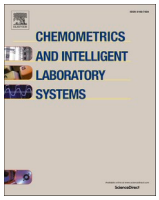
Chemometrics and Intelligent Laboratory Systems 251 (2024) 105166

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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



Latent variable model inversion for intervals. Application to tolerance intervals in class-modelling situations, and specification limits in process control

M.S. Sánchez [a], M.C. Ortiz [b,*], S. Ruiz [a], O. Valencia [a], L.A. Sarabia [a]

[a] Dpto. Matemáticas y Computación, Universidad de Burgos, Facultad de Ciencias, Pza. Misael Bañuelos s/n, 09001, Burgos, Spain
[b] Dpto. De Química, Universidad de Burgos, Facultad de Ciencias, Pza. Misael Bañuelos s/n, 09001, Burgos, Spain

ARTICLE INFO

Keywords:
PLS
LVMI
Tolerance intervals
Specification limits

ABSTRACT

The paper deals with the inversion of intervals when a PLS (Partial Least Squares) model is used. However, instead of discretizing the interval, it is proved that the region resulting from the inversion of a PLS model is a convex set bounded by two parallel hyperplanes, each corresponding to the direct inversion of each endpoint of the given interval.

When the domain of the input variables is a convex set, any feasible solution with predictions within the interval set in the response can be obtained as a convex combination of a point on each of the two hyperplanes. In this way, the new solutions preserve the internal structure of the input variables.

This methodology can be of interest in several domains where the response under study is defined in terms of an interval of admissible values, such as specifications for a product in an industrial process, or tolerance intervals for computing compliant class-models.

The inversion of the corresponding fitted model defines a region in the input space (predictor variables) whose predictions fall within the specified interval. Then, estimating and exploring this region will increase the information about the problem under study.

1. Introduction

Latent Variable Model Inversion (LVMI) [1] refers to the local inversion of a prediction model, based on latent variables, that has been fitted to predict some response(s) in **Y** from p predictor variables in **X**. Briefly, it involves defining target values in the response space and finding the corresponding settings of the p predictors so that the predicted values are those previously fixed.

The advantage of using latent variable prediction models is that they take into account the structure (rank and correlation) of the predictor variables, which vary in the *input* space. In addition, when using PLS (Partial Least Squares), the correlation with **Y** is also taken into account when computing the model. On the negative side, the mathematical inversion is more difficult, usually the inverse function, properly speaking, does not even exist.

The methodology presented here deals with a single response, so the **Y** matrix is actually a vector, usually denoted as **y**, and the *output* space is one-dimensional. The inversion focuses on a range of values, that is, an interval in the space of the response. The inversion will provide specific regions in the input space associated with the predefined interval.

The potential usefulness of the approach goes beyond, for example, considering the confidence interval on prediction (as in [2]). This is illustrated by some case-studies, where intervals are used in different ways. The ultimate goal of the inversion is to help gain insight by increasing knowledge about the process/procedure by exploring the region resulting from the model inversion for the entire interval.

Two of the developed case-studies correspond to class-modelling [3]. This means that each of the n objects in **X** is known to belong (or not) to a class or category of interest, C. Then, vector **y** will consist of binary values {-1, 1}, where 1 denotes the category under study and −1 identifies the objects that do not belong to C.

When working with classes (categories), there is a difference between discriminant and class-modelling methods. Discriminant methods compute decision rules in the input space to assign each object necessarily to one of the categories under study. In the situation established in the previous paragraph, a discriminant method will assign either 1 or -1

* Corresponding author.
E-mail address: mcortiz@ubu.es (M.C. Ortiz).

https://doi.org/10.1016/j.chemolab.2024.105166
Received 26 April 2024; Received in revised form 28 May 2024; Accepted 13 June 2024
Available online 18 June 2024

(necessarily one of them) to each object in **X**. Therefore, they are usually evaluated in terms of accuracy, i.e., the percentage of correct classifications.

In contrast, class-modelling methods compute regions within the input space, called class-models. Thus, a class-model is a region of the input space, mathematically defined with the predictor variables, in which objects of that class are expected to fall. In this case, an object can be inside one or more class-models, and even outside all of them. Validation of such class-models is done in terms of sensitivity and specificity. Sensitivity reflects the ability of the class-model to contain all objects of the category, while specificity is related to the ability of the class-model to leave out *foreign* objects. Consequently, both parameters are related: larger class-models will have better sensitivity but worse specificity, while more restricted class-models are likely to have better specificity at the expense of reduced sensitivity.

In terms of their estimates, sensitivity is estimated as the percentage of objects of *C* that are inside the class-model, while specificity is estimated as the percentage of objects not belonging to *C* that are outside the class-model. However, if the class-model is built using only objects of *C* (the so-called one-class classifiers [4]), only sensitivity can be estimated, with the risk of overestimating the class-model. To estimate specificity, objects that do not belong to *C* are also needed. When these samples are actually used to build the class-models, they are called compliant class-models [5]. With more than two classes studied together, simultaneous compliant class-models are built in ref [6], and single indices [7,8] have been proposed to assess the goodness of all class-models as a whole, taking into account that there are multiple values of sensitivity (one per class) and specificity (especially when pairwise specificities are of interest).

A different aspect is how these models (whether they are decision rules between categories, or class-models per category) are built. Classical discriminant methods such as *k*-NN (*k* -Nearest Neighbors) or LDA (Linear Discriminant Analysis), or class-modelling methods such as SIMCA (Soft-Independent Modelling by Class Analogy) or UNEQ, build the models without regressing the response (label of the categories) on the predictor variables. However, there are other methods that work as a two-step process for classification: first, a regression model is fitted to the response, and then the predicted values are used to make the decision or define the class-model. Although NNs (neural networks) are common prediction methods, PLS is more frequently used in chemometrics, especially when dimensionality reduction is an issue. When PLS is used as a discriminant method, it is known as PLS-DA (Partial Least Squares Discriminant Analysis [9]), and when the PLS predictions are used to define class-models, it is PLS-CM (Partial Least Squares in Class-Modelling [10,11]).

In the cases studied here via model inversion, PLS will be the prediction model. However, a new alternative for constructing class-models is proposed, namely the use of tolerance intervals. Unlike confidence intervals on a parameter estimate, tolerance intervals are computed to contain, with a certain probability, a given percentage of the entire population. The inversion for the values in this interval will define the class-model for the class under study. In addition, using a tolerance interval with a high percentage of individuals in the population, but not all of them, prevents overestimation of the class-model at the cost, likely, of some loss of sensitivity.

This is applied to two different cases with data taken from the literature. The first, in Section 3.2, is about medical diagnosis. The class of interest, *C*, is sick individuals. The interest is not to make a confidence interval on a mean, but a tolerance interval to contain a given number of sick individuals with a given confidence. The other (Section 3.3) deals with the typification of food. In this case, a tolerance interval is made individually for each of two varieties of rice (two classes). The key idea then is to avoid the intersection of these two intervals in the response, to focus not on the common characteristics but on the distinctive ones. Therefore, the inversion will provide the values of the predictor variables with more influence on the difference.

Finally, latent variable model inversion is more widely used in the field of quality control of industrial processes. For this reason, another case-study is added in this context (Section 3.4), which also serves to illustrate some other properties that may be of interest when the input variables are manipulable, which is not the case in the first two case-studies. When constructing the compliant class-models for different classes, the predictors are not manipulable, only measurable, whereas in an industrial process the predictors usually consist of process variables that can be set to predefined values. The data [12] come from from an industrial chemical process with nine predictors and two responses, although only one of them is studied. In this case, the interval is that defined by the product specification to be met, which is typically bounded inferiorly by the Lower Specification Limit (LSL) and superiorly by the Upper Specification Limit (USL). Therefore, there is an interval [LSL, USL] for the response of interest. The results of inverting the PLS model for the interval are analyzed to explore how to intervene in the process to keep it within specification, or how to move [13] the process variables to "redirect" the process toward compliance with specifications.

2. Theoretical aspects

The paper addresses the inversion of PLS models, not for a single value, but for intervals. The intervals are defined in the response *output* space and represent desired target values. In particular, tolerance and specification intervals are considered, both of which are constructed using predicted values.

The following sections briefly summarize the elements and establish the notation necessary to follow the paper.

2.1. PLS prediction models

In general, let **X** be a matrix of predictors, with *n* objects and *p* variables (input variables), and **Y** be a matrix of responses, with the values quantifying *q* characteristics of interest of the same *n* objects. Unless otherwise specified, both matrices are autoscaled (that is, all variables are mean-centered and have variance one).

In mathematical terms, there is also a domain *D* in the *p*-dimensional space formed by the set of possible or admissible values of the input variables represented in **X.** It encompasses all the relevant aspects of the problem at hand (measured variables to represent the classes in the case of class-modelling, or process variables, material attributes, manufacturing or environmental conditions, etc., in the case of process control).

With matrices **X** and **Y**, a PLS model projects the data onto a space of $a < p$ latent variables chosen to maximize, in decreasing order, the product between the covariance of **X** and the correlation between **X** and **Y**. In this latent space, the regression model between the projections and the responses is computed. The PLS decomposition can be written as:

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\,\mathbf{P}^T + \mathbf{R_X} \\ \mathbf{Y} &= \mathbf{T}\,\mathbf{Q}^T + \mathbf{R_Y} \\ \mathbf{T} &= \mathbf{X}\,\mathbf{W} \end{aligned} \tag{1}$$

where **T** is the orthonormal matrix of the projections, that is, the points in the latent space (scores), **P** is the matrix containing the loadings of the predictors in **X** and **Q** is the matrix of the loadings of **Y**, that is, the coefficients of the regression of **Y** on the common latent space **T** formed by the first *a* latent variables. **W** is a weight matrix (same size as **P**) such that $\mathbf{P}^T\mathbf{W} = \boldsymbol{I}_a$, where $\mathbf{I}_a$ is the identity matrix of size $a \times a$. As usual, superscript T denotes transposition, and $\mathbf{R_X}$ and $\mathbf{R_Y}$ are the matrices containing the residuals of the decomposition.

With the notation in equation (1), where the objects and their associated values of response are in rows, a vector of predictors (a configuration of the process) **x** is projected into the space defined by the latent variables by means of the weight matrix **W**, so that the vector of

scores in the a latent variables is the row vector $\mathbf{t}^T = (t_1,\ldots,t_a) = \mathbf{x}^T\mathbf{W}$. In the usual notation for vectors, this is

$$\mathbf{t} = \begin{pmatrix} t_1 \\ \ldots \\ t_a \end{pmatrix} = \mathbf{W}^T\mathbf{x} \tag{2}$$

Therefore, the predicted response (related to the characteristics in the output space) according to equation (1) will be the row vector $\hat{\mathbf{y}}^T = \mathbf{t}^T\mathbf{Q}^T$, that is:

$$\hat{\mathbf{y}} = \mathbf{Q}\,\mathbf{t} \tag{3}$$

Composing the two maps by substituting equation (2) into equation (3), for each input vector $\mathbf{x}$, the predicted responses are:

$$\hat{\mathbf{y}} = \mathbf{Q}\ \mathbf{W}^T\mathbf{x} = \mathbf{B}^T\mathbf{x} \tag{4}$$

From equation (4), the fitted PLS model is a linear mapping between the p-dimensional input space and the q-dimensional output space, defined by the matrix of estimated coefficients $\mathbf{B}^T$ ($q \times p$).

However, it is not convenient to use the coefficient matrix $\mathbf{B}$ directly because there are constraints that must be met, such as the projection of $\mathbf{x}$ must be a score $\mathbf{t}$ to which the regression model can be appropriately applied. To define the appropriateness of applying the PLS model, confidence levels are used to bound the Q and T^2statistics.

For each object, the value of the T^2 statistic is the Hotelling's distance from the projection to the origin of the latent space and indicates the position of the object within the projection space. The Q-residual, also called SPE (Squared Prediction Error), is the square of the Euclidean distance from the object to the orthogonal projection on the latent space. To impose a threshold on these two statistics is to use probabilistic criteria to limit the variation of the points inside and outside the latent space, defining the so-called *PLS-box* [14]. Thus, the *PLS-box* contains the "distant" points only up to the limit imposed by Q, whose projection is in the part of the variation space of the latent variables delimited by T^2.

Any valid (feasible setting of the process) $\mathbf{x}$ should belong to the intersection of the domain D and the *PLS-box* determined when fitting the PLS prediction model.

Note that the domain D is supposed to be a convex set, meaning that the segment connecting any two points of D must be inside D. An equivalent definition of a convex set is that every convex combination of elements of D belongs to D. Therefore, all predictor variables should be continuous.

A simple way to have a convex domain is to consider the hyper-parallelepiped defined by the individual range of each predictor variable in the dataset $\mathbf{X}$. In fact, this is how the domain is defined in all of the following case-studies.

Alternatively, the convex hull of the points in $\mathbf{X}$ would also define a convex domain, although this is more difficult to compute analytically. Nevertheless, the algorithmic computation for a given set of points is included in several commercial software packages (e.g., Matlab), and some code is also available on various sites, such as https://github.com/qhull/qhull. Additional information on its computation can be found in ref. [15].

The convexity hypothesis is only necessary on D because the definition in terms of Q and T^2 statistics implies that the *PLS-box* is also a convex set and, thus, its intersection with D is also a convex set as well.

2.2. Latent variable model inversion

When speaking about inversion, the interest shifts away from the input space to focus on the output. Thus, a target or desired value of the response(s) is set, and the goal of inversion is to determine the settings of the input variables to obtain the intended response.

In mathematical terms, this means inverting the prediction model to compute the exact values of the input variables that must be used to achieve the desired values. In general, no such inverse mapping exists, but a local (algebraic) inversion of the model can be addressed.

In the situation discussed in the present work, there is a single response, $q = 1$, so defining a target response means setting a value of y_d (a scalar). In view of equation (3), the inversion consists of finding a vector $\hat{\mathbf{t}}_d$ such that $\mathbf{Q}\,\hat{\mathbf{t}}_d = y_d$. With a single response, $\mathbf{Q}$ is in fact a row vector, and thus:

$$\mathbf{Q}\,\hat{\mathbf{t}}_d = (q_1, q_2, \ldots, q_a)\begin{pmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \ldots \\ \hat{t}_a \end{pmatrix} = q_1\hat{t}_1 + q_2\hat{t}_2 + \ldots + q_a\hat{t}_a = \sum_{i=1}^{a} q_i\hat{t}_i = y_d \tag{5}$$

Equation (5) is a linear equation with a single scalar solution $\hat{t}_d$ if $a = 1$ or with infinitely many solutions, that can be explicitly parameterized, if $a > 1$. In the last case, all the solutions of Equation (5) are in a hyperplane in the latent space, the one with $\mathbf{Q}^T$ as normal vector.

This property, clear from equation (5), is in fact consequence of the existence of a null space within the latent space [16]. The null space contains the points that are mapped into zero, so they do not change the predicted value, in other words, it is the space that relates the infinite solutions of equation (5). Precisely, if $\mathbf{t}_s$ denotes one of such solutions, the remaining ones can be written as

$$\hat{\mathbf{t}}_d = \mathbf{t}_s + \mathbf{t}_{\text{null}} \text{ where } \mathbf{t}_{\text{null}} \text{ is such that } \mathbf{Q}\mathbf{t}_{\text{null}} = 0 \in \mathbb{R} \tag{6}$$

To backpropagate these solutions to the input space and *finishing* the local inversion, it suffices to apply the loadings of $\mathbf{X}$ in equation (1) to obtain objects:

$$\hat{\mathbf{x}}^T = \hat{\mathbf{t}}_d^T\mathbf{P}^T \Leftrightarrow \hat{\mathbf{x}} = \mathbf{P}\hat{\mathbf{t}}_d \tag{7}$$

Eq. (1) is in fact an approximation for $\{\mathbf{X}, \mathbf{y}\}$ that works properly for the inversion in equations (5)–(7) when $\mathbf{T}$ is the orthonormal matrix of scores.

Except for the case $a = p$ (meaningless in practice), and even with a unique solution of equation (5), due to an empty null space, there are more solutions to the inversion than the one(s) in equation (7) [14].

They correspond to what can be called the residual space, spanned by the discarded latent variables or, in other words, the solutions due to the null space of the projection defined by the $\mathbf{W}^T$ weight matrix, that is, the set of vectors projecting onto the origin of the latent space. This null space is orthogonal to the null space associated with the latent space, so that the infinite solutions of the inversion all lie also in a hyperplane in the input space, that can be written as:

$$\hat{\mathbf{x}}_d = \mathbf{P}\hat{\mathbf{t}}_d + \mathbf{x}_{\text{null}} \text{ where } \mathbf{x}_{\text{null}} \text{ is such that } \mathbf{W}^T\mathbf{x}_{\text{null}} = \mathbf{0} \in \mathbb{R}^a \tag{8}$$

Therefore, the region resulting for the model inversion for the target y_d (as predicted with the PLS model) is the intersection of that hyperplane with the domain D and the *PLS-box*. More details with some illustrative examples can be found in ref. [14]. Also, an example in low dimension is simulated in section 3.1 of the present paper to illustrate these equations.

In any case, the region computed in this manner corresponds to the solutions of the algebraic equality, that is, imposing that the PLS-prediction $\hat{y}$ coincides with y_d, the so-called direct inversion [2].

2.3. Proof that only the endpoints are needed to invert a linear model for an interval

The focus of this paper is not on a single y_d, but rather on an interval $[y_a, y_b]$. In the cases studied here, the intervals in question are either specification limits for a product quality characteristic or tolerance intervals on predictions for class-modelling situations.

In ref. [2], an interval, a confidence interval on the prediction, has been already used to propagate the uncertainty in the prediction to the latent space. The inversion there is conducted by discretizing the confidence interval and solving the equations algebraically (direct

inversion). Then, the null space is subsequently added to each calculated score on the latent space.

Nevertheless, the linear nature of PLS precludes the necessity for discretization of the interval. It is sufficient to consider its endpoints, as will be demonstrated in the following paragraphs.

With the interval of output values $[y_a, y_b]$, let f denote the PLS model fitted with some training set $\{\mathbf{X}, \mathbf{y}\}$. The direct inversion described in the preceding paragraphs can be applied to both y_a and y_b to obtain parallel hyperplanes in the input space (the space where $\mathbf{X}$ is). These hyperplanes include the solutions of the corresponding inversion due to the two null spaces, that of the latent space and that of the residual space, which are orthogonal to each other. Moreover, the two hyperplanes are parallel because matrices $\mathbf{Q}$ and $\mathbf{W}$ (equations (6) and (8)) do not change, they only depend on the PLS model fitted f. From a practical standpoint, these estimations serve as boundaries of the sought region in the input space, which is also constrained by the domain D and the *PLS-box*. Since D and the *PLS-box* are convex sets, the mathematical intersection of them is also convex with at least two parallel linear faces.

To prove that only the endpoints of the interval need to be taken into account, let y_0 be a number in the interval $[y_a, y_b]$, and $\mathbf{x}_a$ and $\mathbf{x}_b$ two feasible settings in the input space (i.e., p-dimensional vectors belonging to the domain D and the *PLS-box*) which are some individual solutions of the inversion of f for y_a and y_b, respectively. This means that

$$f(\mathbf{x}_a) = y_a, f(\mathbf{x}_b) = y_b \tag{9}$$

As $y_0 \in [y_a, y_b]$, there is a unique $\lambda \in [0, 1]$, such that

$$y_0 = (1-\lambda)y_a + \lambda y_b \tag{10}$$

Let $\mathbf{x}_0$ be the same convex combination but in the input space, namely

$$\mathbf{x}_0 = (1-\lambda)\mathbf{x}_a + \lambda \mathbf{x}_b \tag{11}$$

which is another feasible setting in the segment that joints $\mathbf{x}_a$ and $\mathbf{x}_b$. This is so because there are two convex sets, one in each hyperplane. If $\mathbf{x}_a$ and $\mathbf{x}_b$ each belongs to one of them, then $\mathbf{x_0}$ also fulfills the constraints on the domain (it is in the hyper-parallelepiped) and on the two metrics T^2 and Q (it lies in the *PLS-box*).

PLS is a linear mapping, so the sequential substitution of equations (9)–(11), gives

$$f(\mathbf{x}_0) = (1-\lambda)f(\mathbf{x}_a) + \lambda f(\mathbf{x}_b) = (1-\lambda)y_a + \lambda y_b = y_0 \tag{12}$$

Therefore, $\mathbf{x}_0$ is a solution to the inversion of f for y_0.

Reciprocally, if y_0 is the image of a convex combination of two feasible settings (one in each hyperplane), it holds:

$$y_0 = f((1-\lambda)\mathbf{x}_a + \lambda \mathbf{x}_b) = (1-\lambda)f(\mathbf{x}_a) + \lambda f(\mathbf{x}_b) = (1-\lambda)y_a + \lambda y_b \tag{13}$$

And y_0 belongs to interval $[y_a, y_b]$.

Consequently, a vector $\mathbf{x}$ belongs to the region in the input space resulting from the model inversion for the interval if and only if it is a convex combination of any two solutions, each one corresponding to the model inversion for an endpoint of the interval.

2.4. Tolerance intervals

In the present work, γ-content tolerance intervals will be employed as a means of defining a different type of class-models.

In general, for a random variable X, interval $[l, u]$ is a two-sided γ-content tolerance interval [17] at 1-α confidence level that contains at least 100γ% of the population of interest (values of X) with 1 - α confidence.

Under normal distribution of X, the tolerance interval is defined as the range encompassing the mean value, plus or minus a multiple K of the standard deviation. The value of K depends on three parameters: sample size, confidence level (1–α), and content γ. The hypothesis of normal distribution renders the interval more precise than its non-parametric version.

2.5. Specification limits in process control

Quality characteristics are typically evaluated against specifications. In the case of a manufactured product, these specifications consist of desired measurements for the quality characteristics of the components and subassemblies that make up the product, as well as the desired values for the quality characteristics in the final product [12].

The desired value for a given quality characteristic is called the nominal or target value for that characteristic. Target values are typically bounded by a range of values that are believed not to affect the function or performance of the product when the quality characteristic at hand is within that range.

This range defines an interval whose endpoints are the upper and lower specifications limits. The Upper Specification Limit (USL) is the largest allowable value, while the Lower Specification Limit (LSL) is the smallest.

The range of specifications is precisely that of interest for the inversion, to gain more knowledge about how to control the input variables (most likely process variables in this context) to maintain specifications.

3. Results and discussion

Three cases are developed with data from the literature, two within the class-modelling field and one from process control. However, there is an additional introductory example, with data simulated in three dimensions, to illustrate the general procedure of direct inversion of a PLS model for an interval explained in Section 2.2.

3.1. PLS model inversion in a three-dimensional input space

To illustrate the inversion procedure with an interval and the resulting region in the input space, in the general context of LVMI, a data set with 40 points in a three-dimensional space is simulated, following a multivariate normal distribution with a high correlation between two of the variables, namely with a correlation coefficient of 0.85 between X_2 and X_3. The response to be predicted is simply a linear combination of the three predictor variables, 0.66 X_1 –0.41X_2 +0.06 X_3, with the addition of a noise with distribution N(0, 0.25).

Thus, $\mathbf{X}$ is 40 × 3 and $\mathbf{y}$ is 40 × 1, both autoscaled. The fitted PLS model has two latent variables, $a = 2$, which explain 94.91 % of the variance of $\mathbf{X}$, with 98.69 % of the variance in the response $\mathbf{y}$. The second latent variable mostly explains variance of $\mathbf{X}$, which is necessary to reconstruct it via inversion.

To carry out the inversion, the selected interval in the response is $[y_a, y_b] = [-1.07, 0.76]$, which was chosen to be relatively long to better illustrate the position of the different elements in the procedure, as shown in Fig. 1.

Fig. 1a) shows the input space, and Fig. 1c) depicts the interval-related boundaries of the region resulting from the inversion. The different steps can be observed in Fig. 1 in a clockwise direction, and will be explained in the following paragraphs.

In all cases, the black dots represent the samples in $\mathbf{X}$, which serve as a graphical reference for the domain D, which is defined as the rectangular cuboid that is limited by the range of the variables in the training set. The projection of the samples in $\mathbf{X}$ in the two-dimensional latent space (the scores) are in Fig. 1b), also in black dots.

Step by step, matrix $\mathbf{Q}$ is in fact a 1 × 2 vector with coordinates q_1 and q_2 and the inversion for y_a requires two scores t_1, t_2 such that

$$t_1 q_1 + t_2 q_2 = y_a \tag{14}$$

In this case, $6.18\ t_1 + 0.56\ t_2 = y_a \Leftrightarrow t_2 = \frac{y_a - 6.18\ t_1}{0.56} = \frac{y_a}{0.56} - 11.04\ t_1$, which is a straight line in the latent space, the one depicted in

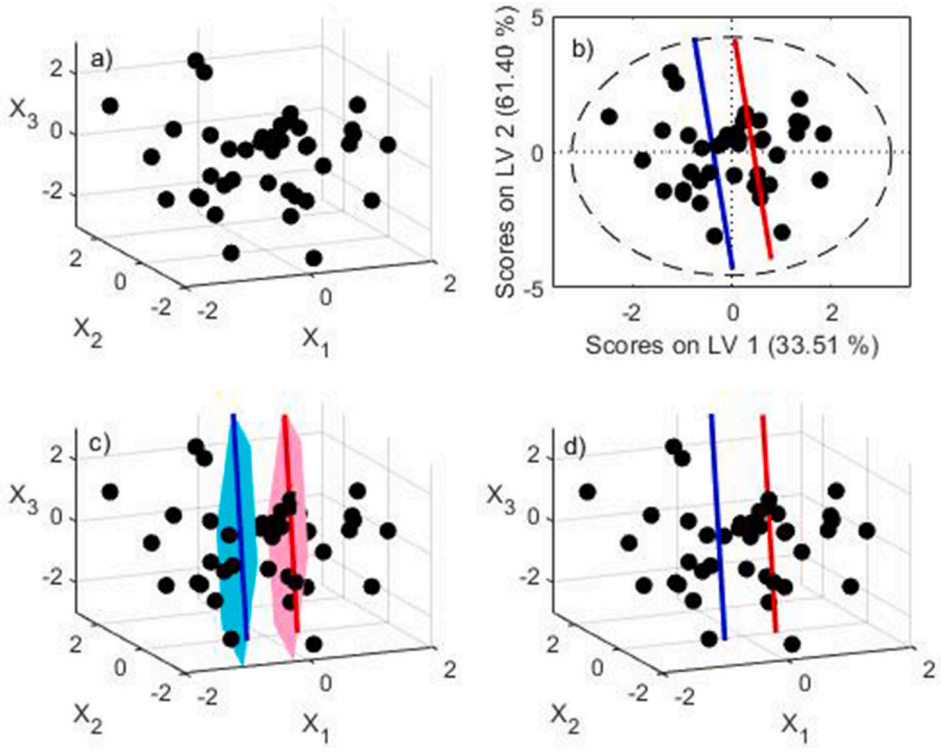


Fig. 1. Simulated study in three dimensions. The black dots represent the objects in **X**, both in a) the three-dimensional input space and b) the two-dimensional latent space (scores). The straight lines in b) and d) are the solutions of the inversion for the endpoints of the interval but constrained to be inside the ellipse in b), whose boundary at 99 % confidence for T^2 is the dashed line. They are backpropagated to the input space in d) with the same color code. Finally, the light-blue and light-red planes in c), containing the blue and red segments, respectively, represent the entire solution of the inversion for the endpoints, because it incorporates both the effect of the projection-related null space and the constraints of belonging to the domain and to the *PLS-box*. The region resulting from the direct inversion for the entire interval is the convex set 'between' the planes, which is obtained by a convex combination of any two elements, one in the blue plane and the other in the red plane.

Fig. 1b) in blue. In fact, only the segment within the limits imposed with the T^2 statistics are acceptable solutions, graphically those inside the ellipse in dashed line (built at 99 % confidence level).

Similarly, the solutions of equation (14) with y_b instead of y_a follow a line with the same slope (determined by **Q**, −11.04 in this case), which is also depicted in Fig. 1b) in red.

Subsequently, by multiplying by **P**, equation (7), the corresponding three-dimensional points in the input space are all aligned. In the present example, the solutions of equation (14) for y_a follow the straight line with parametric equation given by eq. (15).

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4.47 \\ 9.87 \\ 10.56 \end{pmatrix} y_a - \begin{pmatrix} 0.23 \\ 0.67 \\ 0.71 \end{pmatrix} t_1 \tag{15}$$

with parameter t_1 and unitary line's director vector $\mathbf{v} = (-0.23, -0.67 - 0.71)^T$ that, again, only depends on **Q** and **P**. The parallel lines corresponding to those in Fig. 1b) are depicted in Fig. 1d) in the input space, with the same color, i.e., blue for y_a and red for y_b.

Since the entire interval of responses is valid, and PLS is a linear model, all the points in the domain that lie 'between' the aforementioned lines within the π-plane that contains them are inside the *PLS-box* and have their predictions in [y_a, y_b]. The plane π, which contains the red and blue segments, is not shown in Fig. 1d).

However, there are some more solutions to the inversion that originate from the null space of the projection, which is a one-dimensional space, a line, orthogonal to that in equation (15). These two orthogonal directions define a two-dimensional space (a plane) in the input space. In the particular case of y_a, the parametric equation of the plane is:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4.47 \\ 9.87 \\ 10.56 \end{pmatrix} y_a - \begin{pmatrix} 0.23 \\ 0.67 \\ 0.71 \end{pmatrix} t_1 + \begin{pmatrix} 0.16 \\ 0.83 \\ -0.53 \end{pmatrix} \theta \text{ where } t_1, \theta \in \mathbb{R} \tag{16}$$

This is the light-blue plane in Fig. 1c), limiting the values of t_1 and θ to those corresponding to the constraints to be inside the domain *D* and the *PLSbox*. Therefore, this plane is orthogonal to π and contains the blue segment. The light-red plane, parallel to the blue, is the equivalent for y_b. It contains the red line (corresponding to y_b) and is also orthogonal to π.

Consequently, the region resulting from the inversion that contains the settings of the input variables whose predictions will be in [y_a, y_b] is the one between these planes (blue and red) with points that must also belong to the domain *D* and the *PLS-box*, constraints already applied in Fig. 1c).

Therefore, the inversion for the endpoints of the interval provides some boundaries of the convex set of interest. Particular solutions inside this set can be obtained as convex combinations of any two solutions, one in "the blue plane" and the other in the "red plane".

To avoid "tuning" the parameters t_1 and λ, the cross product of the two director vectors gives the normal vector to the plane, whose general equation for y_a is thus:

$$x_1 - 0.25x_2 - 0.09x_3 - 1.03y_a = 0 \tag{17}$$

Therefore, the needed convex combinations are in fact vectors (points) with three coordinates x_1, x_2, and x_3 whose first coordinate is precisely:

$$x_1 = 1.03[\lambda y_a + (1-\lambda) y_b] + 0.25x_2 + 0.09x_3 \tag{18}$$

Varying x_2 and x_3 in the domain D and λ in the interval [0, 1], new points are computed in the input space with predicted values in $[y_a, y_b]$. In higher dimensions, the expressions are perhaps less easily manipulable but the structure of hyperplanes in the boundaries and convex combinations of two points (one in each corresponding hyperplane) is the same.

3.2. Detection of tumors

The first case-study is related to medical diagnosis. A data set comprising 569 tumor tissue samples was obtained from the UCI Machine Learning repository [18]. Of the 569 samples, 357 were identified as benign tumor tissues and 212 as malignant ones. The input variables are obtained by calculating measurements from a digitized fine needle aspiration (FNA) image of a breast sample. Ten measurements in real numbers are obtained for each cell nucleus. Additionally, for each image, the mean, the standard error, and the "worst" or largest (mean of the three largest values) of each case are also found, resulting in a total of 30 measured features. Accordingly, **X** is 569 × 30.

In this case study, the focus is on malignant tumor tissues, which are coded as 1. The matrix of binary codes **y** is completed by coding benign tissues as −1. A PLS model is then fitted with autoscaled **X** and **y**.

In constructing the model, 99 % confidence level was used to bound the Q and T^2 statistics, with absolute values of the standardized residuals limited to 3. The number of latent variables was selected through cross-validation, which entails extracting random subsets, conducting 10 data splits, and performing 5 iterations. The selected model includes some latent variables that do not appear to explain much of the variance in **y**. However, they do explain percentages of the variance in **X**, which is important, particularly because the model should be inverted.

The model with the first six latent variables had 17 samples with values of Q and T^2 statistics or regression residuals that far exceeded the imposed limits. Consequently, these objects were excluded from the training set.

The remaining 552 samples were used to rebuild the PLS model, again with six latent variables that explained 85.11 % of the variance in **X** and 76.32 % of the variance in **y**. Table S1 in the supplementary material contains the details about the behavior of the explained and cumulative variance in both **X** and **y**, as well as the RMSECV (Root Mean Squared Error in Cross Validation) when adding the first latent variables. The coefficient of determination in cross-validation is $R^2_{CV} = 0.747$.

Various goodness-of-fit tests are used to verify that the responses predicted with the PLS model with six latent variables in the class of malignant tumor tissues are adequately fitted by a normal probability distribution with a mean of 0.99 and a standard deviation of 0.5498.

Subsequently, a tolerance interval is established to encompass 99 % of the predictions for malignant tumor tissues, the population from which the relevant samples originate, with 95 % confidence. The interval of predicted values, calculated using the fitted normal distribution, is $[p_a, p_b] = [-0.5576, 2.5378]$. This implies that at 95 % confidence level, 99 % of the response population from which the sample of malignant tumors originates, is within this interval. The tolerance interval is centered on the mean and has been calculated under the assumption of a normal distribution of the data.

The PLS model is inverted for the endpoints of the tolerance interval under the set of constraints previously mentioned. This entails identifying the feasible solutions, which are points (vectors of dimension 30 corresponding to the input variables) that belong to the domain and to the *PLS-box*.

A total of one hundred solutions of the corresponding eq. (5) were computed so that multiplying by the loadings of **X** (eq. (7)) resulted in 100 points in the input space. The addition of another hundred points from the null space of the projection resulted in the generation of 200 solutions. All of the solutions are on a hyperplane, and they all have a PLS prediction value of $p_a = -0.5576$. The solutions were subsequently organized into a matrix $\widehat{\mathbf{X}}_a$.

The equation of the hyperplane can be obtained by regressing one of the columns of $\widehat{\mathbf{X}}_a$ on the remaining ones. For instance, if the first column is regressed on the remaining 29, the regression coefficients **b** (substituting the intercept, −7.75 in this case, by −1 in the first position) define the normal vector of the hyperplane. This regression is, in fact, an interpolation that gives the explicit expression of the first coordinate (first variable) x_1 as a function of x_2, …., x_{30}, precisely $x_1 = -7.75 + \sum_{i=2}^{30} b_i x_i$.

A similar procedure with $p_b = 2.5378$ gives another 200 solutions in the corresponding parallel hyperplane, with the same normal vector but different intercept. For this endpoint, the intercept is 35.28, instead of the −7.75 obtained for the lower endpoint.

Although the input space cannot be depicted in a cartesian diagram, the situation is analogous to that depicted in Fig. 1. Due to the linearity of the model, the objects that are obtained as convex combinations of points in the two hyperplanes corresponding to the inversion for the endpoints of the interval are the part of the input space that correspond to the characteristics of the tumor tissues that will be predicted as malignant. That is to say, the predicted values will be inside the tolerance interval.

These points $(x_1, x_2, \ldots, x_{30})$ all have as first coordinate $x_1 = -7.75\lambda + 35.28(1-\lambda) + \sum_{i=2}^{30} b_i x_i$.

The characteristics of the input variables in this region will provide insight into the commonalities shared by the majority of the samples to be predicted as malignant tumor tissues.

It is important to remember that the structure of the covariance in the space of the 30 predictor variables is critical. Precisely, PLS models this structure and employs the limits of *Q-residual* and T^2 to constrain the tolerable deviations of the projections in the space of the latent variables. Consequently, it is insufficient to merely ascertain that a new vector of 30 values has each coordinate within the admissible range for each of them (the defined domain D). Rather, it is essential to check that it also belongs to the *PLS-box*.

As previously stated, the hyperplanes and the points defined by the 30 values of the predictors are hard to visualize. To illustrate the subsequent analysis, Fig. 2 provides a schematic representation of the elements.

In Fig. 2, the planes π_1 and π_2 'represent' the result of inverting the endpoints of the tolerance interval (those in Fig. 1c when the space is three-dimensional). Two feasible points, $\mathbf{w}=(w_1, w_2)$ and $\mathbf{v}=(v_1, v_2)$, are selected, one in π_1 and the other in π_2. These points share one of their two coordinates, specifically $v_1 = w_1$. Any point **u** in the segment joining

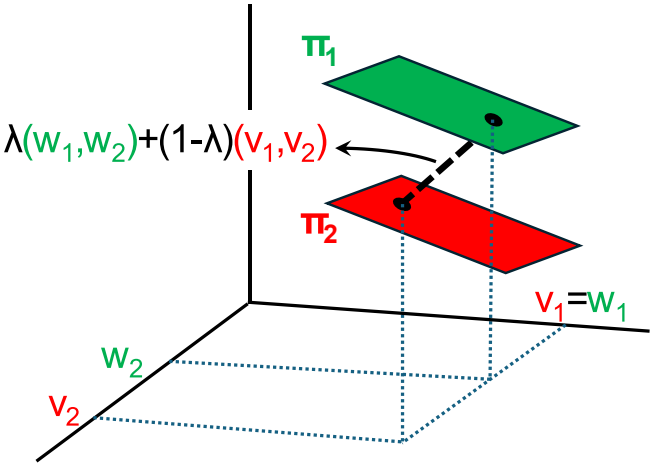


Fig. 2. Illustration of convex combinations with points in the hyperplanes computed by inverting the model for the endpoints of the interval.

v and **w** (black dashed line in Fig. 2) is necessarily a convex combination of **v** and **w**. This implies the existence of a real number λ, $0 \leq \lambda \leq 1$, such that $\mathbf{u} = \lambda\mathbf{w}+(1\text{-}\lambda)\mathbf{v}$.

Using the same colors as in Fig. 2, Fig. 3a) depicts the parallel coordinates plot for a hundred feasible solutions in the hyperplane, analogous to the π_1 plane in Fig. 2, obtained by inverting the model for the upper endpoint of the tolerance interval, $p_b = 2.5378$. Fig. 3b) is the parallel coordinates plot of another hundred feasible points in the hyperplane, related to the π_2 plane, obtained for the lower endpoint, $p_a = -0.5576$.

A parallel coordinates plot is a tool for visualizing high-dimensional data, where each observation is represented by the sequence of its coordinate values plotted against their coordinate indices. The graph is depicted with the standardized data, as otherwise the different magnitudes would prevent the visualization of any meaningful patterns.

In terms of the predictor variables, the tolerance interval calculated for the population of diseased individuals is explored, like in the illustration in Fig. 2, by selecting a point in each hyperplane, **w** and **v,** with specific characteristics that are of interest to the researcher.

For example, solutions with the same value in the 15th predictor variable are selected and marked by black broken lines in Fig. 3a) and b), respectively, for **w** and **v**. Both solutions are plotted together in Fig. 3c), this time **w** is in green and **v** in red, along with the 552 samples (in gray) of **X**, for reference.

Three different points with coordinates in the range defined by **v** and **w** are selected and superimposed in Fig. 3c) to create Fig. 4. In Fig. 4a), the coordinates of point $\mathbf{u_1}$ (depicted in black) were randomly generated using a uniform distribution within the range defined by the minimum and maximum of the corresponding coordinate in **w** and **v**. Consequently, $\mathbf{u_1}$ belongs to the *D* domain but not to the *PLS box*. In fact, $T^2(\mathbf{u_1})/T^2_{crit} = 1.91$ and $Q(\mathbf{u_1})/Q_{crit} = 2.76$, where T^2_{crit} and Q_{crit} are the bounds of the corresponding statistics at 99 % confidence level. Therefore, $\mathbf{u_1}$ is not a feasible setting and the PLS model is therefore unsuitable for application to it. In other words, this point $\mathbf{u_1}$ does not share the internal covariance structure that was modeled by PLS.

For Fig. 4b), the point $\mathbf{u}_2$, again in black, represents a feasible solution because it belongs to both the *D* domain and the *PLS-box*. In particular, its coordinates fall within the range defined by those of **w** and **v** (green and red). Nevertheless, the PLS prediction for $\mathbf{u}_2$ is -0.6024, which is outside the tolerance interval. The reason for this apparent discrepancy is that $\mathbf{u}_2$ belongs to the convex set defined in the input space but it cannot be written as a convex combination of any pair of points, one in each of the hyperplanes. Consequently, its PLS prediction is not in the tolerance interval.

Finally, the point $\mathbf{u}_3$ represented in black in Fig. 4c) was obtained by taking a value of the tolerance interval $[p_a, p_b] = [-0.5576, 2.5378]$ (in this case 1.99) and determining the weights of the corresponding convex combination with the endpoints of the tolerance interval. That is to say, find the value λ such that $1.99 = \lambda p_a + (1\text{-}\lambda)\, p_b$, which is $\lambda = 0.18$. Then, the same convex combination gives point $\mathbf{u}_3 = 0.18\mathbf{v} + 0.82\mathbf{w}$. Consequently, it is a feasible solution and its prediction is inside the tolerance interval, and thus the individual will be declared sick.

The analysis of figures analogous to Figs. 3 and 4, together with the constraints of interest on points **w** and **v** (sharing the value of variable number 15 in this example), permits the exploration of the behavior of the predictor variables that define the region in which 99 % of the population having a malignant tumor, will be found with 95 % confidence.

3.3. Rice typification

Also from the UCI Machine Learning repository, a dataset for classification of certified rice varieties grown in Turkey has 3810 samples, each of which characterized by seven morphological features that were obtained for each grain of rice from images of the grains. These characteristics are: 1, Area (the number of pixels within the boundaries of the rice grain); 2, Perimeter (distance between pixels around the boundaries of the rice grain); 3, 4, Major and Minor Axis Lengths (the longest and

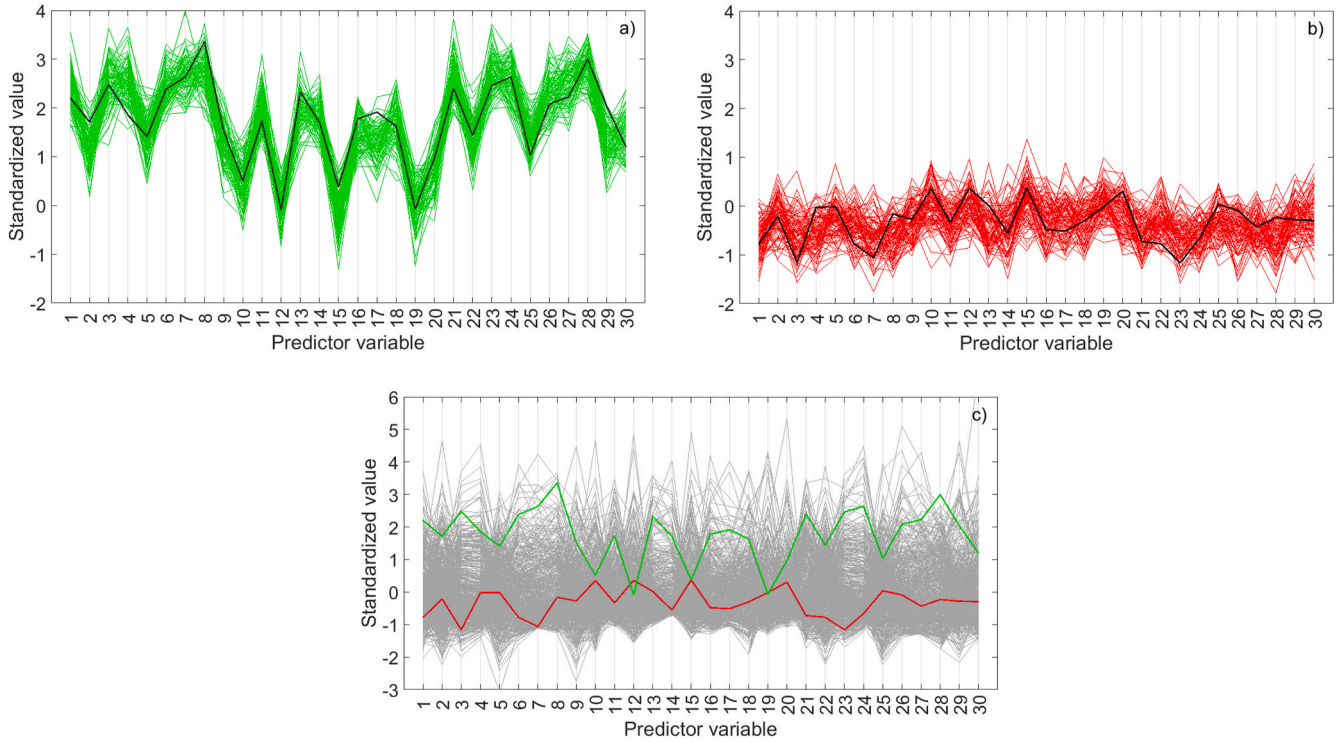


Fig. 3. Parallel coordinates plot of different cases. a) Solutions when inverting the upper endpoint of the tolerance interval, with one of them highlighted. b) Solutions when inverting the lower endpoint of the tolerance interval, with one of them highlighted. c) Samples in **X**, along with the ones selected in a) and b) with the same color code.

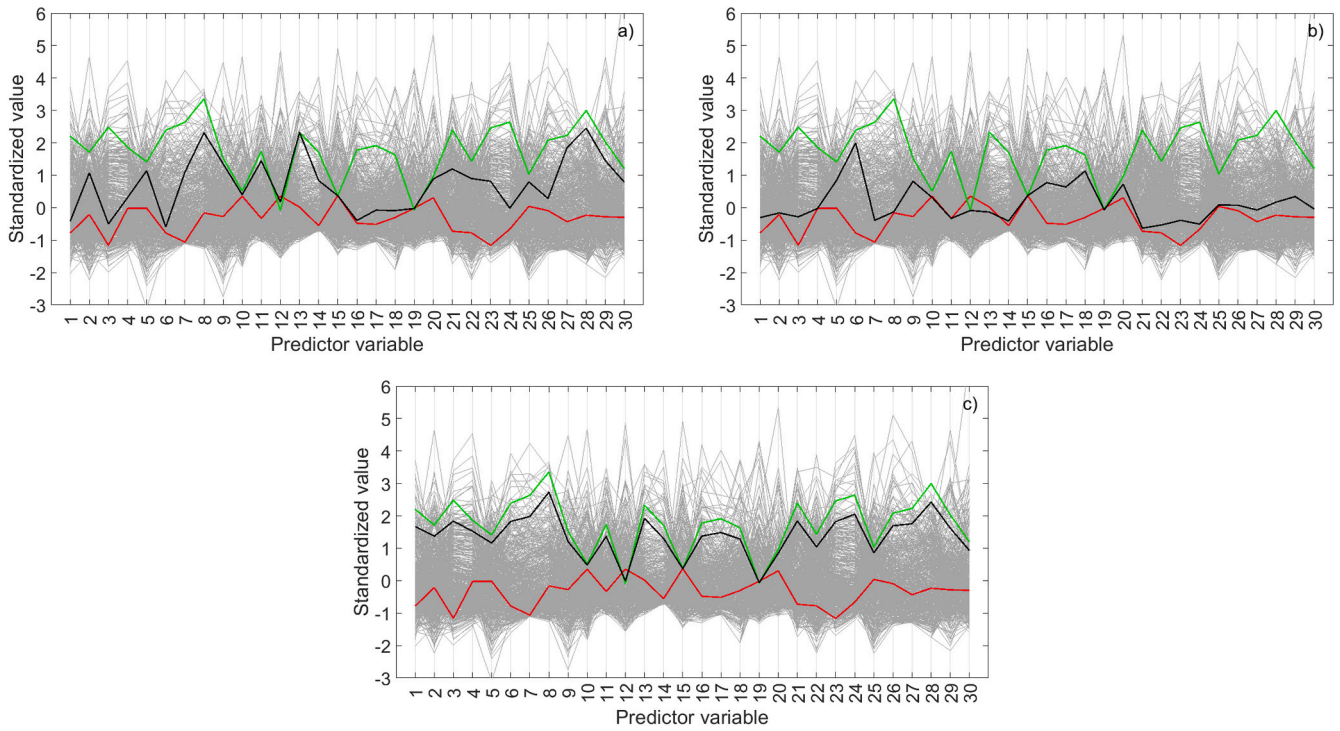


Fig. 4. Parallel coordinates plots, highlighting three solutions, the red and green ones in Fig. 3, and a black one corresponding to a solution: a) outside the *PLS-box*, b) with prediction outside the tolerance interval, c) which is a convex combination of the red and green solutions and thus inside the *PLS-box* and with prediction in the tolerance interval.

shortest lines that can be drawn on the rice grain); 5, Eccentricity (how round the ellipse, which has the same moments as the rice grain, is); 6, Convex_Area (pixel count of the smallest convex shell of the region formed by the rice grain); and 7, Extent (ratio of the region formed by the rice grain to the bounding box). All the details can be found in ref. [19].

There are 2180 samples of the Osmancik species, which will be coded as −1, and 1630 samples of the Cammeo species which will be coded as 1.

Again by cross-validation, two latent variables were selected for the PLS prediction model. The decomposition of variances is presented in Table S2 of the supplementary material. The two latent variables explain 86.09 % of the variance in **X** and 68.38 % in **y**, with $\hat{y}$ varying between −2.46 and 2.26.

The predictions of the PLS model for the Osmancik rice samples are well fitted with a N(-0.59, 0.456). Similarly, the predictions of the Cammeo rice samples are also well fitted with a normal distribution, with a mean of 0.79 and a standard deviation of 0.476.

By modifying the threshold decision value, the fitted distributions can be used to compute sensitivity and specificity of the corresponding class-model. Fig. 5 shows the joint behavior of sensitivity and specificity that can be achieved for the Cammeo variety, which appears to be the same as that computed in ref. [19]. In that study, the authors reported sensitivities, obtained with different classification methods, up to 92.26 % with a corresponding specificity of 93.58 %. Similar performances are highlighted in Fig. 5, where also the typical opposite behavior between sensitivity and specificity is evident.

A different approach, based on the predictions obtained with PLS, is to compute individual tolerance intervals, at 95 % confidence level, containing 93 % of the predictions of Osmancik and also 93 % of the predictions of Cammeo rice. This probability is based on the values of sensitivity and specificity, which are used as a reference when seeking a balanced situation between the two rice species.

The interval computed for Osmancik predictions is [−1.44, 0.26], while that for Cammeo is [−0.099, 1.68]. The intersection of the two

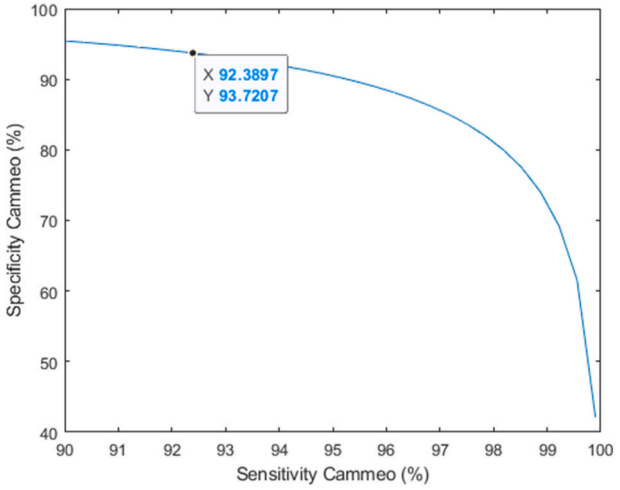


Fig. 5. Sensitivity and specificity of the different class-models for the Cammeo rice species.

intervals encompasses the predictions within [−0.099, 0.26], with probability approximately 0.1 with both distributions (0.109 and 0.100 for the distributions of the predictions for the Osmancik and Cammeo varieties, respectively).

The inversion of the model for the two tolerance intervals yielded four hyperplanes (one for each endpoint of the corresponding interval). A thousand convex combinations of points in the corresponding hyperplanes resulted in 477 feasible points in the input space (that it, in the intersection of the *PLS-box* and the domain D) for Osmancik species and 529 feasible points for Cammeo species. Of these, 142 are located in the common region. To see their position relative to the initial samples in **X**, i.e., visualize the solutions computed in relation to the objects of the two classes in the input space, a Principal Component Analysis (PCA) is

applied, because with seven input variables, the conventional Cartesian representation is not a viable option.

The PCA decomposition is calculated with the predictors in **X**, and then the computed solutions of the inversion of the PLS model for the tolerance intervals are projected onto the PCA model.

Fig. 6 shows the scores on the first two principal components (PCs) of the Osmancik and Cammeo samples, in green and red, respectively. The two PCs explain 86.84 % of the variance of autoscaled **X**. Then, the scores of the solutions obtained by inverting the PLS model are depicted in blue: light blue circles correspond to the Osmancik species and dark blue empty squares represent the Cammeo rice.

Despite the fact that the PCA model is constructed without taking into account the class to which the object belongs, and that the inversion is performed separately for each tolerance interval, the *degree* of confusion between the categories seems to be the same. This is not surprising because the confusion was already seen in the tolerance intervals.

However, there are clear differences in the position of the points, Fig. 6 shows higher scores on the first PC for the Cammeo variety, and also for the blue computed points which are projected around the real samples of the corresponding variety in red.

The loadings on the first PC, namely 0.46, 0.46, 0.45, 0.32, 0.23, 0.46, and −0.058 indicate larger values of the variables for the Cammeo variety, with a different behavior of the variables 4, 5 and especially 7.

In any case, there is a certain degree of confusion between rice varieties when they are characterized by their images, and the proposed procedure can be used to remove this intersection, leaving only the part of the input space that is uniquely related to each variety. Fig. 7a) shows the location of the scores of the computed solutions after removing those at the intersection of the tolerance intervals. Looking at the same points in the input space in Fig. 7b), in the form of a parallel coordinates plot, the separation is not so clear, especially in variable number 7.

Numerically, the regions inside the input space that clearly correspond to the characteristics of each rice variety can be described by using the solutions that are “extreme” in at least one of the variables, that is, that have a coordinate with the maximum or minimum of each variable per variety. These are the solutions in Table 1, and they constitute a kind of edges of the region of interest since they all belong to the boundary. Furthermore, any convex combination of the points in Table 1 is inside the corresponding computed region.

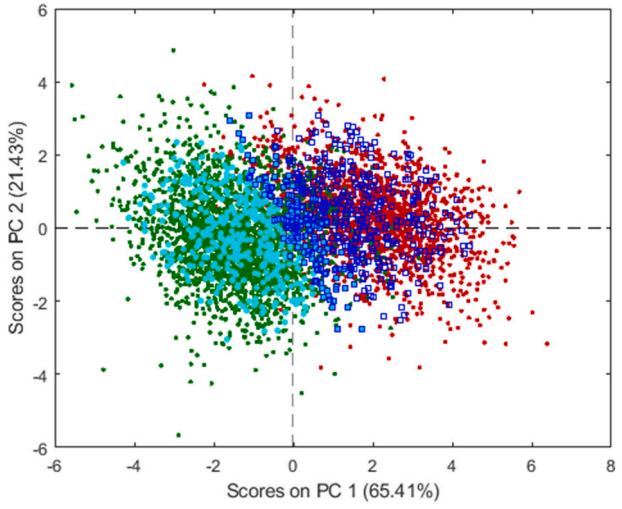


Fig. 6. Scores of PCA on the first and the second PCs. Green points are for Osmancik rice, and red points are for Cammeo rice. The scores of the solutions from the inversion of the PLS model for the tolerance intervals are shown in blue, light blue filled circles for Osmancik rice, and dark blue empty squares for Cammeo rice.

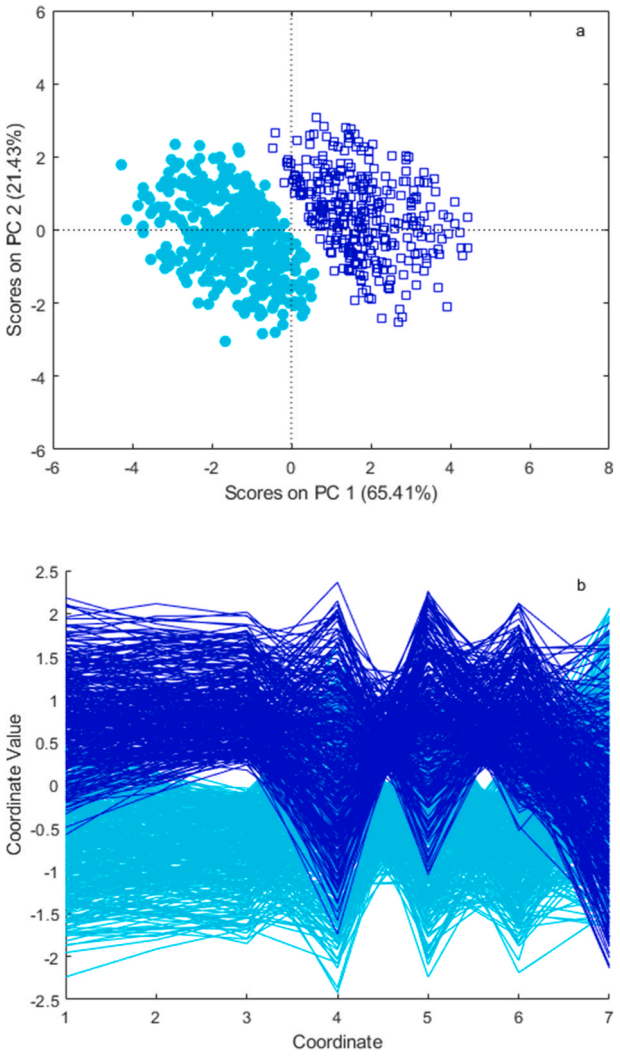


Fig. 7. Rice varieties. a) Scores on the first PC plane and b) parallel coordinates plot of the points in the input space.

3.4. Process control

Montgomery [12] reports a case study with 40 observations from a cascade process, where there are nine input variables and two output variables, which he says is typical of applications in chemical and process plants. For the sake of illustration, some specification limits will be imposed on one of the responses, y_1. Assuming that the data contain solutions outside specifications, the interval defining the lower and upper specification limits [LSL,USL] is shorter than the extent of the process shown in the data, [951, 954] when the bounds of the response values are 948.9 and 956.5.

X is then a 40 × 9 matrix of autoscaled predictors, that is, values of process variables, material properties, environmental conditions, and so on. The response **y** is a 40 × 1 vector of autoscaled values, that is, some characteristic of the objects produced with the values of the nine predictors.

With autoscaled predictors and response, five latent variables are selected by cross-validation. These latent variables explain 87.2 % of the variance in **X** with 80.3 % of the variance in **y**, with a coefficient of determination of 0.80 (0.70 in cross-validation). Details of each individual latent variable are provided in Table S3 in the supplementary material.

One hundred solutions at each endpoint were computed by inverting the model for the two specification limits, those depicted in Fig. 8, in the

Table 1
Some solutions at the boundary of the computed regions.

Area	Perimeter	Major Axis Length	Minor Axis Length	Eccentricity	Convex Area	Extent
Osmancik variety						
8791.8	386.14	159.51	74.429	0.88596	9065.6	0.54803
9870.7	414.76	178.05	72.440	0.90675	10108	0.76688
10039	396.82	156.53	81.937	0.86389	10275	0.73783
10047	401.52	164.73	79.179	0.87832	10306	0.53452
10705	431.84	184.32	74.656	0.91265	11067	0.70358
10998	444.67	189.49	75.302	0.91837	11367	0.50955
12019	419.61	167.68	91.637	0.84024	12284	0.82123
12259	454.98	188.11	84.674	0.88458	12627	0.69707
13336	461.92	187.55	92.003	0.86452	13663	0.70857
13358	442.78	176.66	95.876	0.8534	13661	0.61145
13460	458.54	186.93	92.299	0.8666	13786	0.748
13508	454.93	186.71	92.268	0.86711	13685	0.63703
Cammeo variety						
11662	455.94	193.56	77.809	0.91828	12100	0.54042
11814	450.78	196.98	76.350	0.92740	12005	0.71836
12310	464.72	191.81	82.667	0.90508	12728	0.57377
13151	483.36	209.06	80.706	0.92364	13487	0.49710
15031	485.71	199.25	97.844	0.86505	15319	0.74830
12965	478.19	208.22	79.095	0.93392	13274	0.67497
14840	497.40	206.24	94.115	0.88436	15217	0.80727
15896	522.25	223.94	92.398	0.91439	16242	0.69776
16059	529.47	223.13	93.857	0.90997	16550	0.59719
16327	511.69	210.76	99.848	0.88362	16712	0.75905
16450	521.11	218.65	97.597	0.89723	16723	0.58775

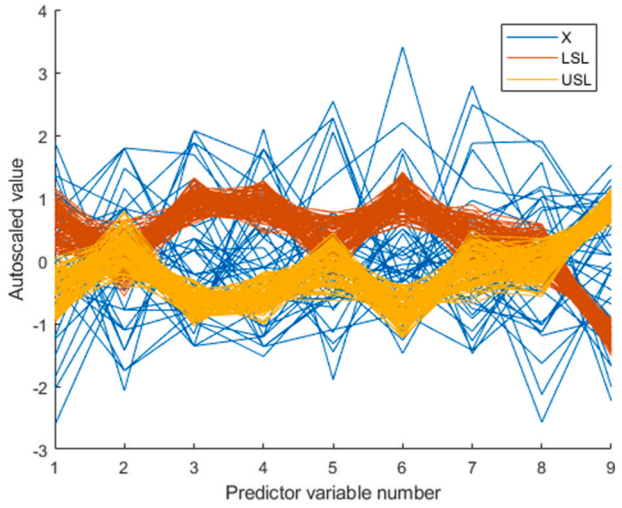


Fig. 8. Parallel coordinates plot of the predictors in blue, and the solutions found by inversion, in red for the Lower Specification Limit (USL) and in orange for the Upper Specification Limit (USL).

form of a parallel coordinates plot in the nine-dimensional input space. As a reference of the variation of the predictor variables, the samples in **X** are also plotted as blue lines in Fig. 8.

A number of procedures have been proposed for the selection of variables in PLS. With the so-called VIP (Variance Importance in Prediction), variables number 2, 5, and 7 are removed because they do not contain relevant information for the prediction model. Interestingly, these variables in Fig. 8 have values of the obtained solutions that are similar in both specification limits. In other words, the selected variables are those that show differential behavior in the solutions obtained for inversion of each specification limit (except for variable number 8), although the model is fitted to predict the whole range of **y** values.

A new PLS model was fitted with the selected variables, with four latent variables that explain 86.67 % of the variance in **X** and 79.78 % in **y**. $R^2 = 0.80, R^2_{CV} = 0.71$, and a single object showed a slight excess of the Q statistic at the 95 % confidence level.

This new model is inverted and a hundred solutions are generated for each specification limit. The resulting solutions are depicted in Fig. 9, where it can be seen that they are analogous to those in Fig. 8. However, in this case, the differences between the corresponding predictor values in each hyperplane are evident. The corresponding values in the LSL are greater than those in the USL for the first four predictor variables. Variable number 8 appears to act as a kind of inflection point, where the observed behavior of greater values for the LSL than for the USL in the first variables changes and is exactly the opposite in variable number 9.

Table 2 shows the mean values of the hundred solutions computed for each specification limit as representative of all of them. They are written in their original scale, rather than the autoscaled values in Figs. 8 and 9, and are rounded to the number of significant digits in the original data set [12]. Note that the linearity of PLS also makes the vector of means a feasible solution.

The usefulness of the values in Table 2 is that any feasible solution with predictions inside the interval defined by the specification limits for y_1 can be obtained as a convex combination of, for example, the two solutions in Table 2.

However, the values whose means are shown in Table 2 do not necessarily define a "hypercube", because further to maintain the predictors within the corresponding bounds, the structure among them must also be maintained. For example, the first column in Table 2 shows that X_1 varies around [14.28, 16.53], let's assume that in a particular situation $x_1 = 15$, then there is a unique coefficient λ ($\lambda = 0.32$ in the example) such that $15 = 16.53\,\lambda + (1-\lambda)14.28$. To keep the product within the specification limits, the rest of the variables must have the same structure, i.e., the same convex combination, namely $x_3 = 87$, $x_4 = 49$, $x_6 = 7.23$, $x_8 = 6.09$, $x_9 = 1.106$. In this way, some deviations in one of the variables can be "compensated" by variations in the others to keep the product within specifications.

Much more possibilities can be obtained with convex combinations of the individual solutions computed. These solutions can be found in the file *solutions_sm.xlsx* in the supplementary material.

In any case, if the nine variables are used to control the process, with the model with all nine variables, the equivalent of Table 2 is Table 3. Note the similarity of the means for predicting each specification limit in the common predictors.

Variable X_2 has the same mean in the solutions corresponding to the different specification limits. However, individual possible values can be seen in Fig. 8.

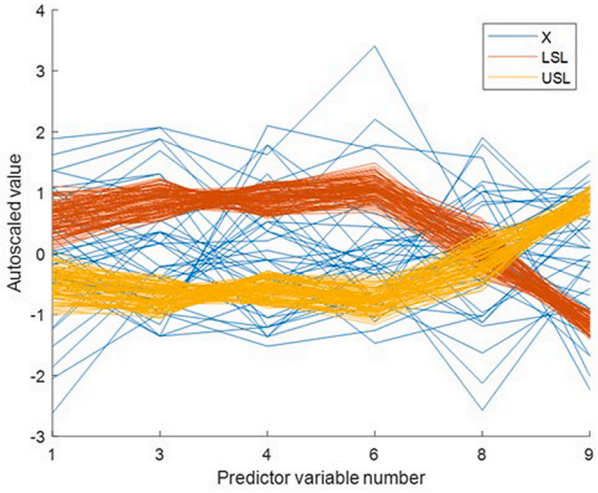


Fig. 9. Parallel coordinates plot of the reduced set of predictors in blue, and the solutions found by inversion, in red for the Lower Specification Limit (USL) and in orange for the Upper Specification Limit (USL).

Table 2
Mean of the predictor variables to obtain predictions at the specification limits.

	X_1	X_3	X_4	X_6	X_8	X_9
LSL	16.53	93	56	7.41	6.11	1.094
USL	14.28	84	46	7.15	6.07	1.112

Table 3
Mean of the predictor variables to obtain predictions at the specification limits.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
LSL	16.66	0.11	93	56	1.65	7.40	1.86	6.12	1.093
USL	14.35	0.11	84	46	1.58	7.15	1.78	6.08	1.112

4. Conclusions

The inversion of latent variable models for an interval of response values provides an approach to defining the design space for a range of acceptable values.

The work shows a general methodology for constructing a set inside the domain (i.e., the region in which the input predictor variables are allowed to vary) and a way of moving within it to select settings of the predictors with a property of interest defined by an interval.

The generalization for uncorrelated response characteristics is straightforward, as well as for correlated quality characteristics (in the responses space), provided that their requirements define a parallelepiped in the output space.

CRediT authorship contribution statement

M.S. Sánchez: Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Data curation, Conceptualization. **M.C. Ortiz:** Writing – review & editing, Funding acquisition, Formal analysis, Conceptualization. **S. Ruiz:** Supervision, Investigation, Formal analysis, Conceptualization. **O. Valencia:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **L.A. Sarabia:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization.

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

Some data are simulated and others are taken from public archives and literature, with appropriate references.

Acknowledgements

Authors acknowledge the funding of Consejería de Educación de la Junta de Castilla y León under project BU052P20, co-financed with European Regional Development Funds.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2024.105166.

References

[1] M. Ottavian, E. Tomba, M. Barolo, Advanced process decision making using multivariate latent variable methods, in: M. Ierapetritou, R. Ramachandran (Eds.), Process Simulation and Data Modeling in Solid Oral Drug Development and Manufacture. Methods in Pharmacology and Toxicology, Humana Press, New York, NY, 2016, https://doi.org/10.1007/978-1-4939-2996-2_6.
[2] P. Facco, F. Dal Pastro, N. Meneghetti, F. Bezzo, M. Barolo, Bracketing the design space within the knowledge space in Pharmaceutical product Development, Ind. Eng. Chem. Res. 54 (2015) 5128–5138, https://doi.org/10.1021/acs.iecr.5b00863.
[3] M. Forina, P. Oliveri, S. Lanteri, M. Casale, Class-modeling techniques, classic and new, for old and new problems, Chemometr. Intell. Lab. Syst. 93 (2008) 132–148, https://doi.org/10.1016/j.chemolab.2008.05.003.
[4] R.G. Brereton, One-class classifiers, J. Chemom. 25 (2011) 225–246, https://doi.org/10.1002/cem.1397.
[5] O. Rodionova, P. Oliveri, A. Pomerantsev, Rigorous and compliant approaches to one-class classification, Chemometr. Intell. Lab. Syst. 159 (2016) 89–96, https://doi.org/10.1016/j.chemolab.2016.10.002.
[6] O. Valencia, M.C. Ortiz, S. Ruiz, M.S. Sánchez, L.A. Sarabia, Simultaneous class-modelling in chemometrics: a generalization of Partial Least Squares class modelling for more than two classes by using error correcting output code matrices, Chemometr. Intell. Lab. Syst. 227 (2022) 104614, https://doi.org/10.1016/j.chemolab.2022.104614.
[7] O. Valencia, M.C. Ortiz, M.S. Sánchez, L.A. Sarabia, A modified entropy-based performance criterion for class-modelling with multiple classes, Chemometr. Intell. Lab. Syst. 217 (2021) 104423, https://doi.org/10.1016/j.chemolab.2021.104423.
[8] D. Castro-Reigía, M.C. Ortiz, S. Sanllorente, I. García, L.A. Sarabia, PLS class modelling using error correcting output code matrices, entropy and NIR spectroscopy to detect deficiencies in pastry doughs, Chemometr. Intell. Lab. Syst. 246 (2024) 105092, https://doi.org/10.1016/j.chemolab.2024.105092.
[9] M. Barker, W. Rayens, Partial least squares for discrimination, J. Chemometr. 17 (2003) 166–173, https://doi.org/10.1002/cem.785.
[10] M.C. Ortiz, J.A. Sáez, J. López Palacios, Typification of alcoholic distillates by multivariate techniques using data from chromatographic analyses, Analyst 118 (1993) 801–805, https://doi.org/10.1039/AN9931800801.
[11] M.C. Ortiz, L.A. Sarabia, R. García-Rey, M.D. Luque de Castro, Sensitivity and specificity of PLS-class modelling for five sensory characteristics of dry-cured ham using visible and near infrared spectroscopy, Anal. Chim. Acta 558 (2006) 125–131, https://doi.org/10.1016/j.aca.2005.11.038.
[12] D.C. Montgomery, Introduction to Statistical Quality Control, seventh ed., John Wiley & Sons, Inc., 2013.
[13] S. Ruiz, L.A. Sarabia, M.S. Sánchez, M.C. Ortiz, Handling variables, via inversion of partial least squares models for class-modelling, to bring Defective Items to non-Defective ones, Front. Chem. 9 (2021) 681958, https://doi.org/10.3389/fchem.2021.681958.
[14] S. Ruiz, L.A. Sarabia, M.C. Ortiz, M.S. Sánchez, Residual spaces in latent variables model inversion and their impact in the design space for given quality characteristics, Chemometr. Intell. Lab. Syst. 203 (2020) 104040, https://doi.org/10.1016/j.chemolab.2020.104040.
[15] H. Sartipizadeh, T. L. Vincent, Computing the Approximate Convex Hull in High Dimensions, arXiv preprint arXiv:1603.04422, 2016•arxiv.org, https://doi.org/10.48550/arXiv.1603.04422.
[16] C.M. Jaeckle, J.F. MacGregor, Industrial applications of product design through the inversion of latent variable models, J. Chemometr. 50 (2000) 199–210, https://doi.org/10.1016/S0169-7439(99)00058-1.
[17] M.C. Ortiz, L.A. Sarabia, M.S. Sánchez, A. Herrero, Quality of analytical measurements: Statistical methods for internal validation, in: second ed. Comprehensive Chemometrics, vol. 1, Elsevier, 2020 https://doi.org/10.1016/B978-0-12-409547-2.14746-8.
[18] D. Dua, C. Graff, UCI Machine Learning Repository, University of California, Irvine, CA, 2019. School of Information and Computer Science. Breast Cancer Wisconsin (Diagnostic) Data Set in, https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29. visited 22/12/2021.
[19] I. Cinar, M. Koklu, Classification of rice varieties using Artificial Intelligence methods, International Journal of Intelligent Systems and Applications in Engineering 7 (2019) 188–194, https://doi.org/10.18201/ijisae.2019355381.