Gaussian Process Regression for Multivariate Spectroscopic Calibration

Tao Chen, Julian Morris and Elaine Martin*

School of Chemical Engineering and Advanced Materials,
University of Newcastle, Newcastle upon Tyne, NE1 7RU, U.K.

* E-mail: e.b.martin@ncl.ac.uk; Tel.: +44 191 222 6231; Fax: +44 191 222 5748.

Abstract

Traditionally multivariate calibration models have been developed using regression based techniques including principal component regression and partial least squares and their non-linear counterparts. This paper proposes the application of Gaussian process regression as an alternative method for the development of a calibration model. By formulating the regression problem in a probabilistic framework, a Gaussian process is derived from the perspective of Bayesian non-parametric regression, prior to describing its implementation using Markov chain Monte Carlo methods. The flexibility of a Gaussian process, in terms of the parameterization of the covariance function, results in its good performance in terms of the development of a calibration model for both linear and non-linear data sets. To handle the high dimensionality of spectral data, principal component analysis is initially performed on the data, followed by the application of Gaussian process regression to the scores of the extracted principal components. In this sense, the proposed method is a non-linear variant of principal component regression. The effectiveness of the Gaussian process approach for the development of a calibration model is demonstrated through its application to two spectroscopic data sets. A statistical hypothesis test procedure, the paired *t*-test, is used to undertake an empirical comparison of the Gaussian process approach with conventional calibration techniques, and it is concluded that the Gaussian process exhibits enhanced behaviour.

Key words: Bayesian inference; Gaussian process; Markov chain Monte Carlo; Multivariate regression; Spectroscopic calibration.

1. Introduction

As powerful analytical tools, spectroscopic techniques, such as mass and infrared spectroscopy, have seen increasing implementation in sectors as diverse as food, pharmaceuticals and petrochemical. The resulting spectrum of the analyte of interest is a continuous curve measured at hundreds (even thousands) of equally spaced wavelengths and it is assumed that it indirectly captures the chemical/physical properties of the material being analyzed. The subsequent analysis of the spectrum can be qualitative or quantitative. Qualitative analysis generally addresses classification problems, e.g. the classification of ovarian cancer using mass spectrometry [1]; and detection problems, e.g. the detection of process transitions between steady states in oil sand extraction plant [2]. In contrast, quantitative analysis focuses on the determination of the value of the chemical/physical properties (e.g. weight percentage of active substance in a sample of tablet, [3]) of the analyte from its measured spectrum. This is referred to as calibration and is the focus of this paper.

Traditional calibration techniques have been based on linear regression methodologies. Consider the case where a calibration model is to be developed from a set of *N* samples of analyte. Let $\mathbf{x}_i$ denote the spectrum of the *i*-th sample, a vector comprising values at *p* wavelengths. Collating this data across the *N* samples materialises in the matrix, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^{\mathrm{T}}$. Likewise, let $\mathbf{y}_i$ be a *q*-dimensional vector, where *q* is the number of outputs of interest, and let $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^{\mathrm{T}}$. The calibration task is thus to build a multivariate regression model, of the form $\mathbf{Y} = f(\mathbf{X})$. More specifically for linear regression:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E} \tag{1}$$

where **B** is a matrix of regression coefficients, and **E** is the residual matrix. This linear regression model satisfies Beer-Lambert's law [4] and the regression coefficients are calculated as follows:

$$\mathbf{B} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y} \tag{2}$$

One of the major problems with this approach is that, in some situations, the number of training samples is smaller than the number of predictors, i.e. $N < p$, resulting in a singular matrix for $\mathbf{X}^T\mathbf{X}$. Hence it is not possible to invert $\mathbf{X}^T\mathbf{X}$ to obtain the regression coefficients in Equation (2). This is a common situation when developing a calibration model since the number of predictors (wavelengths) varies from several hundreds to thousands, but the number of samples is normally less than one hundred.

The typical solution to this problem has been to develop a regression model from a reduced number of wavelengths by projecting them onto lower dimensional sub-space, using a multivariate statistical projection based approach such as principal component regression (PCR) [5] or partial least squares (PLS) [6] [7]. Furthermore not all systems satisfy Beer-Lamberts law and hence to address the inherent non-linearity in the data, non-linear PCR [8] and non-linear PLS [9] [10] have been proposed. The typical approach has been to integrate polynomial functions or artificial neural networks, for example, into the PCR and PLS algorithms.

More recently, there has been a significant increase in interest in Gaussian process regression. Initially proposed by O'Hagan [11], Gaussian process regression was viewed as an alternative approach to artificial neural networks, primarily as a result of the seminal research of Neal [12]. Neal showed that a large class of Bayesian regression models, based on artificial neural networks, converged to a Gaussian process, in the limit of an infinite network [12]. Gaussian processes can also be derived from the perspective of non-parametric Bayesian regression [13] [14], by directly placing Gaussian prior distribution over the space of regression functions without parameterizing $f(\mathbf{X})$. As a result of its good performance in practice and desirable analytical properties, Gaussian process models have been widely applied [13-15], but to date are less well known in the chemometric community.

This paper introduces the application of Gaussian process modelling as an alternative solution to the spectroscopic calibration problem. The derivation of a Gaussian process regression model, in this paper, will be presented from the non-parametric regression perspective [13], through a Bayesian framework. As a general approach to Bayesian inference, the Markov chain Monte Carlo (MCMC) method is utilised to approximate the posterior distribution of the model hyper-parameters (defined in the subsequent section), and to make inferences of the outputs for new samples. By designing the model structure appropriately, a Gaussian process can handle both linear and non-linear data. To handle the challenge of the high dimensionality of the predictors (wavelengths), principal

component analysis is first performed on the original data, prior to applying Gaussian process regression to the scores from the extracted principal components. Thus the proposed method is aligned with that of non-linear principal component regression, where the non-linearity is addressed through the Gaussian process. The use of the Gaussian process based approach will be justified by demonstrating its improved performance over traditional methods, with respect to the development of a calibration model for two sets of near infrared (NIR) data.

2. The Gaussian processes from the perspective of Bayesian regression analysis

The primary objective of this paper is to establish a regression model based on the data set {**X**, **Y**} for the inference of the response variables given any new predictors. Bayesian inference considers the posterior distribution of the model parameters, $\boldsymbol{\theta}$ (or hyper-parameters in a Gaussian process, as introduced subsequently), which is proportional to the product of the prior and likelihood distributions:

$$p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{Y}) \propto p(\boldsymbol{\theta})\, p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}) \tag{3}$$

Within this section, the derivation of the Gaussian process regression model for the development of a calibration model from spectroscopic data is introduced. The posterior distribution of the hyper-parameters in the model will first be formulated, prior to attaining the predictions of new data points from the properties of the Gaussian process. The implementation of the Gaussian process will be carried out using Markov chain Monte Carlo sampling. Finally a number of issues will be discussed.

2.1 Overview of Gaussian process

Within this subsection, a linear regression model is first introduced, from which non-linear scenarios are handled by introducing appropriate basis functions. The Gaussian process is introduced by taking a Bayesian non-parametric perspective on the formulation of the basis function regression model. For ease of derivation, a single response variable is considered initially, i.e. $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$ where N is the number of training samples. The development of a calibration model for multiple responses (multiple component calibration) is discussed in Section 2.3.

In the case of a single response variable, the linear regression model in Equation (1) can be re-written as:

$$y_i = \sum_{d=1}^{p} x_{id} b_d + e_i = \mathbf{x}_i^{\mathrm{T}} \mathbf{b} + e_i \tag{4}$$

where x_{id} is the *d*-th variable of vector $\mathbf{x}_i$, b_d is the *d*-th value of the regression vector $\mathbf{b}$, and e_i is the additive noise term. Within a Bayesian framework, the standard definition for the prior distributions of $\mathbf{b}$ and $\mathbf{e}$, where $\mathbf{e} = (e_1, \ldots, e_N)^{\mathrm{T}}$, is Gaussian with zero mean and diagonal covariance matrix:

$$p(\mathbf{b}) = G(\mathbf{0}, \sigma_b^2 \mathbf{I}) \tag{5}$$

$$p(\mathbf{e}) = G(\mathbf{0}, \sigma_e^2 \mathbf{I}) \tag{6}$$

Thus the response, $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$, is a linear function of $\mathbf{b}$ and $\mathbf{e}$, and hence it also has a Gaussian distribution with zero mean, i.e. $p(\mathbf{y}) = G(\mathbf{0}, \mathbf{C})$. This is a Gaussian process [13-15] and the entries of the covariance matrix, $\mathbf{C}$, can be obtained as follows:

$$\begin{aligned} C_{ij} &= \mathrm{COV}(y_i, y_j) = E(y_i y_j) = E(\mathbf{x}_i^{\mathrm{T}} \mathbf{b}\mathbf{b}^{\mathrm{T}} \mathbf{x}_j + e_i e_j) = \mathbf{x}_i^{\mathrm{T}} E(\mathbf{b}\mathbf{b}^{\mathrm{T}}) \mathbf{x}_j + E(e_i e_j) \\ &= \sigma_b^2 \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j + \sigma_e^2 \delta_{ij} = \sigma_b^2 \sum_{d=1}^{p} x_{id} x_{jd} + \sigma_e^2 \delta_{ij} \end{aligned} \tag{7}$$

where $E(.)$ is the expectation operator, and $\delta_{ij} = 1$ if *i*=*j*, otherwise $\delta_{ij} = 0$. C_{ij} is typically referred to as the "covariance function" and denoted by $C_{ij} = C(\mathbf{x}_i, \mathbf{x}_j)$, to emphasize that it is not a conventional covariance term, but a function of $\mathbf{x}_i$ and $\mathbf{x}_j$ given (σ_b^2, σ_e^2). As the regression model can be summarized by the covariance function without referring to the parameter vector, $\mathbf{b}$, the Gaussian process is a non-parametric regression technique. Furthermore, the form of the covariance function is not restricted to that given in Equation (7). For example, by introducing a "basis function", $\phi_m(\mathbf{x})$, the linear regression model in Equation (4) can be generalized to a non-

linear regression model with respect to the predictors $\mathbf{x}$, however it will still be linear with respect to the basis functions:

$$y_i = \sum_{m=1}^{M} \phi_m(\mathbf{x}_i) b_m + e_i \tag{8}$$

where *M* is the number of basis functions. The most widely applied basis functions include radial basis functions, wavelets, and splines. Thus based on the previous derivation, the covariance function defining the Gaussian process is given by:

$$C(\mathbf{x}_i, \mathbf{x}_j) = \sigma_b^2 \sum_{m=1}^{M} \phi_m(\mathbf{x}_i) \phi_m(\mathbf{x}_j) + \sigma_e^2 \delta_{ij} \tag{9}$$

In addition to the two methods described previously, there are many other ways of defining the covariance function [13] [14]. Mackay [13] stated that the only constraint was that the function must generate a non-negative definite covariance matrix for any set of data points. The following covariance function, which has been widely reported in the literature, is used in this paper:

$$C(\mathbf{x}_i, \mathbf{x}_j) = a_0 + a_1 \sum_{d=1}^{p} x_{id} x_{jd} + v_0 \exp\left(-\sum_{d=1}^{p} w_d (x_{id} - x_{jd})^2 \right) + \sigma_e^2 \delta_{ij} \tag{10}$$

where the first two terms represent constant bias (offset) and linear correlation, respectively. The exponential term is similar to the form of the radial basis function, and it recognizes the strong correlation between the outputs and nearby inputs. $\sigma_e^2 \delta_{ij}$ captures the random error effect as discussed previously with respect to Equation (7). By combining linear and non-linear terms in the covariance function, the Gaussian process is capable of handling both linear and non-linear data structures.

Let $\boldsymbol{\theta} = (a_0, a_1, v_0, w_1, \ldots, w_p, \sigma_e^2)^{\mathrm{T}}$ denote the "hyper-parameters" defining the covariance function in Equation (10). The term "hyper-parameter" is used to differentiate between Gaussian process regression and parametric regression where the parameter vector $\mathbf{b}$ is to be estimated. Bayesian inference of the hyper-parameters requires that their joint posterior distribution is

proportional to the product of the prior and likelihood distributions. The likelihood is a Gaussian distribution: $p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X}) = G(\mathbf{0}, \mathbf{C})$, and its logarithm is given by:

$$L = \log p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X}) = -\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y} - \frac{N}{2}\log(2\pi) \tag{11}$$

Prior distributions are assigned to the hyper-parameters according to the approach cited in Rasmussen [16]. The priors for $\log(a_0)$, $\log(a_1)$ and $\log(\sigma_e^2)$ are Gaussian with mean -3 and standard deviation 3, corresponding to fairly vague priors. The prior for $\log(v_0)$ is Gaussian with mean -1 and a standard deviation of unity and the priors for w_d are inverse Gamma distribution:

$$p(w_d^{-1}) = \mathrm{Gamma}(\alpha, \mu) \propto (w_d^{-1})^{\alpha/2-1} \exp\left(-\frac{w_d^{-1}\alpha}{2\mu}\right) \tag{12}$$

To account for the effect of the number of predictors (hence the number of w_d's), the mean of the above Gamma distribution is scaled as $\mu = \mu_0 p^{2/\alpha}$, where $\alpha = \mu_0 = 1$ as in Rasmussen [16]. For a new data point with predictors **x***, the predictive distribution of the response *y**, conditional on the hyper-parameters, is also Gaussian distributed with mean and variance calculated as follows:

$$E(y^*) = \mathbf{k}^{\mathrm{T}}(\mathbf{x}^*)\mathbf{C}^{-1}\mathbf{y} \tag{13}$$

$$Var(y^*) = C(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^{\mathrm{T}}(\mathbf{x}^*)\mathbf{C}^{-1}\,\mathbf{k}\ (\mathbf{x}^*) \tag{14}$$

where $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x}^*, \mathbf{x}_1), \ldots, C(\mathbf{x}^*, \mathbf{x}_N)]^{\mathrm{T}}$.

2.2 Implementation of the Gaussian process using MCMC

The hyper-parameters of a Gaussian process can be obtained by maximizing the posterior distribution, using the conjugate gradient method [16]. However this approach is sensitive to initialization and normally converges to a local optimum. Therefore a number of random initializations are required to guarantee reliable results. Alternatively, Bayesian inference can be

performed using Monte Carlo sampling. The basic idea of Monte Carlo sampling is to generate a large number of random samples, $\{\boldsymbol{\theta}_{(1)},\ldots,\boldsymbol{\theta}_{(K)}\}$, for the hyper-parameters from their joint posterior distribution, $p(\boldsymbol{\theta}\,|\,\mathbf{X},\mathbf{y})$:

For $k = 1 : K$

Generate $\boldsymbol{\theta}_{(k)} \sim p(\boldsymbol{\theta}\,|\,\mathbf{X},\mathbf{y}) \propto p(\boldsymbol{\theta})\,p(\mathbf{y}\,|\,\boldsymbol{\theta},\mathbf{X})$

End

Therefore the prediction of a new data point can be made by taking the average of Equations (13) and (14) with respect to these Monte Carlo samples:

$$E(y^*) = \frac{1}{K}\sum_{k=1}^{K} E_{(k)}(y^*) \tag{15}$$

$$Var(y^*) = \frac{1}{K}\sum_{k=1}^{K} Var_{(k)}(y^*) + \frac{1}{K}\sum_{k=1}^{K}\left(E_{(k)}(y^*) - E(y^*)\right)^2 \tag{16}$$

where $E_{(k)}(y^*)$ and $Var_{(k)}(y^*)$ are calculated from Equations (13) and (14) respectively, using the Monte Carlo samples $\boldsymbol{\theta}_{(k)}$.

Since the posterior is not of the form of a standard distribution such as Gaussian or Gamma, direct sampling is not possible. Thus Markov chain Monte Carlo (MCMC) methods, a class of sequential sampling techniques, are often utilised. For MCMC methodologies, the *k*-th sample generated, $\boldsymbol{\theta}_{(k)}$, is dependent on the previous sample, $\boldsymbol{\theta}_{(k-1)}$. The Metropolis-Hastings algorithm [17] is one such MCMC method and it uses a *proposal distribution*, $p(\boldsymbol{\theta}\,|\,\boldsymbol{\theta}_{(k-1)})$, to generate a candidate $\boldsymbol{\theta}^*$. $\boldsymbol{\theta}^*$ is then accepted according to a certain probability that is dependent on the posterior probability of $\boldsymbol{\theta}^*$ and the proposed distribution. More specifically, the Metropolis-Hastings algorithm is as follows:

For $k = 1 : K$

Generate $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta}\,|\,\boldsymbol{\theta}_{(k-1)})$

Generate u from a uniform distribution $U[0, 1]$

If $u < \min\left\{1, \dfrac{p(\boldsymbol{\theta}^* \mid \mathbf{X}, \mathbf{y})\, p(\boldsymbol{\theta}_{(k-1)} \mid \boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}_{(k-1)} \mid \mathbf{X}, \mathbf{y})\, p(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_{(k-1)})}\right\}$, $\boldsymbol{\theta}_{(k)} = \boldsymbol{\theta}^*$;

else $\boldsymbol{\theta}_{(k)} = \boldsymbol{\theta}_{(k-1)}$

End

The Metropolis-Hastings algorithm is straightforward to implement since it only requires a proposal distribution and the calculation of the posterior probability of generated samples. The proposal distribution is typically selected as a Gaussian distribution with mean $\boldsymbol{\theta}_{(k-1)}$ and pre-defined covariance matrix: $p(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{(k-1)}) = G(\boldsymbol{\theta}; \boldsymbol{\theta}_{(k-1)}, \boldsymbol{\Sigma})$, which suggests that the new sample is generated based on a random search method in the neighbourhood of the previous sample [17]. However, as the number of hyper-parameters in a Gaussian process is fairly large, and the posterior distribution is complex and possibly of multi-modal form [14], the random search strategy of the Metropolis-Hastings algorithm, and other conventional MCMC methods, may converge very slowly. The hybrid Monte Carlo approach was shown to be able to improve convergence by using the gradient information, that is, the derivatives of the posterior distribution with respect to the hyper-parameters [18]. The derivative of the log-likelihood with respect to a single hyper-parameter, θ, can be derived as:

$$\frac{\partial L}{\partial \theta} = -\frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta}\right) + \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta}\mathbf{C}^{-1}\mathbf{y} \tag{17}$$

where $\partial\mathbf{C}/\partial\theta$ can be obtained from the covariance function. Likewise the derivatives of the prior distribution with respect to the hyper-parameters can also be derived. Based on the gradient information, there are a number of different implementations of hybrid Monte Carlo for a Gaussian process that do not differ significantly [14] [16]. The approach of Neal [14] is used in this study.

It should also be noted that the calculation of the likelihood and the derivatives involves a matrix inversion step and takes time of the order $O(N^3)$, which is feasible for a moderate size of training data sets (less than several thousand) on a conventional computer. For larger data sets, sparse training strategies may be required to reduce the overall computational burden, such as data

splitting [19] or mixtures of Gaussian processes [20]. For the spectroscopic calibration problem, the computational aspect is not a real issue.

2.3 Regression with multiple responses

The extension of Gaussian process regression to multiple response variables, i.e., multiple-component calibration, is non-trivial. It is possible to define a Gaussian process with q response variables, but it is not clear how to ensure the covariance matrix is positive definite. This problem can be addressed using the indirect definition of the covariance function, for example by treating the Gaussian process as the output of stable linear filters [21]. However, when dealing with more than two response variables, the definition of the covariance function in Boyle [21] is extremely complicated. In addition, the computational cost increases significantly, since the covariance matrix is of the order, $Nq \times Nq$, and whose inversion takes time $O(N^3 q^3)$.

To avoid the complexity associated with the definition of the covariance function, and the computational issues, the study reported in this paper adopts a simplified solution which models each response independently [13]. An independent modelling strategy is a compromise between inferred performance and algorithmic complexity. By adopting this approach, it can be argued that significant information contained in the correlation structure between the response variables is ignored. However, even in the area of PLS, there is no consensus on whether multi-response modelling (PLS2) can achieve better predictive performance than independent modelling (PLS1). In this paper the goal is to compare the Gaussian process algorithm for calibration modelling with a number of different techniques including principal component regression, neural networks and partial least squares and thus the issue of modelling multiple or independent responses is not considered in detail.

2.4 Data Pre-processing

In the literature it has been proposed, for numerical reasons that the response variables are transformed to have zero mean before the data is used for training a Gaussian process [14]. The rationale for this is that if the mean of the responses moves significantly away from zero, the constant bias (offset) term in the covariance function (a_0 in Equation (10)) will become relatively large, and thus the resultant covariance matrix will have a large condition number [14] and

consequently the precision of the numerical inversion of the covariance matrix will degrade significantly. Hence in the application studies, the response variables are auto-scaled to have zero mean and unit standard deviation before being used to train a Gaussian process.

Finally, as mentioned in the Introduction, to handle the high dimensional predictors, principal component analysis is first performed on the original predictors. Gaussian process regression is then applied to the extracted scores of the principal components. Thus the proposed methodology is a variant of non-linear principal component regression, where the non-linearity is addressed through the Gaussian process. The appropriate number of principal components can be selected based on the model selection criterion used for conventional PCR, such as Bayesian information criterion (BIC) [22], or cross validation.

3. Application studies

3.1 Data sets

The first case investigated was based on the "Tablet" data set that comprised a set of near infrared (NIR) transmittance spectra generated from the analysis of Escitalopram® tablets, manufactured by a pharmaceutical company. The aim of the study was to determine the weight percentage of the active substance in the tablets, based on the NIR spectra recorded over the range 4000 – 14000 cm^{-1}, of which the region, 7400 – 10500 cm^{-1} (corresponding to 404 predictors), was used for the development of the calibration model [3]. The data set comprised 310 tablet samples manufactured at the laboratory, pilot-plant and production scales. To investigate calibration performance on a relatively small data set, the tablet samples from the pilot scale (120 samples) were used in the reported study. These corresponded to the "preliminary calibration set" defined in the original paper [3]. The training and test data sets each comprised 60 samples. The original data set is available at http://www.models.kvl.dk/research/data/Tablets/. Further details about the data set can be found in Dyrby [3], where it was reported that PLS was capable of achieving acceptable performance, indicating that the inherent data structure is approximately linear.

The second case study is based on the "Meat" data set that was recorded on a Tecator NIR spectrometer (Infratec Food and Feed Analyzer) which measured the absorbance at 100 wavelengths across the region 850 - 1050 nm [23]. The samples were finely chopped pure meat and the goal was to infer a number of properties including moisture, fat and protein content from the

NIR spectra. Based on the research of Borggaard [23], 215 samples were available and of these, 173 defined the training data set and 42 the test data set. According to previous studies [23][24], this data set exhibits significant non-linear behaviour with respect to the development of a calibration model for the fat content. Consequently the linear regression techniques of PCR and PLS, are not expected to perform satisfactorily. The original data is available at http://lib.stat.cmu.edu/datasets/tecator.

3.2 Statistical evaluation of calibration methods

The Gaussian process approach is compared with the more conventional calibration techniques of PCR, PLS, quadratic partial least squares (QPLS, a non-linear variant of PLS) [9] and artificial neural network (ANN) regression. The different algorithms were applied to the scores following the application of principal components with the number of latent variables retained in PCR, PLS and QPLS being determined using cross validation. For the ANN and the Gaussian process, the number of retained principal components was taken to be the same as for PCR. A standard feed-forward ANN with one hidden layer (sigmoid transfer function) and one output layer (linear transfer function) was used. The number of neurons in the hidden layer was determined heuristically through the undertaking of preliminary experiments in this study. To avoid over-fitting, the Bayesian regularization approach, in combination with Levenberg-Marquardt training [25] [26], was applied for the training of the ANN. The results from the different approaches were compared in terms of the root mean square error for prediction (RMSEP).

For a reliable evaluation of the performance of these techniques, two issues must be addressed. First, by partitioning the limited amount of available data into training and test data sets uncertainty is introduced, since in each partition, the calibration algorithms are only evaluating a subset of the available data. The test data set may not be representative of the whole data space, on which the calibration is expected to work in the future. Thus the evaluation may not be reliable. For a small data set, this problem can be alleviated by manually designing the test data set such that it covers the entire range of operation. In some situations, it may be appropriate to reserve a test set that is generated during a different time period to that of the training data, to evaluate the calibration methods in terms of robustness against time variations. In this study, a common stochastic strategy is adopted where the random partitioning of training and test sets is repeated I times (I=50 in this paper). The repeated random partitioning will materialise in the calibration methods being evaluated

across a wider range of operation. Thus the calibration methods are evaluated on each partition, and the average RMSEP is used for comparison.

The second issue is to determine whether the different performance of the calibration methods, in terms of RMSEPs, is statistically significant. In other words, a statistical criterion is necessary to determine whether one method is significantly better (or worse) than another. Two possible hypothesis based approaches are the paired-*t* test [27] and the von der Voet's test [28]. In this paper the former approach is adopted. Let r_i and s_i be the RMSEPs of two calibration methods for the *i*-th random partition of the data. To apply the paired *t*-test, the RMSEPs are first mean centred: $\hat{r}_i = r_i - \bar{r}$ and $\hat{s}_i = s_i - \bar{s}$, where $\bar{r}$ and $\bar{s}$ are the means of r_i and s_i ($i = 1, \cdots, I$) respectively. The *t* statistic is then calculated as follows:

$$t = (\bar{r} - \bar{s}) \sqrt{\frac{I(I-1)}{\sum_{i=1}^{I} (\hat{r}_i - \hat{s}_i)^2}} \tag{18}$$

This *t* statistic has a sampling *t* distribution with $I - 1$ degrees of freedom. After obtaining the *t* statistic, the *p*-value, can be determined using a table of confidence intervals for the *t* distribution [27]. A *p*-value lower than a threshold, normally taken as 0.05, indicates that the difference in the performance obtained by the two calibration methods is statistically significant.

Within this paper by testing the predictive performance of several algorithms, the issue of multiple comparison testing is theoretically an issue. Benjamini and Hocheberg [29] discussed the concept of multiple hypothesis testing and proposed that it may be beneficial to control the "false discovery rate." By implementing the procedure proposed in [29] (not reported), similar conclusions were attained and thus the individual paired *t*-test has been used to report the results, for the reasons of concise presentation and its wide adoption in the literature.

3.3 Software

The results for the Gaussian process reported in the subsequent section were produced using C++ code developed by the authors. There are several software packages for the execution of Gaussian processes publicly available from the Internet. The flexible Bayesian modelling (FBM) package, written by Neal (http://www.cs.toronto.edu/~radford/fbm.software.html) is a general toolbox for

various Bayesian modelling techniques, including Gaussian process, artificial neural networks, among other methodologies. The FBM package was implemented in Ansi C and tested under the Unix system. For those researchers who prefer working with Matlab (MathWorks, Inc., Natick, MA, USA), the Netlab toolbox is a well written package to perform Gaussian process (http://www.ncrg.aston.ac.uk/netlab/). Despite some differences in implementation, the FBM toolbox gives similar results to those presented in the next section.

The Matlab PLS Toolbox version 3.5 (Eigenvector Research, Inc., Wenatchee, WA, USA) was used for the implementation of PCR, PLS, and quadratic PLS and the Matlab Neural Network Toolbox was used for the training of the artificial neural networks.

4. Results and discussions

Table 1 gives the calibration results for the "Tablet" data set, where the RMSEP is the value attained by averaging over 50 random partitions of the training and test data. The linear regression methods, PCR and PLS, perform reasonably well on this data set, implying a strong linear relationship between the spectra and the weight content of the tablets. The *p*-value of the paired *t*-test (Table 2) between PCR and PLS is 0.48, suggesting that both methods give similar performance. Artificial neural networks are known for their universal approximation ability with respect to both linear and non-linear regression functions. In this case study, they give significantly (*p*-values smaller than 10^{-4}) lower RMSEP than both PCR and PLS. The better result from the ANN implies that there may be weak non-linearity in this data set. However, QPLS exaggerates this non-linearity by fitting a second-order polynomial function for the inner relationship of PLS [9], and gives the largest RMSEP, 0.534. In theory, if the data is perfectly linear, the coefficient of the quadratic term in QPLS should be zero, and hence QPLS reduces to linear PLS. However, perfect linearity does not exist in practice. Hence the second-order polynomial function would "over-fit" a weak non-linear relationship, and thus generalize poorly to unseen testing data.

(Table 1 and Table 2 about here)

Finally, Table 1 shows that the Gaussian process achieves further improvement over the ANN. Although the RMSEP of the Gaussian process is not substantially lower than that of the ANN, the *p*-value of 8.6×10^{-4} shows that this improvement is still statistically significant. The predicted versus measured plot for one partition of the training and test data is shown in Figure 1. This figure

is selected from the 50 partitions such that it gives a result close to the average RMSEP (shown in Table 1) of each calibration methods. The same criterion is used to illustrate the results in Figure 2. For clarity only the results of PLS and the Gaussian process are shown. Figure 1 clearly illustrates the satisfactory calibration results achieved from the Gaussian process, and its superior performance to that of PLS, even for this approximately linear data set.

(Figure 1 about here)

The second data set ("Meat") involves a multi-component calibration problem, that is, the need to determine multiple properties (response variables) of the analyte. As discussed in Section 2.3, separate Gaussian process models are built for each response variable. Initial experiments (not reported) showed that for PLS, independent modelling (PLS1) and multi-response modelling (PLS2) gave similar prediction errors with there being no significant difference in the results. Consequently in the following studies, the results for PLS also relate to those from PLS2 since the default settings in the Matlab PLS Toolbox were adopted. Furthermore this result confirms the appropriateness of using separate Gaussian process models.

Table 3 gives the results of each calibration method for each response variable. The RMSEP quoted is averaged over 50 random partitions of the training and test data. It can be seen that both PCR and PLS give poor results in terms of predicting the moisture and fat content, while the RMSEP for the protein content is acceptable. This suggests significant non-linearity exists between the NIR spectra and the moisture and fat content, but weak non-linearity is present with respect to the protein content. PLS performs slightly better than PCR for all properties; however this difference is not statistically significant as the *p*-values between them, shown in Table 4, are greater than the conventional 0.05 significance level (the *p*-values are 0.56, 0.92 and 0.30 for moisture, fat and protein, respectively).

(Table 3 and Table 4 about here)

The non-linearity of the "Meat" data realises a real opportunity for the implementation of an artificial neural network. It can be observed that a substantial improvement is evident in terms of the values of the RMSEP over PCR and PLS, for the prediction of moisture and fat content. However the ANN does not give better results than PCR or PLS for protein content: the difference in the RMSEPs is negligible as indicated by the large *p*-values (third column in Table 4 (c)). QPLS

is more capable of handling the non-linearity in the moisture and fat content giving a further reduction of RMSEP over the ANN. However for protein content, which exhibits weak non-linearity, QPLS performs unsatisfactorily as discussed for the "Tablet" data set.

Table 3 clearly shows that the Gaussian process approach gives the lowest RMSEP for all three components. Although for the moisture content, the Gaussian process is only slightly superior to QPLS (*p*-value is 0.076), its improvement over the other methods for the fat and protein content is statistically significant. Figure 2 illustrates the predicted versus measured plot for PLS and the Gaussian process for the prediction of all three components.

(Figure 2 about here)

5. Conclusions

This paper applied a Bayesian non-parametric regression technique, namely the Gaussian process, for the development of multivariate calibration models for spectroscopic data. Formulated from a Bayesian framework, the Gaussian process is flexible as a consequence of the parameterization of its covariance function, and it can be efficiently implemented using Markov chain Monte Carlo methods. The application studies on the calibration of two NIR spectroscopic data show that the Gaussian process is capable of achieving reliable and satisfactory results for both linear and non-linear data sets, and is thus a promising alternative approach to the calibration problem.

Despite its successful applications in many areas, there are still some open issues with Gaussian processes. For example multiple component calibration is handled by building separate Gaussian process model for each response variable. As noted in Section 2.3, the current solution to account for the covariance among the responses is still complex in terms of implementation, and involves significant computational power. Alternative approaches to addressing this issue are currently under investigation.

Acknowledgments

T. Chen would like to acknowledge the financial support from the EPSRC KNOW-HOW (GR/R19366/01) and Chemicals Behaving Badly II (GR/R43853/01), and the UK ORS Award for his PhD study.

References

[1] B. Wu, T. Abbott, D. Fishman, W. McCurray, G. Mor, K. Stone, D. Ward, K. Williams, H. Zhao. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data, Bioinformatics 19 (2003) 1636– 1643.

[2] W. I. Friesen. Qualitative analysis of oil sand slurries using on-line NIR spectroscopy, Applied Spectroscopy 50 (1996) 1535-1540.

[3] M. Dyrby, S. B. Engelsen, L. Norgaard, M. Bruhn, L. Lundsberg-Nielsen, Chemometric quantitation of the active substance in a pharmaceutical tablet using near infrared (NIR) transmittance and NIR FT-Raman spectra, Applied Spectroscopy 56 (2002) 579-585.

[4] B. G. Osborne, T. Fearn, P. H. Hindle. Practical NIR Spectroscopy, Longman, U. K., 1993.

[5] I. Cowe, J. McNicol. The use of principal components in the analysis of near infrared spectra, Applied Spectroscopy 39 (1985) 257-266.

[6] S. Wold, H. Martens, H. Wold. The multivariate calibration problem in chemistry solved by PLS. In: Matrix Pencils (A. Ruhe, B. Kagstrom, Ed.), Heidelberg: Springer, 286-293, 1983.

[7] P. Geladi, B. Kowalski. Partial least-squares regression: a tutorial. Analytica Chimica Acta 185 (1986) 1-17.

[8] P. J. Gemperline, J. R. Long, V. G. Gregoriou. Nonlinear multivariate calibration using principal components regression and artificial neural networks. Analytical Chemistry 63 (1991) 2313 - 2323.

[9] S. Wold, N. Kettaneh-Wold, B. Skagerberg, Nonlinear PLS modelling, Chemometrics and Intelligent Laboratory Systems 7 (1989) 53-65.

[10] G. Baffi, E. Martin, A. Morris. Non-linear projection to latent structures revisited (the neural network PLS algorithm. Computers and Chemical Engineering 23 (1999) 1293-1307.

[11] A. O'Hagan. Curve fitting and optimal design for prediction (with discussion). Journal of the Royal Statistical Society B 40 (1978) 1-42.

[12] R. M. Neal. Bayesian learning for neural networks. New York: Springer-Verlag, 1996.

[13] D. J. C. MacKay. Introduction to Gaussian processes. In: Neural Networks and Machine Learning (C. M. Bishop, Ed.), Springer, 133-165, 1998.

[14] R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report No. 9702, Department of Statistics, University of Toronto, Canada, 1997.

[15] C. K. I. Williams. Prediction with Gaussian processes: from linear regression to linear prediction and beyond. In: Learning and Inference in Graphical Models (M. I. Jordan, Ed.), Kluwer, 599-621, 1998.

[16] C. E. Rasmussen. Evaluation of Gaussian processes and other methods for non-linear regression. Ph. D. thesis, University of Toronto, Canada, 1996.

[17] C. Robert, G. Casella. Monte Carlo Statistical Methods. Springer, New York, 1999.

[18] S. Duane, A. D. Kennedy, B. J. Pendleton, D. Roweth. Hybrid Monte Carlo. Physics Letters B 195 (1987) 216-222.

[19] V. Tresp, A Bayesian committee machine, Neural Computation 12 (2000) 2719-2741.

[20] C. E. Rasmussen, Z. Ghahramani, Infinite mixtures of Gaussian process experts, In: Advances in Neural Information Processing Systems 14 (T. Dietterich, S. Becker, Z. Ghahramani, Eds.), MIT Press, 2002.

[21] P. K. Boyle, M. R. Frean. Dependent Gaussian processes. In: Advances in Neural Information Processing Systems 17, MIT Press, 2004.

[22] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6 (1978) 461-464.

[23] C. Borggaard, H. H. Thodberg, Optimal minimal neural interpretation of spectra, Analytical Chemistry 64 (1992) 545-551.

[24] H. H. Thodberg, A review of Bayesian neural networks with an application to near infrared spectroscopy, IEEE Transactions on Neural Networks 7 (1996) 56-72.

[25] D. J. C. MacKay, Bayesian interpolation, Neural Computation 4 (1992) 415-447.

[26] F. D. Foresee, M. T. Hagan, Gauss-Newton approximation to Bayesian regularization. In: International Joint Conference on Neural Networks, 1930-1935, 1997.

[27] C. H. Goulden, Methods of Statistical Analysis, 2nd Eds. New York: Wiley, 1956.

[28] H. van der Voet, Comparing the predictive accuracy of models using a simple randomization test, Chemometrics and Intelligent Laboratory Systems 25 (1994): 313-323.

[29] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, Journal of the Royal Statistical Society B 57 (1995): 289-300.

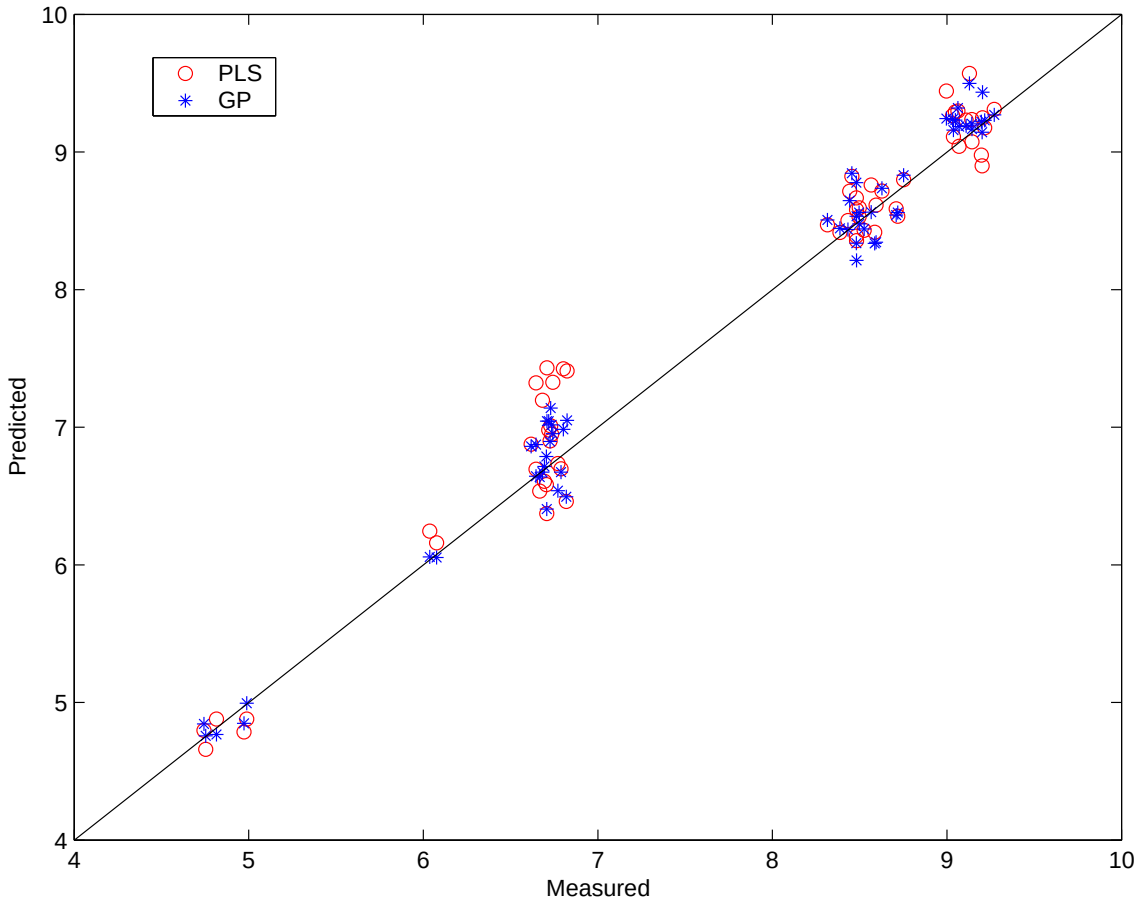

Figure 1: Predicted vs. measured plot for the “Tablet” data set using PLS (RMSEP = 0.261) and Gaussian process (RMSEP = 0.182).

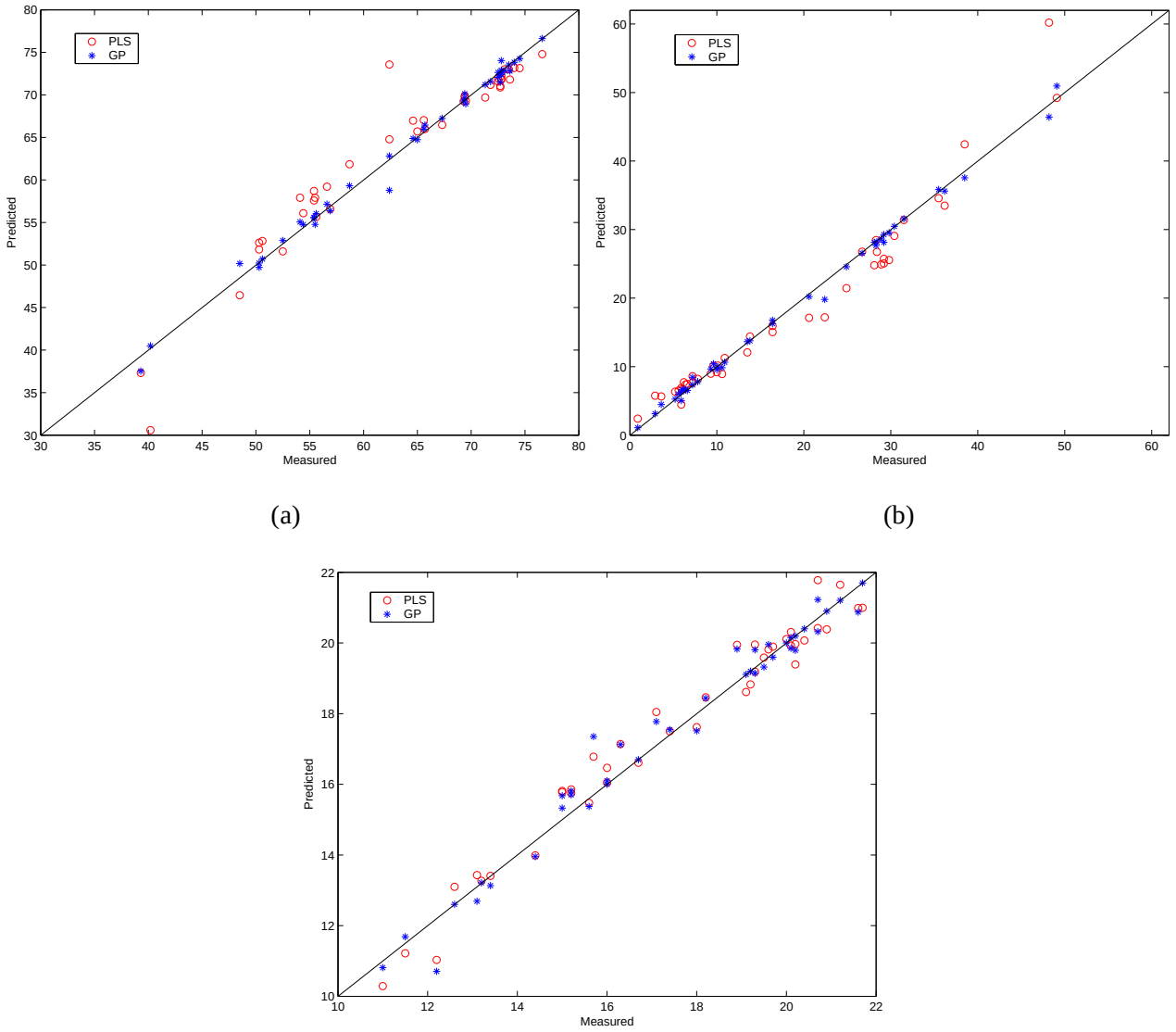

Figure 2: Predicted vs. measured plot for the “Meat” data set using PLS (RMSEP = 2.807, 2.813, 0.561) and Gaussian process (RMSEP = 0.824, 0.732, 0.504). (a): Moisture; (b) Fat; (c): Protein.

Table 1: Calibration results for the “Tablet” data set. Root mean squared errors of prediction (RMSEP) is averaged over 50 random partitions of the training and testing data.

Method	PCR	PLS	ANN	QPLS	GP
RMSEP	0.259	0.260	0.208	0.534	0.189

Table 2: Paired *t*-test on the RMSEP for different calibration methods on the “Tablet” data set.

p-value	PLS	ANN	QPLS	GP
PCR	0.48	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$
PLS		$<10^{-4}$	$<10^{-4}$	$<10^{-4}$
NN			$<10^{-4}$	8.6×10^{-4}
QPLS				$<10^{-4}$

Table 3: Calibration results for the “Meat” data set. Root mean squared errors of prediction (RMSEP) is averaged over 50 random partitions of the training and testing data.

Method	Moisture	Fat	Protein
PCR	2.392	2.556	0.695
PLS	2.368	2.550	0.679
ANN	1.130	1.418	0.674
QPLS	0.900	0.995	0.923
GP	0.820	0.861	0.559

Table 4: Paired *t*-test on the RMSEP for different calibration methods on the "Meat" data set. (a): Moisture; (b): Fat; (c): Protein.

(a)

p-value	PLS	ANN	QPLS	GP
PCR	0.56	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$
PLS		$<10^{-4}$	$<10^{-4}$	$<10^{-4}$
NN			5.2×10^{-3}	$<10^{-4}$
QPLS				0.076

(b)

p-value	PLS	ANN	QPLS	GP
PCR	0.92	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$
PLS		$<10^{-4}$	$<10^{-4}$	$<10^{-4}$
NN			$<10^{-4}$	$<10^{-4}$
QPLS				3.3×10^{-3}

(c)

p-value	PLS	ANN	QPLS	GP
PCR	0.30	0.50	$<10^{-4}$	$<10^{-4}$
PLS		0.87	$<10^{-4}$	$<10^{-4}$
NN			$<10^{-4}$	$<10^{-4}$
QPLS				$<10^{-4}$