ELSEVIER

Chemometrics and Intelligent Laboratory Systems 36 (1997) 95–109

Chemometrics and intelligent laboratory systems

Improving the interpretation of multivariate and rule induction models by using a peak parameter representation

Bjørn K. Alsberg *, Michael K. Winson, Douglas B. Kell

Institute of Biological Sciences, University of Wales, Aberystwyth, Ceredigion SY23 3DA, UK

Received 25 October 1996; accepted 27 January 1997

Abstract

This paper demonstrates that the interpretation of multivariate calibration and rule induction classification models can be significantly improved by adopting a new representation of data profiles (e.g., spectra and chromatograms) containing identifiable peaks. The new representation is based on estimating Gaussian or Lorentzian curve parameters of data profiles by non-linear curve fitting. All modelling is performed on these peak parameters rather than using the traditional approach where each variable is assigned a sampling point in the data profile. Loading weight plots from the multivariate methods and decision trees obtained from rule induction algorithms become more parsimonious and easier to interpret in terms of the new representation.

Keywords: Multivariate calibration; Partial least squares; Rule induction; Infrared spectra; Gaussian and Lorentzian parameters; Non-linear curve fitting; Functional representation

1. Introduction

A one-dimensional spectrum or a curve is typically represented as a vector set of ordinate values using a certain abscissa sampling frequency. However, a more efficient description of a spectrum can be accomplished if it is represented with sufficient accuracy by a *mathematical model* rather than by the individual data points. The use of such representations for the purpose of compressing digital images, speech or data files in general is indeed well known [1–12]. Using compressed representations to enhance interpretation and algorithmic efficiency of multivariate methods is less common, but has been reported in the statistics and chemometrics literature [13–20]. One way to employ a mathematical model representation of curves is to use only the *function parameters* when describing the data profile. The most obvious advantage of using such parameters instead of the original data points is the reduction in the number of bytes needed in storage. Significant reductions in computation time can also be achieved. There is another and even more important advantage with the function parameter representation: a classification or regression model using the compressed function parameters is often easier to understand and interpret for humans. The reason for this is that a data profile is not interpreted by humans in terms of individual sampling points but usually as a combination

[*] Corresponding author.

PII S0169-7439(97)00024-5

of certain patterns. In particular, we will focus on the type of pattern we refer to as a *peak*. In other words, the brain has performed a pattern recognition operation on the data profile and extracted information about it at a *higher abstraction level* than the original sampling points representation (SPR). It should be emphasised, however, that there are many examples of data profiles where a peak description is not applicable. In such situations the SPR may be the best choice or alternatively, other functional descriptions can be used.

In this article we will describe each peak object or shape by the properties height, position and width since this is the required set of input parameters for e.g. Gaussian and Lorentzian functions. By using such parameters in the modelling we have *abstracted away* the details about the individual points making up the peaks in the data profile. The important consequence of working with peak parameters is that the future classification and regression models will also be in terms of that higher abstraction level.

A variety of mathematical methods are now in common use for analysing data sets consisting of data profiles; examples are principal components analysis (PCA) [21–25], principal components regression (PCR) [26–28], partial least squares regression (PLS) [29–34] and artificial neural networks [35–45]. In addition to the traditional chemometric methods we also focus on the rule induction methodology because it gives classification models that are often easier to interpret than classical statistical classification models.

2. The peak parameter representation (PPR)

The traditional use of the original sampled points from scientific instruments to represent curves or spectra is very intuitive. There are, however, several serious drawbacks with this representation. Since each sampled point along the curve is assigned a variable to be analyzed by a multivariate algorithm, the *ordering* of the points will have no meaning. This means that if we randomly permutated all the variable columns in our data matrix containing the data profile information, it would not make any difference to the results generated by the multivariate algorithm. Why should we be concerned about ordering of the sampled points? The answer to the question is related to the fact that all information about the smoothness, continuity, type of critical points etc. are curve properties that are dependent on the ordering of infinitesimally separated points along the curve.

By assuming that the curves we want to analyze satisfy some functional relation, we *abstract away* from the actual individual sample points [14]. In most cases the sampling points used to represent the curves or spectra are redundant to the problem we wish to study. This is easy to show by gradually increasing the sampling frequency we use for collecting points along *smooth* curves. When approaching an infinite sampling frequency the information content in each variable goes to zero!

When we are using a functional representation M of a curve f, we assume that the function has a set P of parameters p_j that can be adjusted to fit close as possible to the original sampling points:

$$f(x) \approx M(x, \{p_1, p_2, \ldots, p_n\}) \tag{1}$$

where the functional representation M satisfy

$$\|f - M\| < \varepsilon \tag{2}$$

The brackets indicate a suitable distance measure between f and M. ε is the lower threshold for what is an acceptable fit.

In all future modelling we let the curve be represented by the parameter set P rather than the original set of sampled points. The ordering of the peak parameters in a multivariate modelling is as before of no importance for the results. Contrary to the sampling point representation, however, the assumptions about smoothness, continuity and other mathematical curve properties are *implicitly contained* in the functional representation and will thus not be affected by any random permutation of the parameters. The information about continuity, smoothness and other properties of the function are not contained in the ordering of the parameters as they are for the sampled points.

There is one way to view the SPR from a functional perspective. We can interpret each sampled point j as a convolution of the true underlying function $f(x)$ with the Dirac delta function:

$$\mathrm{SPR}_j = \int_{-\infty}^{\infty} f(x)\,\delta(x - x_j)\,\mathrm{d}x \tag{3}$$

where x_j are all the centre positions of the points we sampled and the Dirac function $\delta(x)$ is defined from the limit

$$\delta(x) = \lim_{\varepsilon \to 0} \delta_\varepsilon(x) \tag{4}$$

and

$$\delta_\varepsilon(x) = \begin{cases} 1/\varepsilon & 0 \le x < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The number of parameters (or really the number of bytes) in therepresenting function must of course be much less than the number of original sampled points (number of bytes required to store the sampled points). In other words: The functional representation must effectively be a *compression* of the curve. Each function parameter will therefore have significantly more information content than a single sampling point.

The choice of function type reflects the available hypotheses or the level of understanding an investigator has of the curves that are to be analyzed. By choosing a particular functional representation, a *focus* on a limited set of possible function properties is made. It is at this level the investigator can insert a priori information about the problem so that it is accessible to the multivariate algorithm. A very simple example will illustrate this: Assume the area under the curves to be studied is the real interesting variable. If this variable is not explicitly available it will be difficult to recognise from the sampling point representation alone.

In this work we will concentrate on a particular functional relation which is related to the pattern we refer to as a 'peak'. Our hypothesis here is that each data profile in a data set can be represented as a sum of n peaks:

$$\text{spectrum}_k = \sum_j^n \text{peak}_j(x) \tag{6}$$

'peak$_j(x)$' can in principle be any function we would associate with the concept of a peak. In this article, however, we have restricted the choice of peak functions to two types; the first one is the Gaussian function:

$$g(x) = a\exp\left(-\left(\frac{x-b}{c}\right)^2\right) \tag{7}$$

and the second one is the Lorentzian function:

$$l(x) = \frac{a}{4((x-b)/c)^2 + 1} \tag{8}$$

where a refers to the height, b the position and c the width of the peak. By choosing this representation, we are ignoring other important features such as baselines and asymmetrical peak shapes. Such details can of course be added as extensions to our peak representation. For example, baseline effects are often well modelled by simple polynomials and extra parameters would be necessary for this.

Thus, each spectrum k with n peaks is described by a *parameter vector* $\mathbf{p}_k$. These parameter vectors will have the following structure:

$$p_k = [(a_1 b_1 c_1)\ (a_2 b_2 c_2) \ldots (a_n b_n c_n)]. \tag{9}$$

The new data matrix now consists of rows of parameter vectors instead of the original SPR vectors. All the operations in the multivariate modelling, however, will be exactly the same as for the analysis of a matrix with spectra in the SPR.

It may sometimes not be necessary to use the full $\mathbf{p}_k$ vector. If e.g., the position parameter has no relevance to the problem studied, it can be ignored. Actually, as a 'spin-off' effect of the peak parameter representation, we can remove undesirable peak shifts by simply leaving out the position parameters in the multivariate modelling. On the other hand, if there are systematic shifts that convey information important in our modelling it should of course be included.

What are the consequences for the multivariate models generated by algorithms such as the PLS and CART (classification and regression trees) when using the peak parameter representation? In the case of PLS, it is now possible to observe directly the relations between height, position and width of peaks in e.g., loading weight and regression coefficient plots. For spectra containing few peaks, the number of variables to interpret will be much less and more intuitive than for SPR.

For CART the final decision tree has a structure such as IF $a_j < \alpha_j$ AND $c_k < \alpha_k$ THEN class i. This means we can write the final rules in English like 'if the height of peak j is less than α_j and the width of peak k is less than α_k then this data profile belongs to class i'. Such rules would be very similar to how a human expert would explain a classification model based on peak patterns in spectra.

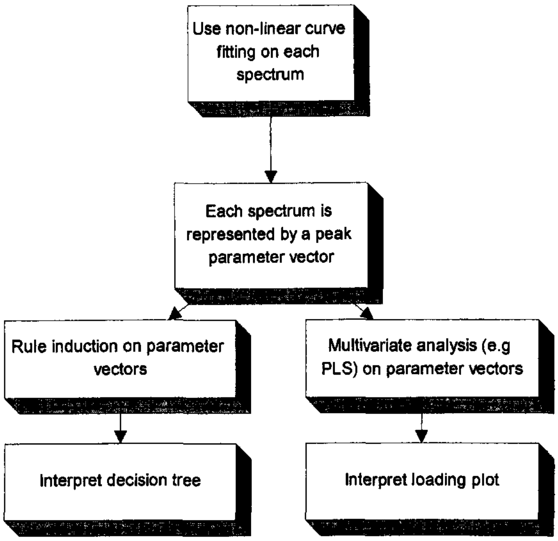
Use non-linear curve fitting on each spectrum
Each spectrum is represented by a peak parameter vector
Rule induction on parameter vectors
Multivariate analysis (e.g PLS) on parameter vectors
Interpret decision tree
Interpret loading plot

Fig. 1. Schematic presentation of the how the PPR is used in this article.

For the suggested representation to work it is necessary to formulate a set of assumptions:

- All peaks in the data set must be present in each data profile. This is to ensure comparability between different peak parameter vectors describing a data profile. If a peak is known not to be present in a data profile it will be assigned zero height and width values.
- The non-linear peak fitting procedure must be sufficiently accurate, i.e. the reconstructed data profile on the basis of the non-linear parameters only, must be very similar to the original data profile.

The second assumption may be difficult to accomplish. It is not always possible to obtain an accurate peak parameter description of a data profile. Often, the lack of accuracy can be traced to the noise level present in the spectra. It is therefore likely that a denoising pre-processing stage may be of benefit. Typical denoising method today are either based on Fourier domain or wavelet transforms [46–48]. It should be noted that curve fitting itself can be regarded as a denoising process. Another question, however, is to what degree of accuracy is needed to solve the problem at hand. Unless minor changes in the peak shapes are crucial for the classification, some deviation from the observed and predicted spectrum can be allowed.

The suggested method as applied in this article is shown in Fig. 1.

3. Methods and algorithms

3.1. Partial least squares regression (PLS)

The theory and properties of the partial least squares algorithms PLS1 (with one dependent (Y) variable) and PLS2 (with several dependent Y-variables) have been extensively studied and reported in the literature [21,22,26,28,33,34,49–57]. We will therefore give only a short description of the PLS method. The central point in the PLS paradigm is to find latent variables in the feature space which have a maximum covariance with the Y-variable(s). Thus, linear combinations of the feature space variables are found that are tilted to have maximum prediction ability for the Y-variable(s). In PLS2 one also uses linear combinations of the Y-space variables. PLS2 therefore has an iterative stage in each of the PLS2 factor calculations. The final PLS can be formulated as a regression equation:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E} \qquad (10)$$

where $\mathbf{E}$ is the matrix of errors and the estimated regression coefficients $\mathbf{B}$ are:

$$\mathbf{B} = \mathbf{X}^{+}\mathbf{Y}. \qquad (11)$$

$\mathbf{X}^{+}$ is a generalised inverse provided by the PLS algorithm. In order to obtain a prediction from the PLS model it is sufficient to use Eq. (11). In this article we compute the regression matrix $\mathbf{B}$ as demonstrated by Martens and Naes [58]:

$$\mathbf{B} = \mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{Q}^{\mathrm{T}} \qquad (12)$$

where $\mathbf{W}$ is the matrix of weights of the X-space, $\mathbf{Q}$ is the weights matrix for the Y space and $\mathbf{P}$ is the X space loading matrix.

The prediction of dependent variables on a new set of objects is done by:

$$\mathbf{Y}_{\text{test}} = \mathbf{X}_{\text{test}}\mathbf{B}. \qquad (13)$$

3.2. Rule induction

Rule induction partitions the space of sample objects into regions of single class memberships [59].

The data set is *recursively split* into smaller subsets where each subset contains objects belonging to as few different classes as possible. The 'purity' of a subset, i.e. the distribution among the classes of the objects within the set, is often measured by using the concept of *entropy* [60]. For each subset there is a set of fractions $P = [p_1, p_2, \ldots, p_J]$, of the objects belonging to the J different classes. p_i is computed as $p_i = n_i(s)/n$ where $n_i(s)$ is the number of objects belonging to class i in subset s and n is the total number of objects in subset s. Such fractions are also referred to as the *probability* of finding an object belonging to class i in the subset.

A subset with objects from one class only will have the highest possible 'purity' and the vector P of probabilities will have a structure $P_{min} = [0, 0, \ldots, 1, \ldots 0]$. The most impure vector P will correspond to equal fractions of each class:

$$P_{\max} = \left[\frac{1}{J}, \ldots, \frac{1}{J}\right] \tag{14}$$

The entropy of P

$$H(P) = -\sum_{i=1}^{J} p_i \log(p_i) \tag{15}$$

has properties in accordance with our intuitive understanding of'impurity': $H_{\min}(P) = 0$ and $H_{\max}(P) = \log_2(J)$ when $p_i = 1/J$. Thus, achieving the highest purity in a subset corresponds to *minimising* $H(P)$ by selecting an optimal partitioning. There are two major strategies for finding the best split/partitioning: *univariate* and *multivariate* rule induction. In the univariate rule induction a single variable x_i at each recursion step is found that gives rise to the purest subsets (i.e. those that have minimum entropy). In this article we will look only at numerical variables, but categorical variables can also be used [61]. In univariate rule induction a split of the input feature space corresponds to a question like 'Is $x_i < c$?' where c is some value chosen from the *finite set* of values variable x_i has among the N calibration objects. All the objects that satisfy the question are grouped into one subset and those that do not into another. Let a_k be the different outcomes of a test on variable x_i. For the numerical tests discussed here we have only two outcomes (a_1 = 'yes' and a_2 = 'no'). The entropy in a given subset of objects will thus be:

$$H(C|a_k) = -\sum_{i=1}^{J} p(c_i|a_k)\log(p(c_i|a_k)) \tag{16}$$

where $p(c_i|a_k)$ is the probability or fraction of the objects satisfying the outcome a_k and belonging to class i. $H(C|a_k)$ is read as the entropy of all the classes in C given the outcome a_k for variable x_i. Often $p(c_i|a_k)$ is computed as the number of objects belonging to class i that has outcome a_k divided by the total number of objects satisfying a_k in the subset. Assuming that the number of outcomes is two, we get two entropies calculated for each variable tested: $H(C|$'yes') and $H(C|$'no'). A measurement of the total impurity (entropy) for the variable x_i selected in the split will be related to the sum of the two individual entropies $H(C|$'yes') and $H(C|$'no'). We multiply each entropy with the fraction of objects ($p(a_i)$) that is present in the current subset, relative to the previous subset:

$$H(C|A) = \sum_{j=1}^{m} p(a_j)H(C|a_j) \tag{17}$$

where we have $m = 2$ for analyses discussed in this article. The symbol A means the set of possible outcomes to a decision question. CART (classification and regression trees) [59] is an example of a univariate rule induction method which will be used in this paper. In CART the split criteria can be changed by using a different objective function to be minimised than the one described in Eq. (16). Two very popular objective functions are referred to as Gini and Twoing. The Gini objective function is:

$$H(C|a_k) = \sum_{i \neq j}^{J} p(c_i|a_k)\left(p(c_j|a_k)\right) \tag{18}$$

which has been used in the analyses in this paper.

In standard *multivariate rule induction* we find a partition of the input feature space that depends on a *linear combination* of all the variables instead of just using one variable. We can formulate this as a question like: 'Is $\sum_{j=1}^{n} w_j x_j \leq c$?'. This type of partitioning of the data space isparticularly useful if there are any collinearities between the variables.

In this article we use the rule induction program OC1 [62] which implements both uni- and multivariate rule induction. Cross validation is used for estimating the size of the pruned decision tree. This means that the rule induction first creates a tree with too many branches and subsequently applies pruning based on cross validation criteria afterwards.

3.3. Non-linear curve fitting

A non-linear curve fitting program was written in the MATLAB version 4.2c.1 (The MathWorks, Natick, MA) language based on the optimisation toolbox [63] which has been used in the experiments. The MATLAB program is based on the original algorithm of non-linear curve fitting by Levenberg and Marquardt [64,65]. For the method to produce useful results it must be provided with

- accurate determination of the number of peaks,
- type of peak shape (e.g., Gaussian, Lorentzian),
- approximate estimate of the peak parameters (e.g. height, position and width).

The non-linear algorithm uses these estimates as a start and improves the parameters by finding the 'best' fit of the sum of the calculated peaks to that of the measured peaks. The optimising function used by the program is:

$$\chi^2 = \frac{\Sigma_{i=0}^{n}(x_i - \hat{x}_i)^2}{(n-f)}$$

where x_i is the measured data and $\hat{x}_i$ is the estimated data profile. n is the number of points in the fitted region and f is the total number of variables from all the peak and baseline functions. $n-f$ is therefore the degrees of freedom of the system. The Levenberg–Marquardt algorithm tries to minimise the χ^2. There are a number of problems associated with non-linear curve fitting. For instance, it can sometimes be difficult to obtain the solution. Further, it is risky to use non-linear curve fitting to estimate the optimal number of peaks. In this article we have used a locally made MATLAB program for peak identification which is based on assigning peak positions to the centres of gravity of the negative second derivative regions see [66], page 250–252. This identification was performed on the mean spectrum in each data set only. All other spectra in the data set used the peak assignment found from analysis of the mean spectrum. After optimising the peak parameters for the mean spectrum, we used these parameters as initial estimate for all the other spectra in the data set.

If the number of peak parameters is too high, there is a risk of the optimisation program to terminate in unfavourable local minima. The reason for this is that the search space becomes too large. If possible, it is recommended that constrained optimisation should be used in these cases which will limit the possible search space.

Another problem with non-linear curve fitting is that the procedure involved is rather time consuming and cannot be trusted to be totally automatic. Nevertheless, in cases where a deeper understanding of the problem at hand is needed we believe the extra effort needed for a non-linear fitting procedure can be justified.

3.4. Duplex data splitting

It is common to split data sets into a calibration and a validation data set. There are several strategies for splitting data, where we have adapted the 'duplex' approach first presented by Snee [67]. The idea behind the 'duplex' algorithm is to divide the original data set into two subsets which cover approximately the same region in the multidimensional feature space and have similar statistical properties. The algorithm is started by finding those two objects that are farthest apart using Euclidean distance metric. This pair of objects is placed in the validation set. In the remaining steps of the algorithm we find all the distances between pairs of objects in the calibration set and the validation set which are closest to each other. Among these pairs we chose the pair with the maximum distance. Of these two objects one is put into the calibration set, and the other is put into the validation set. The process is continued until the number of objects in the original data set is exhausted. The 'duplex' algorithm has been generalised to enable splitting of the original data set into multiple data sets (the 'multiplex' algorithm) by Jones and co-workers [68] but has not been used here since we created only one calibration and one validation set for each data set.

4. Experiments and data sets

To demonstrate the feasibility of the method four data sets will be used. Data set 1 is artificially constructed and consists of 30 spectra where each spectrum has three Gaussian peaks. The 'duplex' method was used to split this data set into a calibration and a validation set; each containing 15 objects. Both heteroscedastic and homoscedastic noise was added to produce a signal-to-noise-ratio (SNR) of 20 ± 3. No variation in the peak position was included here. See Fig. 2 for representative raw data profiles for each of the three classes.

The height of peak #1 and the width of peak #3 are here constructed to be the only variables that determine the class memberships, see Fig. 3.

Data set 2 Table 1 consists of 51 diffuse reflectance FT-IR spectra of a developed culture of the bacterium *Escherichia coli* containing the antibiotic ampicillin at different concentrations. Infra-red spectra (256 coads) for each of these samples were recorded in the wavenumber interval 4000 cm^{-1} to 600 cm^{-1} using a Bruker IFS28 FT-IR spectrometer (Bruker Spectrospin, Coventry CV4 9GH, UK) equipped with a liquid N_2-cooled MCT (mercury–cadmium–telluride) detector and a diffuse-reflectance absorbance TLC accessory (4 cm^{-1} wavenumber resolution, spectra collected at 20 s^{-1}). ASCII data were exported from the Opus software used to control the FT-IR instrument and imported into MATLAB. For this analysis we used only a subregion (1881–916 cm^{-1}) of the wavenumber range. The dependent variable for the data set is the ampicillin concentration which is in the region 0–5000 μg ml^{-1}. PLS was used for constructing the calibration model using 33 spectra for the calibration set and 18 for the validation set.

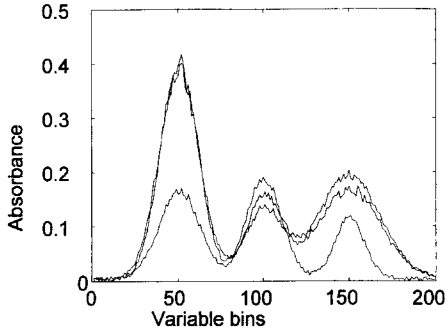

Fig. 2. Here we see one representative spectrum from each of the three classes in Data set 1.

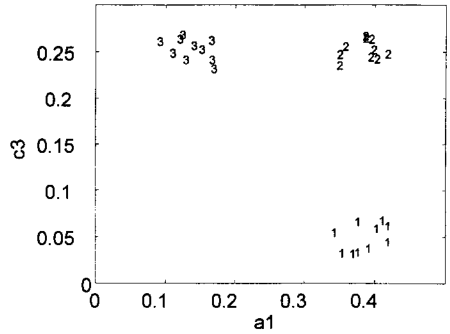

Fig. 3. The pure values of the two significant peak parameters in Data set 1: *a*1 is the height of the first peak and *c*3 is the width of the third peak. The numbers plotted correspond to the three different classes in this data set.

Data set 3 is constructed to compare the PLS loading weight plots of SPR and PPR when the underlying calibration model is based on the *width* of one of the peaks. In this case we have two Gaussian peaks where the second has a variation of the peak width in the interval [0.01, 0.4]. Both peaks are located at the same position on the axis and therefore contain no shifts.

Data set 4 consists of 150 FT-IR spectra of the three compounds histidine, glycine and sucrose in different concentrations. The span of 27 different concentration distributions of each compound is shown in Table 2. Six replicate 5 μl aliquots of 27 samples consisting of different combinations of histidine (100 mM), glycine (300 mM) and sucrose (100 mM) solutions were dried into wells in a sandblasted aluminium plate. Infrared spectra were collected and data processed as described for Data set 2 above, but

Table 1
The percentage correct prediction of the validation set for PPR and SPR of Data set 2. The first column indicates the number of PLS factors used, the second indicates the data set using all peak parameters (i.e., peak positions *b* are included for each peak), the third column indicates the PLS model on peak parameters without the positions and the fourth column indicates the SPR results

PLS factor	PPR (with peak pos.)	PPR (without peak pos.)	SPR
1	76.87%	78.19%	68.01%
2	90.81%	89.99%	84.23%

Table 2
Percentage of histidine, glycine and sucrose solutions used in Data set 4. Six replicates of each combination were analyzed by FTIR

Histidine	Glycine	Sucrose
100	0	0
90	10	0
90	0	10
80	10	10
70	20	10
70	10	20
60	30	10
60	20	20
60	10	30
0	100	0
10	90	0
0	90	10
10	80	10
20	70	10
10	70	20
30	60	10
20	60	20
10	60	30
0	0	100
10	0	90
0	10	90
10	10	80
20	10	70
10	20	70
30	10	60
20	20	60
10	30	60

using 16 coads. Initially we had 6 replicates of each concentration distribution, but found that 12 of the glycine replicates were outliers and therefore removed from the data set. This data set was created to be a classification data set where the objective was to identify the three different compounds given a spectrum. The class membership was determined by finding the compound with the maximum concentration. The DUPLEX algorithm was used to create the calibration and validation data sets.

5. Results

5.1. Data set 1

There are three different classes of objects in this data set.Representative spectra in SPR for each of the classes are shown in Fig. 2. The classification rule for this data set was artificially constructed and is shown in Fig. 3. Two peak parameters are sufficient for classification: the height of peak #1 ($a1$) and the width of peak #3 ($c3$). Each spectrum was inserted into the MATLAB non-linear Levenberg–Marquardt optimisation routine to estimate the parameters for the three peaks. The parameters for the first spectrum were used as starting point for the estimation since each of the three peaks had the same positions for all the 30 samples, see Fig. 2. The estimated parameters were subsequently analyzed by the univariate CART algorithm (using the OC1 program implementation). The estimated decision tree is shown in Fig. 4. This is in perfect agreement with the true decision tree and when it was applied to the validation set there was *no classification error*. Note that the decision tree in Fig. 4 displays the same information as Fig. 3: IF the height of peak #1 ($a1$) is less than 0.25 it is CLASS 3. IF the height of peak #1 ($a1$) is more than 0.25

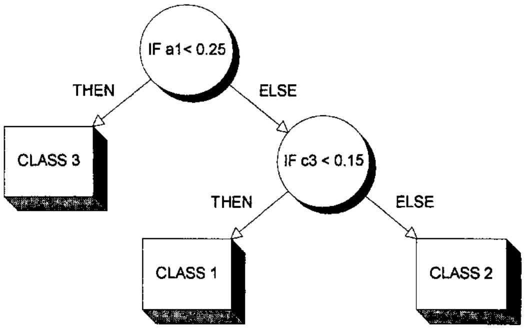

Fig. 4. The CART decision tree on Data set 1 using PPR.

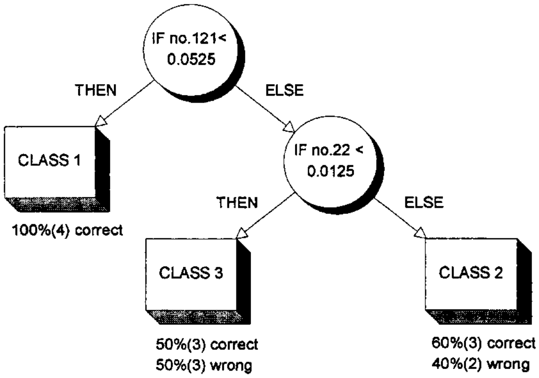

Fig. 5. CART decision tree on Data set 1 in the SPR. The prediction error of this decision tree on the validation set is 33%.

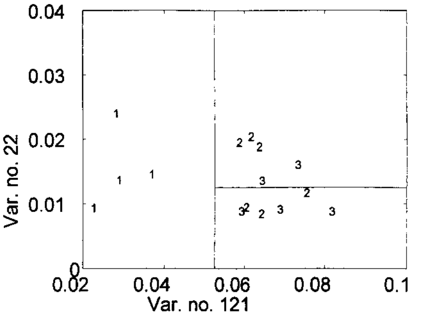

Fig. 6. A 2D illustration of the decision rule for the two significant variables Nos. 121 and 22. The rule was found by applying CART to Data set 1 with the SPR.

AND the width of peak #3 is less than 0.15 it is CLASS 1 ELSE it is CLASS 2.

We now look at the decision tree made on the basis of the SPR. The variables 22 and 121 were found from CART to be the most significant. The prediction error on the validation set was 33%. Fig. 5 is a presentation of the corresponding decision tree. In Fig. 6 we see the result of the prediction on the validation set. The lines added in this 2D plot together with the true class identities of the objects, correspond to the decision rule found. All objects belonging to class 1 have been correctly classified, whereas the rule has problems with distinguishing between class 2 and 3 objects. In addition to the poor predictive ability, the resulting classification model was not unique. This means that in the neighbourhood of variables No. 22 and 121 there are other variables that can also be used in classification models that perform equally or even better (e.g., using variables No.25 and 124 resulted in 20% prediction error). That variable No. 121 is selected to be important does not indicate that peak #3 is important in the modelling. In fact, variable No. 121 is closer to peak #2 than peak #3.

5.2. Data set 2

Eleven Lorentzian peaks were selected as important in each of the 51 IR spectra. These peaks were among those suggested by the peak detection program mentioned above. A typical spectrum (both the observed and that reconstructed on the basis of the eleven Lorentzian peak functions) is shown in Fig. 7 together with all the individual Lorentzian bases making up the spectrum (dashed lines, bottom part of figure). The reason for using Lorentzian functions in the curve fitting is because this is the most commonly encountered band shape in infrared spectrometry [69]. The statistical results of the non-linear curve fitting indicates that the fitting is satisfactory: The mean correlation (between observed spectrum and reconstructed), r^2, for all the spectra was 0.99 ± 0.004 and the mean root mean square (RMS) value between observed spectrum and reconstructed was 2.4 ± 0.4.

PLS was applied to both the SPR and the PPR to observe the differences in model quality and prediction ability. For the PPR variables we applied autoscaling before PLS analysis as peak heights, positions and widths are on different scales and in absolute numbers are often quite different. For example, the peak height parameter would get much more influence in the PLS model than the peak width without autoscaling. Autoscaling is *not* performed for the SPR since the relative absorption variable heights are for specific regions in the spectrum essentially proportional to the ampicillin concentration.

Using the validation set for both representations as a criterion, we found that the PPR using all (a, b, c) parameters (resulting in a total of 33 variables) had an optimal PLS model at $A = 10$ factors with a 8.5% prediction error. The SPR (251 variables) had an optimal PLS model at $A = 7$ factors with a 7.6% prediction error. From a prediction error perspective the

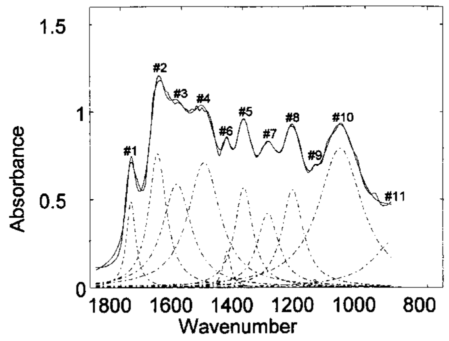

Fig. 7. This figure shows the different peaks modelled by the non-linear curve fitting procedure. Both the observed and estimated spectrum is shown in this figure and as can be seen the fit is very good. Data set 2.

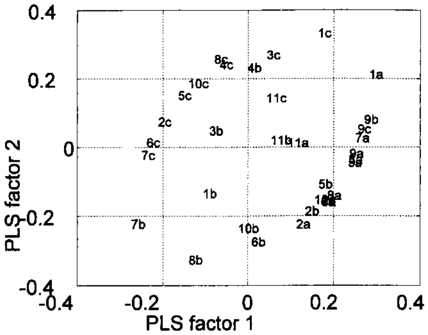

Fig. 8. PLS score plot of Data set 2 using PPR. Here all the *b*-coefficients are included.

SPR is slightly better than the PPR. Since we are here more interested in the interpretation aspects of the models, we are focusing on the first two PLS factors for both representations. The prediction error for the PPR is then 9.2% and for the SPR it is 15.8%. In other words, the new representation has better prediction power for the first two factors. Since the additional PLS factors improve the prediction by just a few percent in both representations we will ignore them in the following discussion. The loading weight plot for the PPR is shown in Fig. 8. The quasi-circular structure of the loading weight plot is caused by the autoscaling procedure [70]. As expected the height of peak #1, which is dominant for ampicillin, has a very high positive loading weight along the first PLS component. The width of peak #1 is dominant for the second PLS component. In fact, most of the variables along the positive side of PLS component No. 2 describe the change in peak *widths*. On the negative side of the second component there is an excess of variables describing peak *shifts*. One surprising result from the loading weight plot is peak #9's importance along PLS component No. 1. From the plot we can see that the height, shift and width of peak #9 is proportional to the ampicillin concentration. This increase is reflected in a corresponding decrease in the widths of peaks #2, #6 and #7 (all have less than −0.2 in loading weight along PLS factor 1).

The corresponding loading weight plots for the SPR are not that conclusive regarding the interpretation, see Fig. 9 for plotting PLS factor 1 versus 2 and Fig. 10 for plotting both PLS factors separately. Both representations agree on the fact that the height of peaks #3 and #5 are important for component one and vary in proportion to the ampicillin concentration. The PPR, however, has given the height of peak #2 much less loading weight than in the SPR. Some of these differences are probably caused by the fact that we use autoscaling in the PPR and no scaling in the SPR (an autoscaling in SPR did not improve the predictive ability). Along the second component for the SPR peaks #1 and #2 again dominate.

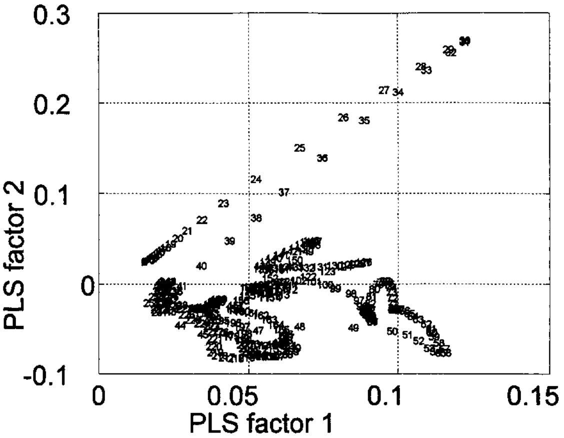

Fig. 9. This is the plot of the two first PLS loading weight vectors for Data set 2. 68% correct prediction after first PLS factor and 84% after the second.

It was surprising to find that some shift parameters were given relatively high loading weights. To investigate whether this was an artefact or not we removed all the *b*-coefficients from the data set and repeated the analysis. Taking out these parameters reduced the predictive ability of the PLS model slightly

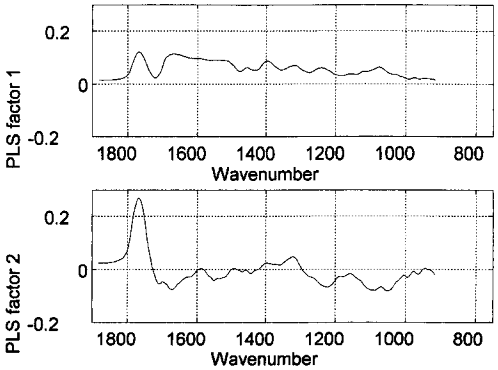

Fig. 10. This is the plot of the two first PLS loading weight vectors for Data set 2.

(10% prediction error after two PLS factors), see Fig. 11 for a plot of the new loading weight plot. Now, peak #9 is not so important for the prediction. Heights of the peaks #2, #5 and #6 are now among the variables with highest loading weight along the first PLS component. There is, however, another more striking effect which now can be seen after removal of the peak position parameters. The height parameters in PPR more or less dominate the first component whereas the widths dominate the second. The peak that does not follow this pattern is #10 which is in fact has a width variable that has the second-to-highest loading weight along component one, height of #1. This 'clustering' of the width and height parameters can be interpreted as when the heights of the peaks are increasing the widths are *decreasing*.

5.3. Data set 3

This data set was constructed to investigate the effects on the loading plot patterns when the dependent variable is influenced by peak *width* changes. The underlying calibration model used in the example was chosen to be:

$$y = a_1 + 2c_2$$

The PLS analyses were performed on both the SPR and the autoscaled peak parameter vectors. Not surprisingly, both models were optimal (using cross validation) for one PLS factor. For visualisation reasons only, the plots in Fig. 12 also include the second PLS component. The upper part of Fig. 12 shows the loading weight plot for the sample point representation and the bottom part the loading weight plot for the PPR. Of course, it is a matter of opinion which plot is the easier to interpret, but we feel confident in saying that the bottom loading weight plot quite easily conveys the message that both the height of peak 1 and the width of peak 2 are very important for predicting y and that they are positively correlated. The same type of information cannot as far as we can see be extracted from the sample point representation loading weight plot.

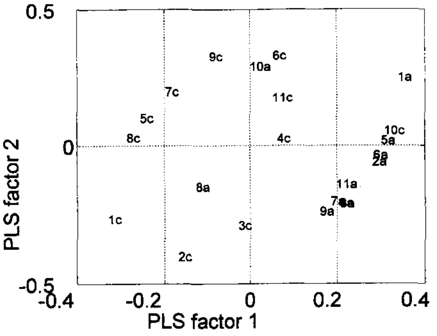

Fig. 11. PLS score plot of Data set 2 using PPR. Here all the *b*-coefficients have been removed.

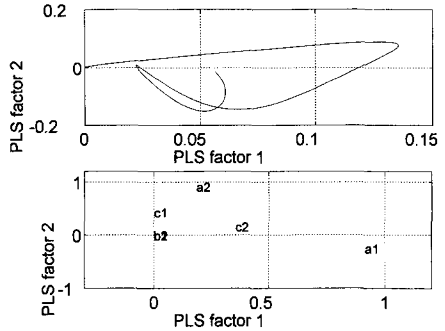

Fig. 12. The PLS result for the SPR and the PPR where the underlying calibration model is dependent on the *width* of the second peak. It is not difficult from the lower PLS loading weight plot to see that the variable marked '*c*2' has a high loading weight along the same latent variable as the height of peak 1 ('*a*1'). This fact is not very obvious from the upper plot for the SPR. Data set 3.

5.4. Data set 4

Each spectrum was recorded in the wavenumber interval 4000 cm^{-1} to 600 cm^{-1}. Curve fitting in the complex fingerprint region was avoided and we selected as subregion 4000 cm^{-1} to 2000 cm^{-1} (see Fig. 13 which shows the spectra for the three compounds in this region). This region is mainly dominated by broad Gaussian like peaks which are due to a continuum of vibration frequencies from various hydrogen bonds. Based on the mean spectrum of this region, the peak finding algorithm mentioned above was used to locate approximately the different peaks. Five Gaussian peaks seem to explain sufficiently the variation in the spectrum. To obtain good starting values for the final curve fitting of all spectra we computed the mean spectrum of the three classes (compounds) histidine, glycine and sucrose. Each of these spectra were fitted to the five Gaussian peaks

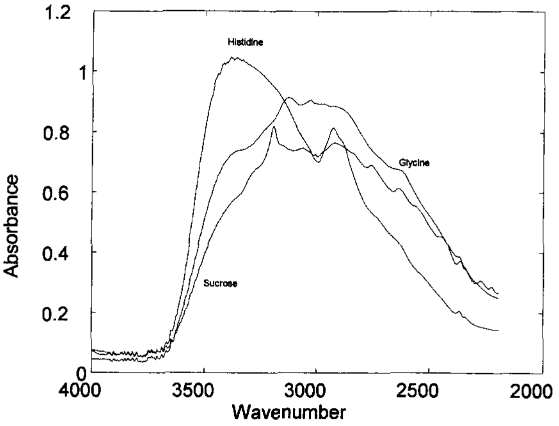

Fig. 13. The three mean vectors of the three classes belonging to histidine, glycine and sucrose in Data set 4.

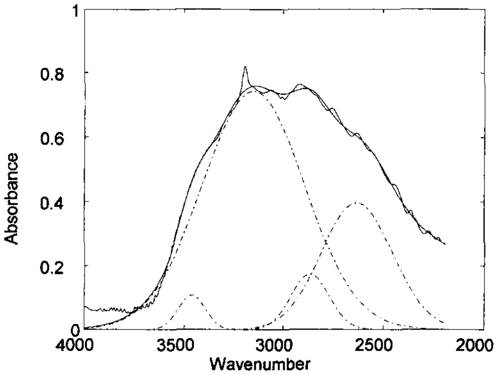

Fig. 15. The mean of the fitted Gaussian curves together with the mean of glycine spectra. The four first of the Gaussian primitives that represents the curve are shown in dashed lines. The fifth Gaussian is very broad and function as a baseline off-set (not shown). Data set 4.

giving start value parameters better suited for each class. The results of the curve fitting for the total data set was satisfactory: The mean RMS error is 3.3 ± 1.0 (median RMS 3.0). The mean correlation (r^2) is 0.997 ± 0.003 (median correlation 0.998). To get a visual impression of the curve fitting, see Fig. 14 (shows the mean vector of class histidine together with the mean of the reconstructed curve and its Gaussian components). Analogous figures for the glycine and sucrose classes are shown in Fig. 15 and Fig. 16.

Both PPR and SPR representations of Data set 4 were analyzed with rule induction. Here the classical univariate CART algorithm was used since its results are usually very simple to interpret. In all analyses we used full cross validation to guide the tree pruning. DUPLEX was used to separate the data into calibration and validation data sets. The pruned classification tree for PPR is shown in Fig. 17. The decision rule is based on one variable only: the height of peak #1. 71 out of the 75 spectra in the validation set were correctly classified. For the SPR we found that the pruned decision tree gave worse predictions (50 out of 75 correct) than the unpruned tree (70 out of 75

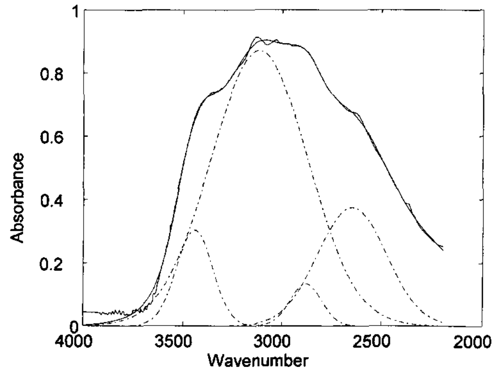

Fig. 14. The mean of the fitted Gaussian curves together with the mean of histidine spectra. The four first of the Gaussian primitives that represents the curve are shown in dashed lines. The fifth Gaussian is very broad and function as a baseline off-set (not shown) Data set 4.

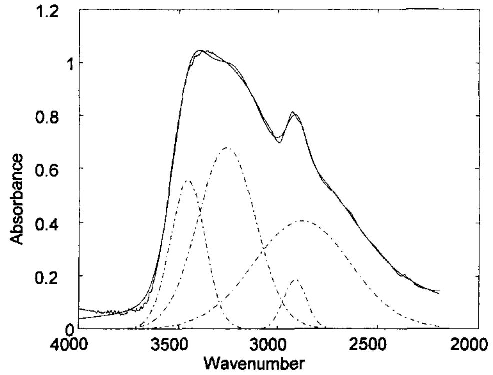

Fig. 16. The mean of the fitted Gaussian curves together with the mean of sucrose spectra. The four first of the Gaussian primitives that represents the curve are shown in dashed lines. The fifth Gaussian is very broad and function as a baseline off-set (not shown).

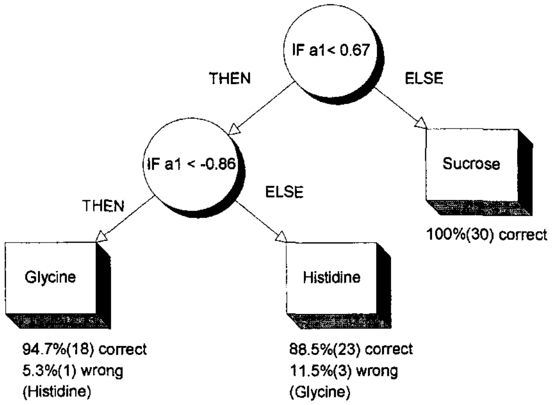

Fig. 17. The final CART decision tree of the PPR of Data set 4. The rule is very simple since it depends only on the height value of the first peak. 71 out of 75 objects in the validation set were correctly classified. The word 'correct' corresponds to the percentage overlap between the model prediction in the square output boxes and what was observed in the validation set. Below the word 'wrong' is the class assigned by the model which was not correct.

correct). The final decision tree for SPR is shown in Fig. 18. Here three wavenumber bins are found to be important, 142 (3456 cm^{-1}), 157 (3398 cm^{-1}) and 260 (3005 cm^{-1}). In other words, SPR needed a more complex model which in addition also has about the same prediction ability. What is the reason for this? We found this to be a typical example of where PPR should perform better than SPR. In Fig. 19 we have plotted the position of peak #1 for the different spectra. We see clearly a shift which is located approximately in the wavenumber region 3500 cm^{-1}

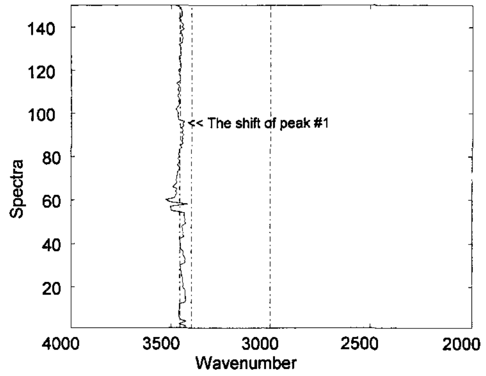

Fig. 19. A demonstration of why the SPR performs worse than PPR on Data set 4. Peak #1 has a region of shift between approximately 3500 cm^{-1} and 3400 cm^{-1}. The SPR cannot follow the shift but can try to 'cover' the region of shift by using more variables.

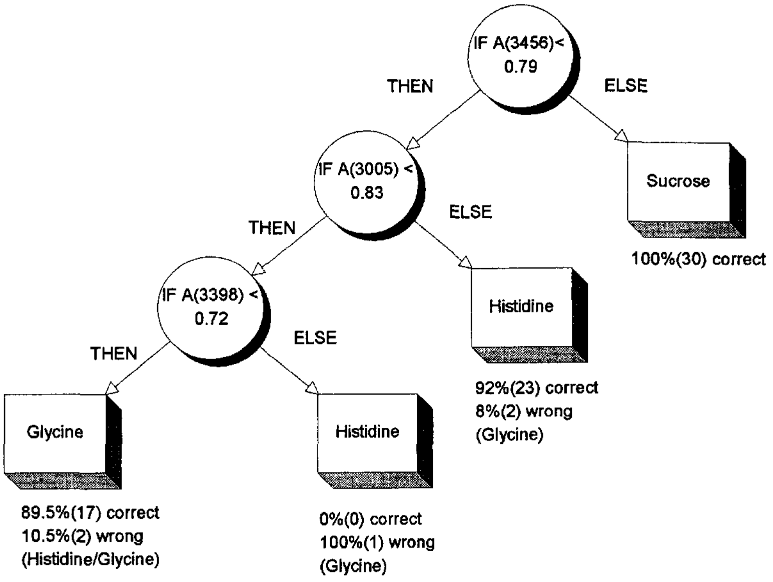

Fig. 18. CART decision tree for the SPR on Data set 4. This rule is more complicated than the corresponding rule for the PPR and has less predictive ability: 70 out of 75 objects in the validation set were correctly classified. The word 'correct' corresponds to the percentage overlap between the model prediction in the square output boxes and what was observed in the validation set. Below the word 'wrong' is the class assigned by the model which was not correct. $A(x)$ is here used to indicate that the absorption at wavelength x is used in the model.

and 2800 cm^{-1}. As a rough rule, peak #1 of the three classes are ordered sucrose < glycine < histidine. Apparently, this shift tendency is not significant enough for the CART algorithm to pick out as a stable SPR variable for classification. It is theoretically impossible for the SPR explicitly to contain the shift information of a peak. The second best thing any algorithm working on a SPR can do is to localise several variables spanning the region of the shift. This is most likely what has happened in this case. We point out in particular the two SPR variables 142 (3456 cm^{-1}) and 157 (3398 cm^{-1}) in the plot of Fig. 19 and we observe that they are located approximately in the variable region of the peak #1 position. In a sense both CART models on SPR and PPR contain approximately the same information, but PPR manages to present it in a compact and much more accessible form.

6. Discussion

The method presented is not primarily constructed to increase the prediction accuracy, but rather to make the final multivariate classification model easier to interpret. Of course, too much prediction error cannot be tolerated and the limit for tolerance must be predetermined. If minor changes in the peak shapes are important for the classification, it is necessary to employ different strategies to solve the problem. In such cases it may be possible to use other functional representations of the spectra, e.g. B-splines or wavelets. Any failure of using the height, position and width as peak parameters in a modelling indicates that the underlying modelling relations do not depend on these particular geometrical shape indicators.

We have, however, seen examples of improved prediction ability (compared to SPR) by using the PPR. The reason for improved prediction ability may be similar to what happens when variable selection is performed. The PPR is definitely a reduction of the original number of variables but could also be viewed as a variable selection procedure where the necessary features needed in the modelling are more *concentrated* than in the traditional SPR.

When the level of representational abstraction is discussed it may be more fruitful to use a classification scheme employed in mathematics rather than chemistry. In the introduction we suggested that sampling points are at lower abstraction level than peaks. This kind of hierarchy of abstraction levels can also be found in object-oriented programming. Here one starts with a baseclass where all objects within have certain properties. The baseclass has subclasses where *additional* properties are specified. For analysis of spectra, we could have defined a baseclass called 'function'. All properties concerned with functions in general would be included here, but this is too general for our purposes and thus subclasses would be necessary. Within the class of functions we have peak functions like the Gaussian and the Lorentzian, but also polynomial and fractal functions. By classifying according to mathematical properties such as types of critical points and smoothness we can include a rich pool of knowledge into final classification models. It would then be possible to 'zoom-in' on specific functional properties in relation to a calibration or classification problem.

References

[1] W.Y. Zou, Smpte J. Soc. Motion Pict. Telev. Eng. 102 (1993) 127–131.

[2] P.S. Pincetl, J.R. Merril, T.E. Piemme, M. D. Comput. 10 (1993) 42–49.

[3] G.J. Lu, Comput. Commun. 16 (1993) 202–214.

[4] R.A. Haddad, A.N. Akansu, A. Benyassine, Opt. Eng. 32 (1993) 1411–1429.

[5] A.E. Jacquin, Proc. IEEE 81 (1993) 1451–1465.

[6] N. Akrout, R. Prost, R. Goutte, Image Vision Comput. 12 (1994) 627–637.

[7] H.B. Li, A. Lundmark, R. Forchheimer, IEEE Trans. Image Process. 3 (1994) 589–609.

[8] D.L. Ruderman, Network-Comput. Neural Syst. 5 (1994) 517–548.

[9] S. Wong, L. Zaremba, D. Gooden, H.K. Huang, Proc. IEEE 83 (1995) 194–219.

[10] H. Raittinen, K. Kaski, Int. J. Modern Phys. C 6 (1995) 47–66.

[11] C.F. Barnes, S.A. Rizvi, N.M. Nasrabadi, IEEE Trans. Image Process. 5 (1996) 226–262.

[12] M. Unser, A. Aldroubi, Proc. IEEE 84 (1996) 626–638.

[13] B.K. Alsberg, O.M. Kvalheim, J. Chemom. 7 (1993) 61–73.

[14] B.K. Alsberg, J. Chemom. 7 (1993) 177–193.

[15] B.K. Alsberg, O.M. Kvalheim, Chemom. Intell. Lab. Syst. 24 (1994) 43–54.

[16] B.K. Alsberg, O.M. Kvalheim, Chemom. Intell. Lab. Syst. 24 (1994) 31–42.

[17] B.K. Alsberg, E. Nodland, O.M. Kvalheim, J. Chemom. 8 (1994) 127–145.

[18] J.O. Ramsay, X. Wang, R. Flanagan, Appl. Stat. J. R. Stat. Soc. C 44 (1995) 17–30.

[19] J.O. Ramsay, J. R. Stat. Soc. B 58 (1996) 495–508.

[20] J.O. Ramsay, K.G. Munhall, V.L. Gracco, D.J. Ostry, J. Acoust. Soc. Am. 99 (1996) 3718–3727.

[21] A. Höskuldsson, J. Chemom. 6 (1992) 307–334.

[22] A. Höskuldsson, J. Chemom. 9 (1995) 91–123.

[23] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.

[24] E. Oja, H. Ogawa and J. Wangviwattana, IEICE Trans. Inf. Syst., E75D (1992) 366–375.

[25] S. Wold, K. Esbensen, P. Geladi, Chemom. Intell. Lab. Syst. 2 (1987) 37–52.

[26] S. DeJong, J. Chemom. 7 (1993) 551–557.

[27] M. Lipp, Z. Lebensm.-Unters. Forsch. 202 (1996) 193–198.

[28] H. Martens and T. Næs, Multivariate Calibration, Wiley, Chichester, 1989.

[29] I.S. Helland, Scand. J. Stat. 17 (1990) 97–114.

[30] P. Geladi, Chemom. Intell. Lab. Syst. 2 (1987) 257–257.

[31] P. Geladi, Chemom. Intell. Lab. Syst. 5 (1988) 2–3.

[32] S. Wold, H. Martens, H. Wold, Lect. Notes Math. 973 (1983) 286–293.

[33] S. Wold, A. Ruhe, H. Wold, W.J. Dunn, Siam J. Sci. Stat. Comput. 5 (1984) 735–743.

[34] S. Wold, N. Kettanehwold, B. Skagerberg, Chemom. Intell. Lab. Syst. 7 (1989) 53–65.

[35] R. Andrews, J. Diedrich, A.B. Tickle, Knowledge-Based Syst. 8 (1995) 373–389.

[36] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[37] A.B. Bulsari (Ed.), Neural Networks for Chemical Engineers, Elsevier, Amsterdam, 1995.

[38] B. Cheng, D.M. Titterington, Stat. Sci. 9 (1994) 2–30.

[39] S.S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan, New York, 1994.

[40] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Redwood City, CA, 1991.

[41] T. Hrycej, Modular Learning in Neural Networks, Wiley, New York, 1992.

[42] D.B. Kell, C.L. Davey, Bioelectrochem. Bioenerg. 28 (1992) 425–434.

[43] Y. Liu, B.R. Upadhyaya, M. Naghedolfeizi, Appl. Spectrosc. 47 (1993) 12–23.

[44] T. Masters, Signal and Image Processing with Neural Networks, Wiley, New York, 1994.

[45] D. Michie, D.J. Spiegelhalter, C.C. Taylor (Eds.), Machine learning: Neural and Statistical Classification, Ellis Horwood, Chichester, 1994.

[46] D.L. Donoho, I.M. Johnstone, G. Kerkyacharian, D. Picard, J. R. Stat. Soc. B 57 (1995) 301–337.

[47] D.L. Donoho, IEEE Trans. Inf. Theory 41 (1995) 613–627.

[48] D.L. Donoho, I.M. Johnstone, C. R. Acad. Sci. I 319 (1994) 1317–1322.

[49] S. Wold, C. Albano, W.J. Dunn, K. Esbensen, S. Hellberg, E. Johansson, W. Lindberg, M. Sjostrom, Analusis 12 (1984) 477–485.

[50] S. Wold, Technometrics 35 (1993) 136–139.

[51] H. Martens, L. Izquierdo, M. Thomassen, M. Martens, Anal. Chim. Acta 191 (1986) 133–148.

[52] M. Martens, H. Martens, S. Wold, J. Sci. Food Agric. 34 (1983) 715–724.

[53] R. Manne, Chemom. Intell. Lab. Syst. 2 (1987) 187–197.

[54] T.R. Holcomb, M. Morari, Comput. Chem. Eng. 16 (1992) 393–411.

[55] P. Geladi, B.R. Kowalski, Anal. Chim. Acta 185 (1986) 1–17.

[56] I.E. Frank, Chemom. Intell. Lab. Syst. 8 (1990) 109–119.

[57] S. DeJong, J. Chemom. 9 (1995) 323–326.

[58] H. Martens, T. Naes, Multivariate calibration, Wiley, Chichester, 1989.

[59] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, Pacific Grove, CA, 1984.

[60] C.E. Shannon, Bell Syst. Tech. J. 379 (1948) 379–423; 623–656.

[61] J.R. Quinlan, Mach. learn. 1 (1986) 81–106.

[62] S.K. Murthy, S. Kasif, S. Salzberg, J. Artif. Intell. Res. 2 (1994) 1–32.

[63] A. Grace, Optimization Toolbox, The MathWorks Inc., Natick, MA, 1994.

[64] D. Marquardt, SIAM J. Appl. Math. 11 (1963) 431–441.

[65] K. Levenberg, Q. Appl. Math. 2 (1944) 164–168.

[66] D.L. Massart, B.G.M. Vandeginste, S.N. Deming, Y. Michotte, L. Kaufman, Chemometrics: A textbook, Elsevier Science Publishers B.V., New York, 1988, pp. 250–252

[67] R.D. Snee, Technometrics 19 (1977) 415–428.

[68] A.K. Jones, D.B. Rowland, Anal. Chim. Acta (1996), submitted.

[69] P.R. Griffiths, J.A. Haseth, Fourier Transform InfraredSpectrometry, John Wiley and Sons, New York, 1986.

[70] R. Goodacre, M.J. Neal, D.B. Kell, Anal. Chem. 66 (1994) 1070–1085.