Received: 30 August 2022 | Revised: 27 January 2023 | Accepted: 29 January 2023

DOI: 10.1002/aic.18071

AIChE JOURNAL

RESEARCH ARTICLE

Process Systems Engineering

Probabilistic predictions for partial least squares using bootstrap

James Odgers[1] | Chrysoula Kappatou[1] | Ruth Misener[1] | Salvador García Muñoz[2] | Sarah Filippi[3]

[1]Computational Optimisation Group, Department of Computing, Imperial College London, London, UK

[2]Synthetic Molecule Design and Development, Lilly Research Laboratories, Eli Lilly & Company, Indianapolis, Indiana, USA

[3]Department of Mathematics, Imperial College London, London, UK

Correspondence
Sarah Filippi, Department of Mathematics, Imperial College London, London SW7 2AZ, UK.
Email: s.filippi@imperial.ac.uk

Funding information
Eli Lilly and Company; Engineering and Physical Sciences Research Council, Grant/Award Numbers: EP/T005556/1, EP/T518207/1

Abstract

Modeling the uncertainty in partial least squares (PLS) is made difficult because of the nonlinear effect of the observed data on the latent space that the method finds. We present an approach, based on bootstrapping, that automatically accounts for these nonlinearities in the parameter uncertainty, allowing us to equally well represent confidence intervals for points lying close to or far away from the latent space. To show the opportunities of this approach, we develop applications in determining the Design Space for industrial processes and model the uncertainty of spectroscopy data. Our results show the benefits of our method for accounting for uncertainty far from the latent space for the purposes of Design Space identification, and match the performance of well established methods for spectroscopy data.

KEYWORDS
Design Space identification, model uncertainty, partial least squares, PLS, probabilistic prediction

1 | INTRODUCTION

Practitioners working within chemical manufacturing often have to deal with high dimensional data, where a key step in any analysis is reducing the dimensionality of the data to make predictions. One method for combining dimensionality reduction with prediction is partial least squares (PLS).[1–4] PLS projects the input variables to a lower dimensional latent space, that can both predict the output space and describe a significant portion of the variance of the input space. This latent space allows PLS to perform particularly well in situations with large numbers of highly correlated input variables. Building a PLS regression model means finding linear relationships between the input, output and latent spaces. The typical method for calculating PLS parameters is the NIPALS algorithm (see Appendix A). This method finds each dimension of the latent variables (t) iteratively, by finding latent variables that maximize covariance between the input and output, then deflating the data and repeating the process.[3,5]

When we refer to uncertainty of a prediction, we are describing a distribution of the probable values which would be observed if the experiment was carried out and the output was measured. This distribution is calculated based on a new specified input and a set of training data that has historic input values and the measured output in those cases. We note that this approach differs from other approaches where the uncertainty may only be given as a confidence interval. There are many reasons that the predictions from a model will not match up with the measured values, for example there will be noise in the quantities being used for the training data, the PLS prediction may be a linear approximation for a system which is not exactly linear, or the system being predicted may contain a certain inherent amount of unavoidable variation. PLS, and the uncertainty prediction method that we propose to build for it, does not

differentiate between these different sources of variability. Our approach incorporates all of these real sources of uncertainty into two categories. The first one is a random, unpredictable variation that we describe as a Gaussian noise on the output. The other source is uncertainty in the PLS parameters, which arises because the training data are finite. As there are a finite number of data points, a number of different parameters of PLS could match the training data quite well and could be capturing the underlying relationship between inputs and outputs more accurately. The novelty of our method is in how to weight different sets of PLS parameters according to how well they describe the data.

Calculating prediction uncertainty in PLS is nontrivial because of a complicated relationship between noise in the output and the predictions coming from PLS.[5] We discuss some of the many different approaches to overcome this difficulty later in this article. Much of the previous work was done on PLS in the 1980s and 1990s,[3,5–7] when limited computational power was available. This required the use of approximations with known limitations, but that were computationally quick. We use the massive increase in computational power to apply a Monte Carlo Bootstrap method, as described by Fushiki,[8] to PLS. By sampling with replacement to create synthetic data sets, our method allows the nonlinearities in the PLS parameters to be included in any prediction.

Our method produces a continuous distribution that is free from any assumption that the change in the prediction of PLS can be modeled using linear expressions. Instead of assuming linearity, we allow variation in the parameters of PLS to be driven by variation in the data used to train the PLS model. Explicitly, the assumptions of the method that we propose below are that the prediction error can be modeled by a normal distribution, that PLS is a reasonable prediction model for the data of interest, and that the training data are drawn from a similar distribution to the data for which the model will be used to make future predictions.

This article first overviews PLS and the different existing methods for uncertainty prediction. Next, we develop a novel bootstrapping method to provide a probabilistic prediction density for PLS. Finally, we demonstrate the applicability of our approach in five case studies. The first three of these case studies focus on Design Space identification,[9] that is, identifying process parameters which are likely to result in a desired output. The final two case studies focus on generating prediction intervals for real data, with one case study performing similarly to existing techniques and the other illustrating how our method's predictions can differ. Finally, the main outcomes of these case studies are summarized and potential avenues of further research are highlighted.

2 | PARTIAL LEAST SQUARES

PLS is a regression technique for making predictions from an input space ($x \in \mathbb{R}^{n_x}$) to an output space ($y \in \mathbb{R}^{n_y}$). Generally speaking, PLS is used when n_x is large and the training data are highly correlated, that is, when the training data are well-described by a low dimensional vector ($t \in \mathbb{R}^{n_l}$) plus a small amount of unexplained variance. It is common when describing the PLS algorithm to introduce matrices $X = (x_1,...,x_N)^T$, $Y = (y_1,...,y_N)^T$, and $T = (t_1,...,t_N)^T$, where the lower index i indicates a specific data point in a training data set ($\mathcal{D} = \{x_i, y_i\}_{i=1}^N$). This allows the relationships between the training data points to be written as

$$X = \mathbf{1_N}\bar{x}^T + TP^T + E, \tag{1}$$

and

$$Y = \mathbf{1_N}\bar{y}^T + TQ^T + F, \tag{2}$$

where $\bar{x}$ and $\bar{y}$ are the mean of the observed input and output data, $\mathbf{1_N} \in \mathbb{R}^{\mathbf{N}}$ is a vector containing N ones $P \in \mathbb{R}^{n_x \times n_l}$ and $Q \in \mathbb{R}^{n_y \times n_l}$ are least squares mappings from the latent space to the input and output spaces, respectively,[10] $E = [e_1,...,e_N]^T$ represents the variance in the input training data discarded by the PLS model, and $F = [f_1,...,f_N]^T$ represents the prediction error for PLS in each of the training data points.[11] There is an additional relationship between the input variable and the latent variable[12]: T is a linear transformation of X given by

$$T = XW\left(P^TW\right)^{-1}, \tag{3}$$

where W is a $\mathbb{R}^{n_x \times n_l}$ matrix.

In this article, we are interested in making a prediction from any point in the input space ($x \in \mathbb{R}^{n_x}$) to the likely points in the output space ($y \in \mathbb{R}^{n_y}$). When being used to make a prediction for a single data point, the PLS equations become

$$x = \bar{x} + Pt + e, \tag{4}$$

$$y = \bar{y} + Qt + f, \tag{5}$$

and

$$t = \left(W^TP\right)^{-1}W^Tx. \tag{6}$$

For a given data set, once the practitioner has specified the number of latent variables, the values of P, Q, and W are all found deterministically using the NIPALS algorithm (see Appendix A).

2.1 | Existing uncertainty techniques in PLS

The focus of this article is on calculating a distribution for possible output values, conditioned on the training data. Consider a set of training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ that contains measured input values and response values. We wish to compute the probability that a given output y is measured if a new input x is observed. This probability should

take into account the different possible parameters of a model by making the probability a weighted average of the predictions that come from these possible parameters—a process known as marginalization. For a PLS model with normally distributed noise, the parameters contain W, P, Q and an estimate for the standard deviation of the noise (σ).

$$p(y \mid x, \mathcal{D}) = \int p(y|x,\omega) p(\omega|\mathcal{D}) d\omega. \tag{7}$$

In many situations Bayesian methods can be used to find the distribution of parameters. This combines prior information ($p(\omega)$) with the likelihood of a set of parameters leading to the training data ($p(\mathcal{D}|\omega)$) to find a distribution known as the posterior for the parameters ω. In order to maintain normalization, the product of the likelihood and prior needs to be divided by a constant ($p(\mathcal{D})$) which is known as the evidence. The expression for the posterior is therefore

$$p(\omega|\mathcal{D}) = \frac{p(\mathcal{D}|\omega) p(\omega)}{p(\mathcal{D})}. \tag{8}$$

Multiple articles have applied Bayesian techniques to data-sets with low dimensional structures, such as those that PLS excels at. This has lead to a technique referred to as Probabilistic PLS,[13–18] where the data are assumed to be generated according to

$$x = t_s P_s + t_p P_p + e \tag{9}$$

$$y = t_s Q_s + f, \tag{10}$$

where t_s is a set of latent variables which describes both the input and output data through P_s and Q_s, respectively, and t_p is a set of variables which is only responsible for describing the input through P_p. This structure of problem can be solved using standard Bayesian estimation techniques and allows researchers to tackle many types of problems. However, this formulation does not provide the same solution as the standard NIPALS algorithm, and gives up many of NIPALS's advantages such as speed, familiarity and memory efficiency.

A widely used approach for studying the uncertainty of PLS is to treat it like other linear estimators.[7,19–22] For a one dimensional output, there are closed form analytic solutions, which make equivalent predictions to the NIPALS algorithm.[6,12] This closed form expression allows analogies to the techniques used to analyze ordinary least squares and principle component regression to be made. These methods have been widely discussed in the literature and we refer readers to Faber and Kowalski[7] and Zhang and García-Muñoz[23] for a detailed descriptions of several of these estimators. The key idea shared by all of these estimations is that the uncertainty in the linear transformation in PLS can be found by considering the effect of adding Gaussian noise to the observations, typically focusing on the outputs. These methods then simplify the relationship between the prediction from PLS to be only linearly dependant on the changes in the observed outputs. This allows the linear transformation that describes the prediction from PLS to be approximated by a multivariate normal distribution, which in turn allows the prediction output to be approximated by a student-t distribution.[5,19] This idea was later extended by Faber and Kowalski[20] to include perturbations in the input space as well, which also affect the parameters of PLS in a nonlinear way.[24]

An advantage of the methods described above is that they allow standard results from linear regression to be used, meaning that the output prediction is a normal distribution with unknown mean and variance. Combining these two sources of uncertainty results in a student-t distribution.[23] This allows us to write the prediction, conditional on the input x and a training data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, as

$$p(y \mid x, \mathcal{D}) = \frac{1}{s(x)} \mathcal{T}\left(\frac{y - \hat{y}(x; W, P, Q)}{s(x)}; N - n_l\right), \tag{11}$$

where $\mathcal{T}(z, \nu)$ denotes the probability density function (p.d.f.) of a student distribution with ν degrees of freedom evaluated at z, $\hat{y}(x; W, P, Q)$ is the prediction from the PLS model—as calculated by the algorithm given in Appendix B. The derivation for Equation (11) can be found in Appendix C. The approximate change in the prediction from PLS when noise is added is captured by $s(x)$, which is calculated using

$$s(x)^2 = \sigma^2 \left(1 + \frac{1}{N} + h(x)\right), \tag{12}$$

where σ is an estimate for the output noise. The parameter uncertainty in the prediction is captured by the leverage, $h(x)$, which is given by

$$h(x) = x^T \Sigma x, \tag{13}$$

where x is a new point for which we are trying to predict the output and Σ is the covariance of the linear transformation that gives the same prediction as PLS.

The method by which the covariance matrix of the multivariate normal distribution is estimated varies, but we wish to highlight two common approaches. One popular method is to assume that the span of the latent space in the input does not vary. This leads to an estimate of the covariance sometimes called the zeroth-order approximation.[19] This allows $s(x)$ to be calculated using Equation (12) with

$$h(x) = \frac{t(x)^T \left(T^T T\right)^{-1} t(x)}{N - 1} \quad \text{and} \quad \sigma^2 = \frac{1}{N - n_l} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2. \tag{14}$$

These expressions are used in combination with Equation (11) to generate a probabilistic distribution to compare our results to.

Another method we wish to discuss for estimating the covariance matrix for the linear transformation is to use bootstrap techniques. Faber[21] investigated the performances of both bootstrap by residuals and bootstrap by objects in spectroscopy. This study preferred bootstrap by residuals for estimating the covariance of the uncertainty in the linear transformation describing PLS in spectroscopy, noting the similarity of its performance to the zeroth order method. This article removes the assumption that the parametric uncertainty in the

predictions from PLS can be modeled by a normal distribution, which the methods discussed in Faber[21] do not address.

Other authors have also considered methods which do not assume that the prediction uncertainty from PLS can be modeled by a normal distribution. This was first examined by Denham,[5] who proposed a method to form predictive intervals for PLS prediction using bootstrap by residuals (see Tibshirani and Efron[25] for a detailed description of bootstrap by residuals). In Denham's method, synthetic data sets are generated by adding random noise, following the empirical distribution of the prediction errors, to the predictions from PLS. These synthetic data sets are then used to generate different sets of PLS parameters. For a new input, the set of parameters are used to calculate a distribution of predictions from PLS, which can be convoluted with the empirical distribution of errors to generate confidence intervals. A similar method for estimating prediction intervals was also used by Reis and Saraiva,[26] but with the focus on situations with high dimensional outputs that contain a structure that is being found by PLS. One aspect not considered by these methods is the effect of variations in the input. If the inputs are of full rank and go directly to the outputs then simply considering the effects of perturbing the output considers all of the noise in the system and can lead to better estimates for the uncertainty.[25] However, latent variable models, such as PLS, assume that only a portion of the input is of interest to predict the output and estimates which portion should be used from the data. As this is a quantity measured from the data it is important to consider that the wrong portion of the inputs has been selected. This can be seen clearly in probabilistic PLS, where ignoring the uncertainty in P_s and P_p is clearly unjustified. In NIPALS PLS the portion of the input which is selected depends on both the input and the output data, so by not considering possible changes in the input an important source of uncertainty is missed by bootstrap by residuals, our method captures this by using bootstrap by pairs.

The approach proposed in this article builds on the work by Denham.[5] Our work applies the method proposed by Fushiki et al.,[27] where a distribution for the parameters $p(\omega|\mathcal{D})$ is not estimated by the posterior—but by a set of maximum likelihood estimates (MLEs) for different possible data sets that are approximated using the bootstrap by pairs method. Our method lends itself to the NIPALS algorithm as each parameter can individually be seen as a MLE,[4] although we are unaware of any result that indicates that the parameters of PLS are a joint MLE. Our approach uses the regular NIPALS algorithm, does not assume any parametric form of uncertainty from the predictions of PLS, and considers all sources of uncertainty.

3 | METHOD

Given a training data set $\mathcal{D}=\{(x_i,y_i)\}_{i=1}^{N}$, we wish to predict the output for a new observation $x\in\mathbb{R}^{n_x}$ using a PLS model. More specifically, we want to construct a predictive distribution $p(y\,|\,x,\mathcal{D})$ taking into account the uncertainty in the estimation of the PLS model parameters due to finite training data. To do so we use bootstrapping, a widely used technique for uncertainty quantification of statistical inference procedure. Efron and Gong[28] present a good overview of bootstrapping.

Figure 1 depicts our approach to construct the predictive distribution $p(y\,|\,x,\mathcal{D})$ using bootstrapping. The key idea is that the randomness in the training data set is a good approximation of the randomness that would be observed in any future observations. Therefore, the uncertainty of the values found in any parameters that arise from the data can be evaluated by examining how the randomness from the observed training data effects the estimated parameters. The first step of our method therefore consists of randomly simulating B new synthetic data sets of length N, denoted by $\mathcal{D}^*_{(b)}=\left\{(x_i,y_i)^*_{(b)}\right\}_{i=1}^{N}$, $b=1\ldots B$, by sampling from the empirical distribution. Note that generating a N sample synthetic data set from the empirical distribution simply requires sampling with replacement N data points from the training data set. For each of these new synthetic data sets $\mathcal{D}^*_{(b)}$, we estimate the PLS parameters $W^*_{(b)}$, $P^*_{(b)}$, and $Q^*_{(b)}$ as well as the MLE for the variance of the prediction noise defined by

$$\sigma^{*2}_{(b)}=\frac{1}{N}\sum_{i=1}^{N}\left(y^*_{i,(b)}-\widehat{y}\left(x^*_{i,(b)};W^*_{(b)},P^*_{(b)},Q^*_{(b)}\right)\right)^2, \tag{15}$$

where $\widehat{y}\left(x^*_{i,(b)};W^*_{(b)},P^*_{(b)},Q^*_{(b)}\right)$ is the prediction from the PLS algorithm for input $x^*_{i,(b)}$ using parameters $W^*_{(b)}$, $P^*_{(b)}$, and $Q^*_{(b)}$. Appendix B describes the method for finding this $\widehat{y}$ prediction. Given these estimates, we compute the prediction probability of observing y given the new input x for each of the B sets of parameters. Here we assume that the prediction error is normally distributed, so

$$p\left(y\mid x,W^*_{(b)},P^*_{(b)},Q^*_{(b)},\sigma^*_{(b)}\right)=\mathcal{N}\left(y;\widehat{y}\left(x;W^*_{(b)},P^*_{(b)},Q^*_{(b)}\right),\sigma^*_{(b)}\right) \tag{16}$$

where $\mathcal{N}(y;\mu,\sigma)$ denotes the p.d.f of a normal distribution with mean μ and variance σ^2 evaluated at y. The overall prediction is found by averaging the predictions of the B individual bootstrapped predictions. Algorithm 1 presents the method.

Mathematically, our method constructs a bootstrap predictive distribution by computing the expectation of the predictive distribution over different possible bootstrap data sets, similar to Fushiki et al.[27] The resulting predictive distribution is given by

$$p(y\,|\,x,\mathcal{D})=E_{\mathcal{D}^*}\left[p(y|x,\widehat{\omega}(\mathcal{D}^*))\right]=\int p(y|x,\widehat{\omega}(\mathcal{D}^*))\;p(\mathcal{D}^*|\widehat{p})\;d\mathcal{D}^*, \tag{17}$$

where $\widehat{\omega}(\mathcal{D}^*)=(P^*,Q^*,W^*,\sigma^*)$ denote the estimated parameters for a given data set $\mathcal{D}^*$. The expectation is taken over the boostrap data set $\mathcal{D}^*$, which consists of N samples drawn independently from the empirical distribution

$$\widehat{p}(x,y)=\frac{1}{N}\sum_{i=1}^{N}\delta\left((x,y)^T-(x_i,y_i)^T\right), \tag{18}$$

where $\delta(\cdot)$ is the Dirac delta function. The integral in Equation (17) is approximated using Monte Carlo integration as follows

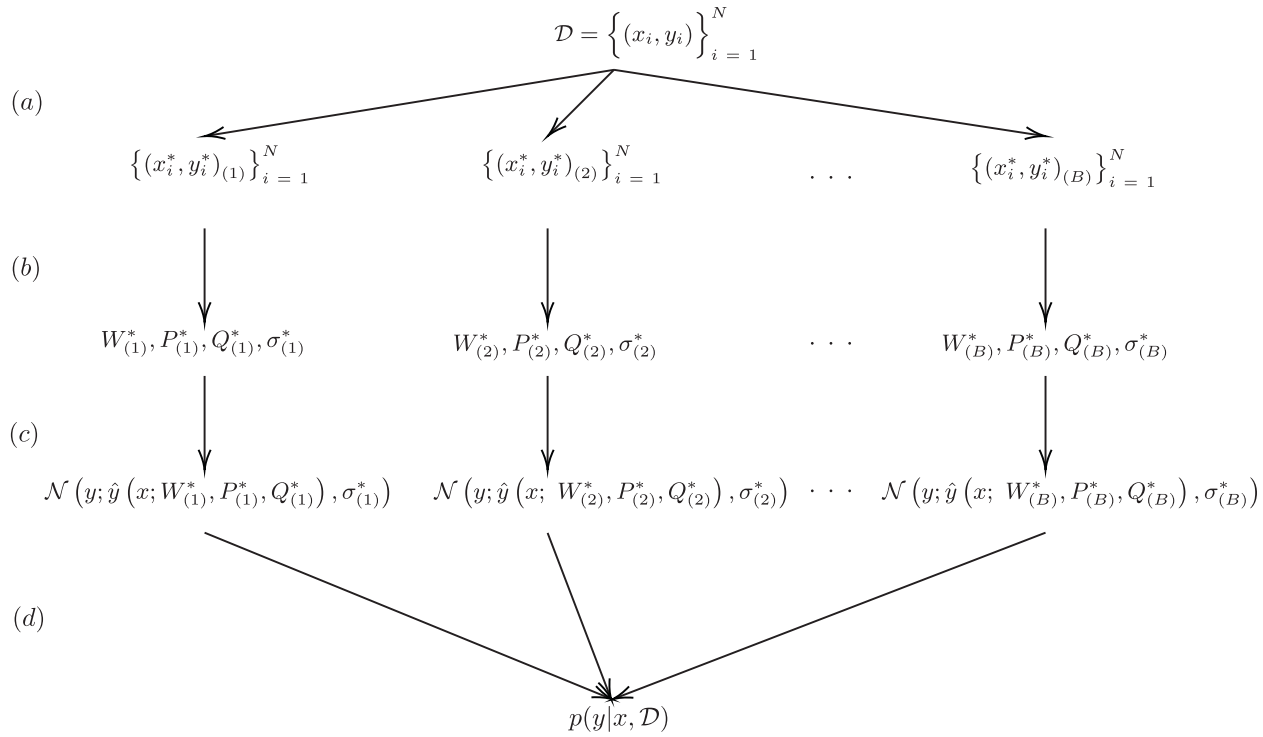


FIGURE 1 Diagram of the proposed method for generating a probabilistic prediction from PLS using bootstrap. (a) The original data set is sampled with replacement to produce bootstrapped data sets. (b) The bootstrapped parameters are estimated using the standard NIPALS algorithm described in Appendix A. (c) The bootstrapped parameters are used to generate a prediction for the output of PLS using a normal distribution with mean given by the prediction from PLS and variance given by the mean squared error of prediction. (d) The estimates from each of the bootstrapped samples are averaged out to give an estimate for the overall probabilistic distribution of the output given the input and observed data.

$$p(y \mid x, \mathcal{D}) \approx \frac{1}{B}\sum_{b=1}^{B} p\left(y|x, \widehat{\omega}\left(\mathcal{D}^*_{(b)}\right)\right), \tag{19}$$

where $\mathcal{D}^*_{(b)}$, $b = 1 \ldots B$, are bootstrapped data sets.

One aspect to note about our method is that the standard deviations of any individual predictions do not depend on the location of the input. Instead, we achieve a prediction uncertainty that is dependant on the inputs by changes in the mean of the predicted distributions. Points further from the mean of the training data will produce prediction means that are further from one another, resulting in a wider overall distribution for $p(y \mid x, \mathcal{D})$, despite each individual prediction having constant variance.

4 | APPLICATIONS

4.1 | Setup

This section illustrates the performance of the proposed approach (see Section 3) on different case studies. We claim that the bootstrapping approach is especially strong in generating good probabilistic predictions in low-data settings that do not follow a standard linear model, that is, a more complex model than ordinary least squares regression. These low-data, nonstandard settings frequently arise in practice. To show the potential of the bootstrapping approach, we proceed by demonstrating its performance on a series of examples: the purpose of these examples is to show that the probabilistic bootstrapping approach performs how we would expect in straightforward examples and offers compelling advantages for real-world test instances.

This section studies two major applications for the bootstrapping approach, which constructs a probabilistic prediction for the output y given an input x and a set of training data $\mathcal{D}$. The first application is identifying a Design Space, that is, the multidimensional set of input variables (including both material and process parameters) that results in a product of the desired quality.[9] Our simple example and two industrially-relevant case studies in Michael Addition and Reductive Amination consider Design Space identification. The second application, calculating probabilistic prediction densities for the output y space, is demonstrated for applications in High Shear Wet Granulation and Spectroscopy.

We compare the bootstrapping approach to the results achieved using the zeroth order approximation in Equations (11) and (12). Due to the linear approximation in the zeroth order method, authors have typically used PLS only as an intermediate step to find the Design Space.[29–32] However, we compare to the zeroth order method because it is widely used when considering uncertainty in PLS.[23,29,30] We use these comparisons to illustrate

ALGORITHM 1 Method for calculating probabilistic bootstrap PLS predictions

1. For $b = 1$ to B
   a. Generate a data set $\mathcal{D}^*_{(b)} = \left\{\left(x^*_i, y^*_i\right)_{(b)}\right\}_{i=1}^{N}$ from the empirical distribution by sampling with replacement from the training data. $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
   b. Calculate the PLS parameters $\left(P^*_{(b)}, Q^*_{(b)}, W^*_{(b)}\right)$ and estimate the variance of the noise from a single bootstrapped data set ($\sigma^*_{(b)}$), using

$$\sigma^{*2}_{(b)} = \frac{1}{N}\sum_{i=1}^{N}\left(y^*_{i,(b)} - \hat{y}\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)\right)^2, \tag{20}$$

2. Calculate the predictive distribution

$$p(y \mid x, \mathcal{D}) \approx \frac{1}{B}\sum_{b=1}^{B}\mathcal{N}\left(y; \hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \sigma^*_{(b)}\right), \tag{21}$$

where $\mathcal{N}(y;\mu,\sigma)$ is the p.d.f. of a normal distribution with mean μ and standard deviation σ evaluated at position y, and $\hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)$ is the prediction from the PLS algorithm for input x using parameters $W^*_{(b)}$, $P^*_{(b)}$, and $Q^*_{(b)}$.

the performance of our method under a range of different circumstances.

All of the computations were carried out on a 2020 M1 MacBook Pro with 16GB of ram. The code can be found at https://github.com/jamesacodgers/bootstrapPLS. The code for the third Design Space example and the spectroscopy example, is not included for confidentiality reasons.

4.1.1 | Identifying a Design Space

The Design Space is the set of input values to a manufacturing process which will produce a desired set of output parameters with a probability above a given threshold based on a given model and training data set. Peterson[33] defines the Design Space mathematically and probabilistically as

$$DS = \{x \in \mathbb{R}^{n_x} \mid p(y \in y_{des} \mid x, \mathcal{D}) > 1 - \alpha\}, \tag{22}$$

where $1-\alpha$ is a minimum acceptable probability. Sections 4.2–4.4 consider applications identifying a Design Space.

Identifying Design Spaces has received much study within the literature.[34–37] There has also been notable work using PLS to help identify the Design Space, for example, by identifying a region, known as the Experimental Space, within which the Design Space is predicted to fall.[29,30,32] Another Design Space identification example uses PLS to perform dimensionality reduction to find suitable samples to examine using a Bayesian model.[31]

ALGORITHM 2 Monte Carlo method for calculating the probability of an output falling in the desired range

1. For m = 1 to M:
   a. Randomly select integer b between 1 and B.
   b. Draw a random number y'_m from the multivariate normal $\mathcal{N}\left(\hat{y}\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \Sigma^*_{(b)}\right)$.
2. The probability can then be calculated using

$$p = \frac{1}{M}\sum_{m=1}^{M}\prod_{k=1}^{n_y} I\left(y_{min,k} < y'_m < y_{max,k}\right), \tag{25}$$

where $I(\cdot)$ is the indicator function and the subscript k indicates which output dimension is being referred to.

Applying the bootstrapping approach to Design Space identification requires integrating the probability distribution function over all acceptable values of the output (y). When the output is one dimensional, Equation (21) can be simply adapted to replace the normal probability distribution functions with the difference between the cumulative distribution function (c.d.f.) values at the minimum and maximum acceptable values for the output, resulting in the equation

$$p(y \mid x, \mathcal{D}) \approx \frac{1}{B}\sum_{b=1}^{B}\Phi\left(y_{max} \mid \hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \sigma^*_{(b)}\right) - \Phi\left(y_{min} \mid \hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \sigma^*_{(b)}\right), \tag{23}$$

where $\Phi(\cdot \mid \hat{y}_b, \sigma_b)$ is the c.d.f. of the normal distribution with mean $\hat{y}_b$ and standard deviation σ_b.

For multidimensional outputs a Monte Carlo approach is used, where samples are randomly drawn from the predicted probability distribution and the fraction of the samples which are within the desired output range is taken as the estimate for the probability of the output falling in the desired range. To draw random samples first a random integer (b) between 1 and B is drawn, then a point is drawn from the p.d.f. of a multivariate normal distribution using the parameter values found using the bth parameter values. In order to fully define this multivariate normal distribution the variance $\sigma^*_{(b)}$ needs to be replaced by the covariance matrix $\Sigma^*_{(b)}$ with elements defined by

$$\sigma^*_{(b),kl} = \frac{1}{N}\sum_{i=1}^{N}\left(y^*_{i,(b),k} - \hat{y}_k\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)\right)\left(y^*_{i,(b),l} - \hat{y}_l\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)\right), \tag{24}$$

where k and l indicate the output dimension that is being referred to. This process is described in Algorithm 2.

One point which we would like to acknowledge is that Design Space identification problems often do not follow the latent structure implicitly assumed by PLS. We believe that considering PLS in this type of setting is reasonable, as in these cases the dimensionality reduction acts as a type of variable selection, which can allow PLS to perform well. In fact, PLS has been studied in the literature for similar applications for a significant period of time.[38] However, there is a requirement that any practitioner interested in applying any PLS uncertainty technique first considers whether a PLS model is appropriate for a given task, or whether a task specific model—such as the ones put forward by Reis et al.[39]—performs better.

4.1.2 | Calculating probabilistic prediction densities

One downside of using real data for Design Space identification is that we are unable to compare to the true Design Space, as this is an unknown we are trying to estimate. To show the validity of our method for real data, Sections 4.5–4.6 compare the calculated prediction densities to the observed outputs in High Shear Wet Granulation and Spectroscopy. This comparison allows us to show that our method is able to make predictions that meaningfully capture the uncertainty of the model for real data.

4.2 | Simple simulated example

A very simple Design Space identification example illustrates the technique. In this example, a two dimensional input (x_1, x_2) generates a one dimensional output (y), via the equation

$$y = x_2 + \epsilon, \tag{26}$$

where ϵ is a Gaussian noise term with variance 0.2. A training data set $\mathcal{D}$ with $N = 40$ samples was generated by sampling input values from a multivariate normal distribution $(x_1, x_2) \sim \mathcal{N}\left((5,5)^T, 5\,\mathbb{I}\right)$ and the output y was generated using Equation (26).

To identify the Design Space for $y_{\text{des}} = [8,12]$ with $\alpha = 0.9$ based on the training data $\mathcal{D}$, we estimate the bootstrap predictive distribution using $B = 1000$ bootstrap data sets and construct the Design Space. Figure 2 represents the resulting Design Space. The contours depicted in Figure 2 come from calculating the probability that the given input results in the desired output of each point in a 100×100 grid, which was interpolated to produce the contours. The full details of the implementation can be found in the provided code. An important feature of our approach is that it considers uncertainty in the latent space through bootstrapped data sets. Figure 2 illustrates how the latent space estimated using all the training data differs from the latent spaces estimated using each of the 1000 bootstrapped data sets. We observe that, by incorporating the uncertainty via bootstrapping, the Design Space constructed using our approach is naturally restricted to be close to the latent space.

We now compare the Design Space found by our method to the one obtained using the zeroth order method as well as to the so-called true Design Space, that is the Design Space if one knew the data-generating process, that is,

$$\left\{x \in \mathbb{R}^2 \mid p(y \in [8,12] \mid x) > 1 - \alpha\right\} = \left\{x \in \mathbb{R}^2 | p(8 - x_2 \le \epsilon \le 12 - x_2) > 1 - \alpha\right\}.$$

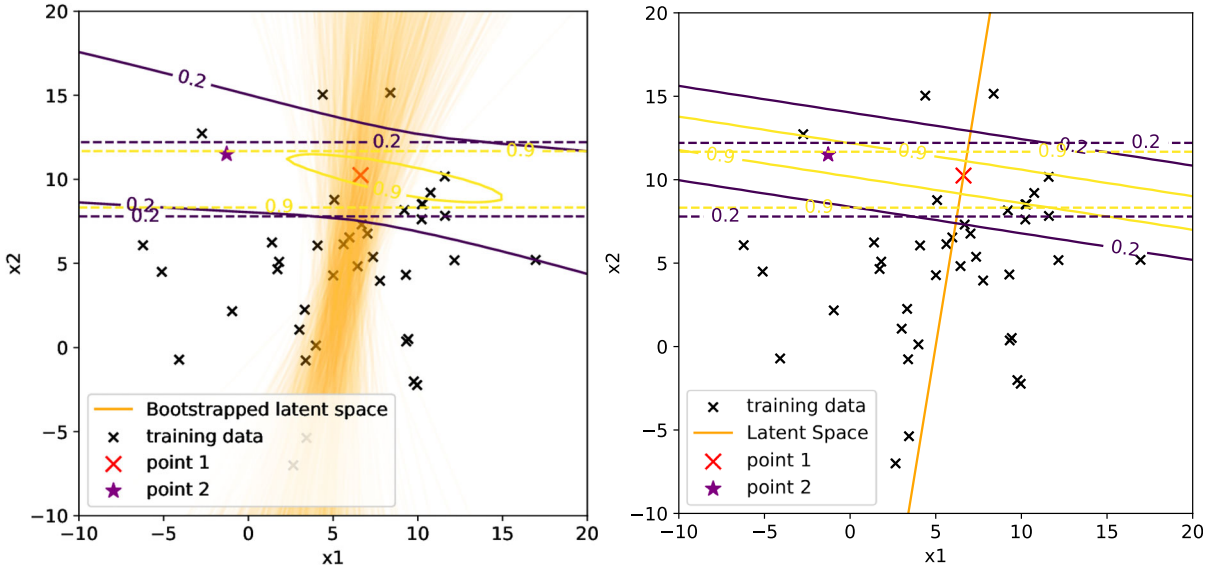
Figure 2 shows that the Design Space constructed using the probabilistic bootstrap approach is included within the true Design Space, while the Design Space constructed using the zeroth order method only overlaps with the true Design Space. Recall that the zeroth order method makes predictions that are invariant with respect to the distance of the point from the latent space. In other words, the zeroth order method assigns equal probability to all points within the input space (x) that map to the same point in the latent space (t). Therefore, the Design Space resulting from the zeroth order method is a band orthogonal to the estimated latent space. In contrast, the probabilistic bootstrap approach produces curves that reflect the uncertainty arising from the portion of the input not typically considered by a PLS model.

Figure 3 shows the estimated predictive distributions $p(y \mid x, \mathcal{D})$ for two different values of x, indicated by the purple and red points in Figure 2, using both the zeroth order and probabilistic bootstrap prediction methods. The two values of x have identical positions in the latent space. As the zeroth order method described in Equations (11) and (12) only considers the projections of the two points on the latent space, the two posterior predictive distributions using the zeroth order method are identical. In contrast, the bootstrap method, which does consider the uncertainty in the position of the latent space, produces two distinct posterior predictive distributions. The probabilistic bootstrap approach emphasizes the importance of the distance to the latent space as a source of uncertainty.

The prediction arising from the bootstrap method is strongly negatively skewed. This is in contrast to any uncertainty method for PLS models that relies on the student-t distribution, including the zeroth order method. As discussed earlier, the student-t distribution arises from assuming that the distribution can be described by just an unknown mean and variance. By relaxing the assumption that uncertainty in PLS parameters, plus Gaussian noise with unknown variance results in a student-t distribution, we see that the resulting distribution is markedly skewed and different from any student-t distribution. This skew is due to the known nonlinearities in the parameter estimation of PLS.

4.3 | Michael addition reaction

The next example explores Design Space identification for the Michael Addition reaction. The full details on the simulation of this reaction are given by Kusumo et al.[37] For this article, the real data generating process will be treated as a black box that accepts two process inputs (the molar ratio of two reactants, and time in a continuous stirred reactor vessel) and predicts two outputs (a



(a) Bootstrap Method (b) Zeroth Order Method

FIGURE 2 Figure showing the Design Space for a simple simulated example. The dashed lines indicate the true probabilities of data generated at that point falling in the desired range. The red and purple points indicate points of interest discussed in Figure 3. (a) The latent spaces (orange lines) found by the bootstrapped data sets, and the resulting Design Spaces using probabilistic bootstrap prediction (shown as purple and yellow sollid lines) —whose shape depends on the distance to the latent space. (b) The latent space (orange line) found by the regular PLS model using the training data set, the resulting Design Spaces (purple and yellow solid lines), do not account for the distance to the latent space.

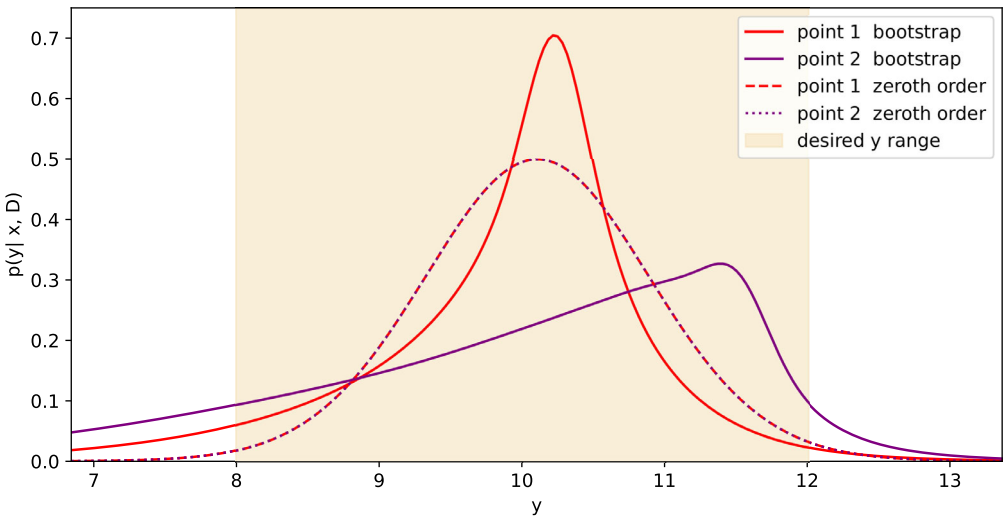
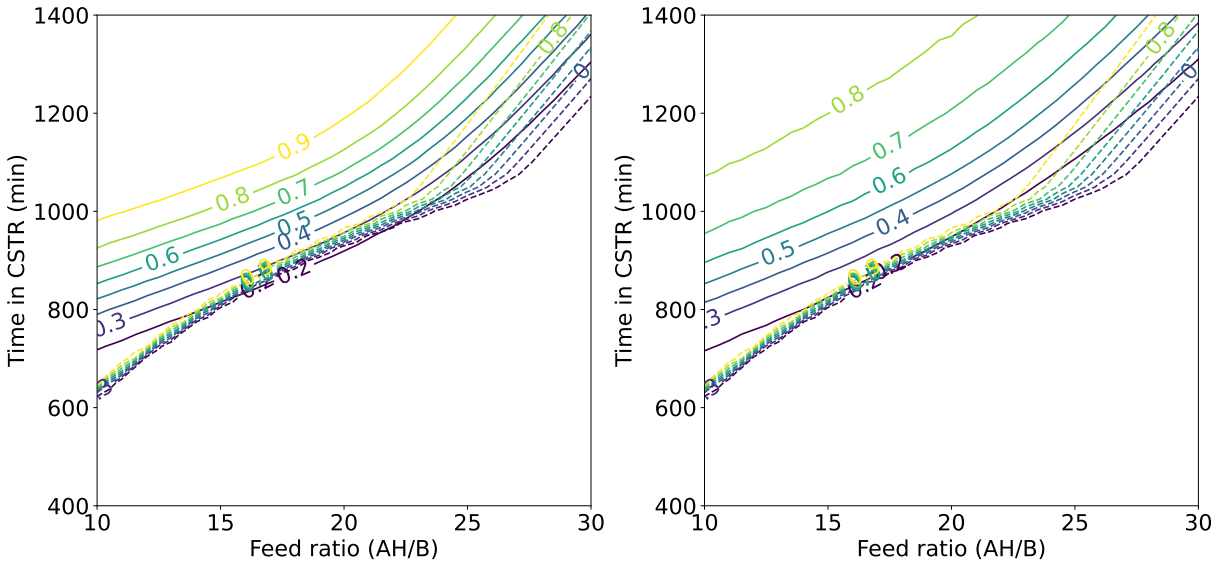


FIGURE 3 Comparison of the two predictions of the probability density function from the zeroth order approximation method and the bootstrap prediction. The solid lines show the bootstrap method proposed here, the dashed line shows the prediction from the zeroth order approximation method, and the shaded area is the desired output range for y. As can be seen above, there is no difference in the predictions from the two points using the zeroth order approximation, whereas the bootstrap method applies significant weight to a far wider range of values.

greater than 90% conversion of the carbon, and a residual molar concentration of an intermediate product). Twenty-five training data points were simulated in a grid arrangement across the inputs, with molar ratio varying between 10 and 30, and the time in the continuous stirred reactor vessel was varied between 400 and 1400 minutes. Training data were initially generated without noise, then a Gaussian error term was added. The standard deviation of the error term was set as 1% of the mean of the training data before the noise was added.

These data were fit with a two component PLS model, which was used to calculate a Design Space with a greater than 90% conversion of carbon and a residual molar concentration of less than 2 mmol per liter. The results of these are shown Figure 4, with the true probabilities of the outputs being in the Design Space shown as



(a) Bootstrap Method (b) Zeroth Order Method

FIGURE 4 Design Spaces for the Michael addition reaction using bootstrapped and zeroth order methods. (a) The Design Space from the bootstrap method and (b) the zeroth order method. The axes indicate different values for two independent inputs to the manufacturing process. The contours indicate the different probabilities of given inputs being within the Design Space, with the solid lines indicating the probabilities from the PLS methods, while the dashed lines indicating the true probabilities of that input resulting in an acceptable output.

dashed lines, and the Design Spaces found by the two models shown in solid lines. In this case it is obvious that the bootstrap method performs significantly better, correctly capturing a significant portion of the inputs in this range as having a greater than 90% chance of being within the Design Space, while the zeroth order approximation fails to predict any region of these inputs as having a greater than 90% chance of being within the Design Space. While it is not possible to identify the exact causes of the significant difference in the performance between the two methods, previous results from Fushiki[8] have shown that in general bootstrap prediction method performs remarkably well in cases where no set of input parameters for the model would recapitulate the true data generating process, as is the case here.

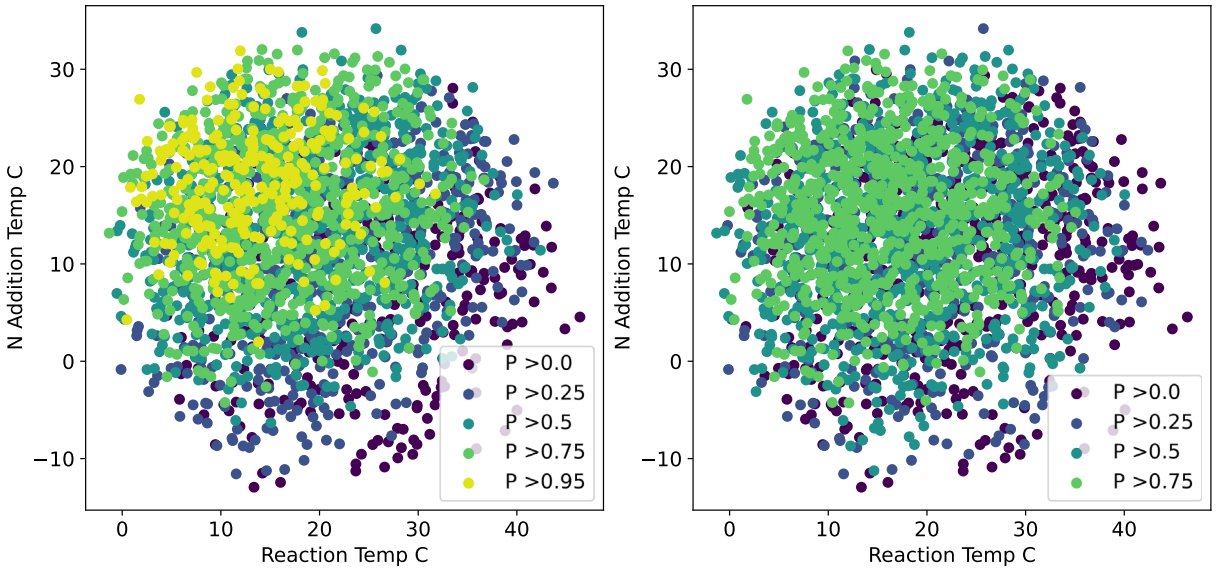
4.4 | Reductive amination

The final Design Space identification example is an industrial case study. In this example there are 15 input dimensions and two output dimensions of interest, one conversion percentage for outputs with a minimum required conversion rate and one contaminant with a maximum allowable presence in the final product. Due to the high dimensional inputs, grid sampling, as was used in the Michael Addition reaction is not possible, so instead a Markov Chain Monte Carlo (MCMC) was used to sample the areas of higher probability of being within the Design Space. In order to constrain the samples to be close to the historic data, samples were only drawn from the Knowledge Space defined by Facco et al.,[29] and described in more detail in Appendix D.

The results of both the bootstrap and the zeroth order approximation are shown in Figure 5. This plot shows the projection of points down to a two dimensional subspace of the true input dimension—a temperature of a reactant when added and the temperature of the reaction. These points are color coded by the probabilities of the given inputs producing an output fulfilling both a greater than 90% conversion to the desired output, and a less than 1% presence of an impurity generated by the reaction. For these acceptance criteria the zeroth order method fails to find any points that would constitute a Design Space at a confidence level of 90%. Meanwhile the bootstrap method proposed here predicts that a large number of the points will fulfill the required Design Space criteria. While it is not possible to comment on the accuracy of this Design Space, there are many situations in which it is preferable to have a larger Design Space, even at an inevitable risk of incorrectly assigning some points that should not truly be in the Design Space.

4.5 | High shear wet granulation

This data set arises from a wet granulation process.[40] Vemavarapu and Badawy[40] consider how six input dimensions: aqueous solubility, contact angle, water holding capacity, two distinct measures of particle size, and surface area affected a number of response variables. In this article, we only consider the percentage of particles larger than 1.4 mm as the response of interest. We abstract away specifics and refer to the inputs as x_j, $j \in \{1,\ldots,6\}$ and the output as y. We scale the data as described by Vemavarapu and Badawy[40] in order to improve the model performance, that is, we



(a) Bootstrap Method (b) Zeroth Order Method

FIGURE 5 Projection of randomly sampled points in the input space to the two dimensional space of just reaction temperature and nitrogen addition temperature. The differently colored points indicate the different probabilities for the given input resulting in an acceptable output. (a) The probabilities found using the bootstrap method proposed here, (b) the probabilities using the zeroth order method. The same point appears to have different probabilities of being within the Design Space as they have been projected down from a higher dimension—in reality these points are separated along one or more of the other dimensions not shown here.

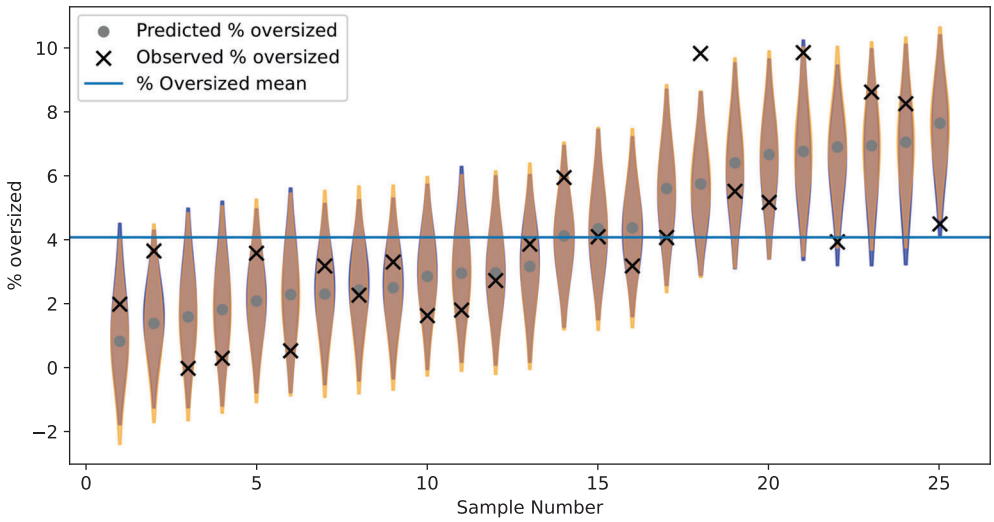
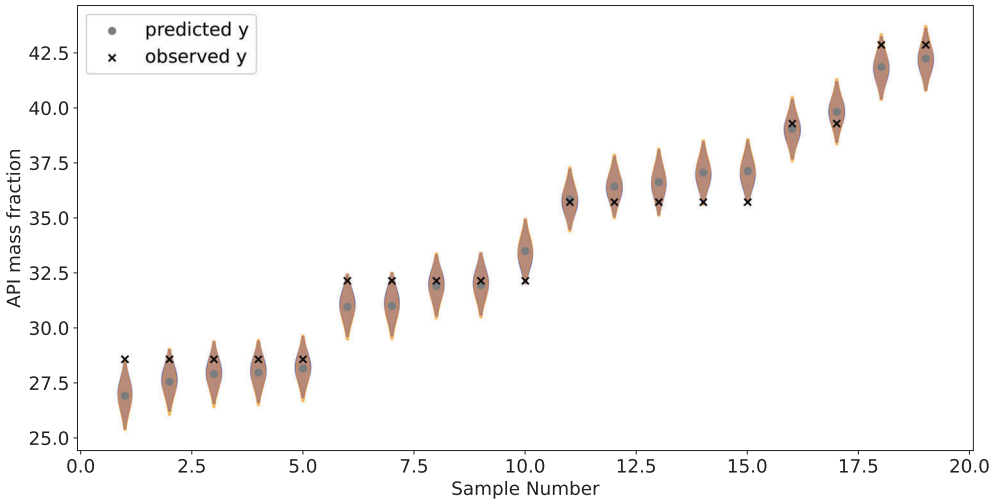


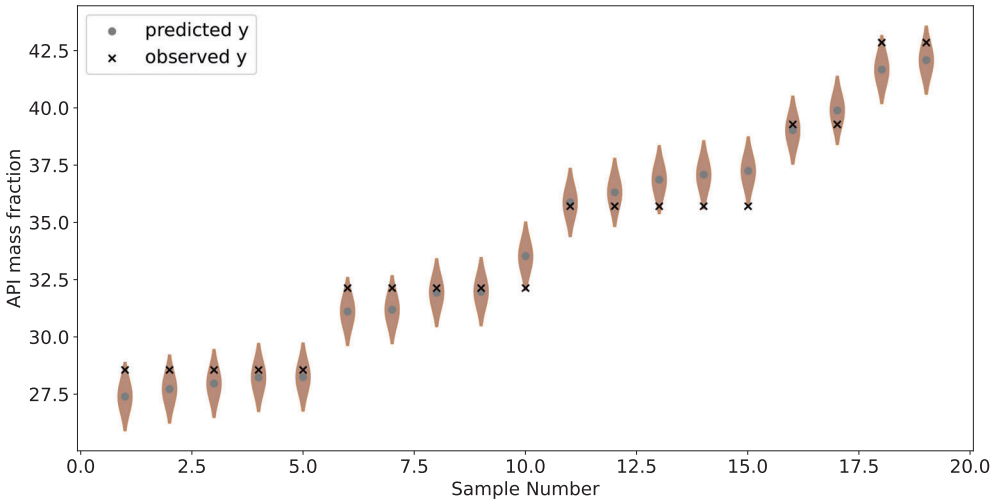
FIGURE 6 Violin plot for the data given in Vemavarapu and Badawy.[40] The blue shapes indicate the probability distribution found by the bootstrap prediction, whereas the orange show the prediction from the zeroth order approximation method. The points are the predictions from PLS before uncertainty is included and the crosses are the true values of the % oversized particles. The blue horizontal line is the mean value of the training data outputs. For the sake of clarity both violin plots are truncated at the 0.025 and 0.0975 quantiles of the plot to remove tails from both distributions.

apply a logarithm to x_j for $j \in \{1,3,4,5\}$ and a cube root to x_6. For each prediction a different training set, excluding the point that is being predicted, but including all other points was created. This new data set was used to train a PLS model and find the uncertainty predictions.

Figure 6 shows a violin plot of the prediction density between the 0.025 and 0.975 quantiles of the predicted distribution, along with the true observed value and the deterministic prediction from PLS. The prediction from the zeroth order method is also included for the sake of comparison. Both of these methods produce credibility



(A) 50 training data points



(B) 1000 training data points

FIGURE 7 The prediction from bootstrap and zeroth-order approximation for spectroscopic data. The blue shapes indicate the probability distribution found by the bootstrap prediction, whereas the red show the prediction from the zeroth-order approximation method. The points are the predictions from PLS before uncertainty is included and the crosses indicate the observed values for the API fraction. The two predictions are indistinguishable for the 1000 point training data set, and nearly identical for the 50 point training data set.

intervals that contain the majority of data points, with the bootstrap prediction only failing to predict the interval for sample number 18 and the zeroth order approximation missing samples 18 and 25. The primary difference between these predictive densities is that the bootstrap prediction generally places more weight on the samples being closer to the mean observed values in the training set - shown as a blue horizontal line in the figure. This preference for predicting the mean by the bootstrap method shifts the intervals upward when the prediction is significantly below the mean and downward when the prediction is above the mean. This effect is the reason why the bootstrap prediction finds sample number 25 within its predicted interval, while the zeroth order approximation does not.

4.6 | Spectroscopy

The last example looked at here uses spectroscopic data.[41,42] The raw data were preprocessed with the methods described by Hetrick et al.,[41] where only wavelengths from 1626 to 1993 nm are used with Standard Normal Variate transformation applied (transforming the data to have mean zero and variance one), and a three latent variable PLS model. Spectroscopy was chosen to evaluate how our method performed in high dimensional settings. Additionally, this case study provided a good chance to evaluate our method in comparison to the zeroth order approximation method, where it has already been shown to perform well.[21] Note again that the data for this case study has not been released for confidentiality reasons.

Two PLS models were fit from this preprocessed data, one using 50 training data points and one using 1000 training data points, these are shown in Figure 7. As can be seen for this data set the zeroth order approximation and the prediction from the bootstrapped PLS method are indistinguishable. This is a boon for our method as it has been found on multiple occasions that for spectroscopy data sets the zeroth order approximation method is very effective in finding accurate predictions,[21] suggesting that even in cases where established methods perform well, our method is not out performed.

5 | CONCLUSIONS

This article applies the techniques described by Fushiki et al.,[27] which allows us to make principled probabilistic predictions from PLS. The probabilistic bootstrapping approach considers the uncertainty arising from the interaction between the parameter uncertainty and the portion of the input variables typically ignored by PLS. In addition to the theoretical justifications for this method, the probabilistic bootstrapping approach finds meaningful for prediction intervals for real world data, and runs efficiently enough to provide useful Design Spaces.

Although prediction and Design Space identification are the two topics chosen to demonstrate the power of our method here, they are by no means the only uses that our method could be put to. For example, our method could also be adapted to identify regions where it is likely that the Design Space could be—similar to finding experimental spaces as described by Facco et al.[29] and Bano et al.[30] This differs from finding a Design Space, as the Design Space combines both parametric uncertainty and uncertainty from inherent random fluctuations, however regions with high parametric uncertainty should still be considered when looking for inputs that potentially result in the desired outputs.

Another direct application of the method presented in this article is in pharmaceutical manufacturing, specifically in the implementation of a Real Time Release Testing solution using a PLS model. The method presented here allows the uncertainty of the model to be accounted for in the decision making process (e.g., to accept or to reject product), without making (invalid) assumptions regarding the trustworthiness of the predictions. The consideration of uncertainty can be of particular interest in applications near a physical boundary, like a decay curve in continuous manufacturing, or the estimation of drug content for a low dose product.

One short fall of our method is that it assumes that the system can be well approximated by a linear prediction and that the difference between the predicted and the true values of the output are explainable by noise. However, many relationships in chemical manufacturing and chemometrics cannot be well modeled by a linear relationship between input and output variables. The current work is unable to distinguish the models uncertainty due to noise and uncertainty arising from mismatch between PLS and the true system being investigated. An open avenue for PLS research is to consider methods to incorporate the uncertainty arising from model mismatch.

NOTATION

x	input variable
y	response variable
$\{(x_i, y_i)\}_{i=1}^{N}$	training data set
$\left\{\left(x_i^*, y_i^*\right)_{(b)}\right\}_{i=1}^{N}$	bootstrapped training data set
P	least squares transformation from t to x
Q	least squares transformation from t to y
W	concatenated, transposed values of w_l
$\bar{x}$	sample mean of X
$\bar{y}$	sample mean of Y
$P^*_{(b)}$	bootstrapped calculation of P using data set b
$Q^*_{(b)}$	bootstrapped calculation of Q using data set b
$W^*_{(b)}$	bootstrapped calculation of W using data set b
$\bar{x}^*_{(b)}$	bootstrapped sample mean of the input data in data set b
$\bar{y}^*_{(b)}$	bootstrapped sample mean of the output data in data set b
t	latent variable found from PLS
e	variance in input variable not considered by PLS
f	noise from the Prediction to the true value of PLS
w_l	direction of maximum covariance between X_l and Y
p_l	least squares prediction between the lth component of t and X_l
q_l	least squares prediction between the lth component of t and Y
X	matrix of input training data
Y	matrix of output training data
X_l	transformed input variables for the $j_t h$ component of PLS
$\widehat{C}_{a,b}$	MLE of the covariance matrix of vectors a and b
$\widehat{y}$	prediction of response variable from PLS given an input x
n_x	number of dimensions of the input variables
n_y	number of dimensions of the output variables
n_L	number of latent variables of PLS
N	number of training data points
$s(x)$	standard error of sample output variable
t_ν	student-t distribution with ν degrees of freedom
DS	Design Space
y_{des}	desired range of response variables for a manufacturing process
α	error acceptance rate for the Design Space
B	number of bootstrap replications
$\mathcal{N}(y;\mu,\sigma)$	probability density function for a normal distribution with mean μ and variance σ^2 evaluated at y
$\delta(\cdot)$	Dirac delta function

AUTHOR CONTRIBUTIONS

James Odgers: Conceptualization (equal); data curation (equal); formal analysis (equal); investigation (lead); methodology (equal); software (lead); validation (lead); visualization (lead); writing – original draft (equal); writing – review and editing (equal). **Chrysoula D. Kappatou:** Conceptualization (supporting); data curation (supporting); formal analysis (supporting); investigation (supporting); methodology (supporting); software (supporting); writing – original draft (equal); writing – review and editing (equal). **Ruth Misener:** Conceptualization (equal); formal analysis (equal); funding acquisition (equal); investigation (equal); methodology (equal); project administration (equal); resources (equal); supervision (equal); writing – original draft (equal); writing – review and editing (equal). **Salvador García Muñoz:** Conceptualization (equal); data curation (supporting); formal analysis (supporting); funding acquisition (equal); investigation (supporting); methodology (supporting); project administration (equal); resources (equal); supervision

(supporting); writing – original draft (equal); writing – review and editing (equal). **Sarah Filippi:** Conceptualization (equal); formal analysis (equal); funding acquisition (equal); investigation (equal); methodology (equal); project administration (equal); resources (equal); supervision (equal); writing – original draft (equal); writing – review and editing (equal).

ACKNOWLEDGMENTS

We would like to thank Kennedy Kusumo, who provided the code for producing the data used in the Michael Addition reaction,[37] and Shankar Vaidyaraman, who provided the data for the reductive amination example.

The authors gratefully acknowledge financial support from Eli Lilly and Company and the Engineering and Physical Sciences Research Council of the UK via Prosperity Partnership (grant number EP/T005556/1, EP/T518207/1).

DATA AVAILABILITY STATEMENT

Data subject to third party restrictions. The data that support the findings of this study are available from Eli Lilly. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the authors with the permission of Eli Lilly.

ORCID

Ruth Misener https://orcid.org/0000-0001-5612-5417
Salvador García Muñoz https://orcid.org/0000-0003-0067-9773
Sarah Filippi https://orcid.org/0000-0001-8652-358X

REFERENCES

1. Wold S, Sjöström M, Eriksson L. PLS-regression: a basic tool of chemometrics. *Chemom Intel Lab Syst*. 2001;58(2):109-130.
2. Geladi P, Kowalski BR. Partial least-squares regression: a tutorial. *Anal Chim Acta*. 1986;185:1-17.
3. Helland IS. On the structure of partial least squares regression. *Commun Stat Simul Comput*. 1988;17(2):581-607.
4. Gustafsson MG. A probabilistic derivation of the partial least-squares algorithm. *J Chem Inf Comput Sci*. 2001;41(2):288-294.
5. Denham MC. Prediction intervals in partial least squares. *J Chemometr*. 1997;11(1):39-52.
6. Höskuldsson A. PLS regression methods. *J Chemometr*. 1988;2(3):211-228.
7. Faber K, Kowalski BR. Propagation of measurement errors for the validation of predictions obtained by principal component regression and partial least squares. *J Chemometr*. 1997;11(3):181-238.
8. Fushiki T. Bootstrap prediction and Bayesian prediction under misspecified models. *Ther Ber*. 2005;11(4):747-758.
9. ICH Harmonized Tripartite Guideline. Pharmaceutical development Q8 (R2). International Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use 2009.
10. Bro R, Elden L. PLS works. *J Chemometr*. 2009;23(2):69-71.
11. Vinzi VE, Chin WW, Henseler J, Wang H. *Handbook of Partial Least Squares*. Springer; 2010.
12. De Jong S. SIMPLS: an alternative approach to partial least squares regression. *Chemom Intel Lab Syst*. 1993;18(3):251-263.
13. Nounou MN, Bakshi BR, Goel PK, Shen X. Process modeling by Bayesian latent variable regression. *AIChE J*. 2002;48(8):1775-1793.
14. Chen H, Bakshi BR, Goel PK. Bayesian latent variable regression via Gibbs sampling: methodology and practical aspects. *J Chemometr*. 2007;21(12):578-591.
15. Zheng J, Song Z, Ge Z. Probabilistic learning of partial least squares regression model: theory and industrial applications. *Chemom Intel Lab Syst*. 2016;158:80-90.
16. Li S, Gao J, Nyagilo JO, Dave DP. Probabilistic partial least square regression: a robust model for quantitative analysis of Raman spectroscopy data. 2011 IEEE International Conference on Bioinformatics and Biomedicine; 2011; 526-531; IEEE.
17. Bouhaddani S, Uh HW, Hayward C, Jongbloed G, Houwing-Duistermaat J. Probabilistic partial least squares model: Identifiability, estimation and application. *J Multivar Anal*. 2018;167:331-346.
18. Castillo E, Reis MS. Bayesian predictive optimization of multiple and profile response systems in the process industry: a review and extensions. *Chemom Intel Lab Syst*. 2020;206:104121.
19. Phatak A, Reilly P, Penlidis A. An approach to interval estimation in partial least squares regression. *Anal Chim Acta*. 1993;277(2):495-501.
20. Faber K, Kowalski BR. Prediction error in least squares regression: further critique on the deviation used in the Unscrambler. *Chemom Intel Lab Syst*. 1996;34(2):283-292.
21. Faber K. Uncertainty estimation for multivariate regression coefficients. *Chemom Intel Lab Syst*. 2002;64(2):169-179.
22. Olivieri AC, Faber NM, Ferré J, Boqué R, Kalivas JH, Mark H. Uncertainty estimation and figures of merit for multivariate calibration (IUPAC Technical Report). *Pure Appl Chem*. 2006;78(3):633-661.
23. Zhang L, García-Muñoz S. A comparison of different methods to estimate prediction uncertainty using partial least squares (PLS): a practitioner's perspective. *Chemom Intel Lab Syst*. 2009;97:152-158.
24. Elden L. Partial least-squares vs. Lanczos bidiagonalization—I: analysis of a projection method for multiple regression. *Comput Stat Data Anal*. 2004;46(1):11-31.
25. Tibshirani RJ, Efron B. An introduction to the bootstrap. *Monogr Stat Appl Probab*. 1993;57:1-436.
26. Reis MS, Saraiva PM. Prediction of profiles in the process industries. *Ind Eng Chem Res*. 2012;51(11):4254-4266.
27. Fushiki T, Komaki F, Aihara K. Nonparametric bootstrap prediction. *Ther Ber*. 2005;11(2):293-307.
28. Efron B, Gong G. A leisurely look at the bootstrap, the jackknife, and cross-validation. *Am Stat*. 1983;37(1):36-48.
29. Facco P, Dal Pastro F, Meneghetti N, Bezzo F, Barolo M. Bracketing the design space within the knowledge space in pharmaceutical product development. *Ind Eng Chem Res*. 2015;54(18):5128-5138.
30. Bano G, Facco P, Meneghetti N, Bezzo F, Barolo M. Uncertainty back-propagation in PLS model inversion for design space determination in pharmaceutical product development. *Comput Chem Eng*. 2017;101:110-124.
31. Bano G, Facco P, Bezzo F, Barolo M. Probabilistic design space determination in pharmaceutical product development: a Bayesian/latent variable approach. *AIChE J*. 2018;64(7):2438-2449.
32. Palací-López D, Facco P, Barolo M, Ferrer A. New tools for the design and manufacturing of new products based on latent variable model inversion. *Chemom Intel Lab Syst*. 2019;194:103848.
33. Peterson JJ. A Bayesian approach to the ICH Q8 definition of design space. *J Biopharm Stat*. 2008;18(5):959-975.
34. Ochoa MP, García-Muñoz S, Stamatis S, Grossmann IE. Novel flexibility index formulations for the selection of the operating range within a design space. *Comput Chem Eng*. 2021;149:107284.
35. Laky D, Xu S, Rodriguez JS, Vaidyaraman S, García Muñoz S, Laird C. An optimization-based framework to define the probabilistic design space of pharmaceutical processes with model uncertainty. *Processes*. 2019;7(2):96.
36. Lebrun P, Boulanger B, Debrus B, Lambert P, Hubert P. A Bayesian design space for analytical methods based on multivariate models and predictions. *J Biopharm Stat*. 2013;23(6):1330-1351.

37. Kusumo KP, Gomoescu L, Paulen R, et al. Bayesian approach to probabilistic design space characterization: a nested sampling strategy. *Ind Eng Chem Res*. 2019;59(6):2396-2408.
38. Jaeckle CM, MacGregor JF. Industrial applications of product design through the inversion of latent variable models. *Chemom Intel Lab Syst*. 2000;50(2):199-210.
39. Reis MS, Pereira AC, Leça JM, Rodrigues PM, Marques JC. Multiresponse and multiobjective latent variable optimization of modern analytical instrumentation for the quantification of chemically related families of compounds: case study—solid-phase microextraction (SPME) applied to the quantification of analytes with impact on wine aroma. *J Chemometr*. 2019;33(3):3103.
40. Vemavarapu C, Badawy SI. Chapter 11—Role of drug substance material properties in the processibility and performance of wet granulated products. In: Narang AS, Badawy SI, eds. *Handbook of Pharmaceutical Wet Granulation*. Elsevier; 2019:387-419.
41. Hetrick EM, Shi Z, Barnes LE, et al. Development of near infrared spectroscopy-based process monitoring methodology for pharmaceutical continuous manufacturing using an offline calibration approach. *Anal Chem*. 2017;89(17):9175-9183.
42. García-Muñoz S, Hernandez TE. Supervised extended iterative optimization technology for the estimation of powder compositions in pharmaceutical applications: method and lifecycle management. *Ind Eng Chem Res*. 2020;59(21):10072-10081.

How to cite this article: Odgers J, Kappatou C, Misener R, García Muñoz S, Filippi S. Probabilistic predictions for partial least squares using bootstrap. *AIChE J*. 2023;e18071. doi:10.1002/aic.18071

APPENDIX A: NIPALS ALGORITHM

The description of PLS is often not given by a single objective function, but rather by an algorithm. Presented below is a description of the PLS algorithm adapted from the work of Gusstafsson (2001),[4] which frames the algorithm in a probabilistic way.

For a PLS model with n_l latent variables and N data points

1. Estimate the mean of both x and y, using

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{A1}$$

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{A2}$$

where x_i and y_i are the data points in the training set.

2. Set $l = 1$

$$x_{i,l} = x_i - \overline{x} \tag{A3}$$

$$y_{i,l} = y_i - \overline{y}. \tag{A4}$$

3. Calculate $w_l \in \mathbb{R}^{n_x}$, which is the component in x_l that is believed to have the maximum covariance with y. This is equivalent to finding the first left singular vector of the estimated covariance matrix. This can be solved using the following maximization problem:

$$max_{w_l} \quad w_l^T \widehat{C}_{x_l y} \widehat{C}_{x_l y}^T w_l \quad s.t. \quad \left\| w_l \right\| = 1, \tag{A5}$$

where the notation $\widehat{C}_{ab} \in \mathbb{R}^{dim(a)\times dim(b)}$ denotes the MLE of the covariance matrix between variables a and b, with components given by

$$c_{j,k} = \frac{1}{N}\sum_{i=1}^{N} x_{i,j} y_{i,k}, \tag{A6}$$

where $x_{i,j}$ and $y_{i,k}$ are the input component j and response component k for the ith sample.

4. Calculate the latent variable $t_{i,l}$

$$t_{i,l} = w_l^T x_{i,l}. \tag{A7}$$

5. Calculate the ordinary least squares regressors ($p \in \mathbb{R}^{1\times n_x}$ and $q \in \mathbb{R}^{1\times n_y}$) for $X_l = [x_1, ..., x_N]^T$ and $Y = [y_1, ..., y_N]$ given the latent variable $t_l = [t_{1,l}, ..., t_{N,l}]$ using

$$p_l = \frac{t_l^T X_l}{t_l^T t_l} \tag{A8}$$

$$q_l = \frac{t_l^T Y}{t_l^T t_l}, \tag{A9}$$

where $T_l \in \mathbb{R}^{N\times 1}$ is a matrix of the lth latent variable of the training data.

6. If $l < n_L$ then transform the input and response variables using

$$\widetilde{x}_{l+1} = \widetilde{x}_l - p_l t_l^T \tag{A10}$$

$$\widetilde{y}_{l+1} = \widetilde{y}_l - q_l t_l^T, \tag{A11}$$

set $l = l + 1$ and go to Step 3. This step is called deflation.

7. Generate $W \in \mathbb{R}^{n_x \times n_l}$, $P \in \mathbb{R}^{n_x \times n_l}$, and $Q \in \mathbb{R}^{n_y \times n_l}$ by concatenating t_l, p_l, and q_l

$$W = [w_1, w_2 ..., w_{n_L}] \qquad P = \left[p_1^T, p_2^T ..., p_{n_L}^T\right]^T \qquad Q = \left[q_1^T, q_2^T ..., q_{n_L}^T\right]^T. \tag{A12}$$

APPENDIX B: PREDICTING A RESPONSE GIVEN AN INPUT VALUE AND PLS PARAMETERS

PLS is able to predict the response (y) from a new input (x). Here the estimate for y is denoted using $\widehat{y}(x; W, P, Q)$, and is calculated in the following way:

1. Set $l = 1$ and $x_1 = x - \overline{x}$.
2. Calculate the scalar t_l using

$$t_l = w_l^T x_l. \tag{B1}$$

3. If $i < n_L$ transform the input variable using

$$x_{l+1} = x_l - t_l p_l^T, \tag{B2}$$

set $l = l + 1$ and return to Step 2.

4. Calculate the value for y using

$$\widehat{y}(x; W, P, Q) = \overline{y} + Qt, \tag{B3}$$

where $t = [t_1, t_2, ... t_{n_L}]^T$.

APPENDIX C: STUDENT-T DERIVATION

For a given input x, the studentized residuals, denoted here by τ, is defined as

$$\tau = \frac{y - \hat{y}(x)}{s(x)} \tag{C1}$$

and follows a student-t distribution with ν degrees of freedom. Its probability density function is given by

$$p(\tau) = \mathcal{T}(\tau; \nu). \tag{C2}$$

The change of variables formula for probability distributions is

$$p(y) = p(\tau)\left|\frac{d\tau}{dy}\right|. \tag{C3}$$

Substituting in the result that

$$\frac{d\tau}{dy} = \frac{1}{s(x)}, \tag{C4}$$

along with the identities in Equation (C1) and Equation (C2) gives the result in Equation (11).

APPENDIX D: DEFINING THE KNOWLEDGE SPACE

The PLS model is only considered valid in the same region as the training data were from. The valid region is found by assuming training data are drawn from a multivariate normal distribution and finding its confidence limits.

There are two statistics used for this: the T^2 statistic and the Squared Prediction Error (*SPE*) statistic. In order for a point to be considered to have been drawn from the same region the *SPE* and the T^2 statistic need to both be below predefined thresholds.

The T^2 statistic for an input x^* is given by

$$T_i^2 = \sum_{j=1}^{n_l} \frac{t_j^{*2}}{\lambda_j}, \tag{D1}$$

where n_l is the number of latent variables, t_j^* is the value of the data point x^* along the jth latent variable and λ_j is the variance along the jth latent variable.

The value of the squared prediction error (*SPE*) for a data point x^* is given by

$$SPE = e^{*T} e^*, \tag{D2}$$

where e^* is the portion of x^* that was discarded by the PLS model.

The upper confidence limit for T^2 is given by

$$UCL\left(T^2\right)_\alpha = \frac{n_l\left(N^2 - 1\right)}{N(N - n_l)} F_{(n_l,(N-n_l)),\alpha}, \tag{D3}$$

where N is the number of samples in the training data, α is a desired confidence level, and $F_{(n_l,(N-n_l)),\alpha}$ is the α quantile of F distribution with n_l and $N - n_l$ degrees of freedom.

The upper confidence limit for SPE is given by

$$UCL(SPE)_\alpha = \frac{v}{2b}\chi^2_{\left(2b^2/v\right),\alpha}, \tag{D4}$$

where b is the mean of the training points, v is the variance, $\chi^2_{\left(2b^2/v\right),\alpha}$ is the α quantile of a χ^2 distribution with $2b^2/v$ degrees of freedom.