Comparison of Partial Least Squares Regression, Least Squares Support Vector Machines and Gaussian Process Regression for a Near Infrared Calibration

Chenhao Cui and Tom Fearn

Department of Statistical Science,
University College London,London, WC1E 6BT, U.K.

Email:chenhao.cui.14@ucl.ac.uk; Tel:+44 747 838 3032

1 Abstract

This paper investigates the use of least squares support vector machines(LS-SVMs) and Gaussian process regression(GPR) for multivariate spectroscopic calibration. The performances of these two non-linear regression models are assessed and compared to the traditional linear regression model, partial least squares regression(PLSR) on an agricultural example. The non-linear models, LS-SVMs and GPR, showed enhanced generalization ability, especially in maintaining homogeneous prediction accuracy over the range. The two non-linear models generally have similar prediction performance, but showed different features in some situations, especially when the size of the training set varies. This is due to fundamental differences in fitting criteria between these models.

Keywords: Near-Infrared spectroscopy; Multivariate calibration; Partial least squares regression; Least squares support vector machines; Gaussian process regression.

2 Background

The problem of interest is the prediction of the concentration of a chemical constituent in a sample from its near infrared(NIR) spectrum. A set of calibration samples is used to learn the quantitative relationship between the constituent concentration and NIR absorption spectrum. In general case, spectral data sets are high dimensional and require dimensional reduction. Traditional linear regression methods(including Principal component regression(PCR) and partial least squares regression(PLSR)) first transfer raw spectral data into latent variables to reduce dimensionality of regressors, and then directly implement multiple linear regression on latent variables. These methods perform reasonably well in most practical applications. However, in some cases, e.g.when predictions are required over a large range so that nonlinearity occurs, linear models fail to keep their accuracy across the range. Nonlinear regression models are then relevant. Popular nonlinear models include support vector machines(SVMs), artificial neural networks(ANN) and Gaussian process regression(GPR). Previous studies have shown that Support

Vector Machines [1] and Gaussian process regression [2] can be applied to multivariate spectroscopic calibration. Numerous feasibility studies have proven SVMs is a very powerful tool on spectroscopic analysis, such as identification of teas [3], quantification of milk powder [4], classification of colonic tissues [5] and quantitative calibration on tobacco [6]. GPR is a relatively new method for spectroscopic calibration, and previous studies have shown its superiority on modelling instrumental systematics [7], imaging spectroscopic analysis [8] and variable selection for calibration [9]. A comparison research also indicated that SVMs outperformed PLS on some validation tests [10]. In this article, least squares Support Vector Machines(LS-SVMs) and Gaussian process regression(GPR) are further studied. Their performances on a NIR spectroscopic calibration are compared to the traditional linear regression method, PLSR, on several aspects, including global error, local prediction behavior and learning efficiency.

3 Theory

All of the models mentioned here are all based on learning a prediction rule on an observed dataset, which is also called calibration set. However, these models have different learning and prediction strategies. They can be divided into two categories according to the way they make predictions: one, called variable space based prediction, calculates the correlation coefficients between each of the variables and the target analyte; the other, called sample space based prediction, ignores individual variable but focuses on affinity and dissimilarity between the samples. PLSR is the first type and the non-linear methods investigated here, LS-SVMs and GPR, are the second type.

3.1 PLSR

PLSR is a predictive two-block regression method based on estimated latent variables, initially developed by Svante Wold [11] and first applied to NIR data by Martens and Jensen [12]. The purpose of PLSR is to predict the property of interest y from observed NIR spectra $\mathbf{x} = (x_k, k = 1, 2 \ldots, K)$. This is achieved by a linear predictor:

$$\hat{y} = b_0 + \sum_{k=1}^{K} b_k x_k \tag{1}$$

where b_0 is the offset and b_k are regression coefficients calculated by the PLSR algorithm. Notice that for each observed spectrum $\mathbf{x}$ the dimensionality K can be very high (typically hundreds of wavelengths in NIR spectra), and a regression model on all variables will be extremely complicated and noisy, so raw spectral data sets usually need dimension reduction before being regressed. As a result, multiple linear regression is actually performed on a set of PLS factors. These factors are calculated and taken out from the data set sequentially [13]. Considering a training data set with I observations $\mathbf{y} = (y_i, i = 1, 2, \ldots, I)^T$, the corresponding spectra $\mathbf{X} = (x_{ik}, i = 1, 2, \ldots, I; k = 1, 2, \ldots, K)$ are taken as sum of several orthogonalized PLS factors, indexed by $a = 1, 2, \ldots, A$. $\mathbf{X}$ and $\mathbf{y}$ are first mean centred, producing $\mathbf{X}_0$ and $\mathbf{y}_0$:

$$\begin{aligned} \mathbf{X}_0 &= \mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^T \\ \mathbf{y}_0 &= \mathbf{y} - \mathbf{1}\bar{y} \end{aligned} \tag{2}$$

The residual on $\mathbf{X}$ and $\mathbf{y}$ after elimination of the a^{th} factor are termed $\mathbf{X}_a$ and $\mathbf{y}_a$. The loading weight vector for the a^{th} PLS factor $\mathbf{w}_a$ can be calculated by making use of the variability in $\mathbf{y}_{a-1}$, formulated as:

$$\mathbf{X}_{a-1} = \mathbf{y}_{a-1}\mathbf{w}_a^T + \mathbf{X}_a \tag{3}$$

under the condition that $\mathbf{w}_a^T\mathbf{w}_a = 1$. Minimize $\mathbf{X}_a$ w.r.t $\mathbf{w}_a$, the solution is given by:

$$\hat{\mathbf{w}}_a = c\mathbf{X}_{a-1}^T\mathbf{y}_{a-1} \tag{4}$$

Correspondingly, the PLS score $\mathbf{t}_a$ for the a^{th} factor can be determined as the projection of $\mathbf{X}_{a-1}$ on $\hat{\mathbf{w}}_a$:

$$\hat{\mathbf{t}}_\mathbf{a} = \mathbf{X}_{a-1}\hat{\mathbf{w}}_a \tag{5}$$

Loading vectors $\mathbf{p}_a$ and q_a can be estimated by regressing $\mathbf{X}_{a-1}$ and $\mathbf{y}_{a-1}$ on $\hat{\mathbf{t}}_a$:

$$\begin{aligned} \mathbf{X}_{a-1} &= \hat{\mathbf{t}}_a\mathbf{p}_a^T + \mathbf{X}_a \\ \mathbf{y}_{a-1} &= \hat{\mathbf{t}}_a q_a + \mathbf{y}_a \end{aligned} \tag{6}$$

Finally, the regressors described in Eq.1 can be defined by:

$$\hat{\mathbf{b}} = \hat{\mathbf{W}}(\hat{\mathbf{P}}^T\hat{\mathbf{W}})^{-1}\hat{\mathbf{q}} \tag{7}$$

and the offset $\hat{b}_0$ is given by:

$$\hat{b}_0 = \bar{y} - \bar{\mathbf{x}}^T\hat{\mathbf{b}} \tag{8}$$

3.2 LS-SVMs

PLSR is normally considered as one of the most powerful high-dimensional linear regression methods. However, the performance of PLSR is sometimes limited due to a non-linear relationship between the input variables and the response. In order to extend the regression method to a non-linear framework, new variables representing non-linearity need to be introduced. This can be achieved by mapping the input data into a higher dimensional feature space through a transformation $\phi(.) : \mathbb{R}^k \rightarrow \mathbb{R}^{k_h}$. This mapping process is done by a linear basis expansion in the input variables, including non-linear transformations and combinations of the basis variables. The introduced non-linear feature space potentially has infinite dimension. Function estimation $\hat{y}(\mathbf{x})$ on an observed sample $\mathbf{x}$ is then performed as follows:

$$\hat{y}(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b \tag{9}$$

where b is the offset term and $\mathbf{w}$ is a vector of regression coefficients in the non-linear feature space. Such a coefficient vector also has potentially infinite dimensions.. Given a training set with known responses $\mathbf{y} = (y_i, i = 1, 2, \ldots, I)^T$ and corresponding observations $\mathbf{X} = (x_{ik}, i = 1, 2, \ldots, I; k = 1, 2, \ldots, K)$, one can easily construct a ridge regression cost function in feature space:

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma\frac{1}{2}\sum_{i=1}^{I}(\mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i)^2 \tag{10}$$

where γ controls the regularization procedure, and is related to the signal to noise ratio in the measurements. The bigger the value of γ, the higher the signal to noise ratio, and the less the necessary shrinkage in the coefficients.

However, this problem is not solvable due to potentially infinite dimensionality. Instead we construct the Lagrange dual function [14]:

$$\mathcal{D}(\mathbf{w}, b, e; \boldsymbol{\alpha}) = \mathcal{L} - \sum_{i=1}^{I} \alpha_i \mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i - y_i \tag{11}$$

with Karush-Kuhn-Tucker conditions [15]:

$$\begin{aligned}
\frac{\partial \mathcal{D}}{\partial \mathbf{w}} &= 0 \rightarrow \mathbf{w} = \sum_{i=1}^{I} \alpha_i \phi(\mathbf{x}_k) \\
\frac{\partial \mathcal{D}}{\partial b} &= 0 \rightarrow \sum_{i=1}^{I} \alpha_i = 0 \\
\frac{\partial \mathcal{D}}{\partial e} &= 0 \rightarrow \alpha_i = \gamma e_i \\
\frac{\partial \mathcal{D}}{\partial \alpha_i} &= 0 \rightarrow \mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i - y_i = 0
\end{aligned} \tag{12}$$

Substituting $\mathbf{w}$ and e, the solution on $\boldsymbol{\alpha}$ and b can be expressed as:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \boldsymbol{\Omega} + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \tag{13}$$

where Ω is the kernel function:

$$\begin{aligned}
\Omega_{nm} = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) = \mathcal{K}(\mathbf{x}_n, \mathbf{x}_m) \qquad n, m = 1, 2, \ldots, I. \\
\boldsymbol{\Omega} = \mathcal{K}(\mathbf{X}, \mathbf{X})
\end{aligned} \tag{14}$$

In the following experiment only the radial basis function kernel is considered, formulated as:

$$\mathcal{K}(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\frac{\| \mathbf{x}_n - \mathbf{x}_m \|_2^2}{l^2}\right) \tag{15}$$

The prediction for a new observation $\mathbf{x}_{I+1}$ can be expressed as:

$$\begin{aligned}
\hat{y}(\mathbf{x}_{I+1}) &= \sum_{i=1}^{I} \alpha_i \phi(\mathbf{x}_{I+1}) \phi(\mathbf{x}_i) + b \\
&= \sum_{i=1}^{I} \alpha_i \mathcal{K}(\mathbf{x}_{I+1}, \mathbf{x}_i) + b
\end{aligned} \tag{16}$$

Substituting α with solution in Eq.13, one can get following equation:

$$\hat{y}(\mathbf{x_{I+1}}) = \mathcal{K}(\mathbf{x}_{I+1}, \mathbf{X}) \left[\mathcal{K}(\mathbf{X}, \mathbf{X}) + \frac{\mathbf{I}}{\gamma}\right]^{-1} [\mathbf{y} - b \cdot \mathbf{1}] + b \tag{17}$$

Notice that only two free parameters (γ, l) need to be tuned in the final target formulation. γ is influenced by the signal to noise ratio, and l, which is sometimes called length-scale in the input space, represents how sensitive $\mathbf{y}$ is to changes in $\mathbf{X}$. When l is small, $\mathbf{y}$ is very sensitive to variances in the input variables and thus can potentially produce precise predictions, but with a huge risk of over-fitting. When l is large, $\mathbf{y}$ changes slowly, so predictions are less precise but more robust. These two free parameters can be tuned through leave-one-out cross-validation(LOO-CV) in training set to minimize the target loss function, for example, squared error loss as utilized here.

3.3 GPR

A Gaussian process is a collection of random variables such that any finite subset exhibits a joint Gaussian distribution [16]. First consider that an observation y_i in the calibration set $\mathbf{y} = (y_i, i = 1, 2, \ldots, I)$ consists of a true underlying function f_i, which is also called the latent function, and an additional Gaussian white noise ϵ:

$$y_i = f_i + \epsilon \tag{18}$$

Assume this Gaussian white noise has zero mean and precision of γ, then the probability distribution of the observations given the latent function can be described by a Gaussian likelihood function:

$$P(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \frac{1}{\gamma}\mathbf{I}) \tag{19}$$

There are many other possible likelihood functions for complicated relationship between latent function and observations, depending on the type of the noise, but in the following study only Gaussian likelihood is considered. The latent function described above is the target Gaussian process(GP). Similar to mean and variance in a Gaussian distribution, a Gaussian process is fully defined by its mean function $\mathcal{M}$ and covariance function $\mathcal{K}_{GP}$. Given a set of input variables $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_I$, the latent function variable $\mathbf{f} = f_1, f_2, ...f_I$ has a joint Gaussian distribution:

$$P(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathcal{M}, \mathcal{K}_{GP}(\mathbf{X}, \mathbf{X})) \tag{20}$$

where $\mathcal{M}$ is the mean vector with the size of I and $\mathcal{K}_{GP}$ is a $I \times I$ covariance matrix. Again, the RBF kernel function can be implemented here, as formulated in Eq.15, but with an additional parameter σ_s^2 to determine the signal level. The $(n, m)^{th}$ element of $\mathcal{K}_{GP}(\mathbf{X}, \mathbf{X})$ is formulated as:

$$\mathcal{K}_{GP}(\mathbf{x}_n, \mathbf{x}_m) = \sigma_s^2 \mathcal{K}(\mathbf{x}_n, \mathbf{x}_m) = \sigma_s^2 \exp\left(-\frac{\| \mathbf{x}_n - \mathbf{x}_m \|_2^2}{l^2}\right) \tag{21}$$

Normally, for notational and computational convenience, the mean function is set to $\mathbf{0}$, and this is appropriate if the response is mean centered. So Eq.20 can be modified as by:

$$P(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathcal{K}_{GP}(\mathbf{X}, \mathbf{X})) \tag{22}$$

Multiplying Eq. 22 with the likelihood function in Eq.19, and integrating out $\mathbf{f}$:

$$P(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \frac{1}{\gamma}\mathbf{I}) \tag{23}$$

Similarly to the other prediction models, the main interest is making predictions on new measurements based on the information from a training (calibration) data set $\mathbf{X} = (\mathbf{x}_i, i = 1, 2, \ldots, I)$ and $\mathbf{y} = (y_i, i = 1, 2, \ldots, I)$. Gaussian process regression assumes that the observation $\mathbf{y}$ has a joint Gaussian distribution with mean of $\mathbf{0}$ (after mean centered) and a covariance function given by $\mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \mathbf{I}/\gamma$. One can easily write down the marginal likelihood:

$$\begin{aligned} P(\mathbf{y}|\mathbf{0}, \mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \mathbf{I}/\gamma) =& (2\pi)^{-I/2} |\mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \mathbf{I}/\gamma|^{-1/2} \\ & \times \exp(-\frac{1}{2}\mathbf{y}^T (\mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \mathbf{I}/\gamma)^{-1}\mathbf{y}) \end{aligned} \tag{24}$$

The three free parameters γ, σ_s^2 and l in Eq.23 can be tuned to maximize this marginal likelihood on the training set. This technique is sometimes called evidence maximization. Different from cross-validation in PLSR and LS-SVMs, evidence maximization uses the whole training set at one time, without dividing it into calibration and validation groups. More importantly, the optimization target is not to reduce the squared error loss function ,which only depends on the point prediction. The likelihood is determined by both the mean and the covariance of posterior distribution. This also explains why an additional parameter σ_s^2 is needed in the GPR framework.

For prediction of a new observation $\mathbf{x}_{I+1}$, we construct a set of $I+1$ latent variables $f_1, \ldots, f_I, f_{I+1}$ with a joint zero mean Gaussian distribution and predefined covariance function:

$$P(f_1, f_2, \ldots, f_I, f_{I+1}) = \mathcal{N}\left(\mathbf{0}, \begin{matrix} \mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) & \mathcal{K}_{GP}(\mathbf{X}, \mathbf{x}_{I+1}) \\ \mathcal{K}_{GP}(\mathbf{x}_{I+1}, \mathbf{X}) & \sigma_s^2 \end{matrix}\right) \tag{25}$$

Taking into account the Gaussian noise ϵ in $\mathbf{y}$, the joint Gaussian distribution of $\mathbf{y}$ and f_{I+1} can be expressed as:

$$P(\mathbf{y}, f_{I+1}) = \mathcal{N}\left(\mathbf{0}, \begin{matrix} \mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \boldsymbol{I}/\gamma & \mathcal{K}_{GP}(\mathbf{X}, \mathbf{x}_{I+1}) \\ \mathcal{K}_{GP}(\mathbf{x}_{I+1}, \mathbf{X}) & \sigma_s^2 \end{matrix}\right) \tag{26}$$

The conditional probability $f_{I+1}|\mathbf{X}, \mathbf{y}, \mathbf{x}_{I+1}$ has the expectation and variance of :

$$\begin{aligned} mean(\hat{f}_{I+1}) &= \mathcal{K}_{GP}(\mathbf{x}_{I+1}, \mathbf{X})[\mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \boldsymbol{I}/\gamma]^{-1}\mathbf{y} \\ var(\hat{f}_{I+1}) &= \sigma_s^2 - \mathcal{K}_{GP}(\mathbf{x}_{I+1}, \mathbf{X})[\mathcal{K}_{GP}(\mathbf{X}, \mathbf{X}) + \boldsymbol{I}/\gamma]^{-1}\mathcal{K}_{GP}(\mathbf{X}, \mathbf{x}_{I+1}) \end{aligned} \tag{27}$$

Compared with LS-SVMs, GPR not only gives a prediction mean for the new observation, but also provides a full probabilistic posterior distribution. This additional information is sometimes very useful, for example for outlier detection. However, since interpreting variance on predictions requires one more degree of freedom, it also increases the difficulty of tuning the hyper-parameters, especially when the training set is a small one.

4 Experiment

Three high-dimensional regression models(PLSR, LS-SVMs and GPR) were applied to the same NIR spectral data set. Their performances in prediction were presented and compared. All of the calculations were performed in Matlab R2014b (The MathWorks Inc, MA, USA). PLSR was achieved with the Statistics and Machine Learning Toolbox; LS-SVM was realized with LS-SVMlab package [17] and GPR was calculated with GPML package [18].

4.1 Sample composition

The data set consists of NIR spectra and corresponding target constituent concentration (nitrogen) from 1240 agricultural seeds. Images were recorded using a MCT hyper-spectral camera and then post processing was used to average the spectral points over the seeds to produce a single spectrum per seed. For each spectrum there are 239 valid wavelengths in the NIR range, spanning from 993.0 nm to 2488.4 nm. The constituent content was then measured using a combustion-based chemical analysis.

4.2 Preprocessing

Several preprocessing methods were evaluated by cross-validation using the PLSR model. The combination of SNV + Savitzky-Golay derivative was implemented. Raw spectra were first preprocessed by Standard Normal Variate (SNV), to remove constant offset and scatter effect by centering and rescaling individual spectrum, and then transformed with Savitzky-Golay 2 side points, 2^{nd} polynomial order fitting first derivative.

4.3 Randomization

The 1240 spectra were first randomly divided into two groups: group A(with 360 spectra) as calibration set and group B(with 880 spectra) as validation set. Fig.1 shows the density distribution, for nitrogen of one pair of randomized calibration and validation sets. All of the three regression methods were trained on the same calibration set first. Obtained models were then applied to predict nitrogen concentration in the validation set according to their spectra, and then compared to their true responses. This randomized test was performed 5 times in total. In each single test, the validation set was carefully kept away from optimizing the model so error on the validation set can be seen as a good indicator of generalized prediction performance.

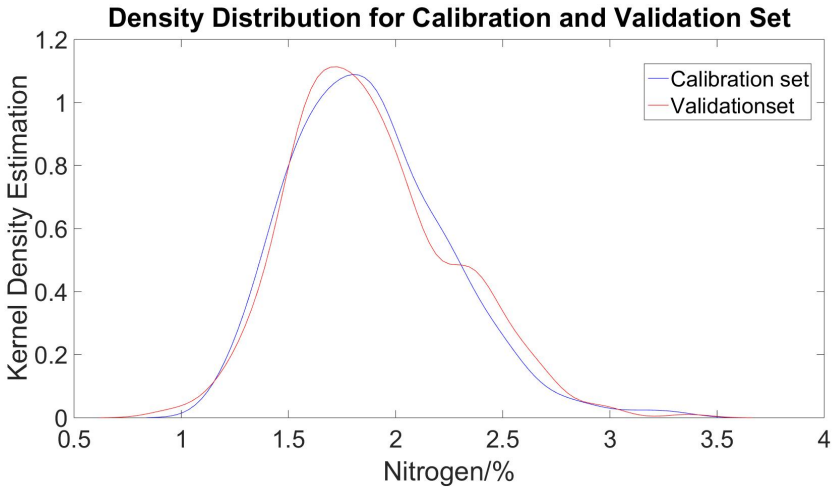


Figure 1: Population distribution for nitrogen in the calibration set and validation set for one random split.

The three regression models were trained on calibration set as follows.

4.4 PLSR

As described in section 3.1, there is only one free parameter in the PLSR model: the number of PLS factors used for prediction. Although higher order factors contain more detailed features from the variance of regressors, this is not necessarily correlated with the response(nitrogen concentration). Using higher order components will inevitably lead to a more complicated model, which may generalize less well. Cross-validation is very useful for choosing a reasonable highest order of PLS factor. An internal 10-fold cross-validation was performed on the PLSR model with different numbers of factors. MSECV(Mean Square Error of Cross-Validation) was taken as an indicator of the quality of the tested model.

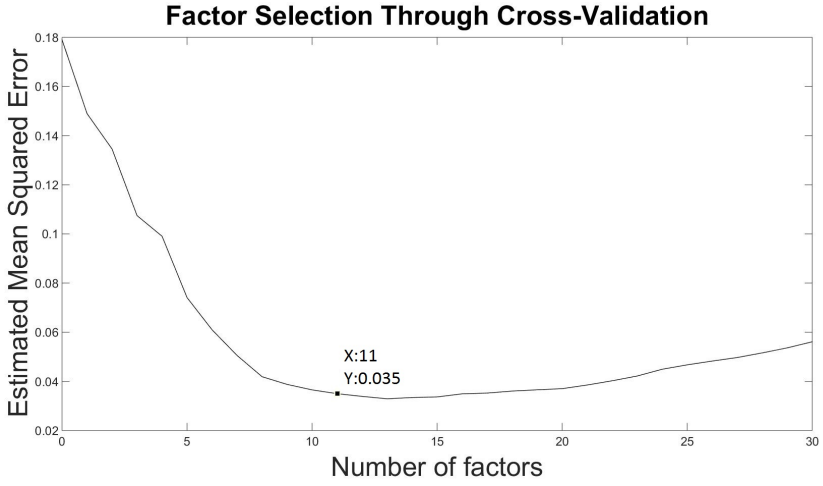


Figure 2: MSECV (% nitrogen) for PLSR model when using different number of components

Fig.2 shows RMSECV values for one randomized calibration set. This parameter selection process was separate from the test set, and was solely used for choosing the number of PLS factors. According to the result, cross-validation suggests using a PLSR model with 11 components. Notice in Figure 2 that 11 and 12 components had very close MSECV values (0.035% and 0.034% respectively) and 11 was adopted, because when two PLSR models have comparable performances, the simpler one is preferred.

4.5 LS-SVMs

For the non-linear regression model LS-SVMs, there are two free hyper-parameters: γ related to the strength of regularization term, and l, determining the length scale of the adopted RBF kernel. Both of these two free parameters need to be tuned through the cross-validation process. Initial values of γ and l were decided by coupled simulated annealing (CSA), a global optimization method composed of several distributed simulated annealing processes with coupled acceptance probability [19], to avoid very bad local minima. The two parameters were then tuned to minimize the squared error loss function $\underset{\gamma,l}{L} = \sum_{i=1}^{I}(\hat{y}_i - y_i)^2$ with 10-fold cross-validation.

4.6 GPR

A Gaussian process is fully defined by a mean function, covariance function and a likelihood function. The options for these three functions in the study are described as follows:

Mean Function in order to simplify the computational work, before Gaussian process regression was carried out all the response was mean centered to 0 by taking out the offset. As a result, the mean function was set to $\mathbf{0}$.

Covariance Function radial basis function(with magnitude) was used as the covariance function in GPR, as described in Eq.21. There are two hyper-parameters in this covariance function: isotropic length scale l and signal magnitude σ_s^2.

Likelihood function Assuming noise on observations has a Gaussian distribution, the Gaussian likelihood function was adopted here. Noise magnitude was specified by $1/\gamma$.

There are three free parameters σ_s^2, l and γ in total. All of them were optimized through evidence maximization using the training set, as described in section 3.3. Notice that no initialization (like CSA in LS-SVMs) on the parameters was done before making inference on hyper-parameters, so potentially there was a very high risk of falling into multiple bad local optima. However, experimental results indicated that local optima were not a severe problem here, especially for large training set cases, because usually there is only one local optimum with a significantly higher marginal likelihood so it will be easy to reject all the other models.

5 Result

5.1 Comparing global RMSEV

As discussed in the previous section, the 1240 data points were randomly divided into two sets: calibration set with 360 samples and validation set with 880 samples. The calibration set and validation set were fully randomized and tested 5 times for each model. Since the predictions from all of the models at each test were correlated because the validation set was a common one, errors in the laboratory measurements for the 880 samples contributed to all sets of prediction errors [20]. T-test for paired samples can be utilized to calculate the true differences on biases and true ratio of root mean square error for validation(RMSEV).
Careful comparisons on biases and RMSEV between the 3 models were performed on 5 separate randomized tests. Results presented in Table 1 shows RMSEV and biases for three models on each randomized validation set. Table 2 presents the 95% confidence intervals for the true difference in biases. Table 3 gives the 95% confidence intervals for the ratio of the true RMSEV.

Index	PLS		LS-SVM		GPR	
Index	RMSEV	Bias	RMSEV	Bias	RMSEV	Bias
1	0.165	0.013	0.134	0.004	0.131	0.004
2	0.166	0.005	0.133	0.002	0.131	0.004
3	0.162	0.01	0.131	0.016	0.123	0.007
4	0.164	0.006	0.127	0.001	0.125	0.006
5	0.168	-0.01	0.126	-0.004	0.125	-0.002
Average	0.165	0.005	0.130	0.004	0.127	0.004

Table 1: RMSEV and Biases (%) for PLSR, LS-SVMs and GPR on each randomized validation set

Index	PLS - LS-SVMs	PLS - GPR	LS-SVMs - GPR
1	(0.002,0.017)	(-0.002,0.036)	(-0.011,0.025)
2	(-0.003,0.024)	(-0.007,0.04)	(-0.013,0.024)
3	(-0.014,0.002)	(-0.014,0.036)	(0.006,0.040)
4	(-0.003,0.012)	(-0.007,0.03)	(-0.01,0.025)
5	(-0.013,0.002)	(-0.031,0.008)	(-0.023,0.011)
Average	(-0.006,0.011)	(-0.012,0.030)	(-0.010,0.025)

Table 2: 95% confidence interval for the true difference in biases (% nitrogen)

Index	PLS / LS-SVMs	PLS / GPR	LS-SVMs / GPR
1	(1.171,1.283)	(1.198,1.311)	(0.994,1.045)
2	(1.176,1.302)	(1.202,1.324)	(0.996,1.045)
3	(1.190,1.308)	(1.266,1.380)	(1.021,1.089)
4	(1.236,1.352)	(1.253,1.374)	(0.990,1.039)
5	(1.274,1.392)	(1.294,1.404)	(0.991,1.033)
Average	(1.209,1.327)	(1.243,1.359)	(0.998,1.050)

Table 3: 95% confidence interval for the ratio of true RMSEV(% nitrogen)

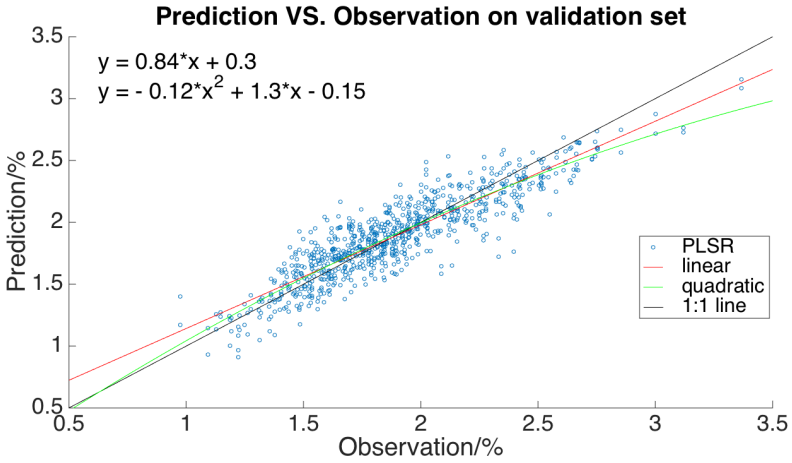
Several important conclusions can be drawn from above results:
1. For all of the randomization tests, there are steady and significant differences on the RMSEV between PLS/LS-SVMs and PLS/GPR. The LS-SVMs and GPR always out-performed PLSR in the sense of RMSEV. Averaged ratios of RMSEV were: PLSR/LS-SVMs = 1.270; PLSR/GPR=1.300.

2. Randomized test 3 showed that the RMSEV and bias of LS-SVMs are significantly larger than GPR, but the magnitudes of differences are both very small. Other tests show no significant differences on RMSEV and bias. These results indicate that the two non-linear models are equivalent on predicting target response.

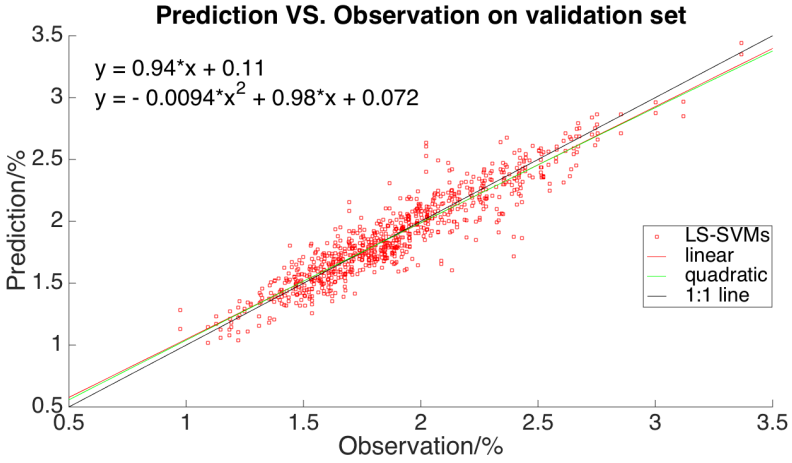
3. Most of the randomized tests showed no significant difference in biases between PLS, LS-SVMs and GPR; this is not a surprise since all of the models were expected to be globally unbiased.

5.2 Local error behavior

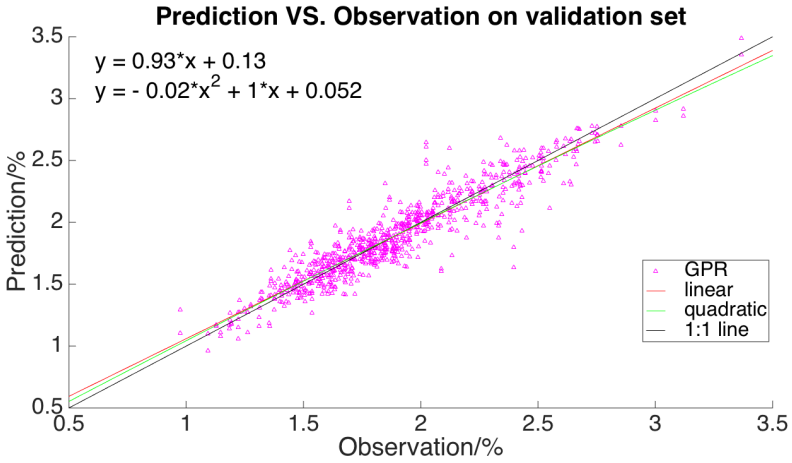
The non-linear models performed better than the linear model in the sense of global error. In this section, predictive behaviors for the three models on each individual sample in the validation set are reported to help understand how linear and non-linear models differ over the whole range. One good way to visualize local error behavior is to plot predictions against their true observations on the whole validation set, as shown in Fig (3).



(a) Predicted vs. Observed for PLSR



(b) Predicted vs. Observed for LS-SVMs



(c) Predicted vs. Observed for GPR

Figure 3: Predictions on each individual sample in the validation set are plotted against their true observations for PLSR(a) , LS-SVMs(b) and GPR(c), with a linear and a quadratic fit to each of them.

For PLSR, even through the performance was acceptable in the middle part of range (roughly

1.5%—2.5%), predictions were heavily biased in the two tails from the regression line. There are two reasonable explanations:

1) The investigated constituent(nitrogen) concentration had an approximately linear impact on NIR spectra within a small range near the population mean, but nonlinear effects become non-ignorable outside this linear range.

2) Prediction power for PLSR was limited. In order to minimize global error, PLSR will try to give best prediction in the center part, which accounts for the majority of the population. As a result, within this range linear correlation between observations and predictions was well maintained. The extreme samples are predicted less well, though this has a limited impact on the overall squared error cost function because there are far fewer of them (see fig 1).

In contrast, the non-linear models (GPR for example here) had fairly consistent standard deviation from the regression line across the range. Non-linear biases at two tails were removed and predictions were then almost linearly related to the observations for all the samples in the validation set. The slope of regression line was also improved from 0.84 to 0.93 by using GPR. Considering any inverse calibration method implicitly uses population distribution in the training set as a prior, all the predictions are naturally shifted to the sample mean [21], in order to minimize global error. For non-linear models, since the fit was much better, this shrinkage to the mean was much less pronounced.

In order to quantitatively compare predictive performance of PLS, LS-SVMs, GPR on the low and high constituent ranges, predictions on the first 5% and the last 5% of the samples in the test set were calculated separately, ranked by their nitrogen concentration. Again, since all the comparisons were carried on a common validation set for each test, the same methods were used to see whether the differences on RMSEV and bias were significant, same as presented in section 5.1. Detailed results on PLSR and GPR are shown in Table 4.

Model	LOW(first 5%)		High(last 5%)		ALL	
	RMSEV	Bias	RMSEV	Bias	RMSEV	Bias
PLSR	0.151	0.009	0.213	-0.156	0.160	0.013
GPR	0.108	0.020	0.160	-0.091	0.130	0.010
CI	(1.250,1.572)	(-0.037,-0.016)	(1.152,1.541)	(-0.094,-0.034)	(1.189,1.268)	(-0.004,0.011)

Table 4: Comparison between PLSR and GPR on different nitrogen ranges. Last row presents 95% confidence interval for the true difference of biases and true ratio of RMSEVs

1) It can be seen, unsurprisingly, that GPR was significantly better than PLS everywhere across the range, in the sense of RMSEV. Especially, the ratio of RMSEV was further enlarged in the two tails.

2) There was no significant difference on the biases on the whole range, and both of them were very close to 0, because the two models are both unbiased decision rules. However, there wee significant differences on biases on low and high concentration range, and PLSR was, surprisingly, less biased on the first 5% of the samples. This can be explained by Figure 4.

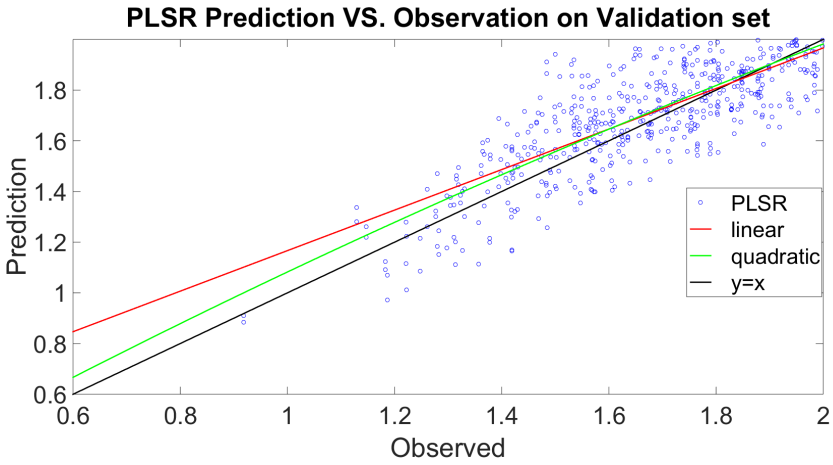


Figure 4: PLSR predictions VS. observations on low constituent range. Red:linear fitting; Green: quadratic fitting; Black: target line(y=x).

The red line is the regression line of prediction on observed. The $y = x$ black line shows the shrinkage to the mean that occurs with PLSR. Because this shrinkage is in the same direction on the curvature of this end of the range, it actually reduces the bias of PLSR. In contrast, at the high end of the range, the shrinkage and curvature are in opposite directions, and PLSR has a very high bias compared to GPR.

5.3 Varying training set size

All of the above analyses are conditional on a fixed size training set. A very important question is how fast each model learns with a growing training set. In practice, the reference measurement could be very expensive and time consuming. Sometimes there will not be sufficient training samples. Then it becomes desirable to understand how these different models perform with varying training set size. In Fig.5, global error on the validation set(RMSEV) is plotted against the number of samples in the training set. For each fixed size of training set, RMSEV is averaged over 5 randomized groupings of calibration and validation sets.

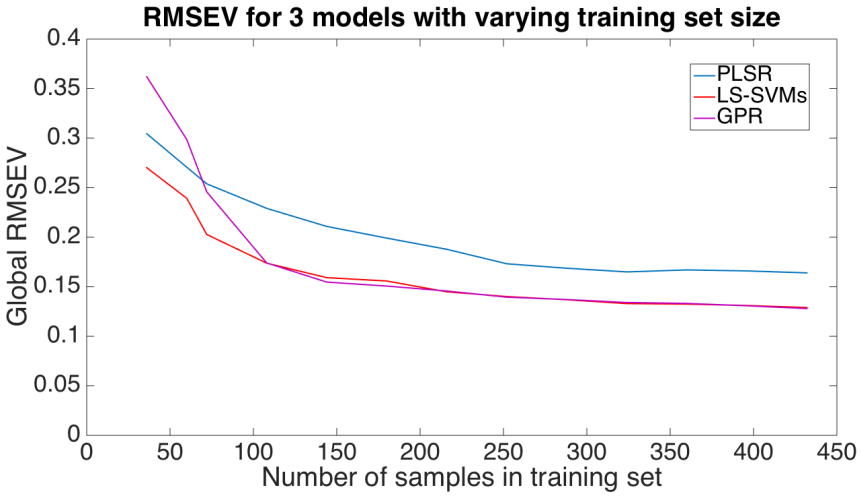


Figure 5: RMSEV for 3 models with varying training set size. PLSR: Cross-validation were performed to choose the number of factors for each test ; GPR and LS-SVMs: RBF kernel.

It can be observed that global errors for all the 3 models decreased when the training set size increased, and then gradually converged to their limits. However, they exhibited different learning speeds. LS-SVMs and PLSR had similar trends, except for a roughly constant offset on RMSEV between the two models. For GPR, when the training set size was small, the prediction power was extremely poor (a RMSEV of 0.38% is almost the width of population distribution in the validation set so the GPR model gave almost homogeneous predictions despite the variation in the spectra), but the predicting power grew quickly with an increase in training set size, and soon became non-distinguishable from LS-SVMs.

This is the first time in this study when GPR and LS-SVMs exhibited significant difference in their predictive performance. Inevitably one wishes to ask a question: what are the differences between LS-SVMs and GPR? By comparing Eq. 17 and Eq. 27, it can be seen that predictions on a new measurement were effectively the same. The covariance function used in GPR was $\mathcal{K}_{GP} = \sigma_s^2 \mathcal{K}$, where $\mathcal{K}$ is the kernel function used in LS-SVMs. If we scale $\frac{1}{\gamma}$ with the same multiplier σ_s^2 then this multiplier actually does not affect $\bar{f}_{I+1}$. By further setting the offset term b in LS-SVMs to 0, they had exactly the same form of prediction. This means, for an appropriate choice of hyper-parameters, the two models can produce identical predictions. However, they use different approaches to find optimal hyper-parameters.

1) In LS-SVMs, the goal is to minimize the squared error loss function, whereas GPR tries to maximize posterior probability. They are quite different optimization targets. Imagine that one set of hyper-parameters gives correct predictions everywhere in the validation set, but quite large variance on them, then it is a perfect predictor for LS-SVMs because it has 0 squared error loss, but a poor predictor for GPR because posterior probability is low. Posterior probability is not a direct indicator of generalization performance, especially when one only cares about predictive mean. In this sense posterior probability might not be suitable to monitor the quality of the model, especially when training set is a small one, because in this case, there is not enough information to estimate the full probability distribution. Squared error loss function depends only on the posterior mean, so it concentrates more on giving close predictions, which helps to

generalize better when the available evidence is weak. However, if probabilistic information is wanted, then one needs a posterior probability, which is also one key advantage of GPR.

2) LS-SVMs use leave-one-out cross-validation(LOO-CV) to tune the parameters, but GPR as implemented here uses evidence maximization(or marginal likelihood maximization). Notice that one can also easily set LOO-CV as inference method in GPR, but these two methods as used here have fundamental differences, which make them distinguishable in the final result. Marginal likelihood involves the probability of the observations given the assumptions of the model, whereas LOO-CV assesses the point prediction only. As a result, LOO-CV is more robust against model mis-specification [22]. Experimental results also show that when using LOO-CV, GPR generalizes slightly better in investigated data set, see Tab.5.

Index	LOO-CV	Evidence Maximization
1	0.287	0.356
2	0.246	0.337
3	0.241	0.304
4	0.248	0.269
5	0.264	0.311

Table 5: RMSEV of GPR with LOO-CV and Evidence maximization for 5 different randomized test. training set size= 60.

It seems that LOO-CV is strongly preferable when the training set is small. When data provided is insufficient, LOO-CV conditions more on the observation and hence uses less knowledge from the prior, which might be mis-specified. Notice that there is no guarantee that cross-validation is always better than evidence maximization in tuning the parameter: since cross-validation conditions too much on observations, there is a higher risk of over-fitting. Since GPR implemented here use both probabilistic loss function and evidence maximization as inference method, unsurprisingly it has relatively worse performance when the training set is a small one. On the other hand, there was no parameter initialization in GPR modeling(where in the LS-SVMs modeling CSA was employed to find suitable start values on hyper-parameters), which makes it more vulnerable to bad local minima for small training set cases. However, this could be an advantage when training set size is large, because it is more computationally efficient.

6 Conclusion

Three models, partial least squares regression, least squares support vector machines and Gaussian process regression were introduced and studied on a large NIR spectroscopic data set. The averaged ratio of PLS/LS-SVMs and PLS/GPR on RMSEV were 1.270 and 1.300 correspondingly. The two non-linear models had similar performances and both improved significantly on PLSR. There was no strong evidence to indicate any significant differences in global bias between the three models. In PLSR, fitting power was not adequate to model the genuine non-linear relationship between NIR and target analyte, which led to a large bias and error when predicting samples with extremely high concentration. Non-linear models including LS-SVMs and GPR can effectively correct the non-linear deviation for extreme samples. Prediction accuracy for all of three models grows with an increasing training set. LS-SVMs is strictly better than PLSR, independent of the size of calibration set. GPR has poor prediction performance when the training set is small, due to the characteristic of the evidence maximization process. However, the

performance improves quickly when more samples are included in the calibration process, and soon becomes non-distinguishable from LS-SVMs.

References

[1] Y. Roggo, P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond, and N. Jent, "A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 44, no. 3, pp. 683–700, 2007.

[2] T. Chen, J. Morris, and E. Martin, "Gaussian process regression for multivariate spectroscopic calibration," *Chemometrics and Intelligent Laboratory Systems*, vol. 87, no. 1, pp. 59–71, 2007.

[3] Q. Chen, J. Zhao, C. Fang, and D. Wang, "Feasibility study on identification of green, black and oolong teas using near-infrared reflectance spectroscopy based on support vector machine (svm)," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 66, no. 3, pp. 568–574, 2007.

[4] A. Borin, M. F. Ferrao, C. Mello, D. A. Maretto, and R. J. Poppi, "Least-squares support vector machines and near infrared spectroscopy for quantification of common adulterants in powdered milk," *Analytica Chimica Acta*, vol. 579, no. 1, pp. 25–32, 2006.

[5] E. Widjaja, W. Zheng, and Z. Huang, "Classification of colonic tissues using near-infrared raman spectroscopy and support vector machines.," *International journal of oncology*, vol. 32, no. 3, pp. 653–662, 2008.

[6] Y. Zhang, Q. Cong, Y. Xie, B. Zhao, *et al.*, "Quantitative analysis of routine chemical constituents in tobacco by near-infrared spectroscopy and support vector machine," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 71, no. 4, pp. 1408–1413, 2008.

[7] N. Gibson, S. Aigrain, S. Roberts, T. Evans, M. Osborne, and F. Pont, "A gaussian process framework for modelling instrumental systematics: application to transmission spectroscopy," *Monthly Notices of the Royal Astronomical Society*, vol. 419, no. 3, pp. 2683–2694, 2012.

[8] J. Verrelst, L. Alonso, J. P. R. Caicedo, J. Moreno, and G. Camps-Valls, "Gaussian process retrieval of chlorophyll content from imaging spectroscopy data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 867–874, 2013.

[9] T. Chen and B. Wang, "Bayesian variable selection for gaussian process regression: Application to chemometric calibration of spectrometers," *Neurocomputing*, vol. 73, no. 13, pp. 2718–2726, 2010.

[10] U. Thissen, M. Pepers, B. Üstün, W. Melssen, and L. Buydens, "Comparing support vector machines to pls for spectral regression applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 73, no. 2, pp. 169–179, 2004.

[11] S.Wold, H.Martens, and H.Wold, "The multivariate calibration problem in chemistry solved by pls method," *In: Proc. Conf. Matrix Pencils(A. Ruhe and B. Kagstrom, eds.), Lecture Notes in Mathematics, Springer-Verlag, Heidelberge*, pp. 286–293, March 1982.

[12] H.Martens and S.A.Jensen, "Partial least squares regression: a new two-stage nir calibration method," *In:Progress in Cereal Chemistry and Technology,Eleveier, Amsterdam*, vol. 5a, pp. 607–647, 1983.

[13] H. Martens and T. Naes, *Multivariate Calibration*. John Wiley & Sons, 1992.

[14] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, J. Suykens, and T. Van Gestel, *Least Squares Support Vector Machines*, vol. 4. World Scientific, 2002.

[15] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, (Berkeley, Calif.), pp. 481–492, University of California Press, 1951.

[16] C. E. Rasmussen, *Gaussian Process for Machine Learning*. The MIT Press, 2006.

[17] K. Pelckmans, J. A. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, B. De Moor, and J. Vandewalle, "Ls-svmlab: a matlab/c toolbox for least squares support vector machines," *Tutorial. KULeuven-ESAT. Leuven, Belgium*, 2002.

[18] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

[19] S. Xavier-de Souza, J. A. Suykens, J. Vandewalle, and D. Bollé, "Coupled simulated annealing," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 2, pp. 320–335, 2010.

[20] T. Fearn, "Comparing standard deviations," *NIR news*, vol. 7, no. 5, pp. 5–6, 1996.

[21] T. Fearn, D. Pérez-Marín, A. Garrido-Varo, and J. E. Guerrero-Ginel, "Inverse, classical, empirical and non-parametric calibrations in a bayesian framework," *Journal of Near Infrared Spectroscopy*, vol. 18, no. 1, p. 27, 2010.

[22] G. Wahba, *Spline models for observational data*, vol. 59. Siam, 1990.