

# Gaussian basis functions for chemometrics

Tuomas Kärnä<sup>a</sup>, Francesco Corona<sup>a</sup> and Amaury Lendasse<sup>a\*</sup>

**High-dimensional data are becoming more and more common, especially in the field of chemometrics. Nevertheless, it is generally known that most of the commonly used prediction models suffer from curse of dimensionality that is the prediction performance degrades as data dimension grows. Therefore it is important to develop methodology for reliable dimensionality reduction. In this paper, we propose a method that is based on functional approximation using Gaussian basis functions. The basis functions are optimised to accurately fit the spectral data using nonlinear Gauss–Newton algorithm. The fitting weights are then used as training data to build a least-squares support vector machine (LS-SVM) model. To utilise the reduced data dimension, relevant variables are further selected using forward–backward (FB) selection. The methodology is experimented with three datasets originating from the food industry. The results show that the proposed method can be used for dimensionality reduction without loss of precision. Copyright © 2008 John Wiley & Sons, Ltd.**

**Keywords:** dimensionality reduction; regression; nonlinearity; LS-SVM; variable selection

## 1. INTRODUCTION

In the field of multivariate data analysis high-dimensional data are becoming more and more common. This is especially true in areas related to spectral data due to constant development of more accurate spectrometers. However, in the machine learning perspective, the constantly growing data dimension causes severe problems. First of all, computational complexity of many commonly used analysis methods grows exponentially with respect to the number of variables [1]. But what is more important, the analysis suffers from the curse of dimensionality, which states that the theoretical lower bound of error increases with data dimensionality. For example it has been shown that pairwise distances in high-dimensional spaces tend to be concentrated in a small interval, which makes it difficult to measure similarity with a distance metric [2,3]. Still, most of the commonly used prediction methods such as k-nearest neighbour (k-NN) and most of kernel methods such as radial basis function networks (RBFN [1]), support vector machines (SVMs [1]) and kernel partial least squares (K-PLS [4]), rely on pairwise distances and are thus bound to suffer from the curse of dimensionality [5].

The growth of data dimension also implies that more training examples are needed for building a reliable prediction model [2,5]. However, the datasets in chemometrics tend to be small due to high costs of data acquisition. In some cases the number of spectral variables exceed the number of training examples which is a poor starting point for machine learning and it very likely leads to poor generalisation performance.

To overcome the curse of dimensionality, one can focus on studying only a small subset of the data or project the data into a smaller dimensional space. Although the first alternative is often effective, it is not efficient: finding a relevant subset can be very time consuming. On the other hand, projecting the data on a small dimensional function space often provides a straightforward way for dimensionality reduction. In the functional data analysis (FDA) approach, the samples are treated as discretised functions that are approximated by some finite function space [6]. In the case of spectrometric data the functional approach seems especially

appealing because the spectral curves are often relatively smooth and low on noise. Often standard function bases, such as the B-splines [7,8] or wavelets [9,10], are used for the approximation. However, instead of using a fixed basis it seems appealing to tune the basis functions according to the data at hand so that minimal number of coefficients (or weights) are needed for representing the data.

We propose that quasi-Newton optimised Gaussian functions are a good choice for the basis [11]. The locations and widths of the Gaussian functions are optimised for an accurate fit in the entire dataset. Consequently, the basis follows the statistical nature of the data and a good representation is obtained with a small number of basis functions. Furthermore, there is only one unknown parameter to tune: the number of Gaussian functions.

In this paper the proposed methodology is experimented with three datasets from the food industry. The goal is to predict some analytical values (such as fat content) using infrared absorption spectra. Gaussian fitting coefficients are used to train a least-squares support vector machine (LS-SVM [12]) model for the final prediction. For comparison partial least squares (PLS [13]) and principal component regression (PCR [14]) models are also experimented. To fully utilise the reduced dimensionality, forward–backward (FB) variable selection is used to select relevant basis functions. A flow chart of the proposed method is presented in Figure 1.

We have chosen LS-SVM model due to its capability to learn nonlinear phenomena. Use of a linear model is based on the assumption that the prediction problem is linear in the sense that absorption is directly proportional to concentration of the substance and a spectrum of a mixture is a linear combination of

\* Laboratory of Computer and Information Science, Helsinki University of Technology, P.O. Box 5400, FI-02015, Finland.  
E-mail: lendasse@hut.fi

a T. Kärnä, F. Corona, A. Lendasse  
Laboratory of Computer and Information Science, Helsinki University of Technology, P.O. Box 5400, FI-02015, Finland



**Figure 1.** Outline of the method.

the spectra of its components [15]. In practise, however, the assumption is not always true. Wülfert *et al.* tested additivity of NIR spectra and showed that there is significant deviation especially with water-rich mixtures [15]. Furthermore, environmental conditions (such as temperature, light and instrument variation) may also affect the measurement and introduce nonlinearities [16]. LS-SVM regression has already been used to model nonlinear dependencies in chemometrics [16,17].

The paper is organised as follows. The Gaussian fitting and nonlinear optimisation is explained in Section 2. Section 3 describes the LS-SVM model. FB variable selection is explained in Section 4. Finally, the experiments and results are presented in Sections 5 and 6, respectively.

## 2. GAUSSIAN BASIS

Consider a set of  $N$  measured absorption spectra  $\{S_i^j\}_{i=1}^m$  and corresponding analytically measured target values  $y_i$ ,  $i = 1, \dots, N$ . It is assumed that  $S_i^j$  are (possibly noisy) measurements of some continuous absorption function  $s(\lambda)$ , that is  $S_i^j = s_i(\lambda_j)$ ,  $j = 1, \dots, m$ , where  $\lambda$  is wavelength. To approximate  $s(\lambda)$  we define a set of Gaussian basis functions:

$$\varphi_k(\lambda) = e^{-\|\lambda - t_k\|^2 / \sigma_k^2}, \quad k = 1, \dots, q \quad (1)$$

where  $t_k$  is the centre and  $\sigma_k$  is the width parameter. If all the Gaussian functions are distinct, the set  $\varphi_k(\lambda)$  spans a  $q$  dimensional vector space  $\mathcal{A} = \text{span}\{\varphi_k\}$ .

Now  $s(\lambda)$  can be approximated using the basis representation:

$$\hat{s}(\lambda) = \sum_{k=1}^q \omega_k \varphi_k(\lambda) = \omega^T \boldsymbol{\varphi}(\lambda)$$

where  $\boldsymbol{\varphi}(\lambda) = [\varphi_1(\lambda), \varphi_2(\lambda), \dots, \varphi_q(\lambda)]^T$ . The weights  $\omega$  are chosen by minimising the square fitting error. The problem is formulated as

$$\min_{\omega} \sum_{i=1}^m |S_i - \omega^T \boldsymbol{\varphi}(\lambda_i)|^2 \quad (2)$$

When the locations and widths of the Gaussian functions are known, the weights  $\omega$  are obtained easily from Equation (2). The solution is the pseudoinverse  $\omega = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{y}$ , where  $\mathbf{y} = [y_1, y_2, \dots, y_m]^T$  are the values to be fitted and the matrix elements  $\mathbf{G}_{i,j} = \varphi_j(\lambda_i)$  [1].

Using this basis, any spectrum  $\hat{s} \in \mathcal{A}$  is uniquely determined by the weight vector  $\omega$ . This suggests that it is equivalent to analyse the discrete weight vectors instead of the continuous functions  $\hat{s}$ .

However, it should be noted that the functional fitting distorts distances, which can be seen by computing the distance of two arbitrary functions  $\hat{v}(\lambda) = \omega^T \boldsymbol{\varphi}(\lambda)$  and  $\hat{u}(\lambda) = \xi^T \boldsymbol{\varphi}(\lambda)$  in the func-

tion space:

$$\|\hat{v} - \hat{u}\|_{\mathcal{A}}^2 = \int_a^b ((\omega - \xi)^T \boldsymbol{\varphi}(\lambda))^2 d\lambda = (\omega - \xi)^T \boldsymbol{\Phi} (\omega - \xi)$$

$$\boldsymbol{\Phi}_{i,j} = \int \varphi_i(\lambda) \varphi_j(\lambda) d\lambda$$

Clearly, if the functions are orthonormal,  $\boldsymbol{\Phi}$  becomes an identity and the norm is equal to the norm in (Euclidian) weight space,  $(\omega - \xi)^T (\omega - \xi)$ . Generally this is not the case and distances in function space are not equivalent to those in the weight space. The distortion can be circumvented by applying a linear transformation  $\hat{\omega} = \mathbf{U}\omega$ , where  $\mathbf{U}$  is the Cholesky decomposition of  $\boldsymbol{\Phi} = \mathbf{U}^T \mathbf{U}$  [18].

### 2.1. Optimisation of locations and widths

Since the basis functions are differentiable, the locations and widths can be optimised for a better fit. The average fitting error of all functions is obtained by averaging Equation (2) over all of the sample inputs  $i = 1, \dots, N$ . Using the matrix notation given above, it can be formulated as

$$E = \frac{1}{2N} \sum_{i=1}^N (\mathbf{G}\omega_i - \mathbf{y}_i)^T (\mathbf{G}\omega_i - \mathbf{y}_i)$$

The partial derivatives are

$$\frac{\partial E}{\partial t_k} = \frac{1}{N} \sum_{i=1}^N (\mathbf{G}\omega_i - \mathbf{y}_i)^T \mathbf{G}_k^{(t)} \omega_{i,k}$$

$$\frac{\partial E}{\partial \sigma_k} = \frac{1}{N} \sum_{i=1}^N (\mathbf{G}\omega_i - \mathbf{y}_i)^T \mathbf{G}_k^{(\sigma)} \omega_{i,k}$$

Here, the matrix  $\mathbf{G}$  is considered as a continuous matrix-valued function  $\mathbf{G} = \mathbf{G}(\mathbf{t}, \boldsymbol{\sigma})$  and therefore its elements are differentiable with respect to  $t_k$  and  $\sigma_k$ . The notation  $\mathbf{G}_k^{(t)}$  and  $\mathbf{G}_k^{(\sigma)}$  stand for the  $k$ th column of  $\mathbf{G}$  differentiated with respect to  $t_k$  and  $\sigma_k$ , respectively.

Knowing the gradient, the locations and the widths can be optimised using a standard unconstrained nonlinear optimisation method. Actually, the problem is constrained to  $\sigma > 0$  but the Gaussian (1) is an even function with respect to  $\sigma$  and thus the constraint can be relaxed. In this paper, Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton method with line search is used. Similar to Newton's method, also BFGS is based on minimising second-order function approximation, but the Hessian matrix is approximated rather than computed explicitly [19].

Nonlinear optimisation requires an initial set of Gaussian functions. Since there are many local minima involved in the optimisation problem, the choice of initialisation is not trivial. However, for the sake of simplicity, initially the basis functions are distributed evenly on the data interval and the width is set to the distance between neighbouring centres.

Although the central idea in this work is to use the function fitting for dimension reduction, the nonlinear optimisation of Gaussian functions bears close resemblance to optimisation RBFN. For example Wettschreck and Dietterich [20] have presented a generalised RBFN where the locations and widths of the Gaussian kernels are optimised by gradient descent based back-propagation algorithm.

### 3. LS-SVM FOR REGRESSION

LS-SVM is a least square modification of the SVM introduced by Suykens [12]. The LS-SVM has two advantages over SVM: first, the computationally demanding quadratic optimisation problem of SVM is simplified so that it reduces to a set of linear equations which greatly decreases the computational costs. Secondly, regression SVM involves three unknown parameters while LS-SVM has only two, the regularisation parameter  $\gamma$  and the kernel width  $\theta$ , which significantly simplifies the parameter optimisation phase. SVM is sometimes presented as a method that can handle high-dimensional data, but at least when it is coupled with a kernel based on pairwise distances, such as most widely used Gaussian kernel, it cannot overcome the curse of dimensionality [5].

Consider a set of  $N$  training examples  $(\mathbf{x}_i, y_i)_{i=1}^N \in \mathbb{R}^m \times \mathbb{R}$ . The LS-SVM model is  $\hat{y} = \mathbf{w}^T \psi(\mathbf{x}) + b$ , where  $\psi: \mathbb{R}^m \mapsto \mathbb{R}^n$  is a mapping from the input space onto a higher dimensional hidden space,  $\mathbf{w} \in \mathbb{R}^n$  is a weight vector and  $b$  is a bias term. The optimisation problem is formulated as

$$\min_{\mathbf{w}, b} J(\mathbf{w}, e) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{2} \gamma \sum_{i=1}^N e_i^2$$

$$\text{so that } y_i = \mathbf{w}^T \psi(\mathbf{x}_i) + b + e_i$$

where  $e_i$  is the prediction error and  $\gamma \geq 0$  is the regularisation parameter. The dual problem is derived using Lagrangian multipliers which leads into a linear KKT system that is easy to solve [12]. Using the dual solution, the original model can be reformatted as

$$\hat{y} = \sum_{i=1}^N \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b$$

where the kernel  $K(\mathbf{x}, \mathbf{x}_i) = \psi(\mathbf{x})^T \psi(\mathbf{x}_i)$  is a continuous and symmetric mapping from  $\mathbb{R}^m \times \mathbb{R}^m$  to  $\mathbb{R}$  and  $\alpha_i$  are the Lagrange multipliers. It should be emphasised that although we formally define the high-dimensional hidden space  $\mathbb{R}^n$  and the mapping  $\psi(\mathbf{x})$ , there is no need to compute anything in the hidden space; the knowledge of the kernel  $K$  is enough. A widely used choice is the standard Gaussian kernel:

$$K(\mathbf{x}_1, \mathbf{x}_2) = e^{-\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 / \theta^2}$$

In order to obtain reliable performance measures the dataset is divided into two pieces, learning set  $C_L$  and test set  $C_T$ . The regression model is trained using the learning set data and cross-validation. The obtained model is then simulated on the independent test set  $C_T$  in order to detect possible over-fitting [1].

### 4. FORWARD-BACKWARD VARIABLE SELECTION

Selection of relevant variables is an important yet difficult task in machine learning. Irrelevant inputs introduce noise to the prediction model which decreases performance. As number of variables reduce, the parameter optimisation becomes easier. Moreover, variable selection can provide valuable information about the data at hand.

The most simple variable selection method is exhaustive search, that is trying out all the possible variable combinations. However, exhaustive search quickly becomes impossible as number of variables grow. FB selection is a faster algorithm but there is no guarantee that the optimal set of variables is found.

In FB algorithm, each variable can be in two states: 'on', meaning that it belongs to the set of selected variables or 'off' meaning that it is discarded. Given a certain initial state vector (states of all variables), the algorithm proceeds by flipping the state of each variable at a time and by computing the corresponding error measure. The flip operation that improved performance the most is accepted, and the states are flipped again (excluding the previously accepted change). The process is continued until no improvement is found. FB selection can be seen as descent in a graph where neighbouring state vectors differ with exactly one state. Such a graph contains many local minima and therefore it is advisable to initialise the process with random state vectors in addition to the ordinary 'all on' and 'all off' states.

In this work, the error measure is LS-SVM prediction error. Since the number of variables changes during FB iteration, the LS-SVM is retrained in each step.

## 5. EXPERIMENTS

The proposed methodology was experimented with three different datasets from food industry. PLS, PCR and LS-SVM regression was tested using the original inputs. On top of that the three regression models were experimented using compressed spectra that were obtained using both Gaussian and B-spline basis functions.

### 5.1. Datasets

The Tecator dataset consists of NIR absorption spectra and fat contents of 215 samples of minced pork meat [21]. Each spectrum was measured at 100 wavelengths ranging from 850 to 1050 nm using Tecator Infratec Food and Feed Analyser. The fat content ranges from 0.9 to 49.1%. First 172 spectra were used as a learning set  $C_L$  and the remaining 43 were used as a test set  $C_T$ . The training set is illustrated in Figure 2.

The second dataset contains 124 mid-infrared absorption spectra of wine samples and the goal is to determine the percentage of alcohol. The 256 spectral variables relate to wavenumbers from 400 to 4000  $\text{cm}^{-1}$ . Alcohol content ranges from 7.48 to 18.5% and the accuracy is three digits. First 94 spectra were used as a learning set  $C_L$  while the remaining 30 were regarded as a test set  $C_T$ . The spectra are illustrated in Figure 3.

The third dataset is related to the prediction of saccharose content of orange juice samples. The dataset contains absorption

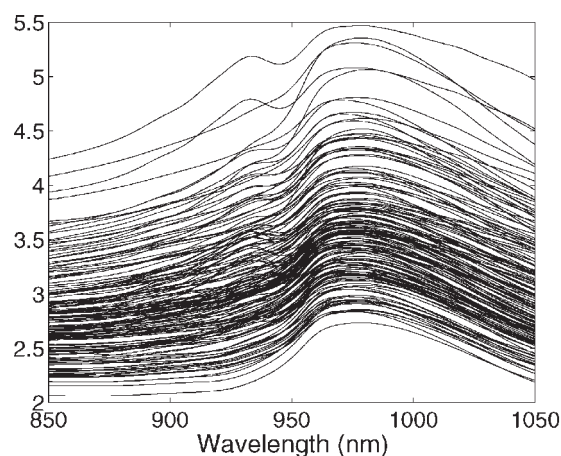
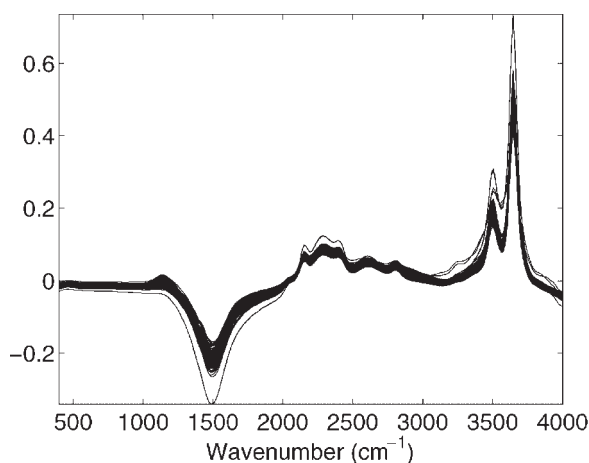


Figure 2. Spectra of the Tecator dataset.



**Figure 3.** Spectra of the wine dataset.

spectra of 700 variables measured in range 1000–2500 nm. The training and learning set contain 146 and 67 values, respectively. The saccharose content ranges from 0 to 78.8%. The spectral data are illustrated in Figure 4.

Both the wine and juice datasets are examples of cases where the number of spectral variables exceeds the number of training examples.

## 5.2. Error measures

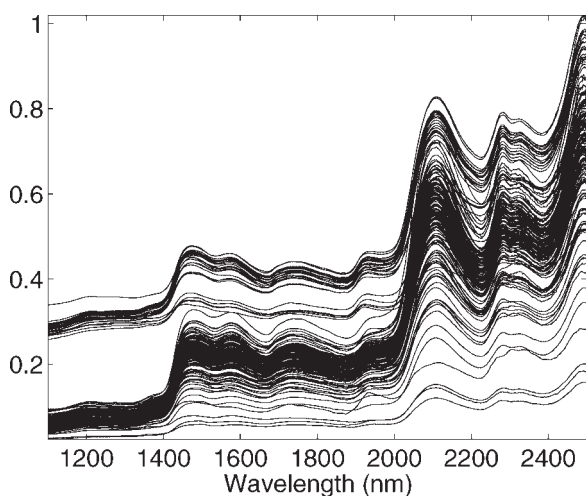
In this work we use mean square error (MSE) to measure the quality of the prediction. In the training phase, 10-fold cross-validation is used. Denoting the 10 subsets of  $C_L$  as  $C_{L,1}, C_{L,2}, \dots, C_{L,10}$ , the cross-validation MSE is defined as

$$\text{MSE}_V = \frac{1}{10} \sum_{j=1}^{10} \text{MSE}_{V,j}$$

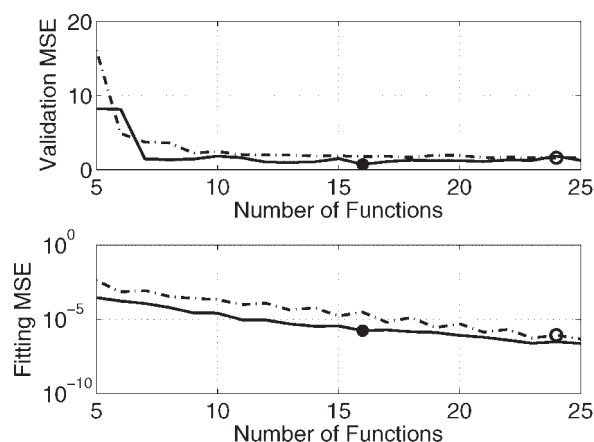
$$\text{MSE}_{V,j} = \sum_{i \in C_{L,j}} (y_i - \hat{y}_i)^2$$

where  $\hat{y}_i$  represents the estimated concentration and each  $\text{MSE}_{V,j}$  is associated to a model that has been trained with the set  $C_L \setminus C_{L,j}$ .

The generalisation performance is estimated by simulating the prediction model on the test set  $C_T$ . In this case the MSE is



**Figure 4.** Spectra of the juice dataset.



**Figure 5.** Tecator dataset. Above: LS-SVM validation error versus number of basis functions. Below: accuracy of the Gaussian function approximation versus number of kernels. Solid line stands for the Gaussian fitting and dash-dotted line stands for B-splines. The selected number of basis functions is marked with a dot (closed for Gaussian, open for B-splines).

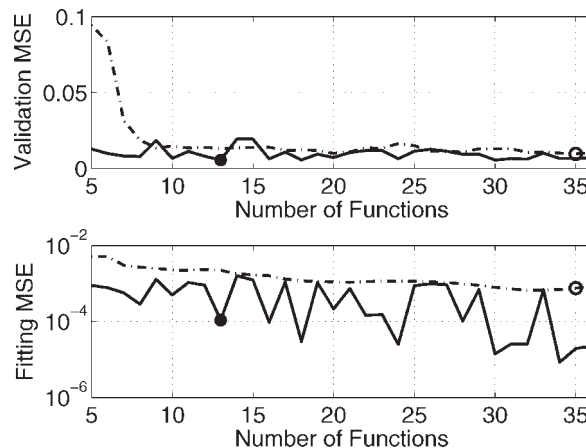
simply

$$\text{MSE}_T = \sum_{i \in C_T} (y_i - \hat{y}_i)^2$$

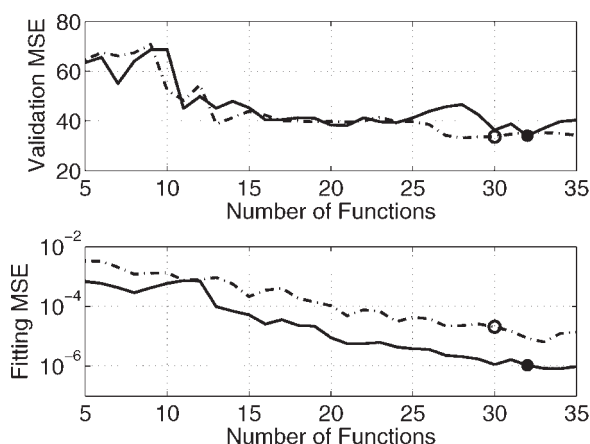
However, since the test sets are rather small, the  $\text{MSE}_T$  is not very reliable error measure and thus the validation errors must also be taken into consideration when interpreting the results.

## 5.3. Selecting number of basis functions

For each dataset, we computed both the Gaussian fitting (as explained in Section 2) and a standard cubic B-spline fitting (see Reference [7], for example). In both cases the Cholesky decomposition was used to orthonormalise the fitting weights. The number of basis functions was selected by trying out several values and selecting the one that gave the smallest LS-SVM validation error. The maximum number of functions was 25, 36 and 35 for Tecator, wine and juice datasets, respectively. The evolution of fitting accuracy and LS-SVM validation error is presented in Figures 5–7 for the three datasets, respectively.



**Figure 6.** Wine dataset. Above: LS-SVM validation error versus number of basis functions. Below: accuracy of the Gaussian function approximation versus number of kernels. Solid line stands for the Gaussian fitting and dash-dotted line stands for B-splines. The selected number of basis functions is marked with a dot (closed for Gaussian, open for B-splines).



**Figure 7.** Juice dataset. Above: LS-SVM validation error versus number of basis functions. Below: accuracy of the Gaussian function approximation versus number of kernels. Solid line stands for the Gaussian fitting and dash-dotted line stands for B-splines. The selected number of basis functions is marked with a dot (closed for Gaussian, open for B-splines).

Fitting accuracy should decrease monotonically as the number of basis functions increase, but in practise this may not always be true. It can be seen that in the case of the wine dataset the Gaussian fitting set produces a very spiky curve. This behaviour is due to the fact that the spectra of the Wine dataset is rather flat with some local spikes and therefore the optimisation of the Gaussian functions gets stuck in local minima more easily. However, it should be noted that the Gaussian basis functions provide a more accurate fit than B-splines of the same dimension. Also, especially with Tecator and wine data, the validation error tends to be smaller than with the B-splines.

The LS-SVM models were trained using 10-fold cross-validation. To acquire reliable cross-validation errors, the learning sets were randomly permuted. The same permutation was used in all tests to obtain fully comparable results. The LS-SVM parameters  $\gamma$  and  $\theta$  were optimised using four sequential  $10 \times 10$  grid searches, starting from a coarse grid and moving to a finer one near the minimum value.

As benchmarks, widely used PLS and PCR regression models were trained on the spectral data. The number of latent variables (in PLS) and number of principal components (in PCA) were selected using the same 10-fold cross-validation. The number of basis functions was also selected using the same criterion. Furthermore, to illustrate the effect of the Gaussian fitting, all the three models, PCA, PLS and LS-SVM, were trained with the raw spectral data as well. In all cases, the input variables were scaled to zero mean and unit variance before training<sup>†</sup>.

#### 5.4. FB variable selection

To initialise FB variable selection, 100 random selections were computed, out of which the 15 best were tuned with the actual FB algorithm. After the FB process, final LS-SVM validation error was computed with full grid search as described in Subsection 5.3.

In the case of the Tecator dataset, the number of Gaussian basis functions was 16, out of which 10 were selected by the FB process. Using the B-spline basis, 24 functions were chosen, out of which

11 were selected. For the Wine dataset, Gaussian fitting resulted in 13 variables out of which 8 were selected while B-spline fitting resulted in 35 variables out of which 15 were selected. The results obtained with the juice dataset are similar: Gaussian fitting resulted in 32 variables out of which 9 were selected and B-spline fitting resulted in 30 variables out of which 10 were selected. Thus in all the cases spectra were compressed remarkably, resulting in only 8–15 significant variables.

## 6. RESULTS

The prediction errors obtained with the three datasets are presented in Tables I–III.

**Table I.** Results for the Tecator dataset. Number of latent variables are given in parenthesis

Method	Number of variables	MSE <sub>V</sub>	MSE <sub>T</sub>
PLS	100 (14)	6.41	4.04
PCR	100 (20)	12.88	4.98
LS-SVM	100	1.67	1.54
Gaussian + PLS	23 (13)	6.80	5.29
Gaussian + PCR	23 (17)	7.61	5.02
Gaussian + LS-SVM	16	0.74	0.95
Gaussian + LS-SVM + FB	10	<b>0.57</b>	<b>1.16</b>
B-spline + PLS	23 (15)	6.63	4.85
B-spline + PCR	23 (23)	6.81	4.68
B-spline + LS-SVM	24	1.62	1.77
B-spline + LS-SVM + FB	11	1.00	1.22

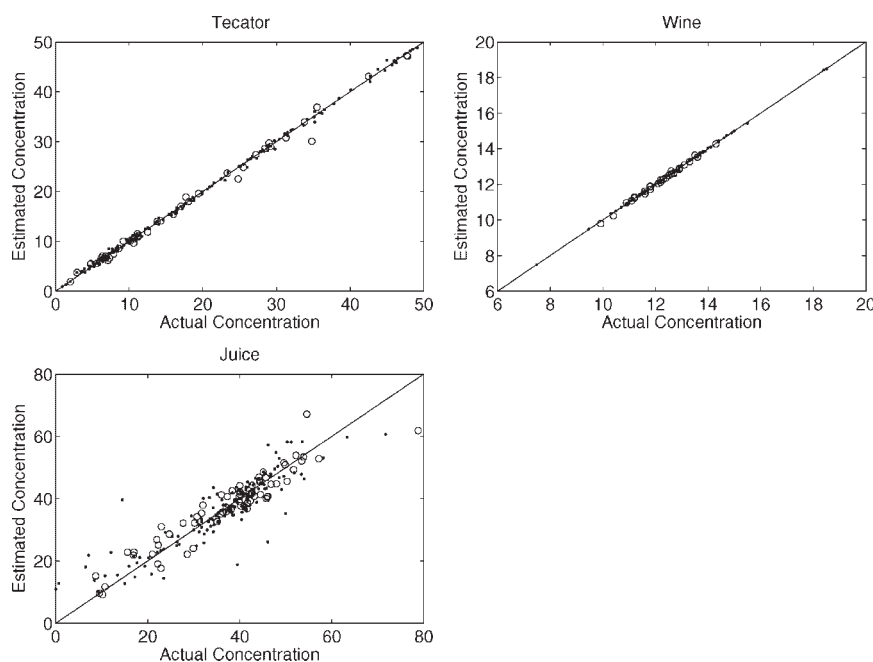
**Table II.** Results for the wine dataset

Method	Number of variables	MSE <sub>V</sub>	MSE <sub>T</sub>
PLS	256 (8)	0.0054	0.0099
PCR	256 (30)	0.0066	0.0094
LS-SVM	256	0.0285	0.0078
Gaussian + PLS	31 (20)	0.0046	0.0130
Gaussian + PCR	26 (24)	0.0054	0.0098
Gaussian + LS-SVM	13	0.0054	0.0121
Gaussian + LS-SVM + FB	8	0.0043	0.0090
B-spline + PLS	33 (18)	0.0046	0.0150
B-spline + PCR	33 (30)	0.0046	0.0313
B-spline + LS-SVM	35	0.0096	0.0064
B-spline + LS-SVM + FB	15	<b>0.0036</b>	<b>0.0062</b>

**Table III.** Results for the juice dataset

Method	Number of variables	MSE <sub>V</sub>	MSE <sub>T</sub>
PLS	700 (9)	<b>28.6311</b>	<b>18.8016</b>
PCR	700 (15)	35.9	26.5
LS-SVM	700	61.5	32.9
Gaussian + PLS	23 (20)	28.1	19.5
Gaussian + PCR	24 (24)	29.1	19.7
Gaussian + LS-SVM	32	34.1	13.6
Gaussian + LS-SVM + FB	9	23.1	18.1
B-spline + PLS	28 (18)	28.1	17.7
B-spline + PCR	28 (12)	29.6	21.1
B-spline + LS-SVM	30	38.0	18.6
B-spline + LS-SVM + FB	10	25.9	18.3

<sup>†</sup> To be more specific, the learning set was scaled to zero mean and unit variance and the same scaling was used in the test set.



**Figure 8.** Plots of actual concentration versus estimated concentration obtained with the best models. Learning set is marked with dots and the test set is marked with circles. The best model for Tecator was Gaussian + LS-SVM + FB, for Wine it was B-Spline + LS-SVM + FB and for Juice it was plain PLS.

### 6.1. Tecator

In the case of Tecator data, using nonlinear prediction method is clearly advantageous. Using LS-SVM divides the MSE roughly by 5 compared to any of the linear models. This suggests that there are some nonlinearities involved in the input–output relation. The nonlinearity of this dataset has been discussed also in the original work by Borggaard *et al.* [21].

Using the Gaussian fitting improves the performance even more: the test MSE was decreased roughly by one-third. However, combining FB variable selection to the Gaussian + LS-SVM model decreases the validation error but increases the test error. Since both the learning set and the test set are small, it is not clear whether this is due to over fitting: the validation error is an average of 10 separate estimates, so theoretically it should be more accurate than the test error. On the other hand, however, the validation error has been minimised during the training process, so it is not independent and thus not as reliable as the test error. All in all, one can only conclude that the FB selection reduces the number of variables from 16 to 10 while the overall performance stays quite the same. The performance of Gaussian + LS-SVM + FB is presented in Figure 8 where the actual target concentrations are plotted against the predicted ones.

Considering both the validation and the test error, B-splines do not perform quite as well as the Gaussian fitting. The variable selection, however, is clearly advantageous in this case.

### 6.2. Wine

In the case of Wine dataset, all the models, including plain PLS and PCR, give good results which suggests that the problem is highly linear. The good performance is clearly observable in Figure 8.

Despite the linearity of the problem, combining the function fitting with LS-SVM model yields good results. Again, considering both  $MSE_V$  and  $MSE_T$  one can conclude that the B-spline + LS-SVM + FB is the best model. However, the Gaussian + LS-SVM

(+FB) results in significantly smaller number of variables while the performance remains satisfactory.

### 6.3. Juice

If the wine dataset was an example where all the models perform reasonably well, the juice dataset is quite the opposite. Predicting the saccharose content is more difficult task, and none of the models produces satisfactory results, which is clearly seen in Figure 8.

Among the plain regression models, plain PLS performs the best. Gaussian + LS-SVM + FB and B-spline + LS-SVM + FB give slightly better results, but considering the size of the dataset, the difference can hardly be considered as significant. Therefore, due to the simplicity of PLS, one can conclude that plain PLS is the best model.

## 7. CONCLUSIONS

We have proposed optimised Gaussian basis functions as a tool for dimensionality reduction for spectrometric data. When combined with LS-SVM model, our experimental results on three datasets suggest that dimensionality can be reduced dramatically without loss of prediction accuracy.

Gaussian fitting combined with LS-SVM model performed best on the Tecator dataset and it was observed that the Gaussian fitting tends to compress the data more than the B-splines. However, B-splines gave better results in the wine dataset. Furthermore, the results obtained with the juice dataset suggest that in some, rather difficult cases, the function fitting cannot improve the prediction performance. Therefore, the choice of method depends on the dataset and the goal of the analysis. Reducing data dimensionality simplifies the regression task and speeds up variable selection considerably. Both the function fitting and the variable selection can provide the researcher

valuable information on the problem at hand, even though it would not yield clear benefits in performance.

Finally, it should be noted that in the light of these results, the importance of learning set and test set size becomes very clear. Therefore, in future the methodology should be tested on various large datasets in order to be able to obtain more reliable performance measures.

### Acknowledgements

The authors are grateful to Professor Marc Meurens for providing the juice and wine dataset.

### REFERENCES

- Haykin S. *Neural Networks: A Comprehensive Foundation* (2nd edn). Prentice-Hall: New York, USA, 1999.
- Verleysen M, François D. The curse of dimensionality in data mining and time series prediction. In *Proceedings of 8th International Work-Conference on Artificial Neural Networks*, Cabestany J, Prieto A, Sandoval DF (eds). IWANN'2005, LNCS, 2005; **3512**: 758–770.
- Beyer K, Goldstein J, Ramakrishnan R, Shaft U. When is "nearest neighbor" meaningful? In *Proceedings of 7th International Conference on Database Theory*, Beerl C, Buneman P (eds). ICDD'99, LNCS 1998; **1540**: 217–235.
- Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel Hilbert space. *J. Mach. Learn. Res.* 2001; **2**: 97–123.
- Bengio Y, Delalleau O, Le Roux N. The curse of highly variable functions for local Kernel machines. In *Proceedings of Neural Information Processing Systems*, Weiss Y, Schölkopf B, Platt J (eds). NIPS 2005, *Adv. Neural Inf. Process. Syst.* 2006; **18**: 107–114.
- Ramsay J, Silverman B. *Functional Data Analysis*. Springer-Verlag: New York, USA, 1997.
- De Boor C. *A Practical Guide to Splines*. Springer-Verlag: New York, USA, 1978.
- Alsberg BK, Kvalheim OM. Compression of nth-order data arrays by B-splines. I: theory. *J. Chemometr.* 1993; **7**: 61–73.
- Shao XG, Leung AK, Chau FT. Wavelet: a new trend in chemistry. *Acc. Chem. Res.* 2003; **36**: 276–283.
- Trygg J, Wold S. PLS regression on wavelet compressed NIR spectra. *Chemometr. Intell. Lab. Syst.* 1998; **42**: 209–220.
- Kärnä T, Lendasse A. Gaussian fitting based FDA for chemometrics. In *Proceedings 9th International Work-Conference on Artificial Neural Networks*, Sandoval F, Prieto A, Cabestany J, Graña M (eds). IWANN'2007, LNCS, 2007; **4507**: 186–193.
- Suykens J, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J. *Least Squares Support Vector Machines*. World Scientific Publishing Co.: Singapore, 2002.
- Härdle W, Liang H, Gao JT. *Partially Linear Models*. Physica-Verlag: Heidelberg, Germany, 2000.
- Vigneau E, Devaux MF, Qannari EM, Robert P. Principal component regression, ridge regression and ridge principal component regression in spectroscopy calibration. *J. Chemometr.* 1997; **11**: 239–249.
- Wülfert F, Kok WT, Smilde AK. Influence of temperature on vibrational spectra and consequences for the predictive ability of multivariate models. *Anal. Chem.* 1998; **70**: 1761–1767.
- Chauchard F, Cogdill R, Roussel S, Roger JM, Bellon-Maurel V. Application of LS-SVM to non-linear phenomena in NIR spectroscopy: development of a robust and portable sensor for acidity prediction in grapes. *Chemometr. Intell. Lab. Syst.* 2004; **71**: 141–150.
- Rossi F, Lendasse A, François D, Wertz V, Verleysen M. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometr. Intell. Lab. Syst.* 2006; **80**: 215–226.
- Rossi F, Delannay N, Conan-Guez B, Verleysen M. Representation of functional data in neural networks. *Neurocomputing* 2005; **64**: 183–210.
- Bazaraa MS, Sherali HD, Shetty CM. *Nonlinear Programming, Theory and Algorithms* (2nd edn). John Wiley and Sons: New York, NY, USA, 1993.
- Wettschereck D, Dietterich T. Improving the performance of radial basis function networks by learning center locations. *Adv. Neural Inf. Process. Syst.* 1992; **4**: 1133–1140.
- Borggaard C, Thodberg H. Optimal minimal neural interpretation of spectra. *Anal. Chem.* 1992; **64**: 545–551.