# Procrustes cross-validation — a bridge between cross-validation and independent validation set

Sergey Kucheryavskiy,[*,†] Sergei Zhilin,[‡] Oxana Rodionova,[¶] and Alexey Pomerantsev[¶]

†*Aalborg University, Department of Chemistry and Bioscience, Esbjerg, Denmark*

‡*CSort Ltd., Barnaul, Russia*

¶*Semenov Federal Research Center for Chemical Physics, RAS, Moscow, Russia*

E-mail: svk@bio.aau.dk

**Abstract**

In this paper we propose a new approach for validation of chemometric models. It is based on k-fold cross-validation algorithm, but, in contrast to conventional cross-validation, our approach makes possible to create a new dataset, which carries sampling uncertainty estimated by the cross-validation procedure. This dataset, called *pseudo-validation set*, can be used similar to independent test set, giving a possibility to compute residual distances, explained variance, scores and other results, which can not be obtained in the conventional cross-validation. The paper describes theoretical details of the proposed approach and its implementation as well as presents experimental results obtained using simulated and real chemical datasets.

# Introduction

Model validation is a crucial part of chemometric analysis. A substantial amount of research has been dedicated to investigation and comparison of different validation techniques and approaches.[1–4] However, in spite of the diversity, in general there are two main options — test set and cross-validation. The test set validation has been proven to be the most reliable for assessment of the final model performance. The cross-validation, in its turn, is mostly used for optimization of chemometric models.

Cross-validation is an iterative procedure based on re-use of calibration set several times[5] by resampling. Despite its simplicity and popularity, cross-validation is rather a controversial technique being heavily criticized, especially when it is applied to final model testing.[6] There are also several research,[2,3] which in contrast justify the use of cross-validation (in this case in form of double repeated cross-validation) for model testing. It should be also noted that the cross-validation procedure can be implemented and assessed in many different ways, see for example a review by Bro *et al.*[7]

Regardless implementation, cross-validation has several rather obvious drawbacks, including the following:

1. It is an iterative procedure and, therefore, time-consuming. Using cross-validation in, for example, grid search, for hyperparameter optimization, often requires a lot of computational time, especially for large datasets.

2. Cross-validation combines results from several local models instead of validating the global one. This gives certain limitations. For example, in case of Principal Component Analysis (PCA), we can not compute explained variance or scores for cross-validated results. Even score distances can not be compared directly as they are related to different PC spaces.

To solve these issues, several researches proposed to use a "simulated" test set, which is based on calibration set, but with some random noise added on top of that.[8] In this paper we propose an alternative approach, whose main idea is to measure the variation among local cross-validation models (in form of angles between latent variable subspaces) and introduce this variation to the calibration set thus creating a new dataset — *pseudo-validation set* (PV-set). This approach was named as *Procrustes Cross-Validation* for reasons to be explained later in this text.

On the one hand, the pseudo-validation set enables validation of a global model and gives most of the possibilities of the independent test set validation. On the other hand, the model performance results are identical to what we can get using conventional cross-validation, but there is no need for repeating cross-validation iterations more than once.

Another very important outcome of the proposed approach is a possibility to investigate the quality of cross-validation by looking at distribution of the new values and comparing it with distribution of values from the calibration set. So far it was possible only for response values in supervised methods, such as regression, but our approach extends this capability. Thus, investigation of similarity between the calibration and the pseudo-validation sets allows us to assess and compare different cross-validation strategies and parameters directly and find the optimal ones.

It should be also noted that the approach itself and the results shown in this paper are related to Principal Component Analysis and methods based on PCA (e.g. SIMCA). We are working on generalizing this approach to regression models (PCR, PLS) which will be reported separately.

The manuscript describes the proposed approach in detail and shows results obtained by using the approach for analysis of simulated and real chemical datasets of different nature. This analysis, in particular, aims at three main objectives:

1. Compare various calibration and corresponding pseudo-validation sets in terms of similarity. Our hypothesis here is that the two sets resemble two collections of instances taken from the same population.

2. Assess that the difference between a pseudo-validation and a calibration set is enough to use the first one for validation purposes. In case of PCA, the pseudo-validation set should behave similarly to the calibration set, when number of components in a model corresponds to optimal complexity, and should show clear signs of overfitting otherwise.

3. Compare (visually and statistically) the results for pseudo-validation set with results obtaining for real independent test set.

We would like to underline, that we do not propose our approach as an alternative to the test set validation as the pseudo-validation set is built on top of the calibration set and hence can not be considered as independent. Therefore, from our point of view, its usage should be limited to optimization and exploration of models and independent test set must be applied for assessing performance of the final model when possible.

# Theory

When data points are fitted by any theoretical model (e.g. simple linear regression, multiple linear regression, principal component analysis, etc.) there are two kinds of errors related to the quality of the model — *fitting error* and *sampling error*.[9,10] The fitting error tells how well the model captures variation of the data values or its relevant part (for example, variation of response values in case of regression). This error can be estimated from a calibration set assuming that the corresponding distribution is known.

The sampling error is related to uncertainty of the model parameters due to sample-to-sample variation. Thus, if we take another set of instances from the same population, and create a new model using measurements made for the instances, the model parameters (e.g. regression coefficients or PCA loadings) will be different. The variation of the parameters depends on several factors, including complexity of the model, heterogeneity of the population and number of the instances.

One of the ways to estimate and account for the sampling error is to apply the model to a new set of instances.[11] However, if the model also needs an additional optimization step (for example to find optimal model complexity, sequence of preprocessing methods, do variable selection, etc.), this requires a use of two additional sets — one for optimization (usually called as validation set) and one for the final testing of the optimized model (test set), which is not always possible. Therefore, using cross-validation for the optimization step is a reasonable and thus widely used alternative.

The general idea of cross-validation is to create several models for different subsets of the calibration set. The variation among the models can be considered as an estimate (although, in some cases, this estimation is sub-optimal) of the sampling error thus cross-validation assesses both the fitting and the sampling errors together. This is employed, for example, in Jack-Knifing method for estimation of standard error of model parameters[12]

Our goal is to measure this variation and introduce it to the calibration set. The next sections describe this procedure in detail.

## Method description

Two kinds of PCA models — global and local — are considered in this manuscript. The global model is the one based on the whole calibration set. All results obtained for the global

model (scores, loadings, etc.) appear in the text without any indices (e.g. $\mathbf{P}$ is a matrix with loadings computed using the whole calibration set, $\mathbf{X}$). Here we assume that we deal with already preprocessed (e.g. mean centered or autoscaled) data.

In terms of the global model we consider PCA decomposition of data matrix $\mathbf{X}$ with $I$ rows (observations) and $J$ columns (variables) using $A$ principal components as:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{P}^{\mathrm{T}} + \mathbf{E} = \hat{\mathbf{X}} + \mathbf{E} \tag{1}$$

Here $\mathbf{T}$ is a matrix of scores ($I \times A$), $\mathbf{P}$ is a matrix of loadings ($J \times A$), $\mathbf{E}$ is a matrix of residuals ($I \times J$). The matrix $\mathbf{\Lambda}$ ($A \times A$) is a diagonal matrix with eigenvalues for the scores:

$$\mathbf{\Lambda} = \mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathrm{diag}(\lambda_1, ..., \lambda_{\mathrm{A}}) \tag{2}$$

And the matrix $\mathbf{U}$ ($I \times A$) is a matrix with normalized scores:

$$\mathbf{U} = \mathbf{T}\mathbf{\Lambda}^{-1/2} \tag{3}$$

Principal components form a subspace $\mathbb{S}$, spanned by the loading vectors (columns of $\mathbf{P}$). Thereby the score matrix $\mathbf{T}$ contains coordinates of projections of the original data points (rows of $\mathbf{X}$) to this subspace. These projections can be also represented in the original variable space by computing matrix $\hat{\mathbf{X}}$:

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{\Pi} \tag{4}$$

Where $\mathbf{\Pi} = \mathbf{P}\mathbf{P}^{\mathrm{T}}$ is a projection matrix. The size of $\hat{\mathbf{X}}$ is $I \times J$ but the rank is $A$.

Let $\mathbf{x}$ be a row-vector either from the calibration set or from a test set. Its relation with the subspace $\mathbb{S}$ can be described by two statistics: the squared orthogonal Euclidean distance to the $\mathbb{S}$, $q$:

$$q(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \|\mathbf{x}(\mathbf{I} - \mathbf{\Pi})\|^2 \tag{5}$$

and the squared Mahalanobis distance (also called as *score distance*), $h$, between the projection of the vector and the origin of the $\mathbb{S}$:

$$h(\mathbf{x}) = \|\mathbf{u}\|^2 = \|\mathbf{x}\mathbf{P}\mathbf{\Lambda}^{-1/2}\|^2 \tag{6}$$

The two distances are usually visualized graphically in form of the *Distance plot* and can be used to assess how well a particular measurement/observation is fitted by the corresponding PCA model. By comparing the distances with critical limits, computed at given significance level, $\alpha$, one can classify the observations as extremes or outliers. Both distances follow $\chi^2$-distribution with parameters (degrees of freedom and scaling factor) which can be computed from the distance values.[13]

Another important application of the orthogonal and score distances is the *Extreme plot*,[13,14] where the number of observed extreme objects is plotted against the theoretically expected number for different significance levels, $\alpha$. As it was shown in,[15] this plot is efficient both for estimation of the correct model complexity and for assessing if a new set of measurements/observations comes from the same population as the calibration set.

**Cross-validation of PCA model**

In case of $K$-fold cross-validation, the rows of the original data matrix, $\mathbf{X}$, are split (either randomly or systematically) into a set of $K$ segments $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\mathrm{K})^T$ with $I/K$ rows in each. Then, at every step $k$, the $k$-th segment, $\mathbf{X}_\mathrm{k}$, is taken out from the set and a local PCA model is developed using the other segments $\tilde{\mathbf{X}}_\mathrm{k} = (\mathbf{X}_1, ..., \mathbf{X}_{\mathrm{k}-1}, \mathbf{X}_{\mathrm{k}+1}, \ldots, \mathbf{X}_\mathrm{K})^\mathrm{T}$ as a calibration set:

$$\tilde{\mathbf{X}}_\mathrm{k} = \tilde{\mathbf{T}}_\mathrm{k}\mathbf{P}_\mathrm{k}^\mathrm{T} + \tilde{\mathbf{E}}_\mathrm{k} = \tilde{\mathbf{U}}_\mathrm{k}\mathbf{\Lambda}_\mathrm{k}^{1/2}\mathbf{P}_\mathrm{k}^\mathrm{T} + \tilde{\mathbf{E}}_\mathrm{k} \tag{7}$$

The principal components of the local model form another subspace, $\mathbb{S}_\mathrm{k}$, spanned by the columns of $\mathbf{P}_\mathrm{k}$. Matrix $\tilde{\mathbf{T}}_\mathrm{k}$ contains coordinates of projections of the data points to this subspace. The points from $\mathbb{S}_\mathrm{k}$ can also be represented in the original variable space as a result of projection: $\tilde{\mathbf{X}}_\mathrm{k}\mathbf{\Pi}_\mathrm{k}$ with projection matrix $\mathbf{\Pi}_\mathrm{k} = \mathbf{P}_\mathrm{k}\mathbf{P}_\mathrm{k}^\mathrm{T}$.

Now let's take a row-vector $\mathbf{x}$ from the excluded segment $\mathbf{X}_\mathrm{k}$. Its relation with the subspace $\mathbb{S}_\mathrm{k}$ can be described using the same two statistics. The orthogonal distance, $q_\mathrm{k}$:

$$q_\mathrm{k}(\mathbf{x}) = \|\mathbf{x}(\mathbf{I} - \mathbf{\Pi}_\mathrm{k})\|^2 \tag{8}$$

and the score distance, $h_\mathrm{k}$:

$$h_\mathrm{k}(\mathbf{x}) = \|\mathbf{x}\mathbf{P}_\mathrm{k}\mathbf{\Lambda}^{-1/2}\|^2 \tag{9}$$

### Creating a pseudo-validation set

We would like to create a new dataset $\mathbf{X}_{\text{pv}}$, with the same $I$ observations as $\mathbf{X}$, such that its relationship with $\mathbb{S}$ (in terms of the coordinates of projections as well as the score and orthogonal distances) is the same as relationship of segments $\mathbf{X}_{\text{k}}$ with $\mathbb{S}_{\text{k}}$.

The idea is visualized using a trivial case based on two dimensional variable space and full cross-validation (leave-one-out) presented in Figure 1. First, a global PCA model is created using the calibration set, $\mathbf{X}$, as it is shown on the left plot. The thick black line represents the principal component space, $\mathbb{S}$, which in this case consists of one PC only.

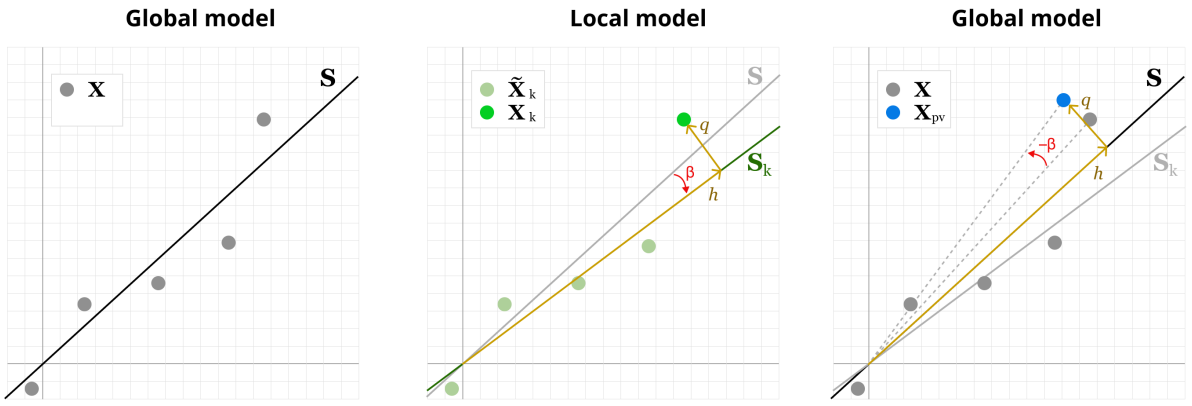


Figure 1: Illustration of pseudo-validation set concept. The left plot shows the global PCA model. The middle plot represents a local model created for one step of cross-validation, angle $\beta$ is a measure of difference between the models. The right plot shows how sample for pseudo-validation set is created by rotating the original sample to angle $-\beta$.

Then, for each cross-validation iteration, $k$, the calibration set is split into $\tilde{\mathbf{X}}_{\text{k}}$ and $\mathbf{X}_{\text{k}}$ and a local model is created using $\tilde{\mathbf{X}}_{\text{k}}$ as it is shown in the middle plot. The difference between the local model, $\mathbb{S}_{\text{k}}$, and the global model, $\mathbb{S}$, can be measured as an angle, $\beta$, between the corresponding principal components. We consider this angle as an estimate of the sampling error — uncertainty in model parameters due to sample-to-sample variation.

This error can then be introduced to the calibration set if we rotate the object from $\mathbf{X}_{\text{k}}$ by

$-\beta$, as it is shown in the right plot. The rotated object is a part of the new set, $\mathbf{X}_{\text{pv}}$, which we call a *pseudo-validation* set. To complete the pseudo-validation set we need to take all iterations in the cross-validation procedure. In case of full cross-validation, every object is rotated to its own angle, as it is shown in the right plot. For segmented cross-validation, objects from the same segment are rotated to the same angle.

As one can see from the Figure 1, the orthogonal and score distances computed for $\mathbf{X}_{\text{k}}$ and the local model (middle plot) are identical to the distances computed for $\mathbf{X}_{\text{pv}}$ and the global model (right plot). If model contains more than one PC, the angles are measured for each pair of the components and the rotations are carried out consequently from the first component to the last, keeping the effect of the previous steps unchanged.

The concept illustrated in the Figure 1 can be generalized as follows. For a given cross-validation procedure with number of segments, $K$, and number of components $A$, on each step $k = 1, ..., K$ do the following:

1. Exclude $k$-th segment from the data, to create a local calibration set $\tilde{\mathbf{X}}_{\text{k}}$.
2. Use $\tilde{\mathbf{X}}_{\text{k}}$ to develop a local model, represented by loading matrix $\mathbf{P}_{\text{k}}$.
3. Compute $J \times J$ rotation matrix $\mathbf{R}_{\text{k}}$, such as $\mathbf{P}_{\text{k}} = \mathbf{R}_{\text{k}}\mathbf{P}$. The matrix keeps length of vectors in the original space and angles between them invariant.
4. Apply the matrix to the excluded segment $\mathbf{X}_{\text{k}}$: $\mathbf{X}_{\text{pv}_{\text{k}}} = \mathbf{X}_{\text{k}}\mathbf{R}_{\text{k}}^{\text{T}}$

Finally, all matrices $\mathbf{X}_{\text{pv}_{\text{k}}}$ are combined together to form the pseudo-validation set $\mathbf{X}_{\text{pv}}$: $\mathbf{X}_{\text{pv}} = (\mathbf{X}_{\text{pv}_1}, \mathbf{X}_{\text{pv}_2}, ..., \mathbf{X}_{\text{pv}_{\text{K}}})^{\text{T}}$. One can imagine that on each step we find a way to put the current local model on to the "Procrustean bed" of the global model — hence the name.

It can be also noted, that the computation of the rotational matrix is carried out step by step, from one component to another. First, the angle between PC1 of the global model and

PC1 of the local model is found and the corresponding transformation is computed. Then the procedure is repeated for PC2, but the transformation is computed to keep distances related to the rotation of the PC1 unchanged. So, in creating the PV-set, application of extra components does not influence the results from the previous ones

Or, in more formal way, if we create two pseudo-validation sets, one using $A = A_1$ components and second one using $A = A_2$ components, and apply each for validation of a PCA model developed for $A = A_3$ components, then the results of the validation will be identical if $A_3 \leq A_2 \leq A_1$.

More details about computation of the rotation matrix $\mathbf{R}_\mathrm{k}$ can be found in supplementary materials. Both algorithms (for calculation of the rotation matrix and computing the pseudo-validation set) are implemented as R and MATLAB code and can be freely downloaded from GitHub: https://github.com/svkucheryavski/pcv.

# Datasets

Three datasets are selected for testing the proposed approach and investigate properties of the corresponding pseudo-validation sets.

### NIRSim

The first dataset consists of simulated spectral data. It is obtained by computing linear combinations of NIR spectra of 6 chemical components and added Gaussian noise using procedure described in.[15] Three subsets, $100 \times 200$ each, are created by using Gaussian noise with $\mu = 0$ and three different values for $\sigma$: $\sigma = 0$ (no noise), $\sigma = 0.01$ and $\sigma = 0.04$.

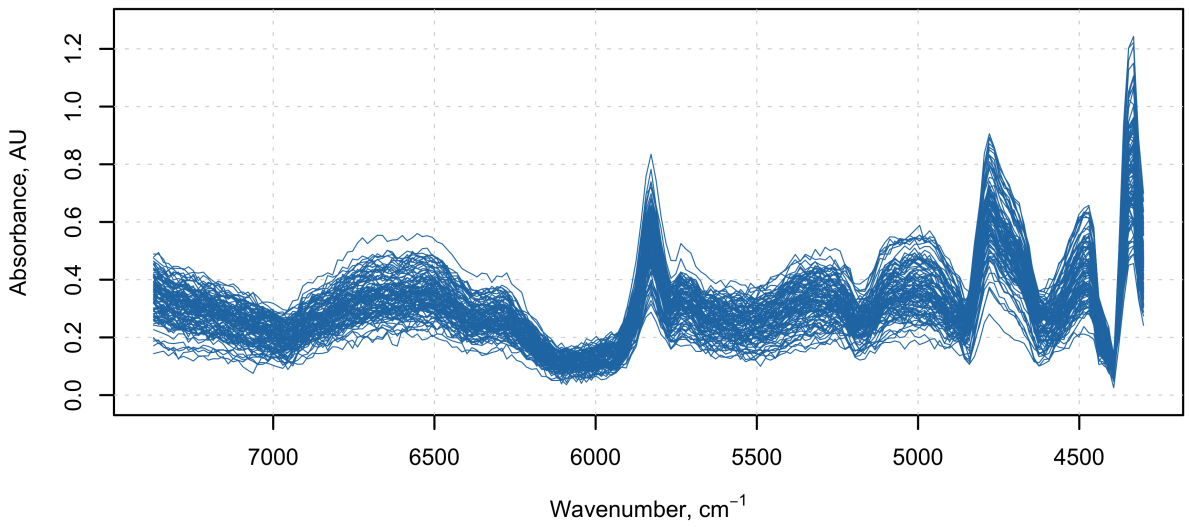


Figure 2: Simulated NIR spectra (NIRSim) for $\sigma = 0.01$.

If PCA is applied to the subset without noise, the corresponding eigenvalues are $\lambda = [56.25, 25, 6.25, 1, 0.25, 0.01]$, which means that four PCs explain most of the systematic variation for this data. However, adding noise obscures some of the variation, thus for subset with $\sigma = 0.01$ the optimal number of components is 3 and for $\sigma = 0.04$ this number is reduced to two. Figure 2 shows the raw spectra from the second subset ($\sigma = 0.01$) as an example. More details about the dataset can be found in.[15]

**Olives**

The second dataset consists of NIR spectra of olives in brine measured in range of 4000—9000 cm$^{-1}$ with 4 cm$^{-1}$ resolution (1 243 wavenumbers in total) in diffusion reflectance mode. For the purpose of this paper, spectra from the same population (same olive type) are selected, cleared for outliers, and divided randomly into calibration ($n = 72$) and test ($n = 40$) sets. The spectra are normalized using Standard Normal Variate (SNV) prior to the

analysis. Figure 3 shows the normalized spectra from the calibration set as an example. More information about the data can be found elsewhere.[14,16]

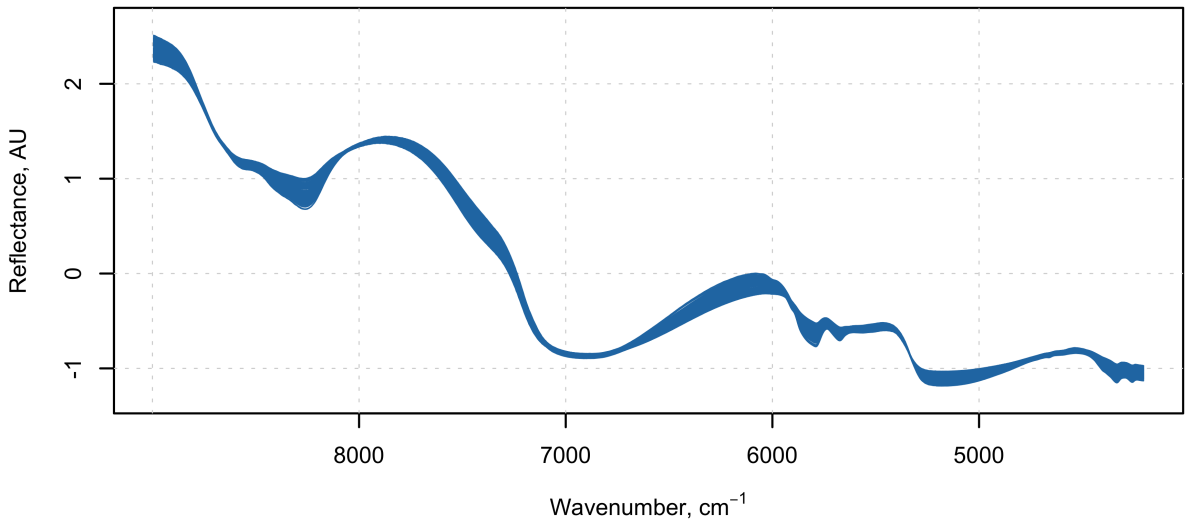


Figure 3: SNV corrected spectra of Olives.

**Wines**

The third dataset contains chemical and physical characteristics (27 variables in total) of 178 wine samples from three different origins: *Barolo* (59 samples), *Grignolino* (71 samples), and *Barbera* (48 samples). The dataset is described in detail in.[17] It has been widely used in various research mostly related to classification methods, for example.[18,19]

# Experiment and results

## Comparing calibration and pseudo-validation sets

The main goal of this part is to explore the similarity between calibration and pseudo-validation sets using PCA tools described in the theoretical section. The *NIRSim* dataset is employed for this purpose. To carry out the investigation, a pseudo-validation set is computed for every *NIRSim* subset using cross-validation procedure with systematic splits (venetian blinds), $K = 4$ and $A = 6$. So, for every noise level, there are two sets of spectra — the original simulated values and the spectra from the corresponding pseudo-validation set.

In theory, the same row, taken from the calibration and the pseudo-validation sets, can be considered as data values obtained for two individuals with similar properties taken from the same population. So the difference between the individuals should be random and magnitude of the difference should be related directly to the heterogeneity of the population — one can think of NIR spectra of two apples taken from the same tree and having similar sugar and water content, for example.

Formally speaking, this means that the pseudo-validation set comes from the same population as the calibration set, however has its own sampling error. For PCA this means the following:

1. The pseudo-validation set should be well fitted by a PCA model trained on the original data values if number of components does not exceed an optimal value (so the model is not overfitted).

2. Swapping calibration and pseudo-validation sets should give similar PCA models and indicate the same number of optimal components.

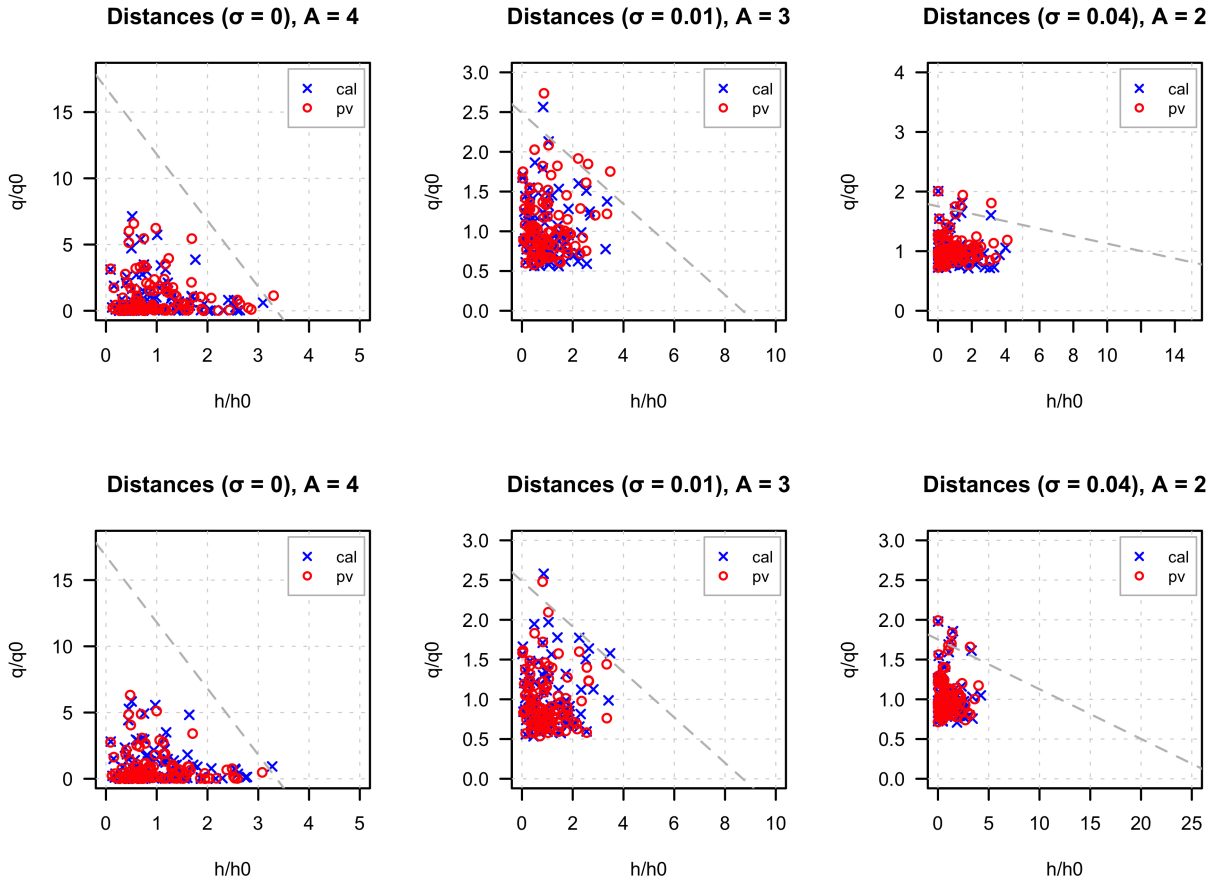


Figure 4: PCA distance plot made for *NIRSim* data with different noise level. The top plots represent models trained using original simulated spectral values. The bottom plots are made using models trained on the corresponding pseudo-validation sets.

Since for *NIRSim* dataset the number of optimal components is known *a priori* we can test both statements using distance and extreme plots for assessing similarity of the two sets in a multivariate sense. Figure 4 demonstrates distance plots created for the PCA models using optimal number of components for each case. Plots on the top were created using models trained on the original simulated spectra and then applied to the corresponding pseudo-validation set. The bottom plots were created for models trained on the pseudo-validation set and applied to the original spectra for validation.

As one can see, there is a very good agreement between the two sets on all six plots. In none

of the cases one set of points looks more extreme comparing to the other. The total number of extreme values lying outside the critical limit, shown as a dashed line, does not exceed theoretically expected value of 5%.

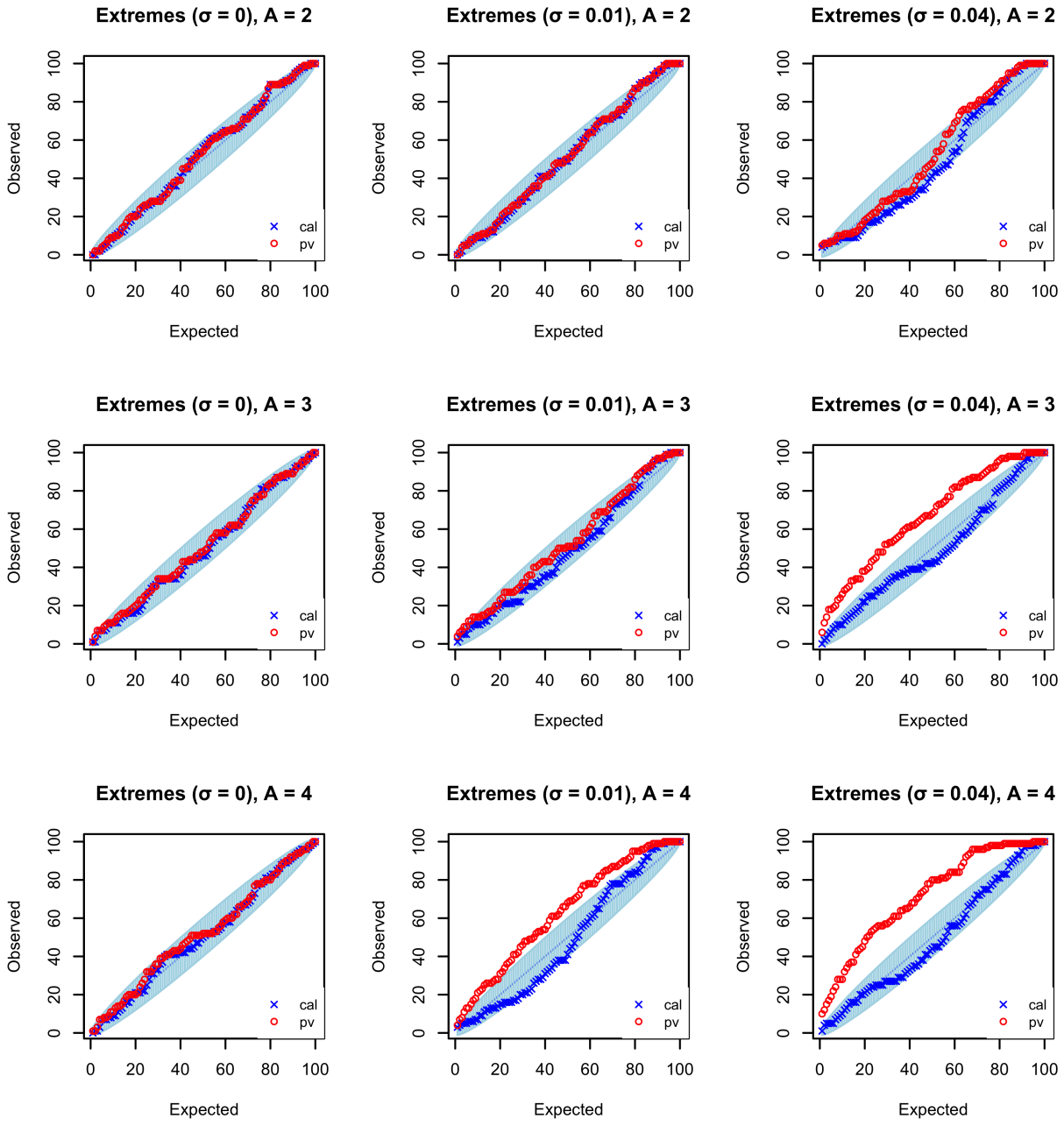


Figure 5: Extreme plots made for PCA models trained on the original simulated spectra.

However, despite the similarity, we can show that if number of components in PCA model

exceeds the optimal value, the pseudo-validation set indicates this clearly. Figure 5 shows extreme plots created for the three PCA models trained using the original simulated spectra and then applied to the pseudo-validation sets.

The plots in the first (left) column are created using the spectra without noise. As we know, all six PCs are important for explaining the systematic variation in this case. It can be also seen in the extreme plots — both calibration and pseudo-validation sets are well described by the PCA model as corresponding points are located within the tolerance interval shown as a blue ellipse. This means that the number of extreme objects identified by the PCA model based on the score and the orthogonal distances at different significance levels, $\alpha$, (ordinate axis) is in agreement with the theoretical expectations (abscissa axis).

However, if we look at the plots made for the spectra with moderate level of noise ($\sigma = 0.01$), shown in the second column, we can notice a disagreement on extreme plot created for $A = 4$. In this case, the optimal number of PCs is 3 and the 4th PC describes noise in the original data making pseudo-validation set looking as objects from another population. The same pattern can be seen for the data with higher level of noise (right column), with the only difference that the disagreement starts at $A = 3$ in this case and there is even more clear effect in the extreme plots.

Summarizing the results for the simulated spectra we can conclude that the pseudo-validation set indeed behaves similar to the calibration set. However, it shows clearly the effect of overfitting in PCA models, in contrast to the calibration set. Thus pseudo-validation set is suitable for validation.

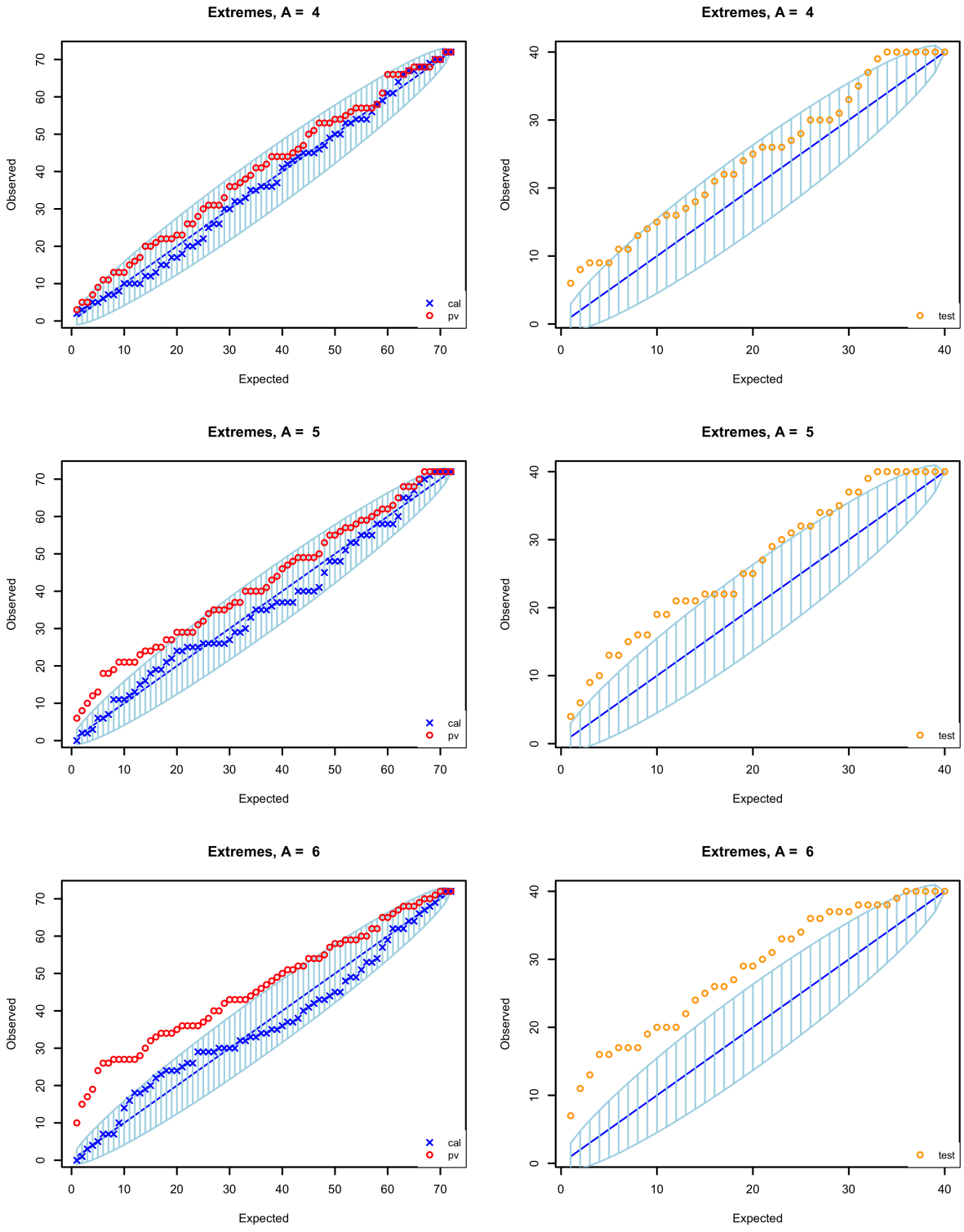


Figure 6: Extreme plots made for the calibration and pseudo-validation sets (left) as well as for the test set (right) using different number of components.

## Comparing pseudo-validation and independent test sets

The main goal of this part is to compare the results obtained using pseudo-validation with the results based on the independent test set. The *Olives* data is used in this case.

First, a PCA model is created using spectra from the calibration set (72 measurements). However, it is difficult to estimate how many components to use in the PCA model. Traditional tools, like total residual variance plot, are not able to point clearly on the optimal number. In this case, the use of the extreme plot made for a new set of observations can be particularly efficient. The plot made for optimal or underfitted model shows similar behavior for the calibration and the test sets. In case of overfitting, the results for the test set clearly demonstrate the lack of fit on the plot, as it is also shown in the previous section.

Figure 6 demonstrates the extreme plots created for the calibration and pseudo-validation sets (left column) and for the test set (right column) using number of PCs in the range of question ($A = 4$, 5, and 6). Apparently, the results for the calibration set do not reveal a clear indication for the overfitting — for all three cases the number of observed extreme objects is within the tolerance intervals.

In contrast, the extreme plot made for the test set, shows clearly that starting from $A = 5$ model is getting overfitted — the number of observed extreme objects is larger than the expected. In case of $A = 6$ the effect is even more clear, also for the large $\alpha$.

The results for the pseudo-validation set also clearly show signs of overfitting on the extreme plot. Visually there is a difference between the behavior of points for the test and the pseudo-validation sets, which is, however, quite small. If we make another random split and repeat the overall procedure the effect is very similar (tried on five iterations, results are not shown here).

## Using PCV on non-spectroscopic data

The *Wines* data is used for this objective. For the sake of simplicity and better visualization, only two classes, *Barolo* and *Barbera*, are taken. In this case we deal with two groups of samples which are part of two different populations — wine origins. At the beginning, the groups were treated independently: the PCV algorithm is applied to the original values from each group, using $A = 20$ and $K = 4$, for creating the corresponding pseudo-validation sets. This procedure resulted in four data sets — the original values and the PV-sets for each population, as it is shown in the top diagram in Figure 7.

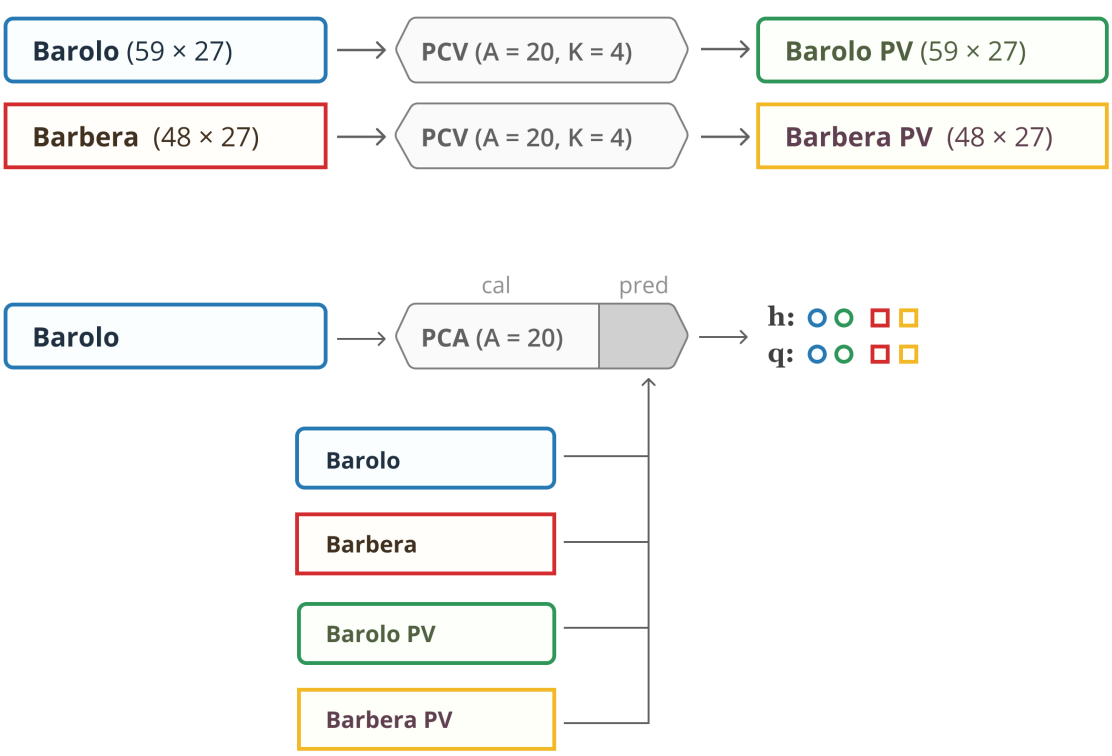


Figure 7: Flow diagrams for the *Wines* experiment. Top — creating of pseudo-validation sets. Bottom — PCA analysis workflow.

Similar to the previous two cases, PCA is employed for analysis of the four sets, however in this case we use the Distance plot as a main tool. The analysis workflow is shown

schematically in the bottom chart on the Figure 7 and consists of the following steps:

1. The original values for the *Barolo* class are taken as calibration set and a PCA model is created.

2. All four sets (the original values and the values from the PV-sets) are projected to the model subspace.

3. The score and the orthogonal distances are computed for each observation using different number of components in the model, $A = 2$ and $A = 20$.

The selection of the number of components are made to get distance values for the model with optimal complexity ($A = 2$) and for the clearly overfitted model ($A = 20$). The results are shown in form of distance plots in Figure 8 (two top plots). Different marker shapes are used to separate the different wine types while colors are employed to distinguish among the four sets.

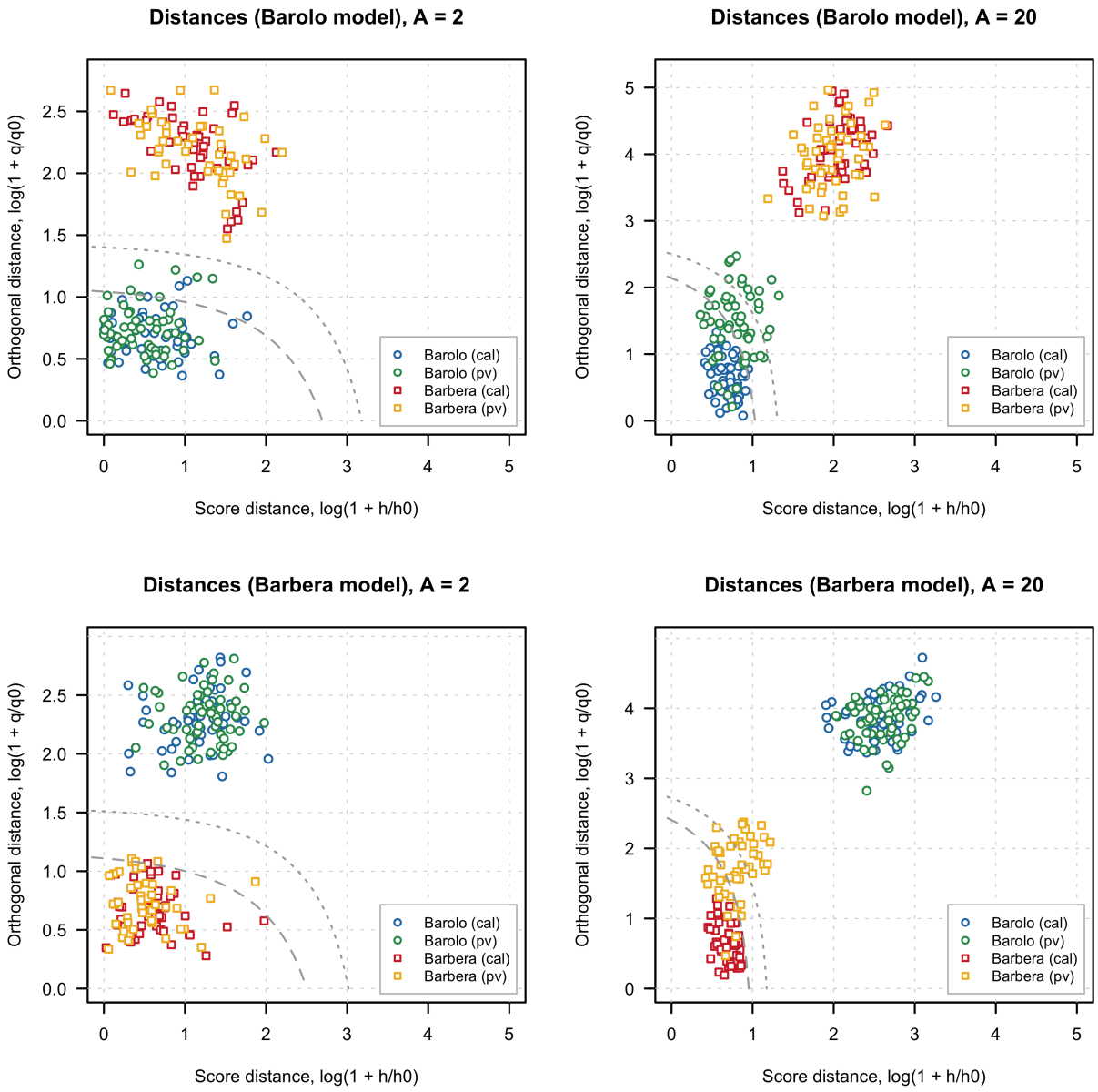


Figure 8: PCA Distance plots for different number of PCs. The top plots are made for PCA model created using *Barolo* original samples as calibration set. The bottom plots show the results for PCA model trained on the *Barbera* original values.

As one can see, in case of model with $A = 2$, there is a very good agreement in behavior of the original data values and the corresponding pseudo-validation sets. In both groups, the points from the two sets are overlapped with the same center and span. However, for the overfitted model, $A = 20$, the situation is different. While the points from *Barbera* group

demonstrate same behavior, points for the *Barolo* samples are split and observations from the pseudo-validation set have much larger score and orthogonal distance which, as we know from the other two cases, is typical sign for overfitting.

The bottom part of the Figure 8 contains similar plots, however in this case PCA model was created using the original values from the *Barbera* class. Obviously the behavior of the points is quite similar to the previous example. We can conclude that despite the different nature of this data, the behavior of the pseudo-validation set is similar to the two other cases discussed before and is inline with our expectations.

# Conclusions

The numerical experiments and results for simulated and real datasets confirm that the pseudo-validation set, computed using the proposed Procrustes cross-validation approach, is an efficient way to validate PCA/SIMCA models on the optimization step. On the one hand, it has main properties of an independent validation set and gives a possibility to get a full stack of results, based on validation of a single model, including scores, distances and explained/residual variance. On the other hand, the performance statistics provided by the pseudo-validation set (e.g. number of objects that has been considered as extreme) are similar to the ones obtained using conventional cross-validation. The latter leads to reproducible and comparable with conventional cross-validation results, which, from our point of view, makes dissemination of the presented approach easier.

# Supplementary materials

## S1. Computing rotation matrix between two latent variable subspaces

Computing the rotation matrix $\mathbf{R}_k$ is the most important step in generating $\mathbf{X}_{pv}$. In this case we suggest to use the algorithm for rotation of vectors in multidimensional space described in.[20] For any given subspaces $\mathbb{S}$ and $\mathbb{S}_k$ represented by a set of orthonormal basis vectors $\mathbf{P}$ and $\mathbf{P}_k$ of dimension $A$ in original variables space, $\mathbb{R}^J$, we can find rotation between the two subspaces, represented as $J \times J$ matrix $\mathbf{R}_k$ as follows.

If there is only one component in each subspace, then the procedure is quite straightforward:

1. Find a $J \times J$ rotation matrix $\mathbf{R}_1$, which aligns the first vector (column) from $\mathbf{P}$ with vector $[1, 0, ..., 0]$ in the $\mathbb{R}^J$.
2. Find a $J \times J$ rotation matrix $\mathbf{R}_{k_1}$, which aligns the first vector (column) from $\mathbf{P}_k$ with vector $[1, 0, ..., 0]$ in the $\mathbb{R}^J$.
3. Compute $\mathbf{R}_k$ as $\mathbf{R}_k = \mathbf{R}_{k_1}^{\mathrm{T}} \mathbf{R}_1$

However, in case of two or more components, we need to repeat this procedure and, at the same time, keep the already aligned vectors unchanged. The consequent rotations should be found in a lower dimensional space ($\mathbb{R}^{J-1}$ for PC2, $\mathbb{R}^{J-2}$ for PC3 and so on), so the vector $[1, 0, ..., 0]$ used in the previous step is a normal vector to this space. Here is an algorithm for PC2.

1. Rotate $\mathbf{P}$ and $\mathbf{P}_k$ using the rotation matrices found for PC1: $\mathbf{P}^{(1)} = \mathbf{R}_1 \mathbf{P}$ and $\mathbf{P}_k^{(1)} = \mathbf{R}_{k_1} \mathbf{P}_k$.

2. Remove first row and first column from both $\mathbf{P}^{(1)}$ and $\mathbf{P}_k^{(1)}$. So this will not affect PC1, which are already aligned, and let us to work in $\mathbb{R}^{J-1}$ space.

3. Find a $(J-1) \times (J-1)$ rotation matrix $\mathbf{R}_2$, which aligns the first vector (column) from the reduced $\mathbf{P}^{(1)}$ (PC2) with vector $[1, 0, ..., 0]$ in the $\mathbb{R}^{J-1}$.

4. Find a $(J-1) \times (J-1)$ rotation matrix $\mathbf{R}_{k_2}$, which aligns the first vector (column) from the reduced $\mathbf{P}_k^{(1)}$ (PC2) with vector $[1, 0, ..., 0]$ in the $\mathbb{R}^{J-1}$.

5. Compute $\mathbf{R}_s$ as $\mathbf{R}_s = \mathbf{R}_{k_2}^{\mathrm{T}} \mathbf{R}_2$

6. Compute $J \times J$ matrix $\mathbf{M}$ as (here $\mathbf{z}$ is a column-vector with zeros.):

$$\mathbf{M} = \begin{bmatrix} 1 & \mathbf{z}^{\mathrm{T}} \\ \mathbf{z} & \mathbf{R_s} \end{bmatrix} \tag{10}$$

7. Compute final rotational matrix $\mathbf{R}_k$ as $\mathbf{R}_k = \mathbf{R}_{k_1}^{\mathrm{T}} \mathbf{M} \mathbf{R}_1$

In case of more than two components, the procedure is repeated recursively until number of columns in $\mathbf{P}^{(l)}$ and $\mathbf{P}_k^{(l)}$ is not equal to one.

# References


(1) Harrington, P. B. Statistical validation of classification and calibration models using bootstrapped Latin partitions. *TrAC Trends in Analytical Chemistry* **2006**, *25*, 1112–1124.

(2) Filzmoser, P.; Liebmann, B.; Varmuza, K. Repeated double cross validation. *Journal of Chemometrics* **2009**, *23*, 160–171.

(3) Baumann, D.; Baumann, K. Reliable estimation of prediction errors for QSAR models under model uncertainty using double cross-validation. *Journal of Cheminformatics* **2014**, *6*, 1–19.

(4) de Boves Harrington, P. Multiple Versus Single Set Validation of Multivariate Models to Avoid Mistakes. *Critical Reviews in Analytical Chemistry* **2018**, *48*, 33–46.

(5) Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. 1995; pp 1137–1143.

(6) Esbensen, K. H.; Geladi, P. Principles of Proper Validation: use and abuse of re-sampling for validation. *Journal of Chemometrics* **2010**, *24*, 168–187.

(7) Bro, R.; Kjeldahl, K.; Smilde, A. K.; Kiers, H. A. Cross-validation of component models: A critical look at current methods. *Analytical and Bioanalytical Chemistry* **2008**, *390*, 1241–1251.

(8) Despagne, F.; Massart, D. L.; De Noord, O. E. Optimization of Partial-Least-Squares Calibration Models by Simulation of Instrumental Perturbations. *Analytical Chemistry* **1997**, *69*, 3391–3399.

(9) Cox, D. R.; Snell, E. J. A General Definition of Residuals. *Journal of the Royal Statistical Society. Series B (Methodological)* **1968**, *30*, 248–275.

(10) Quenouille, M. H. Problems in Plane Sampling. *The Annals of Mathematical Statistics* **1949**, *20*, 355–375.

(11) James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*; Springer-Verlag: New York, 2013; p 426.

(12) Efron, B.; Stein, C. The Jackknife Estimate of Variance. *The Annals of Statistics* **1981**, *9*, 586–596.

(13) Pomerantsev, A. L.; Rodionova, O. Y. Concept and role of extreme objects in PCA/SIMCA. *Journal of Chemometrics* **2014**, *28*, 429–438.

(14) Rodionova, O. Y.; Oliveri, P.; Pomerantsev, A. L. Rigorous and compliant approaches to one-class classification. *Chemometrics and Intelligent Laboratory Systems* **2016**, *159*, 89–96.

(15) Poverantsev, A.; Rodionova, O. Popular Decision Rules in SIMCA: Critical Review. *Journal of Chemometrics* **2020**,

(16) Oliveri, P.; López, M. I.; Casolino, M. C.; Ruisánchez, I.; M. Callao, P.; Medini, L.; Lanteri, S. Partial least squares density modeling (PLS-DM) - A new class-modeling strategy applied to the authentication of olives in brine by near-infrared spectroscopy. *Analytica Chimica Acta* **2014**, *851*, 30–36.

(17) Forina, M.; Armanino, C.; Castino, M.; Ubigli, M. Multivariate data analysis as a discriminating method of the origin of wines. *Vitis* **1986**, *25*, 189–201.

(18) Li, T. H. S.; Guo, N. R.; Kuo, C. L. Design of adaptive fuzzy model for classification problem. *Engineering Applications of Artificial Intelligence* **2005**, *18*, 297–306.

(19) Aeberhard, S.; Coomans, D.; Vel, O. D. Improvements to the classification performance of RDA. *Journal of Chemometrics* **1993**, *7*, 99–115.

(20) Zhelezov, O. I. N-dimensional Rotation Matrix Generation Algorithm. *American Journal of Computational and Applied Mathematics* **2017**, *7*, 51–57.