grid search, while in case of PCV the computational time was around 1 s.

4.5. PLS2 analysis of tecator data

The PLS based method for PCV, suggested in Section 2.5, is versatile and can be applied to both PLS1 and PLS2 factorizations. Although in case of PLS2, the RMSE values computed using the CCV and the PCV are not completely identical.

In order to demonstrate how PCV works for PLS2 and how large the difference between the PCV and the CCV results is, the Tecator data is used again, but this time all three responses (moisture, fat and protein content) are utilized in the model.

Similar to the previous section, the PV-set was generated using the PLS based algorithm with $A = 20$ and $K = 4$, however the response matrix $\mathbf{Y}$ consists of three columns in this case. Fig. 9 shows the RMSE values vs. the number of LVs for each response. As one can see, the RMSE values, which are obtained using PCV and CCV, are almost identical for the first three LVs, which capture 96% of the relative Y-variance in total. There is a larger difference in RMSE values for the later LV; however, it is obvious that the qualitative behaviour of the RMSE vs. number of components is similar for the two approaches.

5. Discussion and conclusions

We demonstrated that Procrustes cross-validation, proposed recently for validation of PCA and SIMCA models, can be presented in a more general way, which makes it more versatile and flexible. The proposed generalization allowed us to develop the more efficient PCA implementation, which is by several orders of magnitude faster, compared to the original version. In addition, it made possible to use PCV approach for validation of the multivariate regression models based on latent variables decomposition, such as PCR and PLS.

The pseudo-validation set, generated by PCV, can be used for validation of a global model providing a full set of outcomes in contrast to CCV in which the outcomes are collected from several local models. At

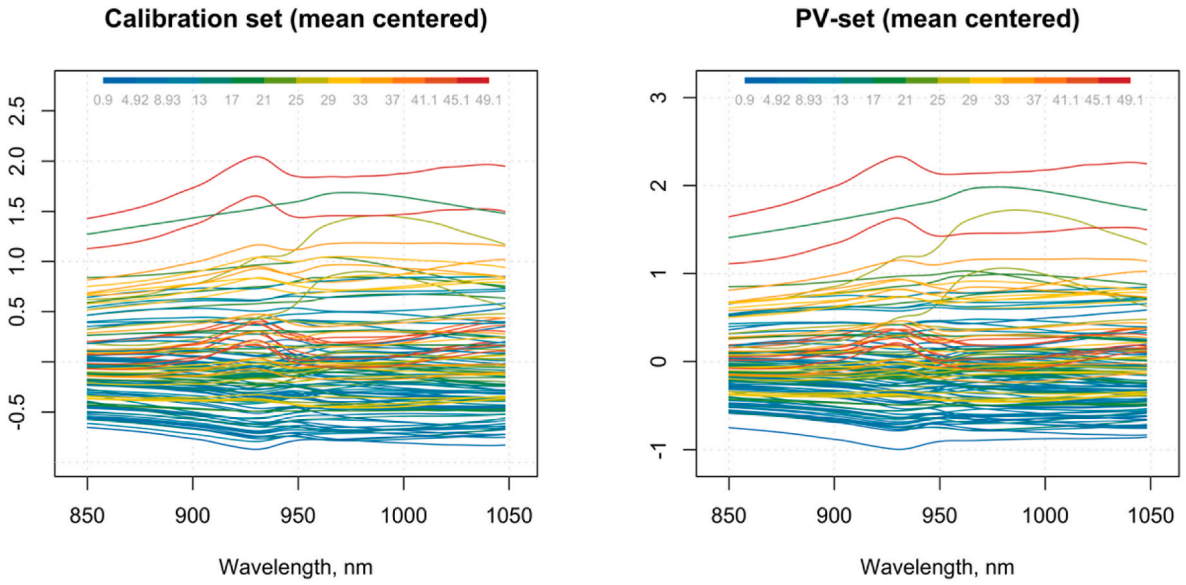

Fig. 7. Calibration set (left) and pseudo-validation set (right) for the Tecator spectra. The PV-set was created using PLS based algorithm with K = 4 and A = 20.

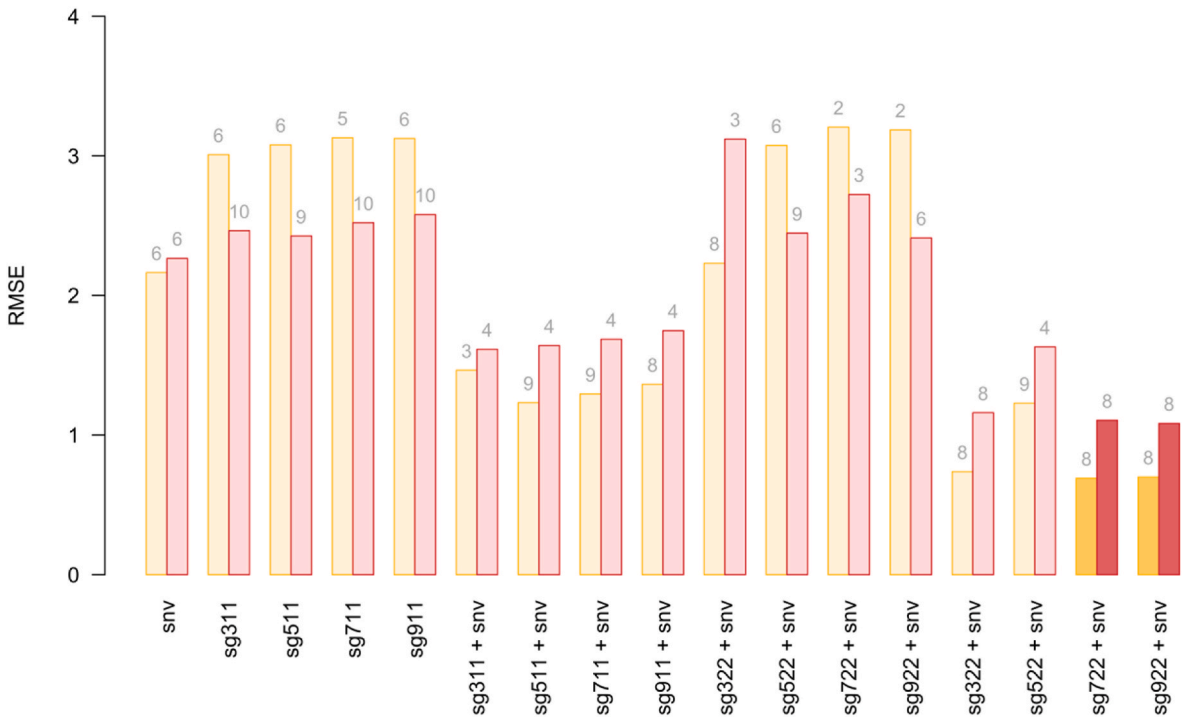

Fig. 8. RMSE values for cross-validation (orange) and pseudo-validation set (red) predictions obtained for PLS model created for preprocessed spectra. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

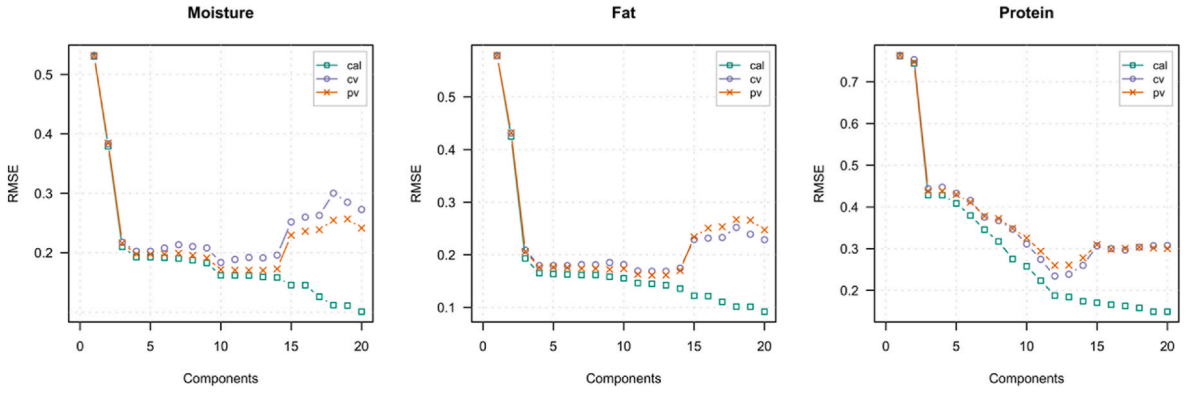

Fig. 9. Root mean square error vs. number of components obtained for each response using PLS2 model for the Tecator dataset.

the same time, using pseudo-validation set makes possible to speed up the process of model optimization, for example, in case of the grid search for selection of the best combination of preprocessing methods.

Procrustes cross-validation also provides additional tools, which help an analyst to get the better insights about the heterogeneity of the dataset, quality of cross-validation splits, presence of outliers, etc.

The general idea of PCV is to emulate a new set of objects/measurements taken from the same population as the training set. In case of multivariate data, it is important to consider the internal structure of the data, in particular, the relationship among variables. Projection methods, such as PCA and PLS makes this possible via utilizing the variance-covariance structure. This makes the PCV algorithms, proposed in the paper, simple and straightforward.

The idea of Procrustean rules, also introduced in this paper, in theory, let PCV be implemented for other machine learning methods as well. However, some of the ML methods, e.g. Support Vector Machines, do not take the internal structure of the whole data into account. Other, such as artificial neural networks, often model this structure in a very complex way, which is difficult to utilize. Based on our experience, PV-sets, generated using the projection methods can be successfully applied for validation and exploration of more sophisticated models, however this requires additional investigation.

CRediT authorship contribution statement

Sergey Kucheryavskiy: Conceptualization, Methodology, Formal analysis, Software, writing. **Oxana Rodionova:** Conceptualization, Writing – review & editing. **Alexey Pomerantsev:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data is from public datasets, the code is available on GitHub

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.aca.2023.341096.

References

[1] K.H. Esbensen, P. Geladi, Principles of proper validation: use and abuse of re-sampling for validation, J. Chemometr. 24 (2010) 168–187, https://doi.org/10.1002/cem.1310.

[2] F. Westad, F. Marini, Validation of chemometric models — a tutorial, Anal. Chim. Acta 893 (2015) 14–24, https://doi.org/10.1016/j.aca.2015.06.056.

[3] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010) 40–79, https://doi.org/10.1214/09-SS054.

[4] S. Kucheryavskiy, S. Zhilin, O. Rodionova, A. Pomerantsev, Procrustes cross-validation—a bridge between cross-validation and independent validation sets, Anal. Chem. 92 (2020) 11842–11850, https://doi.org/10.1021/acs.analchem.0c02175.

[5] M. Arif, G. Chilvers, S. Day, S.A. Naveed, M. Woolfe, O.Ye Rodionova, A. L. Pomerantsev, O. Kracht, C. Brodie, A. Mihailova, A. Abrahim, A. Cannavan, S. D. Kelly, Differentiating pakistani long-grain rice grown inside and outside the accepted basmati himalayan geographical region using a “one-class” multi-element chemometric model, Food Control 123 (2021), 107827, https://doi.org/10.1016/j.foodcont.2020.107827.

[6] E. Boichenko, A. Panchenko, M. Tyndyk, M. Maydin, S. Kruglov, V. Artyushenko, D. Kirsanov, Validation of classification models in cancer studies using simulated spectral data – a “sandbox” concept, Chemometr. Intell. Lab. Syst. 225 (2022), 104564, https://doi.org/10.1016/j.chemolab.2022.104564.

[7] L. Strojnik, D. Potočnik, M. Jagodic Hudobivnik, D. Mazej, B. Japelj, N. Škrk, S. Marolt, D. Heath, N. Ogrinc, Geographical identification of strawberries based on stable isotope ratio and multi-elemental analysis coupled with multivariate statistical analysis: a slovenian case study, Food Chem. 381 (2022), 132204, https://doi.org/10.1016/j.foodchem.2022.132204.

[8] A.K. Pautova, A.S. Samokhin, N.V. Beloborodova, A.I. Revelsky, Multivariate prognostic model for predicting the outcome of critically ill patients using the aromatic metabolites detected by gas chromatography-mass spectrometry, Molecules (2022), https://doi.org/10.3390/molecules27154784.

[9] A.L. Pomerantsev, O.Ye Rodionova, Procrustes cross-validation of short datasets in PCA context, Talanta 226 (2021), 122104, https://doi.org/10.1016/j.talanta.2021.122104.

[10] S. Kucheryavskiy, Mdatools — r package for chemometrics, Chemometr. Intell. Lab. Syst. 198 (2020), 103937, https://doi.org/10.1016/j.chemolab.2020.103937.

[11] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. 2 (1987) 37–52, https://doi.org/10.1016/0169-7439(87)80084-9.

[12] S. Wold, M. Sjöström, L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemometr. Intell. Lab. Syst. 58 (2001) 109–130, https://doi.org/10.1016/S0169-7439(01)00155-1.

[13] S. de Jong, SIMPLS: an alternative approach to partial least squares regression, Chemometr. Intell. Lab. Syst. 18 (1993) 251–263, https://doi.org/10.1016/0169-7439(93)85002-X.

[14] H. Martens, Reliable and relevant modelling of real world data: a personal account of the development of PLS regression, Chemometr. Intell. Lab. Syst. 58 (2001) 85–95, https://doi.org/10.1016/S0169-7439(01)00153-8.

[15] R. Ergon, PLS score-loading correspondence and a bi-orthogonal factorization, J. Chemometr. 16 (2002) 368—373, https://doi.org/10.1002/cem.736.