Available online at www.sciencedirect.com
SCIENCE DIRECT®
Chemometrics and Intelligent Laboratory Systems 70 (2004) 47–53
Chemometrics and intelligent laboratory systems
www.elsevier.com/locate/chemolab

Chemometric calibration of infrared spectrometers: selection and validation of variables by non-linear models

N. Benoudjit[a,*], E. Cools[b,1], M. Meurens[b,1], M. Verleysen[a,2]

[a] *Microelectronics Laboratory (DICE), Université Catholique de Louvain, Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium*
[b] *Spectrophotometry Laboratory (BNUT), Université Catholique de Louvain, 2, (8) Place Croix du Sud, B-1348 Louvain-la-Neuve, Belgium*

Received 11 December 2002; received in revised form 19 September 2003; accepted 3 October 2003

Abstract

Data from spectrophotometers form spectra that are sets of a great number of exploitable variables in quantitative chemical analysis; calibration models using chemometric methods must be established to exploit these variables. In order to design these calibration models which are specific to each analyzed parameter, it is advisable to select a reduced number of spectral variables. This paper presents a new incremental method (step by step) for the selection of spectral variables, using linear regression or neural networks, and based on an objective validation (external) of the calibration model; this validation is carried out on data that are independent from those used during calibration. The advantages of the method are discussed and highlighted, in comparison to the current calibration methods used in quantitative chemical analysis by spectrophotometry.

Keywords: Calibration; Infrared spectrometers; Non-linear

1. Introduction

In recent years, qualitative and quantitative applications of infrared spectroscopy in various chemical field including the pharmaceutical [1], food [2] and textile industries [3] have grown dramatically [4]. The chemical analysis by spectrophotometry rests on the fast acquisition of a great number of spectral data (several hundred, even several thousands).

From a chemometric point of view, the spectral data have remarkable characteristics, which make necessary their treatment by specific methods. The matrix **X** of the data may comprise more variables (spectral data) than observations (spectra). Certain columns (variables) of the matrix **X** can be practically represented as linear combinations of other columns. This situation is called collinearity, and is the source of many problems [5,6]. Studies have shown that if collinearity is present among the variables, the prediction results can get poor (see, for example, Ref. [7]). Therefore, there is a need to select variables among the candidate ones, in order to build a still suitable model with as few variables as possible. Reducing the number of variables also helps to avoid the well-known overfitting phenomenon, encountered in non-linear modelling but also in linear modelling when the number of observations is limited. In principle, all possible combinations of candidate variables should be tried for calculating a suitable model. If the original data set contains n variables, an extensive search of all possible subsets would require the design of $2^n - 1$ different models. This value grows exponentially, making an exhaustive search impractical even for moderate values of n.

In this work, we will first present the usual techniques for the selection of variables in the context of linear regression methods: stepwise multiple linear regression (SMLR), principal components regression (PCR) and partial least squares regression (PLSR). Then, we will propose to incorporate non-linear regression models (RBF networks with radial basis functions) in the variable selection process, through an incremental procedure based on a validation criterion.

* Corresponding author. Tel.: +32-10-47-25-51; fax: +32-10-47-25-98.
E-mail addresses: benoudjit@dice.ucl.ac.be (N. Benoudjit), Etienne.Cools@nefy.ucl.ac.be (E. Cools), meurens@bnut.ucl.ac.be (M. Meurens), verleysen@dice.ucl.ac.be (M. Verleysen).
[1] Tel.: +32-10-47-37-26; fax: +32-10-47-37-28.
[2] Michel Verleysen is a Senior Research Associate of the Belgian F.N.R.S. (National Fund For Scientific Research).

0169-7439/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2003.10.008

Lastly, we will present a comparison of prediction results between the different techniques.

2. Experimental and computational techniques

2.1. Variable selection: state of the art

The problem of variable selection can be defined as follows: given a set of candidate variables, select a subset that performs best (according to some criterion) in a prediction system. More specifically, let **X** be the original set of spectral data, containing n different variables (columns of **X**) and m observations or spectra (rows of **X**). The objective is to find a subset of the columns of **X**, $Z \subseteq X$ containing d variables representing the best model [6].

The methods for spectral variables selection using linear regression models usually used in spectrophotometric analysis are the following ones:

- Stepwise multiple linear regression (SMLR) procedure: the spectral variables are selected among the n available variables by respecting a criterion of optimization such as the test of assumption based on Fisher's law. At each step the F-to-enter values for all variables not yet in the model are checked and the variable with the highest significant F value is entered. Moreover, after each step the F-to-remove values for all variables already in the model are tested. If a variable is detected that does no longer significantly contribute to the regression it is rejected. The procedure is continued until no more variable fulfils the criterion to be entered or to be removed [5,8].

Instead of selecting a subset of all available variables, a number of orthogonal linear combinations of these variables can be used to reduce the dimensionality of matrix **X**.

- Principal component regression (PCR): it first consists of applying a principal components analysis (PCA) to the matrix of the spectral data. The PCA replaces the origin spectral variables, strongly redundant, by principal components (linear combinations of the origin variables), which contain almost the totality of the information; the principal components have the advantage of being uncorrelated. The scores of the most important principal components are then used as inputs for a multiple linear regression (MLR) [6].
- Partial least squares regression (PLSR) is based on principal components of both the matrix of the spectral data **X** and the dependent variables **Y**. The PLS components are calculated in order of importance by maximizing the covariance between **Y** and linear combinations of the **X** variables [6]. The properties of partial least squares regression and examples of its use have been dealt with extensively in the literature [9–12].

All these models make the assumption of the existence of a linear relation between the selected or built variables on one hand, and the characteristic to be predicted on the other. This might not be the case in the reality of certain applications, leading to a need for using non-linear models instead of linear ones. In the literature, some authors use as inputs of the non-linear models the PCA or PLS score vectors [6]. Others use incremental SMLR-like methods, but use a selection criterion based on a learning set instead of a validation one.

2.2. Variables selection and validation by non-linear models

Given these limitations, we propose a method for variables selection based on the three following principles:

1. Use of a non-linear regression model (artificial neural network);
2. Choice of the variables based on an incremental procedure (forward–backward selection);
3. Choice of the variables according to an error criterion computed on a validation set.

The combination of these three principles will lead respectively to enhanced calibration capabilities (compared to linear models), to an efficient compromise between inefficient and exhaustive searches of the variables to select, and to an objective assessment of the performances (and objective comparisons between models as a corollary).

2.2.1. Artificial neural networks

As linear models may not be sufficient to approximate a non-linear, real characteristic, non-linear models should be considered in many cases. With respect to linear methods, non-linear ones offer more capabilities, often at the price of an increased complexity.

Artificial neural networks (ANN) are largely used in applications involving classification or function approximation. Lately, it has been proved that several classes of ANN are universal function approximators [13]. Therefore, they are widely used for function interpolation [14,15] and gain more and more attention from chemists; they have already found numerous applications in data modelling (calibration and pattern recognition problems) [16].

Radial-basis function networks (RBFN) can be used for a wide range of applications primarily because they can approximate any regular function [13] and their training is faster compared to multilayer perceptrons (MLP).

MLP are trained by supervised techniques: the weights are computed by minimizing a non-linear cost function. On the contrary, the training of RBF networks can be split into an unsupervised part and a linear supervised part. Unsupervised updating techniques are straightforward and relatively fast. Moreover, the supervised part of the learning consists in solving a linear problem, which is therefore also fast, with the supplementary benefit of avoiding the problem of local

minima usually encountered when using multilayer perceptrons [16,17].

2.2.2. Radial basis function networks

Several models of RBFN exist: with or without linear and constant terms, with or without normalization, etc.…

An RBF network is a two-layered ANN. Consider an unknown function $f(\mathbf{x}):\Re^d \rightarrow \Re$. RBF networks approximate $f(\mathbf{x})$ by a weighted sum of d-dimensional radial activation functions (plus linear and independent terms). The radial basis functions are centered on well-positioned data points, called centers; the centers can be regarded as the nodes of the hidden layer. Usually, the positions of the centers and the widths of the radial basis functions are obtained by an unsupervised learning rule, whereas the weights of the output layer are calculated by a supervised, single-shot process using pseudo-inverse matrices or singular value decomposition (SVD).

Suppose we want to approximate a function $f(\mathbf{x})$ with a set of K radial basis functions $\phi_j(\mathbf{x})$, centred on the centers $\mathbf{c}_j$ and defined by

$$\phi_j : \Re^d \rightarrow \Re : \phi_j(\mathbf{x}_i) = \phi_j(\| \mathbf{x}_i - \mathbf{c}_j \|), \tag{1}$$

where $\| \cdot \|$ denotes the Euclidean distance, $\mathbf{c}_j \in \Re^d$ and $1 \leq j \leq K$.

The approximation of the function $f(\mathbf{x})$ may be expressed as a linear combination of the radial basis functions [18]:

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{K} \lambda_j \phi_j(\| \mathbf{x}_i - \mathbf{c}_j \|) + \sum_{i=1}^{d} a_i x_i + b, \tag{2}$$

where λ_j, a_i and b are the weights for the radial functions, linear and independent terms, respectively.

A typical choice for the radial basis functions is a set of multi-dimensional Gaussian kernels:

$$\phi_j(\| \mathbf{x}_i - \mathbf{c}_j \|) = \exp\left(-\frac{1}{2} \left(\frac{\| \mathbf{x}_i - \mathbf{c}_j \|}{\sigma_j} \right)^2 \right), \tag{3}$$

where σ_j is the width factor of the jth hidden unit (basis function) in the hidden layer.

2.2.3. RBFN learning strategies

Once the number and the general shape of the radial basis functions $\phi_j(\mathbf{x}_i)$ are chosen, the RBF network has to be trained properly. Given a training data set T of size N_T,

$$T = \{(\mathbf{x}_p, y_p) \in \Re^d \times \Re, 1 \leq p \leq N_T : y_p = f(\mathbf{x}_p)\}, \tag{4}$$

the training algorithm consists in finding the parameters $\mathbf{c}_j$, σ_j, λ_j, a_i and b such that $\hat{f}(\mathbf{x})$ fits the unknown function $f(\mathbf{x})$ as close as possible. Often, the training algorithm is decoupled into a three-stage procedure:

1. determining the centres $\mathbf{c}_j$ of the Gaussian kernels,
2. computing the widths σ_j of the Gaussian kernels,
3. computing the weights λ_j and independent terms a_i and b.

During the first two, stages only the inputs $\mathbf{x}_p$ of the training data set T are used. The parameters are thus adapted according to an unsupervised updating rule. In the third step, the weights λ_j, independent terms a_i and b are calculated with respect to the corresponding desired outputs; meanwhile $\mathbf{c}_j$ and σ_j remain fixed. Moody and Darken [19] proposed to use k-means clustering algorithm to find the location of the centers $\mathbf{c}_j$. Other authors use a stochastic online process (Competitive learning) method, which leads to similar results, with the advantage of being adaptive (continuous learning, even with evolving input data). The principle is:

1. to initialise the centers for example through a random choice in the training data set;
2. to use recursively all data points $\mathbf{x}_p$, and move the closest center $\mathbf{c}_j$ to data point $\mathbf{x}_p$ (best matching unit [BMU]) according to

$$\mathbf{c}_j(t+1) = \mathbf{c}_j(t) + \alpha(t)(\| \mathbf{x}_i - \mathbf{c}_j \|) \tag{5}$$

where $\alpha(t)$ is a time decreasing adaptation factor, $0 < \alpha(t) < 1$.

The second stage of the training process involves the computation of the Gaussian function widths, while fixing the degree of overlapping between the Gaussian kernels. It allows finding a compromise between locality and smoothness of the function $\hat{f}(\mathbf{x})$. First, we compute the standard deviations σ_j^c of each data cluster[4] in a classical way. Subsequently, we determine a *width scaling factor* w, common to all Gaussian kernels. The widths of the kernels are then defined as [20,21]:

$$\forall j, \sigma_j = w\sigma_j^c. \tag{6}$$

By inserting the width scaling factor, the approximation function $\hat{f}(\mathbf{x})$ is smoothed such that the generalization process is more efficient, as we allow an optimal overlapping of the Gaussian kernels. The choice of the optimal width factor is obtained by the following heuristic.

Consider a width factor set Q. We evaluate, successively, for each value $w_l \in Q$ the error criterion, chosen as the normalized mean square error (NMSE) on a validation set (see Eq. (11)). The optimal w_{opt} corresponds to the smallest error:

$$\forall l, \text{NMSE}_V(w_{\text{opt}}) \leq \text{NMSE}_V(w_l). \tag{7}$$

Once the basis function parameters are determined, the transformation between the input data and the corresponding outputs of the hidden units is fixed. The network can thus be viewed as an equivalent single-layer network with linear output units. The output is calculated by

[4] A cluster is a region associated to each center. Such a region is usually called a "Voronoi zone".

a linear combination (i.e., a weighted sum) of the radial basis function plus the independent terms.

$$W = \varphi^{+} y = (\varphi^{T}\varphi)^{-1}\varphi^{T} y, \tag{8}$$

where $W=[\lambda_1\lambda_2\ldots\lambda_K\, a_1 a_2\ldots a_d\, b]^T$ is the column vector of λ_j weight factors, independent terms a_i, and b, and y is the column vectors of y_p training data outputs. Moreover

$$\varphi = \begin{bmatrix} \phi_{11} & \phi_{12} & \ldots & \phi_{1K} & x_{11} & x_{12} & \ldots & x_{1d} & 1 \\ \phi_{21} & \phi_{22} & \ldots & \phi_{2K} & x_{21} & x_{22} & \ldots & x_{2d} & 1 \\ \phi_{31} & \phi_{32} & \ldots & \phi_{3K} & x_{31} & x_{32} & \ldots & x_{3d} & 1 \\ \ldots & & & & & & & & \\ \phi_{m1} & \phi_{m2} & \ldots & \phi_{mK} & x_{m1} & x_{m2} & \ldots & x_{md} & 1 \end{bmatrix} \tag{9}$$

where $\phi_{ij} = \exp\left(\frac{-\|\mathbf{x}_i-\mathbf{c}_j\|^2}{2\sigma_j^2}\right)$ values and $\varphi^{+}=(\varphi^{T}\varphi)^{-1}\varphi^{T}$ denotes the pseudo-inverse of φ. In practice, to avoid possible numerical difficulties due to an ill-conditioned matrix $\varphi^{T}\varphi$, singular value decomposition (SVD) is usually used to find the weights and independent terms instead of computing explicitly the pseudo-inverse matrix.

2.2.4. Forward–backward selection

We propose a method of spectral data selection based on a criterion of validation known as 'forward backward selection'. The selection of the spectral data is divided into two stages, as follows.

The first stage is the forward selection. It starts with the construction of the n possible models each using one single variable. We calculate the error criterion for each of these models and we choose the one that minimizes the criterion. This leads to the choice of the first variable. Secondly, we keep this variable, and build $n-1$ models by adding one of the remaining spectral variables. The error criterion for each one of these models is calculated, and we choose the model that minimizes this criterion. A second variable is then selected. We continue this process until the value of the error criterion increases. As detailed below, it is therefore necessary to evaluate the error criterion on a *validation set*, independent from the training set. By validation set, we mean a set of samples *not used* for training (fitting the calibration model). Depending on the research discipline, some authors use the words 'external validation set', 'external set' or 'prediction set'; the important concept is that the samples used to validate a method must be *independent* from those used for training, regardless of the terminology. Only the use of a validation set will ensure an objective evaluation of the error resulting from each model. Moreover, only the error on a validation set will increase when the number of selected variables is too large, leading to the well-known overfitting phenomenon.

The second stage is the backward selection. It consists in eliminating the least significant spectral data already selected in the first stage. If u spectral variables were selected after of the first stage, u models are built by removing one of the selected variables. The error criterion is calculated on each of these models, and the one with the lower error is selected. Once the model is chosen, we compare its error to the error of the model obtained at the preceding stage. If the new error criterion is lower, then the eliminated spectral variable is not significant and may be removed. The process is then repeated on the remaining spectral variables. The backward selection is stopped when the lower error among all models calculated at one step is higher than the error at the previous step.

As mentioned above, at each step of the forward–backward selection algorithm, the error of several models must be evaluated on data independent from the ones used for learning. This is achieved through the use of a validation set V, containing N_V spectra:

$$V = \{(\mathbf{x}_q, y_q) \in \Re^d \times \Re,\ 1 \le q \le N_V :\ y_q = f(\mathbf{x}_q)\} \tag{10}$$

The error criterion can be chosen as the normalized mean square error defined as [22]:

$$\mathrm{NMSE}_V = \frac{\frac{1}{N_V}\sum_{q=1}^{N_V}(\hat{f}(\mathbf{x}_q) - y_q)^2}{\frac{1}{N_T+N_V}\sum_{j=1}^{N_T+N_V}(y_j - \bar{y})^2}, \tag{11}$$

where N_T, N_V are the number of samples included in the training set and the validation set, respectively, $\hat{f}(\mathbf{x}_q)$ is the value predicted by the model and y_q is the actual value corresponding to spectrum q. Note that Eq. (11) normalizes the errors with respect to the standard deviation of y values in the combined learning and validation sets, the reason being to use as much data as possible to estimate this standard deviation. As this estimation does not depend on the model, the comparison of performances between models remains objective, whatever is the set used to estimate this standard deviation.

The combination of the three principles quoted above (non-linear regression, incremental procedure and choice based on a validation set) allows on one hand to benefit from the capabilities of non-linear methods to predict a physical phenomenon which is probably not linear, and on

Table 1
Comparisons of results on the wine database for the five procedures described (see text)

Calibration model	Number of variables	NMSE_V
PCR	30	0.0030
PLSR	12	0.0052
SMLR	14	0.0390
FBS-lin	17	0.0012
FBS-RBFN	**20**	**0.0009**

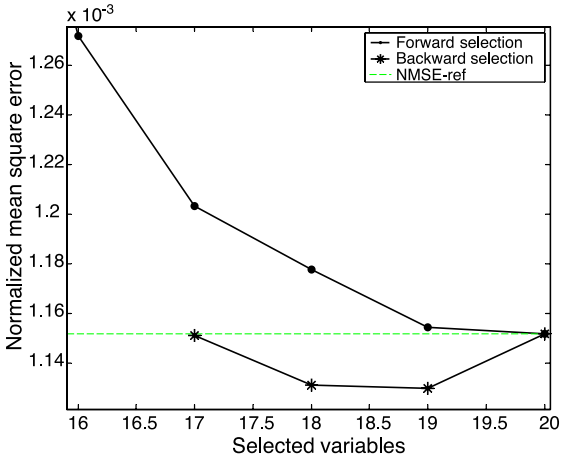

Fig. 1. Forward–backward selection with linear model on the wine samples.

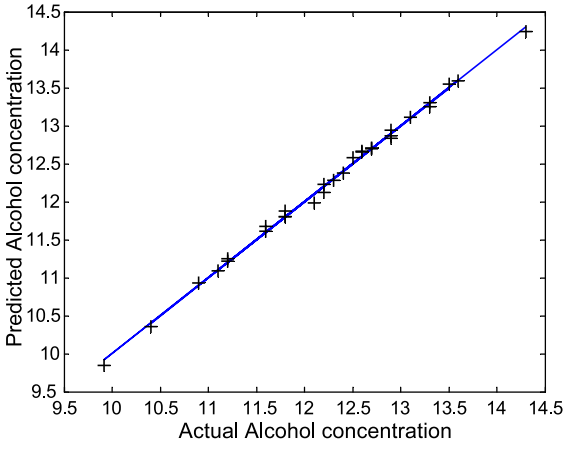

Fig. 3. Predicted alcohol concentration according to actual alcohol concentration for the wine samples with FBS-lin model.

the other hand to avoid overfitting. This procedure of variable selection thus offers, potentially, better performances when these performances are objectively measured on data independent from the learning set. This will be illustrated in the Results section.

2.3. Data sets

Two data sets were chosen to illustrate this study. The first data set relates to the determination of the alcohol concentration in wine samples measured by mid infrared spectroscopy. The training and validation sets contain 94 and 30 spectra, respectively, with 256 spectral variables that are the absorbance (log1/T) at 256 wavenumbers between 4000 and 400 cm^{-1} (where T is light transmittance through the sample thickness). The second data set relates to the determination of the sugar (saccharose) concentration in juice samples measured by near infrared reflectance spectroscopy. In this case, the training and validation sets contain 150 and 68 spectra, respectively, with 700 spectral variables that are the absorbance (log1/R) at 700 wavelengths between 1100 and 2500 nm (where R is light reflectance on the sample surface).

3. Results

The two data sets studied in this work contain 256 and 700 variables, respectively, i.e., more than the number of samples in both cases. Reducing the number of variables is thus mandatory.

3.1. Data set 1

In Table 1, the predictive ability of the PCR, PLSR, SMLR, FBS-lin and FBS-RBFN models is compared in terms of normalized mean square error on a validation set. The three first methods were detailed in Section 2.1, while the two last ones correspond to the forward–backward selection procedure detailed in Section 2.2.4, using a linear model and a non-linear one (RBFN), respectively, for the

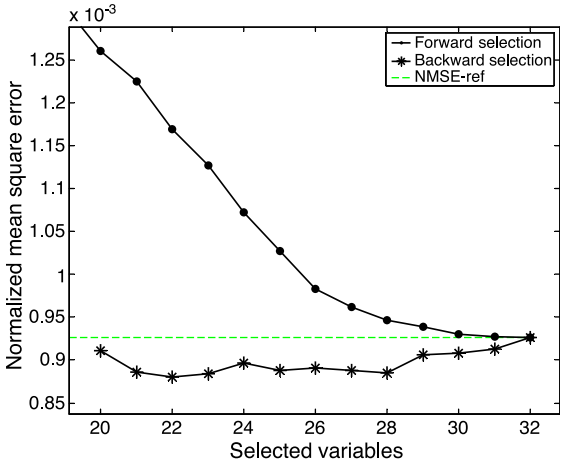

Fig. 2. Forward–backward selection with non-linear model (RBFN) on the wine samples.

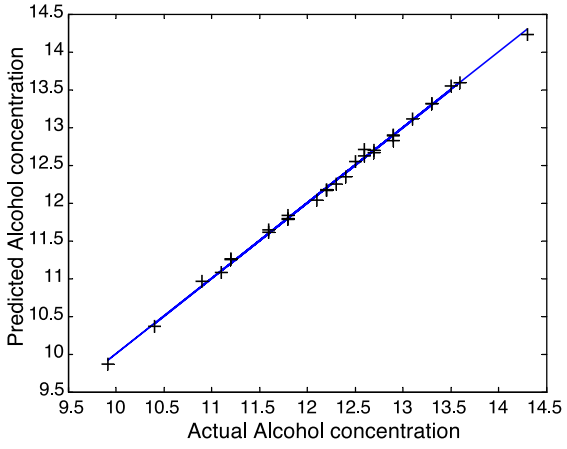

Fig. 4. Predicted alcohol concentration according to actual alcohol concentration for the wine samples with FBS–RBFN model.

Table 2
Comparisons of results on the juice database for the seven procedures described (see text)

Calibration models	Number of variables	$NMSE_V$
PCR	42	0.2596
PLSR	16	0.2435
SMLR	16	0.5137
FBS-lin	7	0.2265
PCA-RBFN	12	0.1407
PLS-RBFN	14	0.1364
FBS-RBFN	**13**	**0.0703**

prediction. Table 1 shows the NMSE on the validation set for each procedure, together with the number of parameters involved in each model.

In the PCR model the lowest $NMSE_V$ value was obtained with 30 principal components. Similarly for the PLSR model, the lowest $NMSE_V$ value was obtained with 12 latent variables. In the FBS-lin case, the initial number of variables selected by the forward selection is 20, but after the application of the backward selection this number is reduced to 17, as illustrated in Fig. 1. It can be seen in this figure that the backward procedure could have been stopped at 19 variables, leading to a smaller $NMSE_V$. We chose here to continue the procedure until the $NMSE_V$ reaches the same value as after the forward stage, thus privileging a lower number of variables. About the FBS-RBFN procedure, we tested radial-basis function networks with 2–9 nodes in the hidden layer. On the Wine database, the best result was obtained with three centers in the hidden layer; the width scaling factor was also optimized on the validation set. Also, in this case, the initial number of variables selected by forward selection is 32, but after the backward selection this number is reduced to 20, as shown in Fig. 2. In Figs. 1 and 2, the validation $NMSE_V$ is shown as a function of the number of forward–backward selected variables. Note that the set of variables obtained by FBS-lin is different from the set of variables obtained by FBS-RBFN.

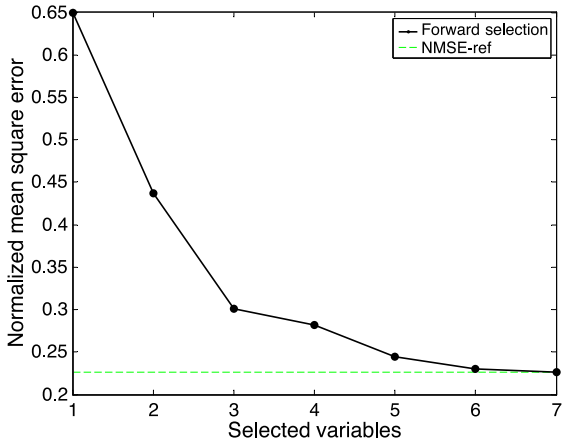

Fig. 5. Forward–backward selection with linear model on the juice samples.

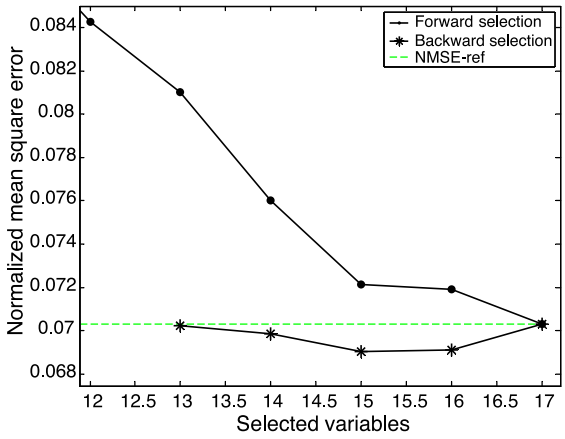

Fig. 6. Forward–backward selection with non-linear model (RBFN) on the juice samples.

Errors obtained with the forward–backward selection (both with linear and non-linear calibration models) are much lower than the errors obtained with the other procedures. The use of RBFN slightly improves the results compared to the use of a linear model. Figs. 3 and 4 show the relation between the predicted alcohol concentration and the actual alcohol concentration with both methods of variables selection.

3.2. Data set 2

On this second dataset (juice samples), we used the five methods already used for the wine dataset, and added two methods: PCA-RBFN and PLS-RBFN, where the selection of variables takes place exactly as in PCR and PLSR, respectively (thus using a linear model for the selection). Nevertheless, once the variables are selected, a non-linear model (RBFN) is used to predict the concentration in sugar (saccharose). Table 2 gives a comparison between the predictive ability of the PCR, PLSR, SMLR, FBS-lin,

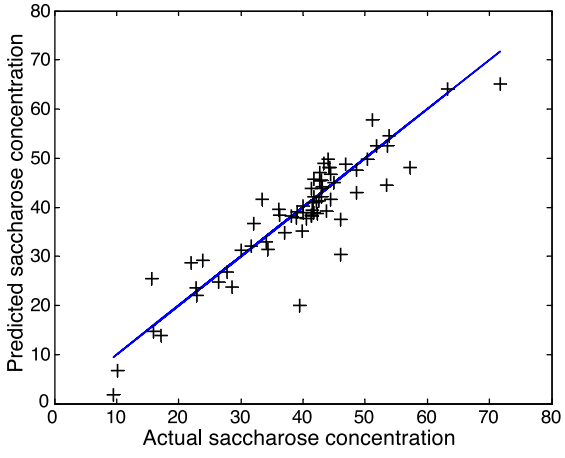

Fig. 7. Predicted saccharose concentration according to actual saccharose concentration with PLS-RBFN model for the samples of juice.

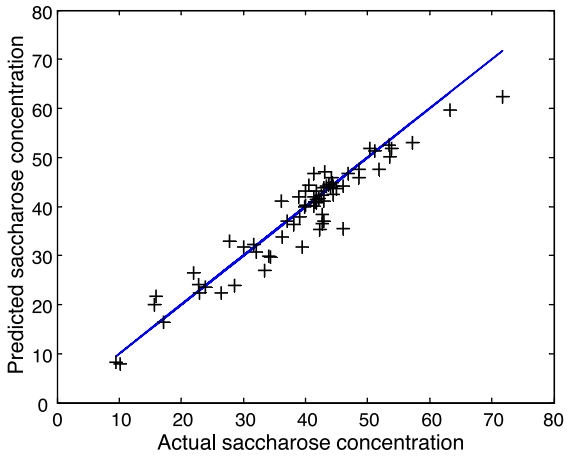

Fig. 8. Predicted saccharose concentration according to actual saccharose concentration with FBS-RBFN model for the samples of juice.

FBS-RBFN, PCA-RBFN and PLS-RBFN models in terms of normalized mean square error of the validation set.

Table 2 shows clearly that the calibration models using non-linear RBFN largely reduce the errors. The smallest error is obtained with the FBS-RBFN calibration model, with eight centers in the hidden layer.

Fig. 5 shows that in the FBS-lin case the backward selection did not reduce the number of variables previously selected by the forward stage (seven variables). On the contrary, Fig. 6 shows the effect of the backward selection on the reduction of the number of variables previously selected by the forward stage in the case of the FBS-RBFN. The number of variables is reduced from 17 to 13 for the same level of $NMSE_V$. The predicted saccharose concentration according to the actual saccharose concentration (both with PLS-RBFN and FBS-RBFN procedures) are presented in Figs. 7 and 8, respectively. These figures show the improvement obtained with the use of forward–backward selection procedure compared to the PLS one, as measured in Table 2.

In these two examples, we notice that in the mid infrared spectroscopy (wine), the calibration model FBS-RBFN decreases slightly the error, with regards to the calibration model FBS-lin. On the other hand, in the near infrared spectroscopy (juice) the error decreases considerably, around 3–4 times lower than the error obtained with the linear calibration models (PCR, PLSR, SMLR and FBS-lin).

We notice in these two examples that the increase of performances due to the use of non-linear (RBFN) calibration models is obvious in both cases, but is higher for the near infrared spectroscopy data (juice) than for the mid infrared ones (wine). Moreover, when comparing methods using similar calibration models (linear or non-linear), the forward–backward selection also clearly improves the performances with regards to other variable selection procedures.

4. Conclusions

In this work, the problem of the variables selection is treated. Two data sets were studied, containing more than 250 variables each. We proposed a procedure for spectral data selection based on the combination of three principles: non-linear regression, incremental procedure for variable selection, and use of a validation set. This procedure allows on one hand to benefit from the advantages of non-linear methods to predict a chemical data which is probably not in completely linear relation with the infrared spectra of the products, and on the other hand to avoid the overfitting phenomenon, i.e., building a model with good performances on the learning set but behaving poorly on new (validation) data. As shown in our study, the FBS-RBFN procedure can efficiently deal with the calibration of multivariate spectral data (mid and near infrared spectroscopy).

References

[1] S. Sekulic, H.W. Ward, D. Brannegan, E. Stanley, C. Evans, S. Sciavolino, P. Hailey, P. Aldridge, Anal. Chem. 68 (1996) 509.
[2] Y. Ozaki, R. Cho, K. Ikegaya, S. Muraishi, K. Kawauchi, Appl. Spectrosc. 46 (1992) 1503.
[3] M. Blanco, J. Coello, H. Iturriaga, S. Maspoch, E. Bertran, Analyst 122 (1997) 777.
[4] M. Blanco, J. Coello, H. Iturriaga, S. Maspoch, J. Pagès, Anal. Chim. Acta 384 (1999) 207–214.
[5] D. Bertrand, E. Dufour, La spectroscopie infrarouge et ses applications analytiques, Editions Tec&Doc, collection sciences et techniques agroalimentaires, 2000.
[6] T. Eklöv, P. Mårtensson, I. Lundström, Anal. Chim. Acta 381 (1999) 221–232.
[7] J.M. Sutter, J.H. Kalivas, Microchem. J. 47 (1993) 60.
[8] D.L. Massart, B.G.M. Vandeginste, L.M.C. Buydens, S. De Jong, P.J. Lewi, J. Smeyers-Verbeke, Handbook of Chemometrics and Qualimetrics: Part A, Elseiver, Amsterdam, 1997.
[9] P. Geladi, B.R. Kowalski, Anal. Chim. Acta 185 (1986) 1.
[10] H. Martens, T. Næs, Multivariate Calibration, Wiley, New York, NY, 1991.
[11] A. Hoskuldsson, J. Chemom. 2 (1988) 211.
[12] A. Lorber, L.E. Wangen, B.R. Kowalski, J. Chemom. 1 (1987) 19.
[13] J. Park, I. Sandberg, Neural Comput. 5 (1993) 305–316.
[14] D.S. Broomhead, D. Lowe, Complex Syst. 2 (1988) 321–355.
[15] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford Univ. Press, 1995.
[16] B. Walczak, D.L. Massart, Chemom. Intell. Lab. Syst. 50 (2000) 179–198.
[17] R.J. Howlett, L.C. Jain, Radial Basis Function Networks: 1. Recent Developments in Theory and Applications, Physica-Verlag, Heidelberg, Germany, 2001.
[18] T. Poggio, F. Girosi, Proc. IEEE 78 (9) (1990) 1481–1497.
[19] J. Moody, C.J. Darken, Neural Comput. 1 (1989) 281–294.
[20] N. Benoudjit, C. Archambeau, A. Lendasse, J. Lee, M. Verleysen, ESANN, Bruges, Belgium, 2002, pp. 425–432.
[21] N. Benoudji, M. Verleysen, Neural Process. Lett. 18 (2) (2003) 139–154.
[22] A.S. Weigend, N.A. Gershnfeld, Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley Publishing, Reading, MA, 1994.