APPENDIX V.4.

Neural Networks Tutorial

1. INTRODUCTION

The aim of this tutorial is to describe in a systematic way the chemometric development of a calibration model for spectroscopic data analysis by Neural Networks (NN). This process consists of many steps, from the pre-treatment of the data to the utilisation of the calibration model, and includes for instance outlier detection (and possible rejection), validation and many other topics in chemometrics. The literature often describes alternative approaches for each step, e.g. several tests have been described for the detection of outliers. The objective of this tutorial is to present some of the main alternatives, to help the reader in understanding them and to decide which ones to apply. Thus, a complete strategy for calibration development is presented. The tutorial is limited to NN of the multi-layer feed-forward type (also called multi-layer perceptron, MLP) with the error back-propagation learning rule that is the most popular.

Tutorials on NN in chemistry were proposed by Smits *et al.* [Smi], Svozil *et al.* [Svo] (the latter contains an extensive list of Internet resources for NN) and different types of applications of NN to spectroscopy were reviewed by Cirovic [Cir]. Bos *et al.* [Bos1] presented an excellent overview of practical aspects of NN in quantitative analysis, and a tutorial review on NN in multivariate calibration was proposed by Despagne and Massart [Des1].

In the following, bold capital letters **X** will denote matrices, bold lowercase letters **x** will denote column vectors and normal lowercase letters x will refer to real numbers.

NN allow to estimate relationships between one or several input variables called independent variables or descriptors $\mathbf{x_i}$ (e.g. absorbances at different wavelengths) and one or several output variables called dependent variables or responses **y** (e.g. concentration of a target analyte), without any *a priori* assumption of a specific model form. Information in an NN is distributed among multiple cells (nodes) and connections between the cells (weights). An example of MLP is displayed in Figure 1, for a model with four descriptors $\mathbf{x_1, x_2, x_3, x_4}$ and a single response **y**.

The descriptors are presented to the NN at the input layer and then weighted by the connections w'_{ij} between the input and hidden layer. Hidden layer nodes receive simultaneously weighted signals from input nodes and perform two tasks: a summation of the weighted inputs followed by a projection of this sum on a non-linear transfer function f_h, to produce what is called an activation. That a non-linear function of the weighted input sums is defined. In turn, hidden nodes activations are weighted by the connections w_j'' between the hidden and output layer and forwarded towards the nodes of the output layer. Output nodes perform a summation of incoming weighted signals and project the sum on their specific transfer function f_o (linear or non-linear). In Figure 1 a single response y is modelled and the output layer contains only one node. The output of this node is the estimated response $\hat{y}$ that can be expressed as:

$$\hat{y} = f_o\left(\theta'' + \sum_{j=1}^{nh} w_j'' f_h\left(\sum_{i=1}^{nd} w_{ij}' x_i + \theta'\right)\right) \qquad (1)$$

where nd and nh are the number of input variables and hidden nodes, respectively.

The NN models are defined by sets of adjustable parameters determined by an algorithm, not *a priori* by the user. Adjustable parameters are the weights w_{ij}', w_j'' and biases θ', θ'' that act as offset terms by shifting the transfer functions horizontally. They are determined with an iterative procedure called training or learning. The adjustable parameters are given initial random values, then training starts and proceeds in two steps. First, a forward pass (Figure 1a) is performed through the NN with a set of training samples with known experimental response y. At the end of the pass, the magnitude of the error between experimental and predicted responses (the cost function) is calculated and used to adjust all weights of the NN, in a backpropagation step (Figure 1b). These two steps constitute an iteration or epoch. A new forward pass is then performed with the training samples and the optimised parameters. The whole procedure is repeated until convergence is reached. This means that a pre-specified or acceptably low error level is reached. The most popular algorithm to adjust weights during training is the gradient descent algorithm based on the estimation of the first derivative of the error with respect to each weight [Flet].

Training an NN is an optimisation problem, where one seeks the minimum of an error surface in a multi-dimensional space defined by the adjustable parameters. Such surfaces are characterised by the presence of several local minima, saddle points or canyons. It must be understood that the NN will probably not find the absolute minimum of the error surface, but a local minimum relatively close to the absolute minimum and acceptable for the problem considered. The most important feature of NN applied to regression is that they are universal approximators: they can fit any continuous function defined on a compact domain (a domain defined by bounded inputs) to a pre-defined arbitrary degree of accuracy [Hor].

In classical calibration the basic equation is:

signal = f (concentration)

In multivariate calibration one often does not know the concentrations of all the compounds that influence the absorbance at the wavelengths of interest so that this model cannot be applied. The calibration model is then written as the inverse:

concentration = f (signal)

The accuracy of the calibration, defined as the deviation between the experimental and the true result and therefore compromising both random errors (precision) and systematic errors (bias), is better for inverse than for classical calibration [Cen1]. Having to use inverse calibration is in no way a disadvantage. In this tutorial, it will be explained how NN can be used to develop inverse calibration models relating a set of concentrations **y** to a set of signals **X**.

The tutorial is written for spectroscopic methods in general, but with specific emphasis on near-infrared (NIR). Figure 2 gives a general flow-scheme of the steps needed to develop a calibration model. Each step is discussed in detail below.

In this flow scheme we have considered a situation in which the minimum of a priori knowledge is available and where virtually no decisions have been made before beginning the measurement and method development. In many cases information is available or decisions have been taken which will have an influence on the flow scheme adopted. For instance, it is possible to decide before the measurement campaign that the initial samples will be collected for developing the model and validation samples will be collected later, so that no splitting is considered (Section 8), or to be aware that there are two types of samples but that a single model is required. In the latter case, the person responsible for the model development knows or at least suspects that there are two clusters of samples and will probably not determine a clustering tendency (Section 5), but verify visually that there are two clusters as expected.

The scheme of Figure 2 is also in a certain sense a chemometric maximum. In this tutorial we wanted to emphasise the chemometrics: it is assumed that the model developer wants to document all aspects of the model development in a rather formal and chemometric way. This is not always the case and, in practice, many decisions are based on visual observation or knowledge of the problem. Whatever the situation and the scheme applied in practice, the following steps are usually present:

- visual evaluation of the spectra before and after pre-treatment: do replicate spectra largely overlap, is there a baseline offset, etc.
- visual evaluation of the **X**-space, usually by looking at score plots resulting from a principal component analysis (PCA) to look for gross outliers, clusters, etc. In what follows, it will be assumed that gross outliers have been eliminated.
- visual evaluation of the y-values to verify that the expected calibration range is properly covered and to note possible inhomogeneities, which might be remedied by measuring additional samples.
- selection of the samples that will be used to train, optimise and validate the model and the scheme which will be followed.
- a first modelling trial to decide whether it is possible to arrive at the expected quality of model and to detect gross non-linearity if it is present.
- refinement of the model by e.g. considering elimination of possible outliers, selecting the optimal number of input variables, etc.
- final validation of the model.
- routine use and updating of the model.

2. REPLICATES

Different types of replicates should be considered. Replicates in **X** are defined as replicate spectroscopic measurements of the same sample. The replicate measurement should preferably include the whole process of measuring, for instance including filling the sample holders. Replicates of the reference measurements are called replicates in **y**. Since the quality of the prediction does not only depend on the measurement but also on the reference method**,** the acquisition of replicates both in **X** and **y**, i.e. both in the spectroscopic measurement and the reference analysis, is recommended. However, since the spectroscopic measurement, e.g. NIR, is usually much easier to carry out, it is more common to have replicates in **X** than in **y**. Replicates in **X** increase the precision of the predictions which are obtained. Precision is used here as a general term. Depending on the way in which the precision is determined, a repeatability, an intermediate precision or a reproducibility will be obtained [Iso1,Iso2]. For instance, if all replicates are measured by the same person on the same day and the same instrument a repeatability is obtained.

Replicates of **X** can be used to select the best pre-processing method (Section 3) and to compute the precision of the predicted values from the multivariate calibration method. The predicted y-values for replicate calibration samples can be computed. The standard deviation of these values includes information about the experimental procedure followed, variation between days and/or operators, etc. The mean spectrum for each set of replicates is used to build the model. If the model does not use the mean spectra, then in the validation step the replicates cannot be split between the calibration and test set (Section 8).

It should be noted that, if the means of replicates were used in the development of the model, means should also be used in the prediction phase and vice versa, otherwise the estimates of precision derived during the modelling phase may be wrong.

Outlying replicates must first be eliminated by using the Cochran test [cen3], a univariate test for comparing variances that is described in many statistics books. This is done by comparing the variance between replicates for each sample with the sum of these variances. The absorbance values constituting a spectrum of a replicate are summed after applying the pre-processing method (Section 3) that will be used in the modelling stage and the variance of the sums over the replicates is calculated for each sample. The highest of these variances is selected. Calling the object yielding this variance i, we divide this variance by the sum of the variances of all samples. The result is compared to a tabulated critical value at the selected level of confidence. When the value for object i is higher than the critical one, it is concluded that i probably contains at least one outlying replicate. The outlying replicate is detected visually by plotting all replicates of object i, and removed from the data set. Due to the elimination of one or more replicates, the number of replicates for each sample can be unequal. This number is not equalised because by eliminating some replicates of other samples information is lost.

3. SIGNAL PRE-PROCESSING

3.1. Detection of non-linearity

Sources of non-linearity in NIR methods are described in [Mil2], and can be summarised as due to:

- violations of the Beer-Lambert law.
- detector non-linearity.
- stray light.
- non-linearity in diffuse reflectance/transmittance.
- chemically-based non-linearities.
- non-linearities in the property/concentration relationship.

As a general rule, one should not try to build an NN model unless the data set is known or suspected to be non-linear. Therefore, some diagnostic tools are necessary to detect the presence of non-linearity in a data set. The simplest approach -which is in many cases sufficient to detect the presence of non-linearity- is to plot the property of interest versus the different measurement variables, or combinations of these variables such as PC scores. If these plots are inconclusive then one should build a linear model with MLR, PCR or PLS. Visual inspection of the residuals ($\mathbf{y}-\hat{\mathbf{y}}$) of the linear model *versus* each descriptor $\mathbf{x}_i$ retained in the model, versus the experimental response $\mathbf{y}$ and versus the estimated response $\hat{\mathbf{y}}$ should then be performed to detect non-linearities.

Recently, Centner *et al.* [Cen2] reviewed a number of more sophisticated graphical and numerical methods to detect non-linearities. They cited the Mallows augmented partial residuals plot (APaRP, Figure 3) [Malo,Coo] combined with a runs test as the most promising approach for detecting non-linearity in a lower PC, which may be masked by the inclusion of higher PCs. A limitation of the APaRP plot is that it allows only the detection of non-linearities that can be described or approximated by a quadratic term. A classical statistical way to check for non-linearities in one or more variables in multiple linear regression is based on testing whether the model improves significantly when a squared term is added. This can be applied also when the variables are PC scores to the linear model [Dra]. One compares

$$\mathbf{y}_i = b_0 + b_1\mathbf{t}_i + b_2\mathbf{t}_i^2 + \mathbf{e}_i \qquad (2)$$

to

$$\mathbf{y}_i = b_0^* + b_1^*\mathbf{t}_i + \mathbf{e}_i^* \qquad (3)$$

$\mathbf{t}_i$ being the score values for the object i on the PC investigated. A one-sided F-test can be employed to check if the improvement of fit is significant. One can also apply a two-sided t-test for checking if b_2 is significantly different from 0. The calculated t-value is compared to the t-test value with (n-3) degrees of freedom, at the desired level of confidence.

The runs test [Mass] examines whether an unusual pattern occurs in a set of residuals. In this context a run is defined as a series of consecutive residuals with the same sign. Figure 3 would lead to 3 runs and the following pattern: “ + + + + + + + - - - - - - + + +“.

From a statistical point of view long runs are improbable and are considered to indicate a trend in the data, in this case a non-linearity. The test therefore consists of comparing the number of runs with the number of samples. Similarly, the Durbin-Watson test [Dra] examines the null hypothesis that there is no correlation between successive residuals. In this case no trend occurs. The runs or Durbin-Watson tests should be carried out as a complement to the visual evaluation and not as a replacement.

All these methods are lack-of-fit methods and it is probable that they will also indicate lack-of-fit when the reason is not non-linearity, but the presence of outliers. Centner *et al.* [Cen2] emphasised the need for careful outlier detection before drawing conclusions about the presence of non-linearity in a data set. Outliers with high leverage can pull the regression line and lead to an incorrect estimation of the number of runs. Conversely, some outlier detection methods can wrongly flag as outliers samples that are high leverage points responsible for non-linearity in the data [Sek]. Caution is therefore required. We prefer the runs or the Durbin-Watson tests, in conjunction with visual evaluation of the partial response plot or the Mallows plot.

3.2. Reduction of non-linearity

It is possible that the non-linearity detected can be easily corrected for instance with an appropriate preprocessing. In such case it will not be necessary to build a NN model and more classical tools like PCR or PLS will be applicable.

A very different type of pre-processing is applied to correct for the non-linearity due to measuring transmittance or reflectance [Osb]. To decrease non-linearity problems, reflectance (R) or transmittance (T) are transformed into absorbance (A):

$$A = \log_{10}\left(\frac{1}{R}\right) = -\log_{10} R \qquad (4)$$

The equipment normally provides these values directly.

For solid samples another possible approach is the Kubelka-Munk transformation [Kub]. In this case, the reflectance values are transformed into Kubelka-Munk units (K/S), using the equation:

$$\frac{K}{S}=\frac{(1-R)^2}{2R} \qquad (5)$$

where K is the absorption coefficient and S the scatter coefficient of the sample at a given wavelength.

3.3. Noise reduction and differentiation

When applying signal processing, the main aim is to remove part of the noise present in the signal or to eliminate some sources of variation (e.g. background) not related to the measured y-variable. It is also possible to try and increase the differences in the contribution of each component to the total signal and in this way make certain wavelengths more selective. The type of pre-processing depends on the nature of the signal.

General purpose methodologies are smoothing and differentiation. By smoothing one tries to reduce the random noise in the instrumental signal. The most used chemometric methodology is the one proposed by Savitzky and Golay [Sav]. It is a moving window averaging method. The principle of the method is that, for small wavelength intervals, data can be fitted by a polynomial of adequate degree, and that the fitted values are a better estimate than those measured, because some noise has been removed. For the initial window the method takes the first 2m+1 points and fits, by least squares, the corresponding polynomial of order o. The fitted value for the point in position m replaces the measured value. After this operation, the window is shifted one point and the process is repeated until the last window is reached. Instead of calculating the corresponding polynomial each time, if data have been obtained at equally spaced intervals, the method uses tabulated coefficients in such a way that the fitted value for the center point in the window is computed as:

$$x_{ij}^{*}=\frac{\sum_{k=-m}^{m} c_k x_{i,j+k}}{\text{Norm}} \qquad (6)$$

where x_{ij}^{*} represents the fitted value for the center point in the window, $x_{i,j+k}$ represents the 2m+1 original values in the window, c_k is the appropriate coefficient value for each point and Norm is a normalising constant (Figure 4a-b). Because the values of c_k are the same for all windows, provided the window size and the polynomial degree are kept constant, the use of the tabulated coefficients simplifies and accelerates the computations. For computational use, the coefficients for every window size and polynomial degree can be obtained in [Gor,Bia]. The user must decide the size of the window, 2m+1, and the order of the polynomial to be used. Errors in the original tables were corrected later [Ste]. These coefficients allow the smoothing of extreme points, which in the original method of Savitzky-Golay had to be removed. Recently, a methodology based on the same technique has been proposed [Bar], where the degree of the polynomial used is optimised in each window. This methodology has been called Adaptive-Degree Polynomial Filter (ADPF).

Another way of carrying out smoothing is by repeated measurement of the spectrum, i.e. by obtaining several scans and averaging them. In this way, the signal to noise ratio (SNR), increases with $\sqrt{n}_s$, n_s being the number of scans.

It should be noted that in many cases the instrument software will perform, if desired, smoothing by averaging of scans so that the user does not have to worry about how exactly to proceed. Often this is then followed by applying Savitzky-Golay, which is also usually present in the software of the instrument. If the analyst decides to carry out the smoothing with other software, then care must be taken not to distort the signal.

Differentiation can be used to enhance spectral differences. Second derivatives remove constant and linear background at the same time. An example is shown in Figure 5b-c. Both first and second derivatives are used, but second derivatives seem to be applied more frequently. A possible reason for their popularity is that they have troughs (inverse peaks) at the location of the original peaks. This is not the case for first derivatives.

In principle, differentiation of data is obtained by using the appropriate derivative of the polynomial used to fit the data in each window (Figure 4c-d). In practice, tables [Sav,Ste] or computer algorithms [Gor,Bia] are used to obtain the coefficients c_k. Alternatively the differentials can be calculated from the differences in absorbance between two wavelengths separated by a small fixed distance known as the gap.

One drawback of the use of derivatives is that they decrease the SNR by enhancing the noise. For that reason smoothing is needed before differentiation. The higher the degree of differentiation used, the higher the degradation of the SNR. In addition, and this is also true for smoothing data by using the Savitzky-Golay method, it is assumed that points are obtained at uniform intervals which is not always necessarily true. Another drawback is that calibration models obtained with spectra pre-treated by differentiation are sometimes less robust to instrumental changes such as wavelength shifts which may occur over time and are less easily corrected for the changes.

Constant background differences can be eliminated by using offset correction. Each spectrum is corrected by subtracting either its absorbance at the first wavelength (or other arbitrary wavelength) or the mean value in a selected range (Figure 5d).

An interesting method is the one based on contrasts as proposed by Spiegelman [Spi2,Wu3]. A contrast is the difference between the absorbance at two wavelengths. The differences between the absorbances at all pairs of wavelengths are computed and used as variables. In this way offset corrected wavelengths, derivatives (differences between wavelengths close to each other) are included and also differences between two peak wavelengths, etc. A difficulty is that the number of contrasts equals p(p-1)/2 which soon becomes very large, e.g. 1000 wavelengths gives 500000 contrasts. At the moment there is insufficient experience to evaluate this method and it has not been used as a pre-treatment for NN.

3.4. Methods specific for NIR

The following methods are applied specifically to NIR data of solid samples. Variation between individual NIR diffuse reflectance spectra is the result of three main sources:

- non-specific scatter of radiation at the surface of particles.
- variable spectral path length through the sample.
- chemical composition of the sample.

In calibration we are interested only in the last source of variance. One of the major reasons for carrying out pre-processing of such data is to eliminate or minimise the effects of the other two sources. For this purpose, several approaches are possible.

Multiplicative Scatter (or Signal) Correction (MSC) has been proposed by [Gel,Isa,Næs2]. The light scattering or change in path length for each sample is estimated relative to that of an ideal sample. In principle this estimation should be done on a part of the spectrum which does not contain chemical information, i.e. influenced only by the light scattering. However the areas in the spectrum that hold no chemical information often contain the spectral background where the SNR may be poor. In practice the whole spectrum is sometimes used. This can be done provided that chemical differences between the samples are small. Each spectrum is then corrected so that all samples appear to have the same scatter level as the ideal. As an estimate of the ideal sample, we can use for instance the average of the calibration set. MSC performs best if first an offset correction is carried out first. For each sample:

$$\mathbf{x}_\mathrm{i} = \mathrm{a} + \mathrm{b}\bar{\mathbf{x}}_\mathrm{j} + \mathbf{e} \qquad (7)$$

where $\mathbf{x}_\mathrm{i}$ is the NIR spectrum of the sample, and $\bar{\mathbf{x}}_\mathrm{j}$ symbolises the spectrum of the ideal sample (the mean spectrum of the calibration set). For each sample, a and b are estimated by ordinary least-squares regression of spectrum $\mathbf{x}_\mathrm{i}$ vs. spectrum $\bar{\mathbf{x}}_\mathrm{j}$ over the available wavelengths. Each value x_ij of the corrected spectrum $\mathbf{x}_\mathrm{i}$(MSC) is calculated as:

$$\mathrm{x}_\mathrm{ij}(\mathrm{MSC}) = \frac{\mathrm{x}_\mathrm{ij} - \mathrm{a}}{\mathrm{b}}; \quad \mathrm{j} = 1,2,\ldots,\mathrm{p} \qquad (8)$$

The mean spectra must be stored in order to transform in the same way future spectra (Figure 5h).

Standard Normal Variate (SNV) transformation has also been proposed for removing the multiplicative interference of scatter and particle size [Barn1,Barn2]. An example of such spectra is given in Figure 5a, where several samples of wheat are measured. SNV is designed to operate on individual sample spectra. The SNV transformation centers each spectrum and then scales it by its own standard deviation:

$$\mathrm{x}_\mathrm{ij}(\mathrm{SNV}) = \frac{\mathrm{x}_\mathrm{ij} - \bar{\mathrm{x}}_\mathrm{i}}{\mathrm{SD}}; \quad \mathrm{j} = 1,2,\ldots,\mathrm{p} \qquad (9)$$

where x_ij is the absorbance value of spectrum i measured at wavelength j, $\bar{\mathrm{x}}_\mathrm{i}$ is the absorbance mean value of the uncorrected ith spectrum and SD is the standard deviation of the p absorbance values:

$$\mathrm{SD} = \sqrt{\frac{\sum_{j=1}^{p}\left(x_{ij} - \bar{x}_i\right)^2}{p-1}} \qquad (10)$$

Spectra treated in this manner (Figure 5e) have always zero mean and variance equal to one, and are thus independent of original absorbance values.

De-trending of spectra accounts for the variation in baseline shift and curvilinearity of powdered or densely packed samples by using a second degree polynomial to correct the data [Barn1]. De-trending operates on individual spectra. The global absorbance of NIR spectra is generally increasing linearly with respect to the wavelength, but it increases curvilinearly for the spectra of densely packed samples. A second-degree polynomial can be used to standardise the variation in curvilinearity:

$$\mathbf{x}_i = a\lambda^{*2} + b\lambda^{*} + c + \mathbf{e}_i \qquad (11)$$

where $\mathbf{x}_i$ symbolises the individual NIR spectrum and λ^* the wavelength. For each sample, a, b and c are estimated by ordinary least-squares regression of spectrum $\mathbf{x}_i$ versus wavelength over the range of wavelengths. The corrected spectrum $\mathbf{x}_i$(DTR) is calculated by:

$$\mathbf{x}_i(\mathrm{DTR}) = \mathbf{x}_i - a\lambda^{*2} - b\lambda^{*} - c = \mathbf{e}_i \qquad (12)$$

Normally de-trending is used after SNV transformation (Figure 5f-g). Second derivatives can also be employed to decrease baseline shifts and curvilinearity, but in this case noise and complexity of the spectra increases.

It has been demonstrated that MSC and SNV transformed spectra are closely related and that the difference in prediction ability between these methods seems to be fairly small [Dha,Hel].

3.5. Selection of pre-processing methods in NIR

The best pre-processing method will be the one that finally produces a robust model with the best predictive ability. Unfortunately there seem to be no hard rules to decide which pre-processing to use and often the only approach is trial and error. The development of a methodology that would allow a systematic approach would be very useful. It is possible to obtain some indication during pre-processing. For instance, if replicate spectra have been measured, a good pre-processing methodology will produce minimum differences between replicates [Noo] though this does not necessarily lead to optimal predictive value. Depending on the physical state of the samples and the trend of the spectra, a background and/or a scatter correction can be applied. If only background correction is required, offset correction is usually preferable over differentiation, because with the former the SNR is not degraded and because differentiation may lead to less robust models over time. If additionally scatter correction is required, SNV and MSC yield very similar results. An advantage of

SNV is that spectra are treated individually, while in MSC one needs to refer to other spectra. When a change is made in the model, e.g. if, because of clustering, it is decided to make two local models instead of one global one, it may be necessary to repeat the MSC pre-processing. Non-linear behaviour between **X** and **y** appears (or increases) after some of the pre-processing methods. This is the case for instance for SNV, that is not a linear transformation. However this does not cause problems when using NN.

4. GRAPHICAL INFORMATION

Certain plots should always be made. One of these is to simply plot all spectra on the same graph (see Figure 5). Evident outliers will become apparent. It is also possible to identify noisy wavelength regions and perhaps exclude them from the model.

Another plot that one should always make is the PCA score plot. We recommend that it is carried out with the centered raw data and on the data after the signal pre-processing chosen in step 3.

Since PCA produces new variables, such that the highest amount of variance is explained by the first eigenvectors, the score plots can be used to give a good representation of the data. By using a small number of score plots (e.g. t_1-t_2, t_1-t_3, t_2-t_3), useful visual information can be obtained about the data distribution, inhomogeneities, presence of clusters or outliers, etc.

Plots of the loadings (contribution of the original variables in the new ones) identify spectral regions that are important in describing the data and those which contain mainly noise, etc. However, the loadings plots should be used only as an indication: there are better methods available to decide on which variables to retain if one wants to eliminate uninformative variables [cen4].

5. CLUSTERING TENDENCY

Clusters are groups of similar objects inside a population. When the population of objects is separated into several clusters, it is not homogeneous. To perform multivariate calibration modelling, the calibration objects should preferably belong to the same population. Often this is not possible, e.g. in the analysis of industrial samples, when these samples belong to different quality grades. The occurrence of clusters may indicate that the objects belong to different populations. This suggests there is a fundamental difference between two or more groups of samples, e.g. two different products are included in the analysis, or a shift or drift has occurred in the measurement technique. When clustering occurs, the reason must be investigated and appropriate action should be taken. If the clustering is not due to instrumental reasons that may be corrected (e.g. two sets of samples were measured at different times and instrumental changes have occurred) then there are two possibilities: to split the data in groups and make a separate model for each cluster or to keep all of them in the same calibration model.

The advantages of splitting the data are that one obtains more homogeneous populations and therefore, one hopes, better models. However, it also has disadvantages. There will be less calibration objects for each model and it is also considerably less practical since it is necessary to optimise and validate two or more models instead of one. When a new sample is predicted, one must first determine to which cluster it belongs before one can start the actual prediction. Another disadvantage is that the range of y-values can be reduced, leading to less stable models. For that reason, it is usually preferable to make a single model. The price one pays in doing this is a more complex and therefore potentially less robust model. Indeed, the model will contain two types of variables, variables that contain information common to the two clusters and therefore have similar importance for both, and variables that correct for the bias between the two clusters. Variables belonging to the second type are often due to peaks in the spectrum that are present in the objects belonging to one cluster and absent or much weaker in the other objects. In [Jou1], an example is presented, where two clusters occur. Some of the PCs are directly related with the property to be measured in both clusters, whereas others are related to the presence or absence of one peak. This peak is due to a difference in chemical structure and is responsible for the clustering.

Clustering techniques have been exhaustively studied (see a review of methods in [Mel]). Their results can for example be presented as dendrograms. However, in multivariate calibration model development, we are less interested in the actual detailed clustering, but rather in deciding whether significant clusters actually occur. For this reason there is little value in carrying out clustering: we merely want to be sure that we will be aware of significant clustering if it occurs.

The presence of clusters may be due to the y-variable. If the y-values are available in this step, they can be assessed on a simple plot of the **y** values. If it is distinctly bimodal, then there are two clusters in **y**, which should be reflected by two clusters in **X**. If y-clustering occurs, one should investigate the reason for it. If objects with y-values intermediate between the two clusters are available, they should be added to the calibration and tests sets. If this is not the case, and the clustering is very strong (Figure 6) one should realise that the model will be dominated by the differences between the clusters rather than by the differences within clusters. It might then be better to make models for each cluster, or instead of NN to use a method that is designed to work with very heterogeneous data such as locally weighted regression (LWR) [Næs2, Næs3].

The simplest way to detect clustering in the **X-**data is to apply PCA and to look at the score plots. In some cases, the clustering will become apparent only in plots of higher PCs so that one must always look at several score plots. For this reason, a method such as the one proposed by Szcubialka *et al*. may have advantages [Szc]. In this method, the distances between an object and all other objects are computed, ranked and plotted. This is done for each of the objects. The graph obtained is then compared with the distances computed in the same way for objects belonging to a normal or to a homogeneous distribution. A simple example is shown in Figure 7 where the distance curves for a clustered situation are compared with that for a homogeneous distribution of the samples.

If a numerical indicator is preferred, the Hopkins index for clustering tendency (H_{ind}) can be applied. This statistic examines whether objects in a data set differ significantly from the assumption that they are uniformly distributed in the multidimensional space [Hop,Cen4,Law]. It compares the distances $\mathbf{w}_i$ between the real objects and their nearest neighbours to the distances $\mathbf{q}_i$ between artificial objects, uniformly generated over the data space, and their nearest real neighbours. The process is repeated several times for a fraction of the total population. After that, the H_{ind} statistic is computed as:

$$H_{ind} = \frac{\sum_{i=1}^{n} \mathbf{q}_i}{\sum_{i=1}^{n} \mathbf{q}_i + \sum_{i=1}^{n} \mathbf{w}_i} \qquad (13)$$

If objects are uniformly distributed, $\mathbf{q}_i$ and $\mathbf{w}_i$ will be similar, and the statistic will be close to 0.5. If clustering are present, the distances for artificial objects will be larger than for the real ones, because these artificial objects are homogeneously distributed whereas the real ones are grouped together, and the value of H_{ind} will increase. A value for H_{ind} higher than 0.75 indicates a clustering tendency at the 90% confidence level [Law]. Figures 8a and 8b show the application of the Hopkins' statistic, i.e. how the q_i- and w_i-values are computed for two different data sets, the first unclustered and the second clustered. Because the artificial data set is homogeneously generated inside a square box that covers all the real objects and with co-ordinates determined by the most extreme points, an unclustered data set lying on the diagonal of the reference axis (Figure 8c) might lead to a false detection of clustering [For]. For this reason, the statistic should be determined on the PCA scores. After PCA of the data, the new axis will lie in the direction of maximum variance, in this case coincident with the main diagonal (Figure 8d). Since an outlier in the **X**-space is effectively a cluster, the Hopkins statistic could detect a false clustering tendency in this example. A modification of the original statistic has been proposed in [Law] to minimise false positives. Further modifications were proposed by Forina *et al*. [For].

Clusters can become more obvious upon data pre-treatment. For instance, a cluster which is not visible from the raw data may become more apparent when applying SNV. Consequently it is better to carry out investigations concerning clustering on the data pre-treated prior to modelling.

6. DETECTION OF OUTLIERS

Actually, the term "outlier detection" encompasses two steps: first, atypical object detection, followed by an outlier identification. Although numerical methods allow flagging of samples that are outliers on statistical grounds, the positive identification of an atypical object as a true outlier requires knowledge of the process or data acquisition procedure, or interaction with the person in charge of this acquisition. It is recommended to keep all flagged samples unless they are positively identified as outliers on experimental grounds.

We distinguish two types of outliers: we call them outliers in the **X**-space and outliers towards the model. Moreover we can consider outliers in **y**. Outliers in the **X**-space are points lying far away from the rest when looking at the x-values only. This means we do not use knowledge about the relationship between **X** and **y**. Outliers towards the model are those that present a different relationship between **X** and **y**, or in other words, samples that do not fit the model. Moreover an object can be an outlier in **y**, i.e. can present extreme values of the concentration to be modelled. If an object is extreme in **y,** it is probably also extreme in **X**.

At this stage of the process, we have not developed the model and therefore cannot identify outliers towards the model. However, we can look for possible outliers in **X** and in **y** separately. The simplest tool to flag atypical objects before modelling is the visual observation of the **X** and **y** data available. Detection of outliers in **y** is a univariate problem that can be handled with the usual univariate tests such as the Grubbs [Gru,Kel,Cen4] or the Dixon [Mil1,Cen4] test. One should look at the original set of sample spectra, the vector of responses and score plots on the first PCs. Outliers in **X** are multivariate and therefore they represent a more challenging problem. Our strategy will be to identify the extreme objects in **X**, i.e. identify objects with extreme characteristics, and apply a test to decide whether they should be considered outliers or not. Once the outliers have been identified, we must decide whether we eliminate them or simply flag them for examination after the model is developed so that we can look at outliers towards the model. In taking the decision, it may be useful to investigate whether the same object is an outlier in both **y** and **X**. If an object is outlying in concentration (**y**) but is not extreme in its spectral characteristics (**X**), then it is probable that at a later stage it will prove an outlier towards the model (Section 14) and it will be necessary at the minimum to make models with and without the object. A decision to eliminate the object at this stage may save work.

Extreme samples in the **X**-space can be due to measurement or handling errors, in which case they should be eliminated. They can also be due to the presence of samples that belong to another population, to impurities in one sample that are not present in the other samples, or to a sample with extreme amounts of constituents (i.e. with very high or low quantity of analyte). In these cases it may be appropriate to include the sample in the model, as it represents a composition that could be encountered during the prediction stage. We therefore have to investigate why the outlier presents extreme behaviour, and at this stage it can be discarded only if it can be shown to be of no value to the model or detrimental to it. We should be aware however that extreme samples always will have a larger influence on the model than other samples.

Extreme samples in the **X-**space have a double effect. Such objects add considerably to the total variance in the data set and, since the PCs try to explain variance, they will influence at least one of the PCs and therefore also the scores on such PCs. They may even lead to the inclusion of additional PCs, which is a problem because PCs are often used as inputs in NN models (see Section 9). In view of the parsimony principle [Sea] (Sections 9 and 11.2) this is considered undesirable. Moreover, extreme objects will probably have extreme scores on at least one PC, so that extreme scores will be present in the **T** matrix. These extreme scores will have an extreme (and possibly deleterious) effect in the range-scaling step (see Section 10).

X-space outlier detection can be performed with Rao's statistic [Mer]. Rao's statistic sums all the variation from a certain PC on. If there are a PCs, and we start looking at variation from PC r on, then:

$$D_i^2 = \sum_{i=r+1}^{a} t_{ij}^2 \qquad (14)$$

A high value for D_i^2 means that the object i shows a high score on some of the PCs that were not included and therefore cannot be explained completely by r PCs. For this reason it is then suspected to be an outlier. The method is presented here because it uses only information about **X**. The way in which Rao's statistic is normally used requires the number of PCs entered in the model. This number is put equal to r. At this stage we do not have a value for r. What can be done therefore is to follow the D value as a function of r, starting from r = 0. High values of r indicate that the object is modelled only correctly when higher PCs are included. If at a later stage it is decided to work with less PC, such an object will be an outlier. A test can be applied for checking the significance of high values for the Rao's statistic by using these values as input data for the single outlier Grubbs' test [Cen3]:

$$z = \frac{D_{test}^2}{\sqrt{\frac{\sum_{i=1}^{n}\left[D_i^2\right]^2}{n-1}}} \qquad (14)$$

Outlier detection is not easy. This certainly is the case if more than one outlier is present. In that case all the above methods are subject to what is called *masking* and *swamping*. Masking occurs when an outlier goes undetected because of the presence of another, usually adjacent, one. Swamping occurs when good observations are incorrectly identified as outliers because of the presence of another, usually remote, subset of outliers. Masking and swamping occur because the mean and the covariance matrix are not robust to outliers.

The potential method for outlier detection was proposed by Jouan-Rimbaud *et al.* [Jou2]. Potential methods first create so-called potential functions around each individual object. Then these functions are summed (see Figure 9). In dense zones, large potentials are created, while the potential of outliers does not add to that of other objects and can therefore be detected in that way. An advantage is that special objects within the **X**-domain are also detected, for instance, an isolated object between two clusters. Such objects (we call them inliers) can in certain circumstances have the same effect as outliers. A disadvantage is that the width of the potential functions around each object has to be adjusted. It cannot be too small, because many objects would then be isolated; it cannot be too large because all objects would be part of one global potential function. Moreover, while the method does allow very well in flagging the more extreme objects, a decision on their rejection cannot be taken easily.

Since outlier detection is not always successful, it is possible to design NN that can handle outliers present in the training set. For instance, Walczak [Wal3] proposed to use error thresholding functions adjusted iteratively during training with respect to the median of residuals. Wang *et al.* [Wan] also applied a thresholding function adjusted with respect to the assumed proportion of outliers among the ranked residuals. In both approaches, the idea is to prevent outlier residuals from influencing weight estimations during training.

7. NUMBER OF SAMPLES FOR MODELLING

Because the model has to be used for the prediction of new samples, all possible sources of variation that can be encountered later must be included in the calibration set. This means that the chemical components present in the samples must be included in the calibration set with a range of variation in concentration at least as wide, and preferably wider than, the one expected for the samples to be analysed; that sources of variation such as different origins or different batches are included and possible physical variations (e.g. different temperatures, different densities) among samples are also covered.

In addition, it is evident that the higher the number of samples in the calibration set, the lower the prediction error [Lor]. In this sense, a selection of samples from a larger set is contra-indicated. However, while a random selection of samples may approach a normal distribution, a selection procedure that selects samples more or less equally distributed over the calibration space will lead to a flat distribution. For an equal number of samples, such a distribution is more favourable from a regression point of view than the normal distribution, so that the loss of predictive quality may be less than expected by looking only at the reduction of the number of samples [Hil]. Also, from an experimental point of view, there is a practical limit on what is possible. While the NIR analysis is often simple and not costly, this cannot usually be said for the reference method. It is therefore necessary to achieve a compromise between the number of samples to be analysed and the prediction error that can be reached. It is advisable to spend some of the resources available in obtaining at least some replicates, in order to provide information about the precision of the model (Section 2).

When it is possible to artificially generate a number of samples, experimental design can and should be used (with more than two levels to allow non-linear modelling) to decide on the composition of the calibration samples [Mass]. When analysing tablets, for instance, one can make tablets with varying concentrations of the components and compression forces, according to an experimental design. Even then, it is advisable to include samples from the process itself to make sure that unexpected sources of variation are included. In the tablet example, it is for instance unlikely that the tablets for the experimental design would be made with the same tablet press as those from the production process and this can have an effect on the NIR spectrum [Jou1].

In most cases only real samples are available, so that an experimental design is not possible. This is the case for the analysis of natural products and for most samples coming from an industrial production process. The number of samples available is often a limiting factor when using NN. Like other regression methods, there are constraints concerning the number of samples required to develop an NN model. The number of adjustable parameters is such that the training set is rapidly overfitted if too few samples are available. We consider that when this number is less than 30, an alternative modelling technique should be considered. This is not always obvious for inexperienced users who can be deceived by the extreme flexibility of NN since they can fit the training data with arbitrary precision. It is possible to obtain excellent training results for the modelling of data sets with less than 15 samples. However, if these models are validated on new independent samples, a significant degradation of the results is observed due to a lack of generalisation ability.

To estimate the minimum number of training samples allowing theoretical generalisation, one can use a parameter called the Vapnik-Cervonenkis dimension (VCDim). For an MLP with one hidden layer, the lower bound of the VCDim is approximated as twice the total number of weights in the NN [Hus]. It is possible to reach good generalisation if the number of training samples is at least equal to this lower bound. When the number of samples available does not fulfil this requirement, NN can still be used to find an acceptable local minimum close enough to the absolute minimum of the error function. However, the ratio of the number of samples to the number of adjustable parameters should be kept as high as possible, in order to avoid under-determination of the problem. The number of samples is generally imposed or limited by practical constraints, but one can partly solve the under-determination problem by reducing the number of weights in the NN as much as possible, as explained in Section 11.

8. SELECTION AND REPRESENTATIVITY OF TRAINING AND MONITORING SAMPLES

An important step in the development of any calibration model is the splitting of the available data into two subsets: a calibration set (used to estimate model parameters) and a validation set or test set (used to check the generalisation ability of the model on new samples). For NN the problem is more complex because they fit to arbitrary precision the training data, provided that the number of hidden nodes is sufficient and the training time long enough. Therefore, an additional monitoring set is necessary to stop the training before the NN learns idiosyncrasies present in the training data [Svo,Tet1]. The usual procedure is to split the calibration set into two subsets, the training and monitoring set respectively. The monitoring set must be representative of the population under study in order to avoid NN overtraining that leads to overfitting (see

Figure 10). One question then arises: how to select the training and monitoring samples so that they are representative for the group?

Ideally, for a number nt of training samples, the monitoring set and the test set (if it is available) should contain between nt/2 and nt samples each. The repartition of samples between these sets and the terminology used in several papers can be a source of confusion. The performance of an NN should not be judged by its performance on training data that can always be fitted perfectly. There is no reason why results obtained on a monitoring set could not be reported, as long as it is made clear that these results were obtained on the data set used to evaluate the training end-point. One must be aware of the limitations of this approach: a true validation error is a better estimator of the NN generalisation ability than a monitoring error [Svo]. If one decides to favour the modelling power of the NN by using only two subsets (training and monitoring) instead of three subsets of smaller size (training, monitoring and validation), very good results may be obtained on the monitoring set but the model has not been truly validated in the sense that the monitoring data were used to optimise one of the model parameters (number of iterations for training). However, the monitoring results can be considered as indicative of the modelling power to expect from the NN model, and they can be compared with PCR or PLS cross-validation results.

Some authors mention leave-k-out (often k=1) cross-validation as a way of estimating the generalisation ability of the NN, for instance, when only few calibration samples are available. This approach is not adapted to NN [Mast,Gem2] because solutions obtained when two different samples are removed from the training set can differ significantly from each other [Svo]. In this case one cannot consider that the global model is validated, and it is even possible that none of the models developed during cross-validation describe the same region of the error surface as the global model. Therefore, if too few calibration samples are available to create a monitoring set, it is better to consider an alternative method to NN.

Several approaches are available for selecting representative calibration samples. The simplest is random selection, but it is open to the possibility that some source of variation will be lost. These are often represented by samples that are less common and have little probability of being selected. A second possibility is based on knowledge about the problem. If one is confident that we are aware of all the sources of variation, samples can be selected on the basis of that knowledge. However, this situation is rare and it is very possible that some source of variation will be forgotten.

Kennard and Stone proposed a sequential method that should cover the experimental region uniformly and that was meant for the use in experimental design [Ken]. The procedure consists of selecting as the next sample (candidate object) the one that is most distant from those already selected objects (calibration objects). The distance is usually the Euclidean distance. As starting points we either select the two objects that are most distant from each other, or preferably, the one closest to the mean. From all the candidate points, the one is selected that is furthest from those already selected and added to the set of calibration points. To do this, we measure the distance from each candidate point i_0 to each point i which has already been selected and determine which is smallest. From these we select the one for which the distance is maximal:

$$d_{\text{selected}} = \max_{i_0} \left(\min_{i} (d_{i,i_0}) \right) \qquad (16)$$

In the absence of strong irregularities in the factor space, the procedure starts first by selecting a set of points on the borderline of the data set (plus the center point, if this is chosen as the starting point). It then proceeds to fill up the calibration space. Kennard and Stone called their procedure a uniform mapping algorithm; it yields a flat distribution of the data which, as explained earlier, is preferable for a regression model.

Næs proposed a procedure based on cluster analysis. The clustering is continued until the number of clusters matches the number of calibration samples desired [Næs1]. From each cluster, the object that is furthest away from the mean is selected. In this way the extremes are covered but not necessarily the center of the data.

Figures 11a and 11b show the results of applying these two algorithms to a 2-dimensional data set of 250 objects, designed not to be homogeneous. Other methods have been proposed such as "unique-sample selection" [Hon]. The results obtained seem similar to those obtained from the previously cited methods.

Selection algorithms like Kennard-Stone ensure that monitoring and/or validation samples are within the domain covered by the training samples, so that the model does not extrapolate. This type of sample selection does not match the not-so-ideal situation sometimes encountered in practice, where it is not guaranteed that all new samples fall within the calibration domain. The duplex algorithm [Sne] allows a more realistic repartition of samples than the two previous methods. In the first step, the two points that are furthest away from each other are selected for the calibration set. From the remaining points, the two objects that are furthest away from each other are included in the test set. In the third step, the remaining point which is furthest away from the two previously selected for the calibration set is included in that set. The procedure is repeated selecting a single point for the test set which is furthest from the existing points in that set. Following the same procedure, points are added alternately to each set. This approach selects representative calibration and test data sets of equal size. The result of applying the duplex method is presented in Figure 11c.

Of the proposed methodologies, the Kennard-Stone and duplex methods need the minimum a priori knowledge. In addition, they provide a calibration set homogeneously distributed in space (flat distribution). If one wants to compare the efficiency of several modelling methods, samples can be selected with Kennard-Stone designs. If a model has to be developed for an application for which there is no guarantee that only interpolation will be performed, then duplex design will lead to more pessimistic but reliable results. Sample selection is often performed in the PC space on the scores matrix **T** instead of on the original matrix **X**, which allows one to reduce the computational burden.

It is also possible to perform the splitting after projecting the samples on a two-dimensional map with a Kohonen NN [Loz,Maj]. The advantage of such a projection is that an estimation of the relevant number of dimensions is not required and the essential topological features of the data set are preserved in two dimensions, which allows rapid visualisation of the data structure.

With strongly clustered data, subset selection should be performed on each cluster separately in order to ensure a good representativity between the training and test data.

Once the calibration set has been selected, several tests can be employed to determine the representativity of the selected objects with respect to the total set [Jou4]. This appears to be unnecessary if one of the algorithms recommended for the selection of the calibration samples has been applied. In practice, however, little attention is often paid to the proper selection. For instance, it may be that the analyst simply takes the first n samples for the calibration set. In this case a representativity test is necessary. One possibility is to obtain PC score plots and to compare visually the selected set of calibration samples to the whole set. This is difficult when there are many relevant PCs. In such cases a more formal approach can be useful. We proposed an approach that includes the determination of three different characteristics [Jou3]. The first one checks if both sets have the same direction in the space of the PCs, where the number of PCs to take into account is determined using the methodology described in Section 6. The directions are compared by computing the scalar product of two direction vectors obtained from the PCA decomposition of both data sets. To do this, the normed scalar product between the vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ is obtained:

$$P = \left| \frac{\mathbf{d}_1'\mathbf{d}_2}{\sqrt{\mathbf{d}_1^2\mathbf{d}_2^2}} \right| \qquad (17)$$

where $\mathbf{d}_1$ and $\mathbf{d}_2$ are the average direction vector for each data set:

$$\mathbf{d}_1 = \sum_{i=1}^{r} \lambda_{1,i}^2 \mathbf{p}_{1,i} \quad \text{and} \quad \mathbf{d}_2 = \sum_{i=1}^{r} \lambda_{2,i}^2 \mathbf{p}_{2,i} \qquad (18)$$

where $\lambda_{1,i}^2$ and $\mathbf{p}_{1,i}$ are the corresponding eigenvalues and loading vectors for data set 1, and $\lambda_{2,i}^2$ and $\mathbf{p}_{2,i}$ are the corresponding eigenvalues and loading vectors for data set 2. If the P value (cosinus of the angle between the direction of each set) is higher than 0.7, it can be concluded that the original variables have similar contribution to the latent variables, and they are comparable.

The second test compares the variance-covariance matrices. The intention is to determine whether the two data sets have a similar volume both in magnitude and direction. The comparison is made by using an approximation of the Bartlett's test. First the pooled variance-covariance matrix is computed:

$$\mathbf{C} = \frac{(n_1 - 1)\mathbf{C}_1 + (n_2 - 1)\mathbf{C}_2}{n_1 + n_2 - 2} \qquad (19)$$

The Box M-statistic is then obtained :

$$M = \nu\left[(n_1 - 1)\ln\left|\mathbf{C}_1^{-1}\mathbf{C}\right| + (n_2 - 1)\ln\left|\mathbf{C}_2^{-1}\mathbf{C}\right| \right] \qquad (20)$$

with

$$\nu = 1 - \frac{2p^2 + 3p - 1}{6(p-1)} \left\{ \frac{1}{n_1 - 1} + \frac{1}{n_2 - 1} - \frac{1}{n_1 + n_2 - 2} \right\} \qquad (21)$$

and the parameter CV is defined as:

$$\mathrm{CV} = e^{-M/n_1+n_2-2} \qquad (22)$$

If CV is close to 1, both the volume and the direction of the data sets are comparable.

The third and last test compares the data set centroids. To do this, the squared Mahalanobis distance D^2 between the means of each data set is computed:

$$D^2 = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)'\mathbf{C}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \qquad (23)$$

C is defined as in Equation 21, and from this value, a parameter F is defined as:

$$F = \frac{n_1 n_2 (n_1 + n_2 - p - 1)}{p(n_1 + n_2)(n_1 + n_2 - 2)} D^2 \qquad (24)$$

F follows a Fisher-Snedecor distribution, with p and n_1+n_2-p-1 degrees of freedom.

As already stated these tests are not needed when a selection algorithm is used. With some selection algorithms they would even be contra-indicated. For instance, the test that compares variances cannot be applied for calibration sets selected by the D-optimal design, because the most extreme samples are selected and the calibration set will necessarily have a larger variance than the original set.

9. DATA COMPRESSION

The ratio of the number of samples to the number of adjustable parameters in the NN should be kept as large as possible. One way of over-determining the problem is to compress input data, especially when they consist of absorbances recorded at several hundred wavelengths. In addition to reducing the size of input data, compression allows one to eliminate irrelevant information such as noise or redundancies present in a data matrix. Successful data compression can result in increased training speed, reduction of memory storage, better generalisation ability of the model, enhanced robustness with respect to noise in the measurements and simpler model representation.

The latent variables calculated with the PLS algorithm are designed to project data points on a lower dimensional subspace describing all relevant sources of variance. PLS latent variables are built so as to maximise the covariance between **X** and **y**. Some authors have used PLS to calculate input scores for NN training [Dup]. However, the latent variables are designed to conserve information linearly correlated with the response and some relevant non-linear information might be rejected in higher order latent variables that are not retained in the model [Borg,Dol]. For this reason, it is not recommend to pre-process data with PLS before NN modelling.

The most often used method for compressing information before calibration modelling with NN is PCA. Orthogonality of input variables is not so critical for NN that can handle collinear input data. However most NN applications in quantitative analysis with spectral data use PC scores as input variables [Borg,Blan1,Gem2,Jia1,Blan2,Popp,Zha,Des2]. For the determination of the optimum number A of input PCs to retain, one can use the same PC selection procedures as for PCR, although the choice is not so critical since NN models are built iteratively by successive optimisations of the NN topology. One possible approach consists in performing initial calculations with a deliberately large number of PCs and progressively reducing this number. This point will be detailed in Section 11.2.

When compressing data with PCA, one must be aware of some theoretical limitations. PCA is a linear projection method that fails to preserve the structure of a non-linear data set. If there is some non-linearity in **X** (or between **X** and **y***)*, this non-linearity can appear as a small perturbation on a linear solution and will not be described by the first PCs as in a linear case. A non-linear transformation of the **X**-matrix or PC scores matrix can be performed to restore the least-squares approximation property, but the resulting non-linear PCs are strongly dependent upon the pre-selected non-linear form and may not ensure the best representation of distances between points in the original space [Jia2]. In practice, PC scores are often successfully used as inputs without transformation because all relevant information about **X** is usually contained in the first 15 PCs.

Alternatively, it is possible to use Fourier analysis [Hin,Gem2], Hadamard transform [Dat] or wavelet analysis [Col] to pre-process spectral data before NN modelling. An attractive feature of wavelets is their ability to describe optimally local information from the spectrum, whereas Fourier decomposition is global. If this localised information is related to the non-linearity present in the data, an improvement can be expected if the input matrix is described with wavelet coefficients instead of PC scores or Fourier coefficients. A difficulty lies in the selection of one of the numerous wavelet bases for spectral decomposition. A scheme based on the optimisation of the minimum description length (MDL) criterion in multivariate calibration was explained by Walczak and Massart [Wal4].

Whatever the compression method retained, the new subspace (PCs, Fourier coefficients, wavelet coefficients) for sample description must be determined on the training set only. Then the monitoring and test samples can be projected in this subspace to calculate their scores or coefficients.

10. DATA SCALING

Once the input variables have been selected or calculated, one must ensure that they can be used for efficiently estimating NN parameters. It is not necessary to mean-center input variables before training since the biases act as offsets in the model (See section 1). NN training is not based on variance-covariance maximisation, and therefore it is not necessary to scale the different variables to unit variance, even when they are heterogeneous.

The only constraint for NN is to scale each input variable so that training starts within the active range of the non-linear transfer functions. Usually, samples are range-scaled with a linear mapping called min-max scaling. Scaling parameters must be determined on the training samples. All samples must be scaled with respect to these parameters. Let x_{min}^{train} and x_{max}^{train} be the extreme values of variable **x** in the training set, and let r_{min} and r_{max} define the limits of the range where we want to scale variable **x**. Any sample x_i (from the training, monitoring or test set) must be scaled to a new value a_i as follows:

$$a_i = \frac{\left(x_i - x_{min}^{train}\right)}{\left(x_{max}^{train} - x_{min}^{train}\right)}\left(r_{max} - r_{min}\right) + r_{min} \qquad (25)$$

For NN with sigmoid or hyperbolic tangent transfer functions (see section 11.4), r_{min} and r_{max} are set to –1 and 1, respectively. One must also ensure that the initial weights w_i^0 are reasonably small to avoid saturating the transfer functions in the first iterations. We suggest setting them so that:

$$0 < \left|w_i^0\right| < 0.1 \qquad (26)$$

If non-linear transfer functions are used in the output layer, the y-values must also be range-scaled so that outputs produced by the NN are not in the flat regions of the transfer function (see Figure 12). In these regions, the derivatives used for weight adjustment are almost zero and learning stops. For a sigmoid transfer function, range-scaling y to [0.2,0.8] is recommended, whereas for hyperbolic tangent range-scaling must be performed in the range [-0.8,0.8]. In theory, when linear transfer functions are used no range-scaling is needed since they are not bounded. In practice, in the early steps of learning there is a risk that unscaled responses lead to divergent wild steps for weight adjustments that can only be slowly recovered, especially with noisy data. Therefore, it is better to also range-scale responses to an arbitrary small range. To calculate training, monitoring or test error one must perform an inverse range-scaling to return the predicted responses to their original scale and compare them with experimental responses.

11. DETERMINATION OF NETWORK TOPOLOGY

The topology of an NN is determined by the number of layers in the NN, the number of nodes in each layer and the nature of the transfer functions.

Optimisation of NN topology is probably the most tedious step in the development of a model. The composite contribution of bias and variance to the mean squared error in a regression model can be represented as a function of model complexity (see Figure 13). NN can perform unbiased estimation of the training set to arbitrary precision and achieve asymptotic consistency. Universal approximation has a cost, however: a truly unbiased NN model (for instance, an NN with an infinite number of hidden nodes) would exhibit a very large variance, would be extremely sensitive to the idiosyncrasies in the training set and could only perform well on noise-free data. To attenuate the influence of noise that affects real analytical measurements, one has to constrain NN topology and allow some bias in the model. This can be done by the following means: reducing the number of layers, nodes and connections in the NN, constraining the form of the transfer functions or using a monitoring set to stop training. The different steps in topology optimisation are summarised in the flow chart in Figure 14.

11.1. Number of layers

The terminology used to describe NN topology can vary according to the authors, some of them considering the input layer as a simple buffer. We designate the NN represented in Figure 1 as a three-layer NN, with a 4-3-1 architecture (four input nodes, three hidden nodes, one output node).

The theoretical property of universal approximation has been proved for NN with only one hidden layer [Hor], and it is recommended that one uses only one hidden layer in multivariate calibration, unless the relationship to the model seems to be discontinuous [Mast]. In this case an additional hidden layer is necessary.

It is possible to add direct connections between the input and output layer of an NN [Borg] as illustrated in Figure 15. When the input variables have a mixed contribution to the response (some linear and some non-linear), direct connections can handle the linear part and the classical NN builds the non-linear part of the model. This approach can be interesting with NIR spectroscopic data where the non-linear effects observed generally correspond to small deviations from a linear solution [Borg]. Direct connections may speed up the learning process and ease model interpretation in situations where descriptors are heterogeneous. It was found that directly connected NN learned more quickly in the initial and intermediate training phases, but NN without direct connections converged to lower calibration and prediction errors [Blan1]. In theory, an NN without direct connections can achieve the same prediction performance as an NN with direct connections, so NN without direct connections should be preferred in order to reduce the number of adjustable parameters.

11.2. Number of input and output nodes

NN can model multiple responses simultaneously, however it is recommended to model only one response at a time and therefore use a single output node. The only exception to this rule is for situations where one wants to predict several correlated responses, such as the concentrations of different constituents of a mixture in a closed system. In that case, all responses can be modelled simultaneously with an NN having one output node per response.

To set the initial number of input nodes, two approaches are possible: the stepwise addition approach consists of starting with a deliberately small number of input variables and adding new variables one at a time until the monitoring and/or prediction performance of the NN does not improve any more; the stepwise elimination approach consists of starting with a deliberately large number of input scores and gradually removing (pruning) some of them until the monitoring and/or prediction performance of the NN stops improving.

Both approaches are used in practice and no definite recommendation can be given as to which one is better, since they both have advantages and limitations. If PCs are selected according to eigenvalues and the scores used as inputs, the stepwise addition method often leads to quick and satisfactory results, because all necessary information is usually contained in the first few PCs. However, it can happen that most information is contained in, e.g., PC1 to PC5, but some important additional information is also contained in PC10. During stepwise addition, the NN performance will stagnate or degrade between PC6 and PC9 and there are few chances that PC10 is included in the final model.

When stepwise elimination is performed, one must include a deliberately large number of input variables in the initial set. Irrelevant variables can be eliminated later, but relevant variables that have not been included in the initial model will not be tested subsequently. Here again, working with PC scores as inputs is advantageous. Using classical techniques (e.g., Malinowski's factor indication function and reduced eigenvalue test [Mal] or cross-validation [Wol]), one can estimate the pseudo-rank of the input data matrix. Then, one selects a few additional PCs (five or six) that may account for possible non-linearity, and the NN training can be started with this initial training set. The size of the initial set should typically vary between 10 and 15 PCs. The drawback of the stepwise elimination approach is that it can be extremely time consuming, if input variables are tentatively removed by trial and error, because of the large number of possible combinations [Loz].

In neural computation, the relevance of a variable to a model is called its sensitivity. The optimisation of the set of input variables can be accelerated if a method to estimate the sensitivity of each variable is implemented. Several methods have been proposed. The most common is often referred to as Hinton diagrams. It consists of ascribing to each input variable a sensitivity proportional to the average magnitude of its associated connections in the NN, represented on a two-dimensional map by square boxes of varying size. Candidate variables to be deleted are those with the lowest sensitivity. In spite of its popularity, this method exhibits severe theoretical and practical limitations [Des2,Tet2]. It is based on an analogy with the classical MLR approach, where the magnitude of a regression coefficient reflects the importance of the relationship between the associated descriptor and the response. In an NN model, input variables that have a linear contribution to the response will be modelled in the linear portion of the sigmoidal transfer function associated with small or medium magnitude weights, whereas the non-linear variables will be modelled in the concave portion of the transfer function associated with large magnitude weights. Therefore, the Hinton diagram ranking method is not based on the intrinsic relevance of a variable to a model, but simply on the nature of its contribution to the response. Linear input variables are systematically flagged as unimportant even when they explicitly contribute to the model. This approach can only give reliable results when the data set is entirely linear, in which case there is no point in using an NN.

Two variance-based approaches for input variable sensitivity determination were proposed recently [Des2]. They are designed for situations where input variables are orthogonal, which is the case with PC scores. The methods are based on the estimation of the individual contribution of each input variable to the variance of the predicted response. In the first approach, this contribution is determined by partial modelling. First, the NN is trained to estimate the parameters of the model:

$$\hat{\mathbf{y}} = f(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) \qquad (27)$$

After training, the sensitivity of each input variable $\mathbf{x_i}$ is calculated as the variance of the response $\hat{\mathbf{y}}(\mathbf{x_i})$ predicted with the trained NN when all input variables except $\mathbf{x_i}$ are set to zero:

$$\hat{\mathbf{y}}(\mathbf{x_i}) = f(\mathbf{x_i}) \qquad (28)$$

$$\mathbf{S_i} = \sigma^2_{\hat{\mathbf{y}}(\mathbf{x_i})} \qquad (29)$$

In the second approach, the separate contribution of each input variable to the variance of the estimated response is derived from a variance propagation equation for non-linear combinations of variables. In the case of a two-variable model ($\mathbf{x_1}, \mathbf{x_2}$), this equation is:

$$\sigma^2_{\mathbf{y}} = \left(\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{x_1}}\right)^2 \sigma^2_{\mathbf{x_1}} + \left(\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{x_2}}\right)^2 \sigma^2_{\mathbf{x_2}} + 2\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{x_1}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{x_2}}(COV)_{\mathbf{x_1 x_2}} \qquad (30)$$

Since PC scores are orthogonal, the covariance term can be neglected and the sensitivity of input variable $\mathbf{x_i}$ is calculated as

$$\mathbf{S_i} = \left(\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{x_i}}\right)^2 \sigma^2_{\mathbf{x_i}} \qquad (31)$$

Applying the chain rule several times, one obtains an analytical expression that allows to determine $\mathbf{S_i}$ at the end of training. The most interesting characteristic of these two variance-based methods (partial modelling and variance propagation) is that they give extremely stable results. When NN with the same topology are trained with different sets of initial random weights, they can converge to different local minima on the error surface that are qualitatively equally good and close to each other. In that case the two variance-based methods give similar results, which is not always the case with Hinton diagrams.

Once the sensitivity of each input variable has been estimated, we recommend that one should first try to remove the variable with the lowest sensitivity, and retrain the NN. If the monitoring error decreases after removing the flagged variable, it can be considered as irrelevant for the model and permanently removed, otherwise it must be replaced and another flagged variable must be tentatively removed. Since parsimonious models should be preferred in multivariate calibration, we propose the following methodology for the stepwise elimination of input variables. Let ME(k) be the monitoring error at the k-th trial, and ME(k+1) the monitoring error at the next trial after removal of a flagged input variable. Then:

If ME(k+1) $\leq \tau \times$ME(k), remove the flagged variable.
Else, replace the flagged variable and try to remove the next variable with lowest sensitivity.

Here τ is a tolerance factor that can be adjusted to different values. It is suggested to set τ =1.1. Increasing this factor will result in removing more input variables from the model, at the risk of losing some relevant sources of variance; τ should not be lower than 1, otherwise the NN could have a poor generalisation ability.

For a given set of input variables, the NN performance will also vary with the number of hidden nodes. Therefore, optimisation of the number of input variables and of the number of hidden nodes should be performed in conjunction: at each step, one should optimise the number of input variables, then the number of hidden nodes, then optimise again the number of input variables and proceed so until the monitoring error stops decreasing.

11.3. Number of hidden nodes

An upper bound on the number of hidden nodes is of the order of the number of training samples used [Hus]. It was further proved that an NN with *n* sigmoidal hidden nodes could approximate the response of 2*n*-1 samples [Son]. These results support the idea that it is not necessary to use large numbers of hidden nodes to fit complex multivariate relationships. On the contrary, large numbers of hidden nodes often accentuate the risk of overfitting [Derk1]. It is therefore recommended to systematically reduce the number of hidden nodes as much as possible, in order to achieve simpler and more robust models.

Kanjilal and Banerjee [Kan] presented a strategy for reducing the number of hidden nodes in an NN. The method is based on orthogonalisation of the hidden layer output matrix with singular value decomposition (SVD), after a crude convergence has been reached. Zhang *et al.* [Zha] presented an algorithm based on a similar concept, that allows one to use all calibration samples for NN training without need for a monitoring set. The initial postulate is that NN with large numbers of hidden nodes are relatively insensitive to initial conditions, but their generalisation ability is worse than NN with a hidden layer of reduced size. The proposed scheme consists of starting NN training with a deliberately large hidden layer until an arbitrarily low error is reached, then perform SVD on the hidden layer output matrix **H**:

$$\mathbf{H}_{k\times h} = \mathbf{U}_{k\times k} \,.\, \mathbf{S}_{k\times h} \,.\, \mathbf{V}^{T}_{h\times h} \qquad (32)$$

where h is the number of hidden nodes and k the number of training samples. The number r of dominant singular values in the diagonal **S** matrix (determined by a variance ratio criterion) is considered as the number of hidden nodes necessary for the NN. A new NN is built, with only r<k hidden nodes, and the new initial weight matrices are determined by least squares fit so that the hidden layer output matrix is

$$\mathbf{H}' = [\mathbf{U}_1 \; \mathbf{U}_2 \; ... \; \mathbf{U}_r] \qquad (33)$$

Training is then resumed on this pruned NN with improved generalisation ability.

Reduction of the number of hidden nodes can also be done by trial and error, like the optimisation of number of input variables. It is always a good idea to compare the performance of a one hidden node model with the performance of a more complex model since many data sets in multivariate calibration are only slightly non-linear. The advantage of models with one hidden node is that the results they produce are stable and independent of the set of initial random weights [Kol]. Moreover, a model with one hidden node reduces to a sigmoidal regression that can be easily interpreted. In an extrapolation calibration study [Est], the prediction error of the NN on one data set was reduced by 50% by using one hidden node only.

11.4. Transfer function

Kolmogorov’s theorem states that an NN with linear combinations of n×(2n + 1) monotonically increasing non-linear functions of only one variable is able to fit any continuous function of *n* variables [Lipp]. The most currently used non-linear transfer functions in the hidden layer are the sigmoid or hyperbolic tangent functions that are bounded, easily differentiable and exhibit a linear-like portion in their center, so that data sets that are only slightly non-linear can also be modelled (see Figure 12). These two functions are popular because they allow to fit a large number of non-linearities, but other functions can be tried. For instance, Gemperline *et al.* [Gem1] performed multivariate calibration with NN on UV/VIS data using in their hidden layer combinations of linear, sigmoid, hyperbolic tangent and square functions, to accommodate different types of non-linear response in different spectral regions.

The transfer function(s) in the output layer can be linear or non-linear. In many situations, if the number of hidden nodes is sufficient, all modelling is done in the hidden layer. It was observed that in some situations where data were mainly linear, non-linear output transfer functions could introduce distortion in the predicted responses [Goo], as illustrated in Figure 16. If a linear output transfer function is used, any linear node in the hidden layer can be replaced with a direct connection between input and hidden layer (because two successive linear transformations can be reduced to a single one), which reduces the number of adjustable parameters in the NN.

The safest procedure is to try both types of output transfer functions (linear and non-linear) during topology optimisation and to base the decision on the shape of residuals for models constructed with the same input variables.

12. TRAINING OF THE NETWORK

12.1. Learning algorithms

Two general modes of learning can be distinguished: incremental learning and batch learning. Incremental learning consists of successively updating the weights in the NN after estimating the error associated with the response predicted for each sample presented in a random order. In the batch learning mode the errors of all training samples over each iteration are first summed and the parameters are adjusted with respect to this sum. The former approach has the advantage that it superimposes a stochastic component on the weight update. This can help the NN escape from local minima on the error surface in the hyperspace of the weights.

A drawback is that the method is prone to the phenomenon of thrashing: the NN can take successive steps in opposite directions that may slow learning. Batch learning provides a more accurate estimate of the gradient vector [Svo] and faster convergence, but it also requires more memory storage capacity. The relative efficiency of both approaches is usually data set-dependent. The incremental approach seems particularly suited for very homogeneous training sets [Hert] or for on-line process control applications [Svo] where the composition of the training set is constantly modified.

Training an NN is an optimisation problem, and several methods are available for this task. It is not possible to review in detail all algorithms available, but the main types of algorithms will be summarised and their particularities outlined.

The gradient descent algorithm performs a steepest-descent minimisation on the error surface in the adjustable parameters hyperspace. This algorithm was described and popularised by Rumelhart and McClelland [Rum] in 1986. The excessively slow convergence of the basic algorithm and its tendency to become trapped in the numerous local minima of the error surface triggered the need for improvements such as the addition of a momentum term in the weight update, that allows one to smooth the error surface and to attenuate oscillations in the bottom of steep valleys. The speed of the algorithm can be significantly enhanced by using adaptive parameters (learning rate and momentum rate) for each weight in the NN. This is the basis of the delta-bar-delta [Jac] and extended delta-bar-delta [Min] algorithms, that have been successfully applied in multivariate calibration [Blan1].

Faster convergence can be reached with second-order optimisation methods, based on the determination or approximation of the Hessian matrix of partial second derivatives of the cost function: these methods typically have a convergence time one order of magnitude smaller than the gradient method or its derivatives. In the Newton-Raphson method, the Hessian matrix is used to adjust the descent direction at each step, and convergence is reached in a single step if the error surface is quadratic, with ellipsoidal contours. Currently, one of the most popular and efficient second-order methods for NN training is the Levenberg-Marquardt algorithm [Flet,Popp,Des2,Derk1] that is a compromise between gradient descent and Newton-Raphson optimisation. At each step, an adaptive parameter allows the algorithm to transit smoothly between the gradient direction and the Newton-Raphson direction. The inverse Hessian matrix is only estimated and iteratively updated to avoid tedious calculations. Conjugate gradient optimisation is an alternative second-order technique that also uses the Hessian matrix, but the algorithm is formulated in such a way that the estimation and storage of the Hessian matrix are completely avoided [Flet]. With conjugate gradient optimisation, each new search direction is chosen so as to spoil as little as possible the minimisation achieved by the previous one, in contrast to the winding trajectory observed with the gradient method. This method is guaranteed to locate the minimum of any quadratic function of *n* variables in at most *n* steps.

Genetic algorithms (GA) have been used for NN training [Jia1,Bos2]. This global search method allows one to overcome the problem of becoming trapped in local minima, but at the expense of a long computing time because each individual in the population represents a different NN model. In addition, a number of parameters must be set to define the population size and evolution mode, and therefore this approach cannot be easily implemented.

Random optimisation consists of taking successive random steps in the weight space and discarding all steps that do not reduce the cost function. In contrast to the classical back-propagation algorithm, random search is guaranteed to find a global minimum [Loo], but the computation time is so high that the method is never used in practice. Instead, GA or random optimisation can be used as preliminary techniques to optimise the initial set of weights in the NN, then the training is continued with a back-propagation-based method.

12.2. When to stop training

As mentioned previously, a monitoring set has to be used in order to reduce the tendency of NN to overtrain and therefore overfit the training data. The evolution of the monitoring error must be followed during training. The frequency of monitoring error estimation has to be determined by the user; ideally it should be performed after each iteration. Consecutive monitoring error values are stored in a vector, and several criteria can be applied to retain the optimum set of weights: train the NN for a pre-defined large number of iterations and retain the set of weights corresponding to the minimum of the monitoring error curve; stop training and retain the last set of weights as soon as the monitoring error is below a pre-specified threshold or stop training and retain the last set of weights as soon as the decrement between two successive monitoring errors is below a pre-specified threshold.

One must also check that the training error is reasonably low at the number of iterations retained, and that the representativity between the training and the monitoring set is ensured. A useful way to detect lack of representativity between training and monitoring set is when the root mean squared error curves for both sets are separated by a large gap in the region where they flatten, as shown in Figure 17a [Smi,Kat].

Alternatively, it is possible that the optimal monitoring error is reached while the training error is still relatively high (Figure 17b.). This can be due to chance correlation, for instance when the initial set of random weights brings the model near a local minimum on the monitoring error surface. In both cases (large gap between monitoring and training error curves, or early minimum for monitoring), a different splitting of data between the two subsets should be considered.

The sensitivity of the NN solution to initial conditions is an important issue [Kol]. To fit a given non-linear data set, a NN can develop several combinations of transfer functions that give similar training errors, yet with different model shapes and different generalisation abilities. The use of a monitoring set reduces the risks of overfitting the training set, but it can happen that the monitoring set is also overfitted by chance correlation with the weights. Since the monitoring error is used as a criterion for determination of the end of training, an apparently low monitoring error can lead to retaining a model with a poor generalisation ability. To overcome effects due to chance correlation, several trials must be performed with different sets of initial random weights [Tet1]. At least ten trials are recommended. To eliminate the influence of isolated solutions with low monitoring error due to monitoring set overfitting, a robust approach consists of retaining the topology corresponding to the lowest median monitoring error over the replicate trials [Des3]. Once the topology has been established, the set of weights that led to the monitoring error value closest to the median can be retained for the final model. It is recommended, however, to test it against a validation set, if available, before performing predictions on unknown samples.

12.3. Model interpretation

NN have more to offer than a simple empirical model. The sensitivity plots that were presented earlier describe the relative influence of the different input variables in the final model. In addition, examination of the projection of the samples on the hidden nodes of the NN is often informative [Mast]. For instance, a calibration model was developed for the quantitative analysis of traces of lead in water, using inductively coupled plasma atomic emission spectrometry (ICP-AES) data as inputs (14 descriptors). At the end of training, if one displays the activation of hidden nodes *versus* each other, one obtains plots comparable to score plots (Figure 18). The five measurement replicates marked with asterisks are easily identified as probable outliers. Such plots are instructive and also allow visualisation of clusters present in the data.

Figures 19a to 19c represent the activation of the three hidden nodes at the end of training for the ICP-AES data NN model. Figures 19d and 19e represent the activations of two hidden nodes used to model the concentration of a mineral charger in a polymer using NIR spectra. The activation of hidden nodes for ICP-AES data indicates that this data set is mainly linear, whereas the transfer functions for the modelling polymer data are activated in their strongly non-linear portion. Thus we obtain information on the degree of non-linearity of a given data set, even when the exact form of the model is unknown.

A method to visualise the form of the contribution of the different input variables in the NN model has also been proposed [Des4]. Like the procedure for input variable sensitivity determination (see 11.2), this approach is also based on the concept of partial modelling. It consists of displaying the partial models for each variable. A statistical procedure based on replicate trials (after addition of random noise to the model input variables) and analysis of variance for lack-of-fit allows to identify variables with a significantly nonlinear contribution to the model.

Recently, several groups have investigated the assessment of statistical confidence intervals for predictions with NN. Dathe and Otto [Dat] derived confidence intervals using the bootstrap method. After finding the optimum topology of the NN, they erase a portion of the calibration matrix and randomly fill it with replicate samples from the remaining portion. An arbitrary number n_{sets} of calibration matrices is created, and n_{sets} models are built with the pre-defined topology. An external test set is used to predict the responses with each of the bootstrapped NN models, and standard deviations of predicted responses can be calculated. Derks and Buydens [Derk2] also worked on the calculation of confidence intervals and compared three forms of bootstrapping. The advantage of the bootstrap approach is that the derived confidence intervals contain all sources of variability (experimental noise, model errors, effect of different sets of random weights), thus yielding a worst case estimation. The drawback is that the confidence intervals derived correspond to an NN topology, not to a single model with a fixed set of weights.

12.4. Measures of predictive ability

Several statistics are used for measuring the predictive ability of a model. In case of n<p the prediction error sum of squares, PRESS, is computed as:

$$\mathrm{PRESS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2 \qquad (34)$$

where y_i is the actual value of **y** for object i and $\hat{y}_i$ the y value for object i predicted with the model under evaluation, e_i is the residual for object i (the difference between the predicted and the actual **y** value) and n is the number of objects for which $\hat{y}$ is obtained by prediction.

The mean squared error of prediction (MSEP) is defined as the mean value of PRESS:

$$\mathrm{MSEP} = \frac{\mathrm{PRESS}}{n} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} = \frac{\sum_{i=1}^{n} e_i^2}{n} \qquad (35)$$

Its square root is called root mean squared error of prediction, RMSEP:

$$\mathrm{RMSEP} = \sqrt{\mathrm{MSEP}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \qquad (36)$$

All these quantities give the same information. In the chemometrics literature it seems that RMSEP values are preferred, partly because they are given in the same units as the **y** variable.

An important question is what RMSEP to expect? If the final model is correct, i.e. there is no bias, then the predictions will often be more precise than those obtained with the reference method [DiF,Cen1], due to the averaging effect of the regression. However, this cannot be proved from measurements on validation samples, the reference values of which were obtained with the reference method. The RMSEP value is limited by the precision (and accuracy) of the reference method. For that reason, RMSEP can be applied at the optimisation stage as a kind of target value. An alternative way of deciding on model complexity therefore is to select the lowest complexity that leads to an RMSEP value comparable to the precision of the reference method.

12.5. Outlying objects in the model

In Section 6 we explained how to detect possible outliers before the modelling, i.e. in the **y-** and/or **X**-space. When the model has been built, we should check again for the possibility that outliers in the **X**-**y**-space are present, i.e. objects that do not fit the true model well. The difficulty with this is that such outlying objects influence (bias) the model obtained, often to such an extent that it is not possible to see that the objects are outliers to the true model. Diagnostics based on the distance from the obtained model may therefore not be effective. Consider the univariate linear case of Figure 20. The outlier A to the true model attracts the regression line (exerts leverage), but cannot be identified as an outlier because its distance to the obtained regression line is not significantly higher than for some of the other objects. Object A is then called influential and we should therefore concentrate on finding such influential objects.

Centner *et al.* [Cen3] proposed a procedure based on the development of PLS leave-one-out cross-validation models after flagging possible outliers with a Grubb's test performed on the Rao's statistic. The idea is to discriminate situations where a true outlier alters the models resulting in a large cumulative cross-validation error, from situations where the large value of the cross-validation error is simply due to the incorrect prediction of a high leverage point that is not an outlier. A similar approach based on cross-validation cannot be performed with NN (see Section 8). As a consequence, a limitation of this approach that uses PLS is that the identification is based on linear cross-validation models. A sample that is an outlier to a linear model might not be an outlier to a non-linear model [Sek]. Therefore one can use this method to flag possible outliers, but not as a positive outlier identification method.

The next step is the study of residuals. A first approach is visual. One can make a plot of $\hat{\mathbf{y}}$ against $\mathbf{y}$. If this is done for the final model, it is likely that, for the reasons outlined above, an outlier will not be visible. One way of studying the presence of influential objects, is therefore not to study the residuals for the final model but the residuals for the model with 1, 2, ..., a input variables, because in this way we may detect outliers on a specific variable. If an object has a large residual on, say, input variable 2, but a small residual when variable 3 is added, it is possible that variable 3 is included in the model only to allow for this particular object. This object is then influential on variable 3. We can provisionally eliminate the object, carry out NN model again and, if a more parsimonious model with at least equal predictive ability is reached, may decide to eliminate the object completely.

Genetic algorithms or simulated annealing can be applied to select subsets (including clean subsets) according to a given criterion from a larger population. This lead Walczak *et al*. to develop their evolution program, EP [Wal1,Wal2]. It uses a simplified version of a genetic algorithm to select the clean subset of objects, using minimalisation of RMSEP as a criterion for the clean subset objects. The percentage of possible outliers in the data set must be selected in advance. The method allows the presence of 49% of outlying points, but the selection of such a high number risks the elimination of certain sources of variation from the clean subset and the model. The clean subset should therefore contain at least 90%, if not 95%, of the objects. Other algorithms based on the use of clean subset selection have been proposed by Hadi [Had] and Hawkins [Haw] and by Atkinson and Mukira [Atk]. Unfortunately none of these methods have been studied to such an extent that they can be recommended in practice.

If possible outliers are detected, the final decision should be made on the basis of a comparison of prediction results for NN models with and without the flagged samples in the training set. NN models should be made removing one of the flagged samples in turn, starting with the most suspect object. If the model obtained after deletion of the candidate outlier has a clearly lower RMSEP, or a similar RMSEP but a lower complexity, the outlier should be removed. If only a few candidate outliers remain after this step (not more than 3) one can also look at NN models in which each of the possible combinations of 2 or 3 outliers was removed. In this way one can detect outliers that are jointly influential. It should be noted however that a conservative approach should be adopted to the rejection of outliers. If one outlier and, certainly, if more than a few outliers are rejected we should consider whether perhaps there is something fundamentally wrong and review the whole process including the chemistry, the measurement procedure and the initial selection of samples.

13. USING THE MODEL

Once the final model has been developed, it is ready for use: the calibration model can be applied to spectra of new samples. It should be noted that the data pre-processing and/or pre-treatment selected for the calibration model must also be applied to the new spectra and this must be done with the same parameters (e.g. same ideal spectrum for MSC, same window and polynomial size for Savitzky-Golay smoothing or derivation, etc.). For mean-centering, the mean used in the calibration stage must be used for the pre-treatment of the new spectra.

Although it is not the subject of this tutorial, which is restricted to the development of a model, it should be noted that to ensure quality of the predictions and validity of the model, the application of the model over time also requires several applications of chemometrics. The following subjects should be considered:

- Quality control: it must be verified that no changes have occurred in the measurement system. This can be done for instance by applying system suitability checks and by measuring the spectra of standards. Multivariate quality control charts can be applied to plot the measurements and to detect changes [Tra,Kre].

- Detection of outliers and inliers in prediction: the spectra must belong to the same population as the objects used to develop the calibration model. Outliers in concentration (outliers in **y**) can occur. Samples can also be different from the ones used for calibration, because they present sources of variance not taken into account in the model. Such samples are then outliers in **X**. In both cases, this leads to extrapolation outside the calibration space so that the results obtained are less accurate. Large extrapolations can lead to unacceptable results. It is therefore necessary to investigate whether a new spectrum falls into the spectral domain of the calibration samples. As stated in Section 6, we can in fact distinguish outliers and inliers. Outliers in **y** and in **X** can be detected by adaptations of the methods we described in Section 6. Inliers are samples which, although different from the calibration samples, lie within the calibration space. They are located in zones of low (or null) density within the calibration space: for instance, if the calibration set consists of two clusters, then an inlier can be situated in the space between the two clusters. If the model is non-linear, their prediction can lead to interpolation error. Few methods have been developed to detect inliers. One of them is the potential function method of Jouan-Rimbaud *et al*. (Section 6) [Jou2]. Another possibility was presented by De Ruyck [Ruy]. If the data set is known to be relatively homogeneous (by application of the methods of Section 5), then it is not necessary to look for inliers.

- Updating the models: when outliers or inliers were detected and it has been verified that no change has occurred in the measurement conditions, then one may consider adding the new samples to the calibration set. This makes sense only when it has been verified that the samples are either of a new type or an extension of the concentration domain and that it is expected that similar new samples can be expected in the future. Good strategies to perform this updating with a minimum of work, i.e. without having to take the whole extended data set through all the previous steps, do not seem to exist.

- Correcting the models (or the spectra): when a change has been noticed in the spectra of the standards, for instance in a multivariate QC chart, and the change cannot be corrected by changes to the instrument, this means that spectra or model must be corrected. When the change in the spectra is relatively small and the reason for it can be established [Bou2], e.g. a wavelength shift, numerical correction is possible by making the same change to the spectra in the reverse direction. If this is not the case, it is necessary to treat the data as if they were obtained on another instrument and to apply methods for transfer of calibration from one instrument to another. A review about such methods is given in [Bou1].

14. CONCLUSIONS

It will be clear from the preceding sections that developing good multivariate calibration models requires a lot of work. There is sometimes a tendency to overlook or minimise the need for such a careful approach. Data pre-treatment and presentation (number of samples, detection of outliers, data compression and splitting) are critical issues. Experience has proved that several failures of NN for modelling where indeed due to inappropriate problem formulation. Such issues can be circumvented by focusing on prior model identification, in particular the detection of non-linearity. Proper *a priori* non-linearity detection is one of the major difficulties and methods existing so far often fail in the presence of outliers. The deleterious effects of outliers are not so easily observed as for univariate calibration and are therefore sometimes disregarded. Problems such as heterogeneity or non-representativity can occur also in univariate calibration models, but these are handled by analytical chemists who know how to avoid or cope with such problems. When applying multivariate calibration, the same analysts may have too much faith in the power of the mathematics to worry about such sources of errors or may have difficulties in understanding how to tackle them. Some chemometricians do not have analytical backgrounds and may be less aware of the possibility that some sources of error can be present. It is therefore necessary that strategies should be made available for systematic method development that include the diagnostics and remedies required and that analysts should have a better comprehension of the methodology involved. It is hoped that this tutorial will help to some degree in reaching this goal.

It must be clear that this tutorial is only limited to the multi-layer feed-forward type of NN. Other types of NN can be used in multivariate calibration. For instance, radial basis function (RBF) networks offer interesting alternatives to more classical NN in the sense that they allow local training and the final models can be interpreted in terms of logical rules [Derk3,Hus,Wal5]. Another approach that allows to gain insight into a complex problem is to combine the use of classical feed-forward NN (for prediction) with a counter-propagation NN to obtain contour plots of the input and output variables [Loz,Maj].

NN should become part of the standard toolkit of analytical chemists concerned with multivariate calibration, but it is important to have a clear understanding of their capabilities and limitations. One should not consider

NN as black boxes, but as regression models whose flexibility will depend on the topology defined by the user.

REFERENCES

[Ast] "*Standard practices for infrared, multivariate, quantitative analysis*". Doc. E1655-94, in "*ASTM Annual book of standards*", vol.03.06, ASTM, West Conshohocken, PA, USA, 1995.

[Atk] Atkinson, A.C. and Mulira, H.M., *Statistics and computing*, **3** (1993) 27.

[Bar] Barak, P., *Anal. Chem.*, **67** (1995) 2758.

[Barn1] Barnes, R.J., Dhanoa, M.S., and Lister, S.J., *Appl. Spectrosc.*, **43** (1989) 772.

[Barn2] Barnes, R.J., Dhanoa, M.S., and Lister, S.J., *J. Near Infrared Spectrosc.*, **1** (1993) 185.

[Bia] Bialkowski, S.E., *Anal. Chem.*, **61** (1989) 1308.

[Blan1] Blank, T.B., and Brown, S.D., *Anal. Chem.*, **65** (1993) 3081.

[Blan2] Blank, T.B., and Brown, S.D., *Anal. Chim. Acta*, **277** (1993) 273.

[Borg] Borggaard, C., and Thodberg, H.H., *Anal. Chem.*, **64** (1992) 545.

[Bos1] Bos, M., Bos, A., and van der Linden, W.E., *Analyst*, **118** (1993) 323.

[Bos2] Bos, M., and Weber, H.T., *Anal. Chim. Acta*, **247** (1991) 97.

[Bou1] Bouveresse, E., and Massart, D.L., *Vib. Spectrosc.*, **11** (1996) 3.

[Bou2] Bouveresse, E., Rutan, S.C., Vander Heyden, Y., Penninckx, W., and Massart, D.L., *Anal. Chim. Acta.*, **348** (1997) 283.

[Cen1] Centner, V., Massart, D.L., and de Jong, S., *Fresenius J. Anal. Chem.*, **361** (1998) 2.

[Cen2] Centner, V., Massart, D.L., and de Noord, O.E., *Anal. Chim. Acta*, **376 (**1998) 153.

[Cen3] Centner, V., Massart, D.L., and de Noord, O.E., *Anal. Chim. Acta*, **330** (1996) 1.

[Cen4] Centner, V., Massart, D.L., de Noord, O.E., de Jong, S., Vandeginste, B.M., and Sterna, C., *Anal. Chem.*, **68** (1996) 3851.

[Cir] Cirovic, D.A., *Trends Anal. Chem.*, **16** (1997) 148.

[Col] Collantes, E.R., Duta, R., Welsh, W.J., Zielinski, W.L., and Brower, J., *Anal. Chem.*, **69** (1997) 1392.

[Coo] Cook, R.D., *Technometrics*, **35** (1993) 351.

[Dat] Dathe, M., and Otto, M., *Fresenius J. Anal. Chem.*, **356** (1996) 17.

[Derk1] Derks, E.P.P.A., and Buydens, L.M.C., *Chemom. Intell. Lab. Syst.*, **41** (1998) 171.

[Derk2] Derks, E.P.P.A., and Buydens, L.M.C., *Chemom. Intell. Lab. Syst.*, **41** (1998) 185.

[Derk3] Derks, E.P.P.A., Sanchez Pastor, M.S., and Buydens, L.M.C., *Chemom. Intell. Lab. Syst.*, **28** (1995) 49.

[Des1] Despagne, F., and Massart, D.L., *Analyst* **123** (1998) 157R.

[Des2] Despagne, F., and Massart, D.L., *Chemom. Intell. Lab. Syst.*, **40** (1998) 145.

[Des3] Despagne, F., Massart, D.L., and Chabot, P., “Development of a robust calibration model for nonlinear on-line process data” (in preparation).

[Des4] Despagne, F., Massart, D.L., and Zanier, N., and de Noord, O.E. “Methods for interpretation of neural network calibration models” (in preparation).

[Dha] Dhanoa, M.S., Lister, S.J., Sanderson, R., and Barnes, R.J., *J. Near Infrared Spectrosc.*, **2** (1994) 43.

[Dol] Dolmotova, L., Ruckebusch, C., Dupuy, N., Huvenne, J.P., and Legrand, P., *Chemom. Intell. Lab. Syst.*, **36** (1997) 125.

[Dra] Draper, N.R., and Smith, H., *Applied regression analysis*, 2nd. edition, John Wiley & Sons, New York, 1981.

[Dup] Dupuy, N., Ruckebush, C., Duponchel, L., Beurdeley-Saudou, P., Amram, B., Huvenne, J.P., and Legrand, P., *Anal. Chim. Acta*, **335** (1996) 79.

[Est] Estienne, F., Pasti, L., Walczak, B., Despagne, F., Jouan-Rimbaud, D., Massart, D.L., and de Noord, O.E., “A comparison of multivariate calibration techniques applied to experimental NIR data sets. Part II: Predictive ability under extrapolation conditions.” (in preparation).

[Flet] Fletcher, R., *Practical Methods of Optimisation, Vol.1: Unconstrained Optimisation*, John Wiley & Sons, New-York, 1980.

[For] Forina, M., Drava, G., Boggia, R., Lanteri, S., and Conti, P., *Anal. Chim. Acta*, **295** (1994) 109.

[Gel] Geladi, P., MacDougall, D., and Martens, H., *Appl. Spectrosc.*, **39** (1985) 491.

[Gem1] Gemperline, P.J., Long, J.R., and Gregoriou, R.V., *Anal. Chem.*, **63** (1991) 2313.

[Gem2] Gemperline. P. J., *Chemom. Intell. Lab. Syst.*, **39 (**1997) 29.

[Gema] Geman, S., Bienenstock, E., and Doursat, R., *Neural Comput.*, **4** (1992) 1.

[Goo] Goodacre, R., Neal, M.J., and Kell, D.B., *Anal. Chem.*, **66** (1994) 1070.

[Gor] Gorry, P.A., *Anal. Chem.*, **62** (1990) 570.

[Gru] Grubbs, F.E., and Beck, G., *Technometrics*, **14** (1972) 847.

[Had] Hadi, A.S., and Simonoff, J.S., *J. Am. Stat. Assoc.*, **88** (1993) 1264.

[Har] Hartnett, M., Lightbody, G., and Irwin, G.W., *Analyst*, **121** (1996) 749.

[Haw] Hawkins, D.M., Bradu, D., and Kass, G.V., *Technometrics*, **26** (1984) 197.

[Hert] Hertz, J., Krogh, A., and Palmer, R., *Introduction to the Theory of Neural Computation*, Addison Wesley, Redwood City, CA, 1991.

[Hin] Hindle, P.H., and Smith, C.R.R. , *J. Near-infrared Spectrosc.*, **4** (1996) 119.

[Hon] Honigs, D.E., Hieftje, G.H., Mark, H.L., and Hirschfeld, T.B., *Anal. Chem.*, **57** (1985) 2299.

[Hop] Hopkins, B., *Ann. Bot.*, **18** (1954) 213.

[Hor] Hornik, K., Stinchcombe, M., and White, H., *Neural Networks*, **2** (1989) 359.

[Hus] Hush, D.R., and Horne, B.G., *IEEE Signal Process. Mag.*, **1** (1993) 8.

[Isa] Isaksson, T., and Næs, T., *Appl. Spectrosc.*, **42** (1988) 1273.

[Iso1] *Statistics - Vocabulary and Symbols Part 1.* ISO standard 3534 (E/F), 1993.

[Iso2] *Accuracy (trueness and precission) of measurement methods and results*. ISO standard 5725 1-6, 1994.

[Jac] Jacobs, R.A., *Neural Networks*, **1** (1988) 226.

[Jia1] Jiang, J.H., Wang, J.H., Song, X.H., and Yu, R.Q., *J. Chemom.*, **10** (1996) 253.

[Jia2] Jiang, J.H., Wang, J.H., Chu, X., and Yu, R.Q., *Anal. Chim. Acta*, **336** (1996) 209.

[Jou1] Jouan-Rimbaud, D., Walczak, B., Massart, D.L., Last, I.R., and Prebble, K.A., *Anal. Chim. Acta*, **304** (1995) 285.

[Jou2] Jouan-Rimbaud, D., Bouveresse, E., Massart, D.L., and de Noord, O.E., “Detection of prediction outliers and inliers in multivariate calibration.”, *Anal. Chim. Acta*, in press.

[Jou3] Jouan-Rimbaud, D., Massart, D.L., Saby, C.A., and Puel, C., *Chemom. Intelligent Lab. Syst.*, **40** (1998) 129.

[Jou4] Jouan-Rimbaud, D., Massart, D.L., Saby, C.A., and Puel, C., *Anal. Chim. Acta*, **350** (1997) 149.

[Kan] Kanjilal, P.P., and Banerjee, D.N., *IEEE Signal Process. Mag.*, **1** (1993) 8.

[Kat] Kateman, G, and Smits, J.R.M., *Anal. Chim. Acta*, **277** (1993) 179.

[Kel] Kelly, P.C., *J. Assoc. Off. Anal. Chem.*, **73** (1990) 58.

[Ken] Kennard, R.W., and Stone, L.A., *Technometrics*, **11** (1969) 137.

[Kol] Kolen, J.F., and Pollack, J.B., in *Advances in neural information processing systems*, vol.3, ed. Lippmann, R.P., Moody, J.E., and Touretzky, D.S., Morgan Kaufmann, San Mateo, CA, 1991.

[Kub] Kubelka, P., *Journal of the Optical Society of America*, **38**(5) (1948) 448.

[Law] Lawson, R.G., and Jurs. P.J., *J. Chem. Inf. Comput. Sci.*, **30** (1990) 36.

[Lipp] Lippmann, R.P., *IEEE Trans. Neural Networks*, **5** (1995) 1061.

[Loo] Looney, C.G., *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists*, Oxford University Press, New York, 1997.

[Lor] Lorber, A., and Kowalski, B.R., *J. Chemom.*, **2** (1988) 67.

[Loz] Lozano, J., Novic, M., Rius, F.X., and Zupan, J. *Chemom. Intell. Lab. Syst.*, **28** (1995) 61.

[Maj] Majcen, N., Rajer-Kanduc, K., Novic, M., and Zupan, J., *Anal. Chem.*, **67** (1995) 2154.

[Mal] Malinowski, E.R., *Factor Analysis in Chemistry*, John Wiley & Sons, New York, 2nd edn., 1991.

[Mass] Massart, D.L., Vandeginste, B.M.G., Buydens, L.M.C., De Jong, S., Lewi, P.J., and Smeyers-Verbeke, J., *Handbook of chemometrics and qualimetrics: part A.* Elsevier, Amsterdam, 1997.

[Mast] Masters, T., *Practical Neural Network Recipes in C++*, Academic Press, Boston, 1993.

[Mel] Meloun, M., Militký, J., and Forina, M., "*Chemometrics for analytical chemistry. Vol.1: PC-aided statistical data analysis*", Ellis Horwood, Chichester (England), 1992.

[Mer] Mertens, B., Thompson, M., and Fearn, T., *Analyst*, **119** (1994) 2777.

[Mil1] Miller, J.C., and Miller, J.N., *Statistics for analytical chemistry*, Ellis Horwood, Chilchester, 1988.

[Mil2] Miller, C.E., *NIR News*, **4 (6)** (1993) 3.

[Min] Minai, A.A., and Williams, R.D., in *International Joint Conference on Neural Networks*, vol.3, 1990.

[Næs1] Næs, T., *J. Chemom.*, **1** (1987) 121.

[Næs2] Næs, T., Isaksson, T., and Kowalski, B.R., *Anal. Chem.*, **62** (1990) 664.

[Næs3] Næs, T., and Isaksson, T., *NIR News*, **5(4)** (1994) 7.

[Noo] de Noord, O.E., *Chemom. Intell. Lab. Syst.*, **23** (1994) 65.

[Osb] Osborne, B.G., *Analyst*, **113** (1988) 263.

[Popp] Poppi, R.J., and Massart, D.L., *Anal. Chim. Acta*, **375** (1998) 187.

[Rum] Rumelhart, D.E., and McClelland, J.L., *Parallel Distributed Processing*, vol.1, MIT Press, Cambridge, MA, 1986.

[Sav] Savitzky, A., and Golay, M.J.E., *Anal. Chem.*, **36** (1964) 1627.

[Sea] Seasholtz, M.B., and Kowalski, B.R., *Anal. Chim. Acta*, **277** (1993) 165.

[Sek] Sekulic, S., Seasholtz, M.B., Wang, Z., Kowalski, B.R., Lee, S.E., and Holt, B.R., *Anal. Chem.*, **65** (1993) 835A.

[Smi] Smits, J.R.M., Melssen, W.J., Buydens, L.M.C., and Kateman, G., *Chemom. Intell. Lab. Syst.*, **22** (1994) 165.

[Sne] Snee, R.D., *Technometrics*, **19** (1977) 415.

[Son] Sontag, E.D., in *Advances in Neural Information Processing Systems*, vol.3, ed. Lippmann, R.P., Moody, J.E., and Touretzky, D.S., Morgan Kaufmann, San Mateo, CA, 1991.

[Ste] Steinier, J., Termonia, Y., and Deltour, J., *Anal. Chem.*, **44** (1972) 1906.

[Svo] Svozil, D., Kvasnicka, V., and Pospíchal, J., *Chemom. Intell. Lab. Syst.*, **39** (1997) 43.

[Tet1] Tetko, I.V., Livingstone, D.J., and Luik, A.I., *J. Chem. Inf. Comput. Sci.*, **35** (1995) 826.

[Tet2] Tetko, I.V., Villa, A.E.P., and Livingstone, D.J., *J. Chem. Inf. Comput. Sci.*, **36** (1996) 794.

[Vog] Vogt, N.B., *Chemom. Intell. Lab. Syst.*, **7** (1989) 119.

[Wal1] Walczak, B., *Chemom. Intell. Lab. Syst.*, **28** (1995) 259.

[Wal2] Walczak, B., *Chemom. Intell. Lab. Syst.*, **29** (1995) 63.

[Wal3] Walczak, B., *Anal. Chim. Acta*, **322** (1996) 21.

[Wal4] Walczak, B., and Massart, D.L., *Chemom. Intell. Lab. Syst.*, **36** (1997) 81.

[Wal5] Walczak, B., and Massart, D.L., *Anal. Chim. Acta*, **331** (1996) 177.

[Wan] Wang, J.H., Jiang, J.H., and Yu, R.Q., *Chemom. Intell. Lab. Syst.*, **34** (1996) 109.

[Wol] Wold, S., *Technometrics*, **20** (1978) 397.

[Zha] Zhang, L. Jiang, J.H., Liu, P., Liang, Y.Z., and Yu, R.Q., *Anal. Chim. Acta*, **344** (1997) 29.

APPENDIX V.4.

APPENDIX V.4.

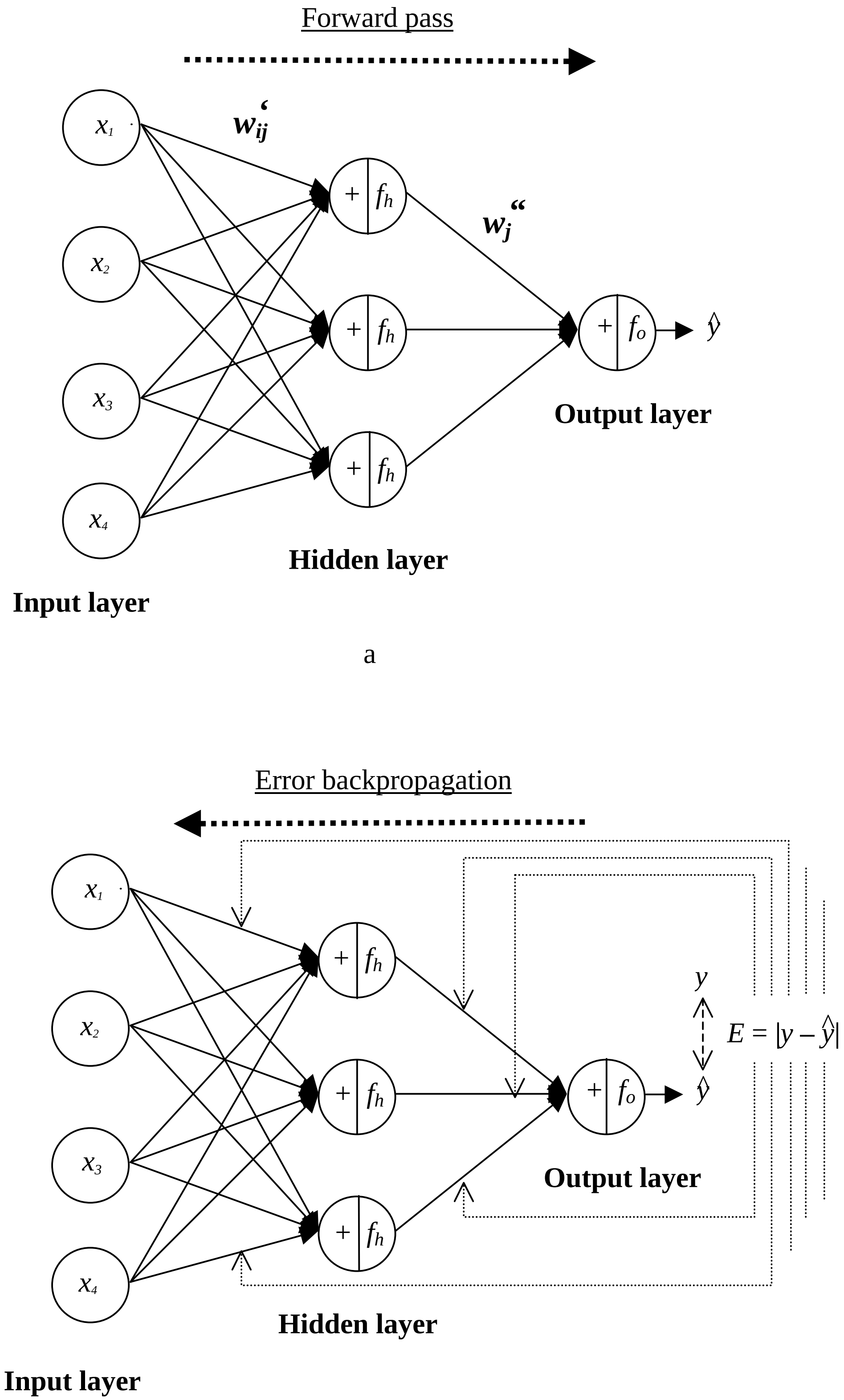
Figure captions

1. Feed-forward NN training.
   a) Forward pass.
   b) Error backpropagation.
2. General flow-scheme of the steps needed to develop a calibration model.
3. APaRP plot for visual detection of nonlinearities. Printed with the permission of *Analytica Chimica Acta* Centner, V., Massart, D.L., and de Noord, O.E., *Anal. Chim. Acta*, **376** (1998) 153.
4. a) Application of the Savitzky-Golay method (window size 7, m=3; cubic polynomial, n=3), o measured data, * smoothed data.
   b) Smoothed results for data set in a: ... original data, o measured data, * smoothed data.
   c) … 1st derivative of the cubic polynomial in the different windows in a, * estimated 1st derivative data.
   d) 1st derivative of the data set in a: ... real 1st derivative, * estimated values (window size = 13, m=6; cubic polynomial, n=3).
5. NIR spectra for different wheat samples and several preprocessing methods applied to them.
   a) Original data.
   b) 1st derivative.
   c) 2nd derivative.
   d) Offset-corrected.
   e) SNV-corrected.
   f) Detrended.
   g) Detrended+SNV-corrected.
   h) MSC corrected.
6. An example of strongly clustered data.
7. a) Plot of 200 objects normally distributed in two variables x1 and x2.
   b) The distance curves of the 200 normally distributed objects.
   c) Clustered data, normally distributed in each cluster.
   d) The distance curves of the clustered data.
8. Hopkins statistics applied to two different data sets. Open circles represent real objects, closed circles selected real objects and asterisks represent artificial objects generated over the data space.
   a) H value = 0.49.
   b) H value = 0.73.
   c) H value = 0.69.

  d) H value = 0.56 (the same data set as in c), after PCA rotation).

9. Contour plot corresponding to k=4 with the 10% percentile method and with (*) the identified inlier (Adapted from Jouan-Rimbaud, D., “Method Development in Linear Multivariate Calibration Applied to Spectroscopic Data”, *Doctoral Thesis* (1997), Vrije universiteit Brussel)
10. Typical evolution of training and monitoring errors as a function of number of iterations.
11. The first 24 points selected using different algorithms.
    a) Kennard & Stone method (closest point to the mean included).
    b) Næs clustering method.
    c) Duplex method with (o) the calibration set and (*) the test set.
12. Usual non-linear transfer functions.
    a) Hyperbolic tangent.
    b) Sigmoid).
13. Evolution of mean squared error of as a function of the complexity of a model.
14. Strategy for NN topology optimisation.
15. Example of three-layer 4-3-1 NN with direct connections.
16. Predictions for linear model with incorrect NN topology.
17. Detection of representativity problems between training and monitoring set on RMS error curves.
    a) Lack of representativity.
    b) Chance correlation with initial set of weights.
18. Visualisation of sample repartitions on hidden nodes (hn) output maps for ICP data.
    a) HN1-HN2.
    b) HN1-HN3.
    c) HN2-HN3).
19. Visualisation of hidden nodes activation.
    a) ICP data, HN1.
    b) ICP data, HN2.
    c) ICP data, HN3.
    d) Polymer data, HN1.
    e) Polymer data, HN2).
20. Illustration of the effect of an outlier (*) to the true model (---) influencing the regression line (—).

APPENDIX V.4.

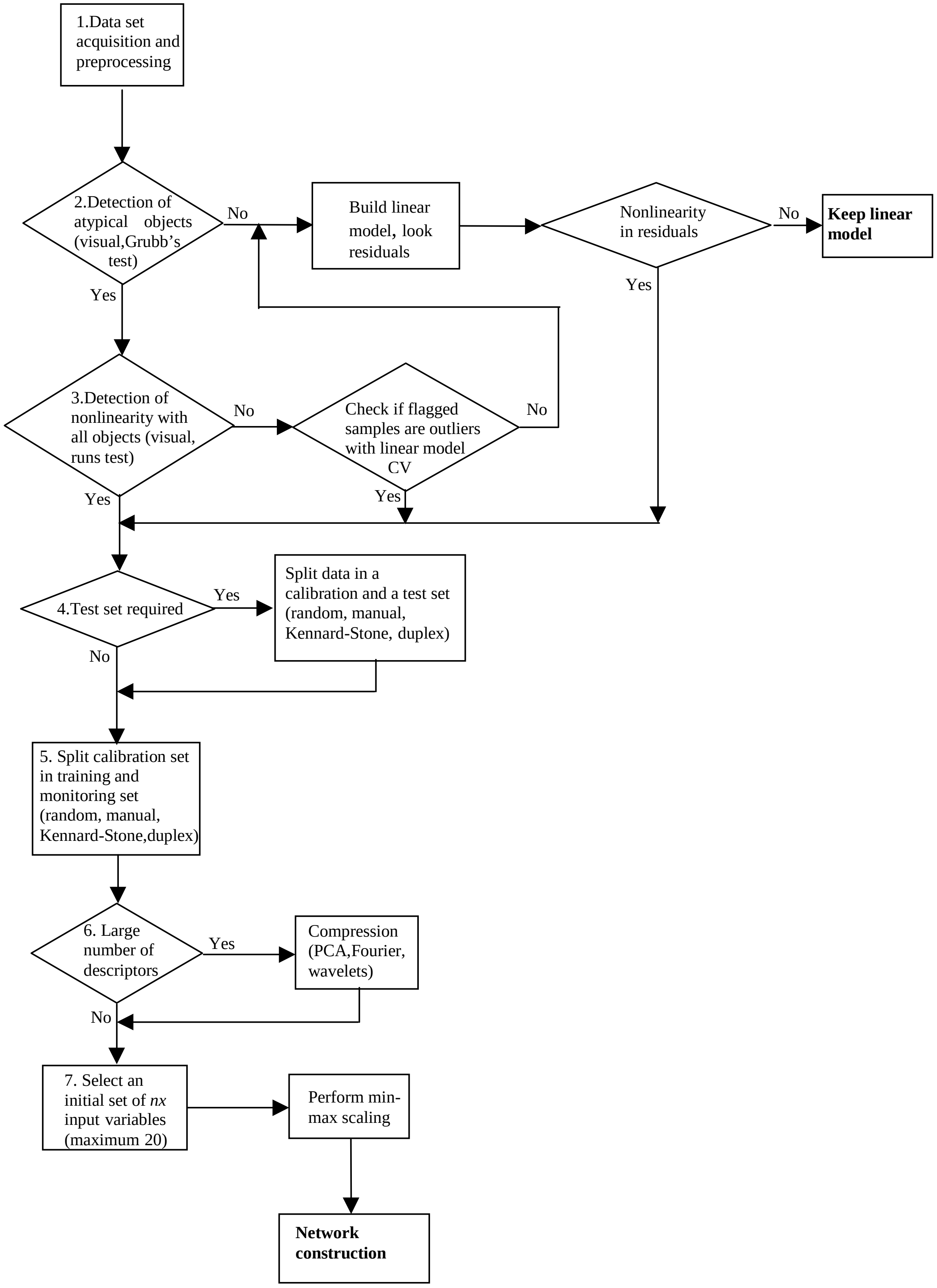


a

b

Figure 1

Figure 2


1.Data set acquisition and preprocessing
2.Detection of atypical objects (visual,Grubb's test)
No
Build linear model, look residuals
Nonlinearity in residuals
No
Keep linear model
Yes
Yes
3.Detection of nonlinearity with all objects (visual, runs test)
No
Check if flagged samples are outliers with linear model CV
No
Yes
Yes
4.Test set required
Yes
Split data in a calibration and a test set (random, manual, Kennard-Stone, duplex)
No
5. Split calibration set in training and monitoring set (random, manual, Kennard-Stone,duplex)
6. Large number of descriptors
Yes
Compression (PCA,Fourier, wavelets)
No
7. Select an initial set of *nx* input variables (maximum 20)
Perform min-max scaling
Network construction

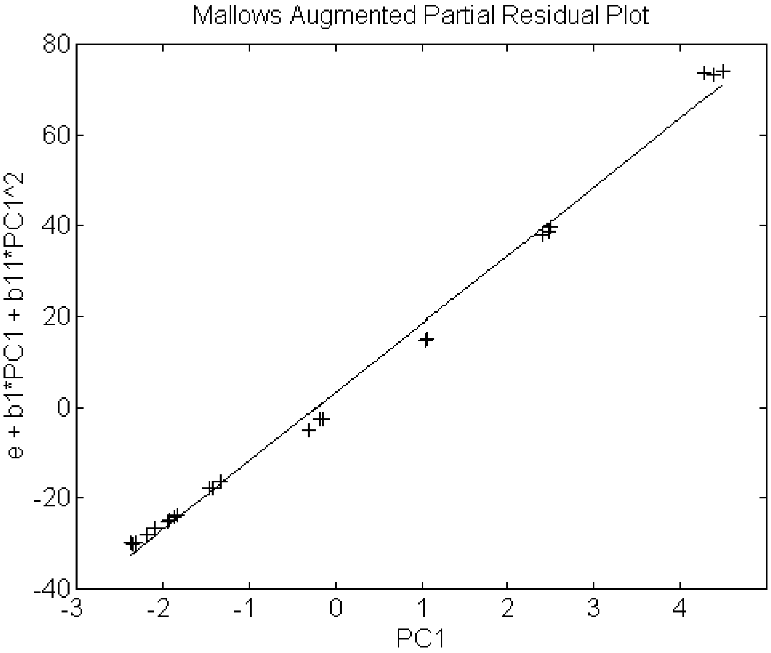
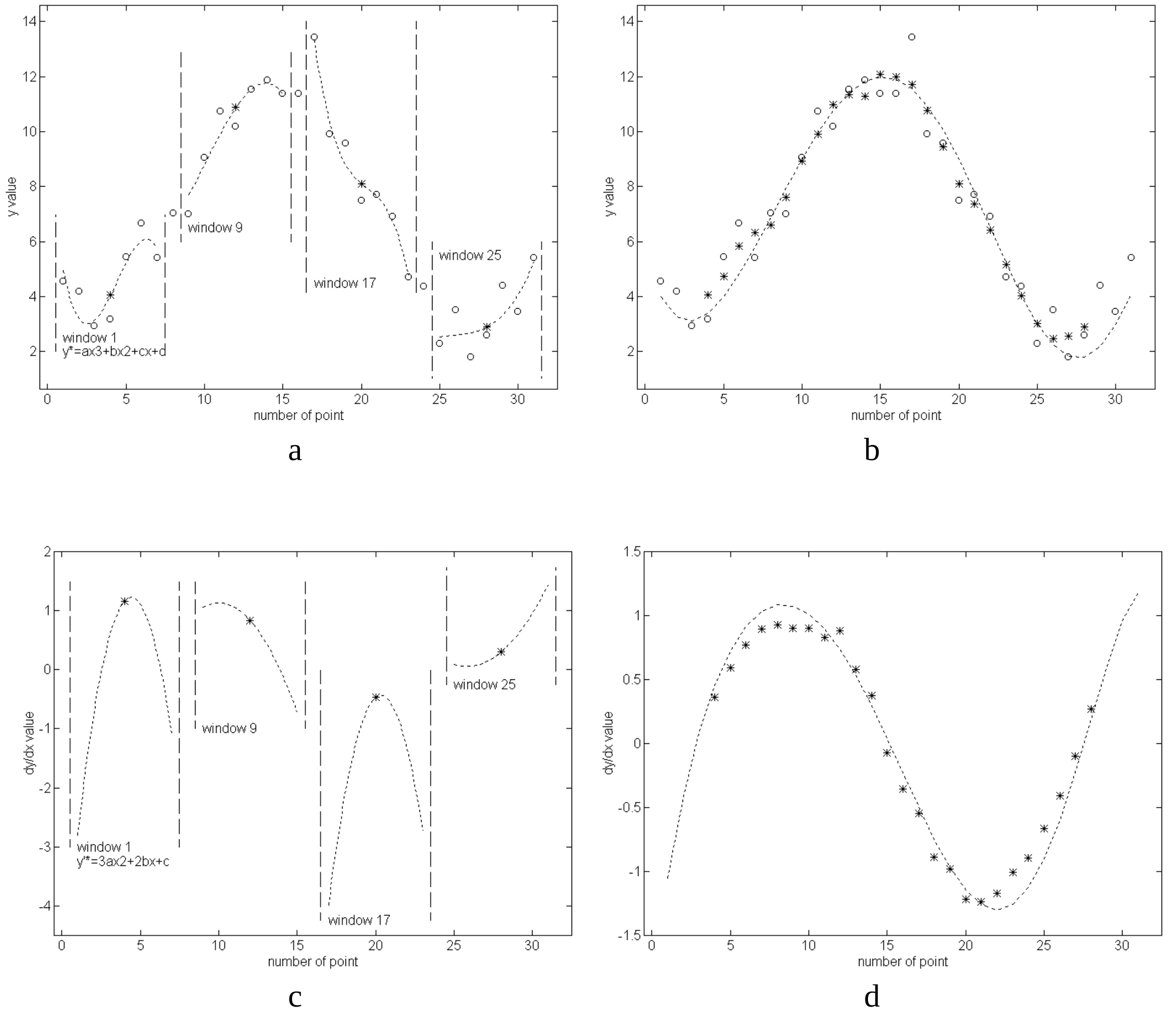

Mallows Augmented Partial Residual Plot
80
60
40
20
0
-20
-40
-3
-2
-1
0
1
2
3
4
PC1
e + b1*PC1 + b11*PC1^2

Figure 3

APPENDIX V.4.



a

b

c

d

Figure 4

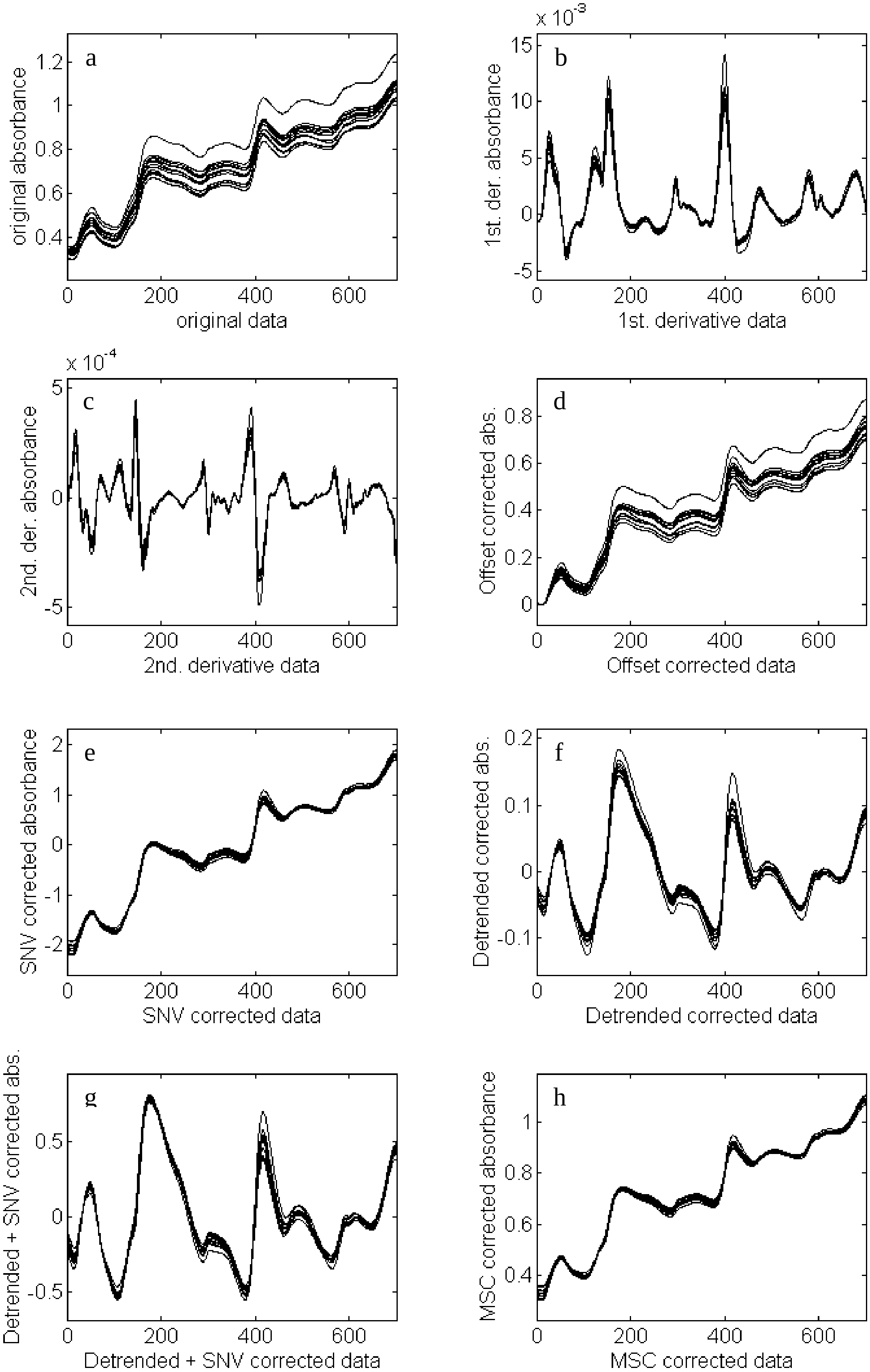

a
original absorbance
1.2
1
0.8
0.6
0.4
0
200
400
600
original data
b
x 10^-3
1st. der. absorbance
15
10
5
0
-5
1st. derivative data
c
x 10^-4
2nd. der. absorbance
5
0
-5
2nd. derivative data
d
Offset corrected abs.
0.8
0.6
0.4
0.2
0
Offset corrected data
e
SNV corrected absorbance
2
1
0
-1
-2
SNV corrected data
f
Detrended corrected abs.
0.2
0.1
0
-0.1
Detrended corrected data
g
Detrended + SNV corrected abs.
0.5
0
-0.5
Detrended + SNV corrected data
h
MSC corrected absorbance
1
0.8
0.6
0.4
MSC corrected data

Figure 5

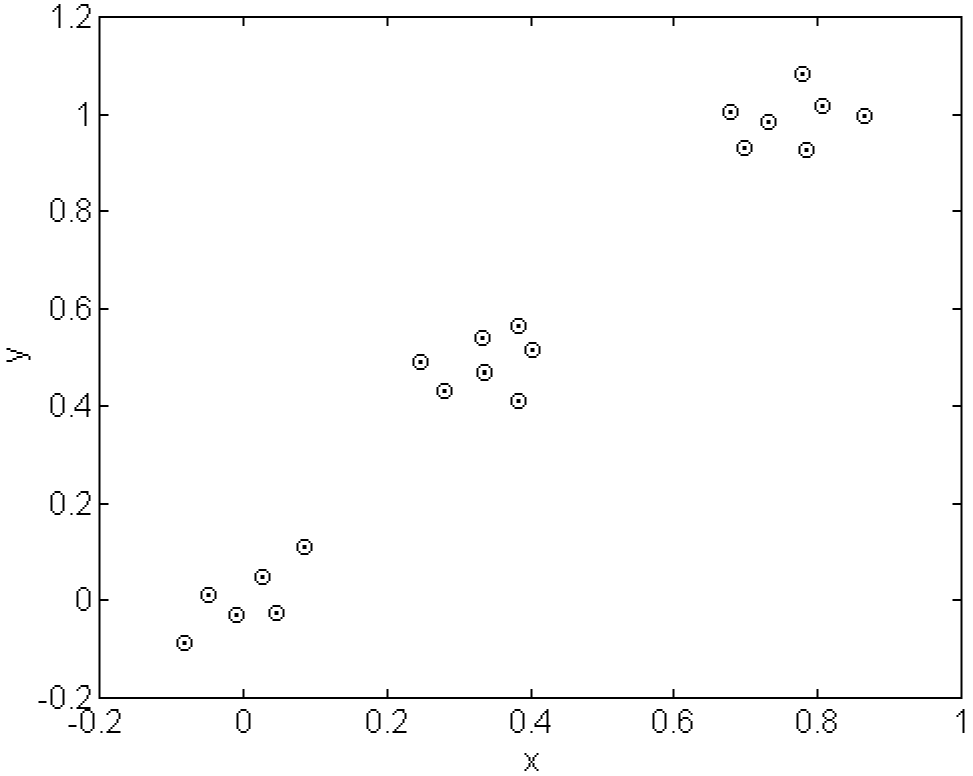


Figure 6

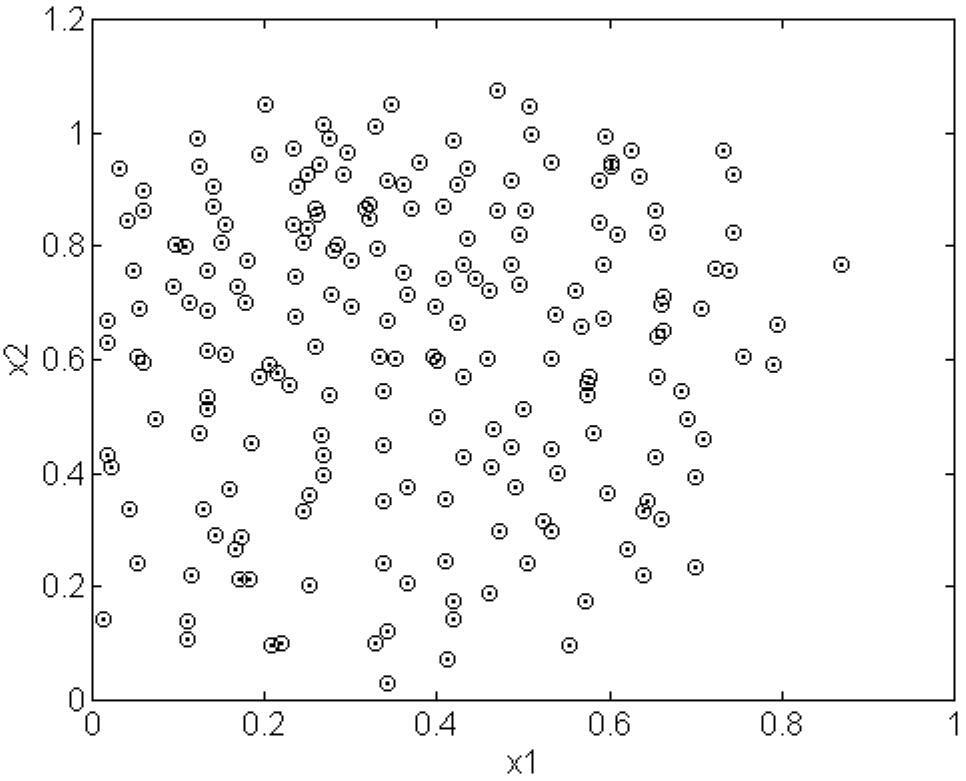


Figure 7a

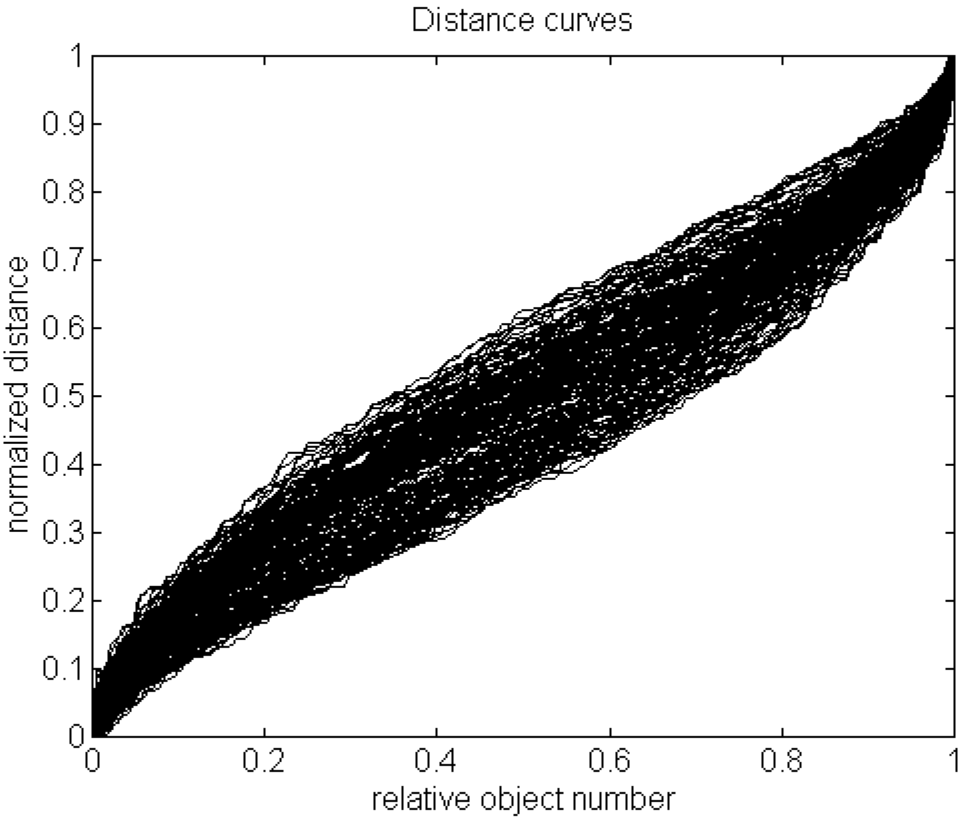

Distance curves
normalized distance
relative object number
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1

Figure 7b

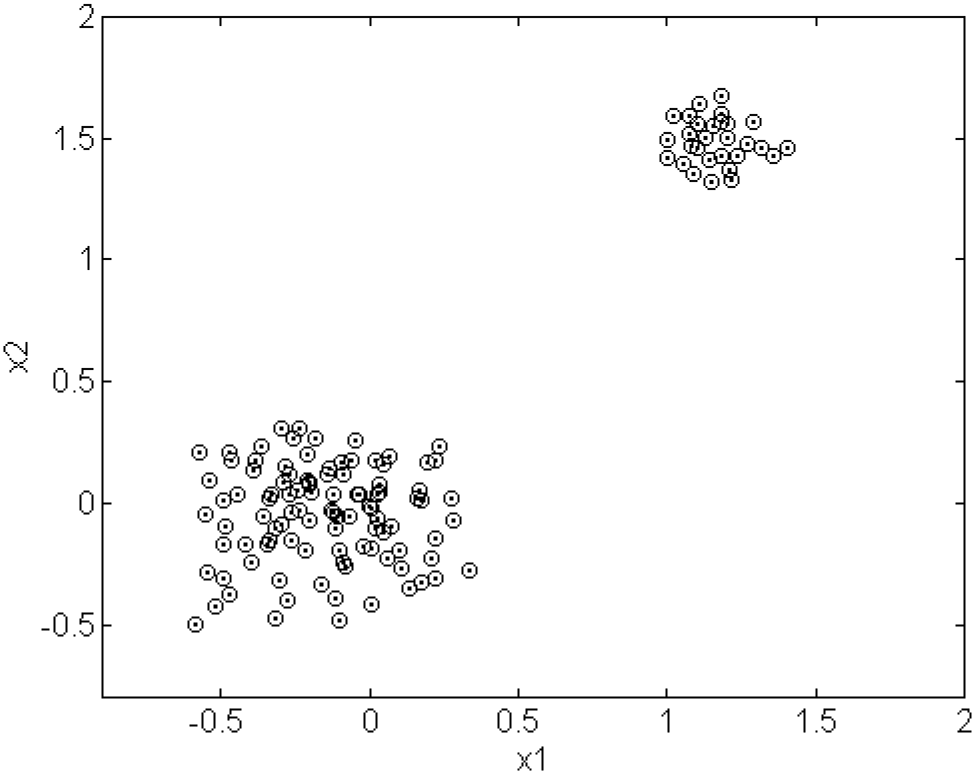


Figure 7c

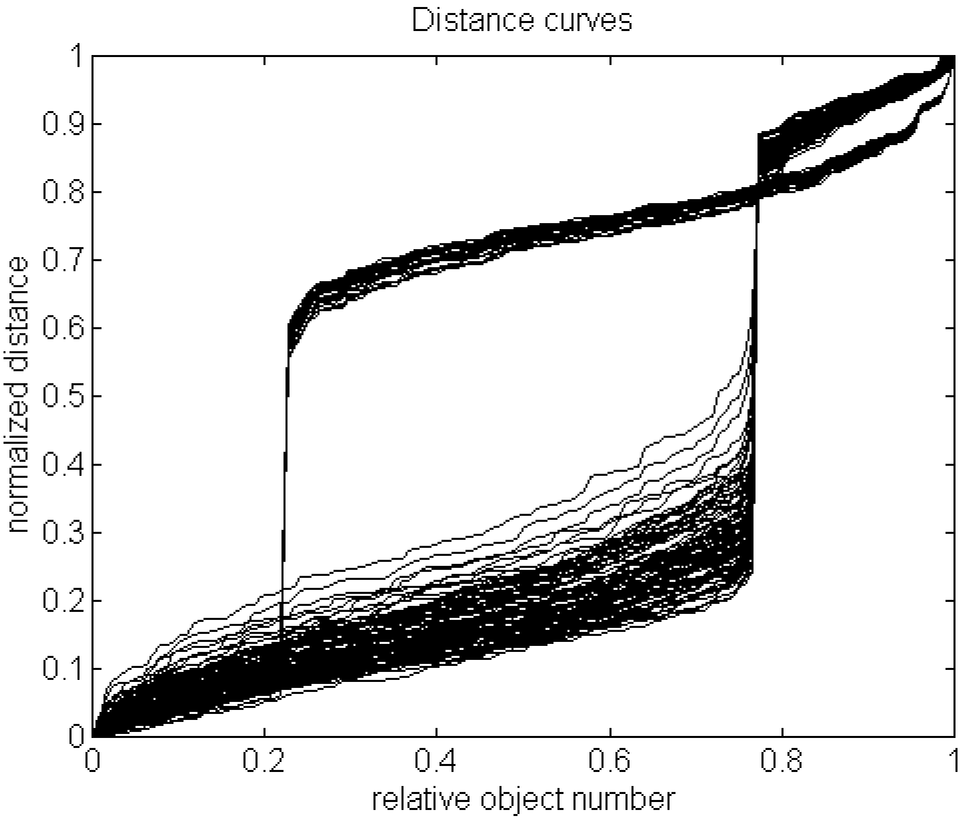
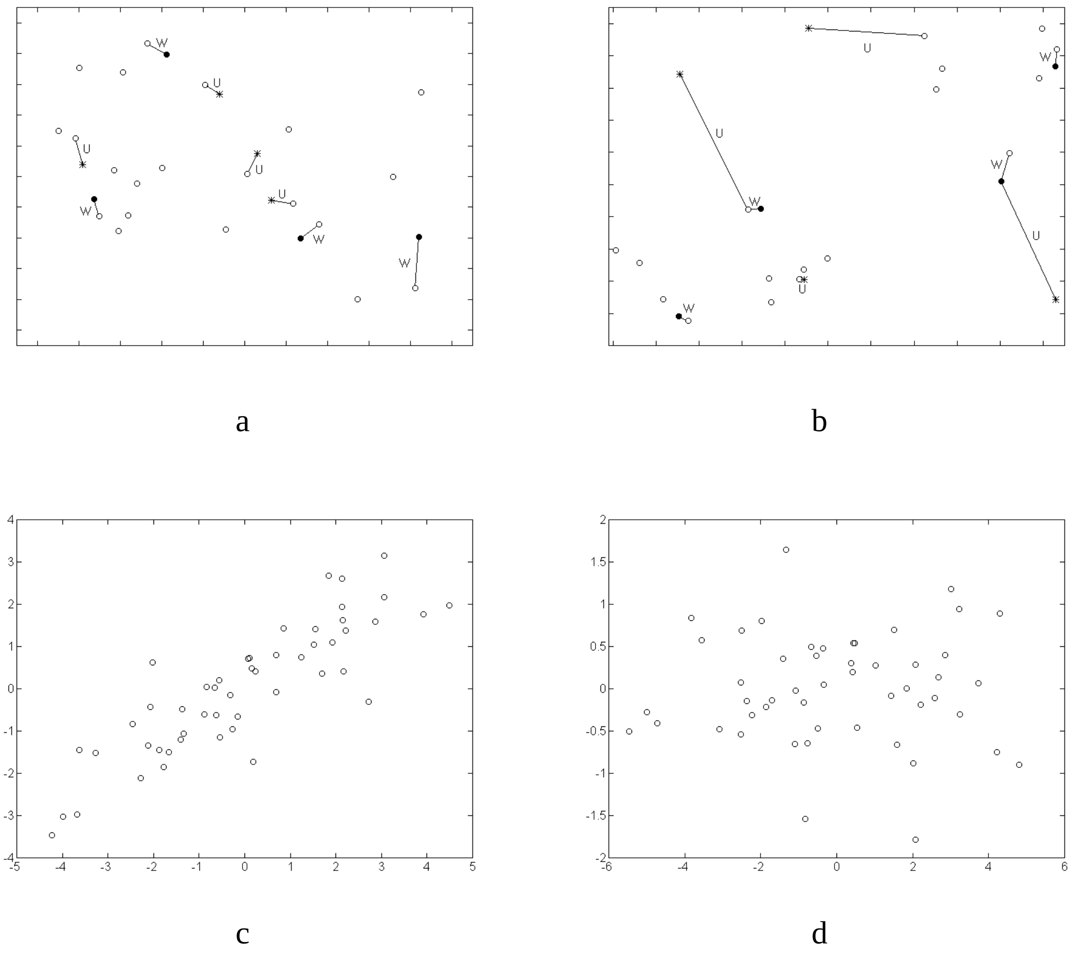

Distance curves
normalized distance
relative object number
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1

Figure 7d

APPENDIX V.4.



a

b

c

d

Figure 8

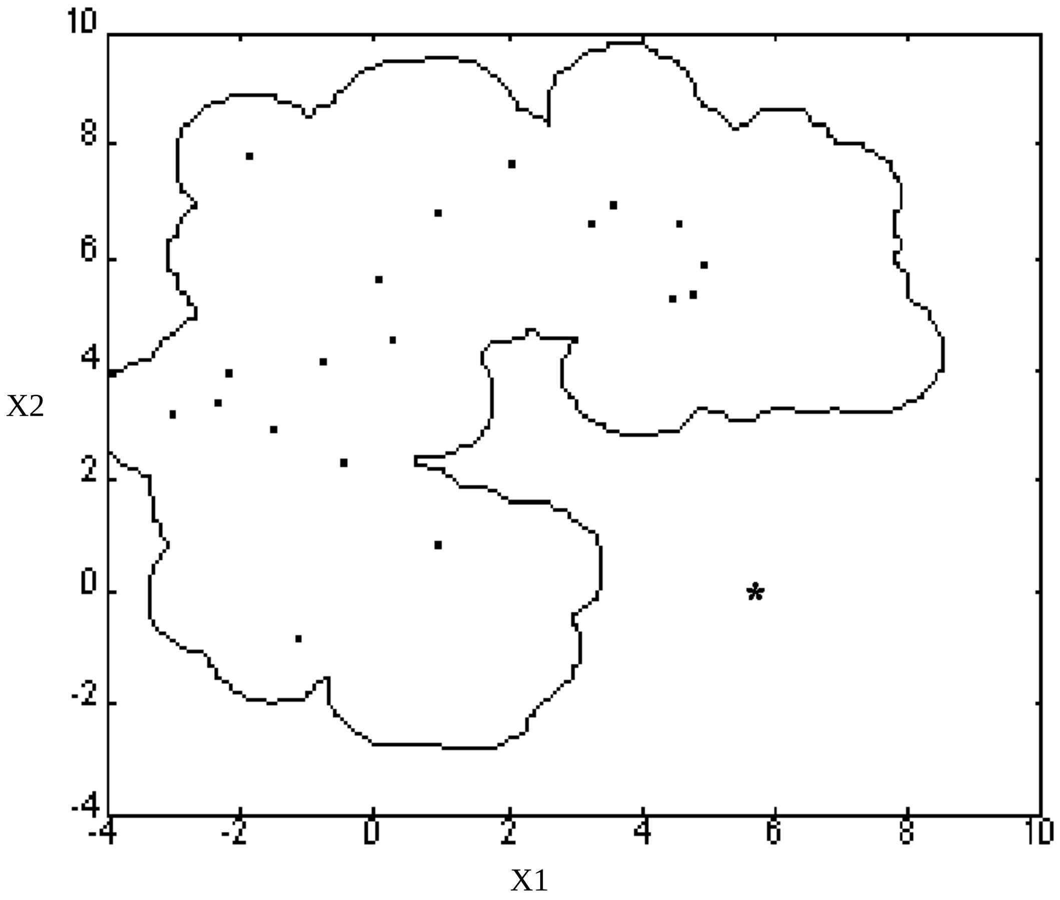


Figure 9

APPENDIX V.4.

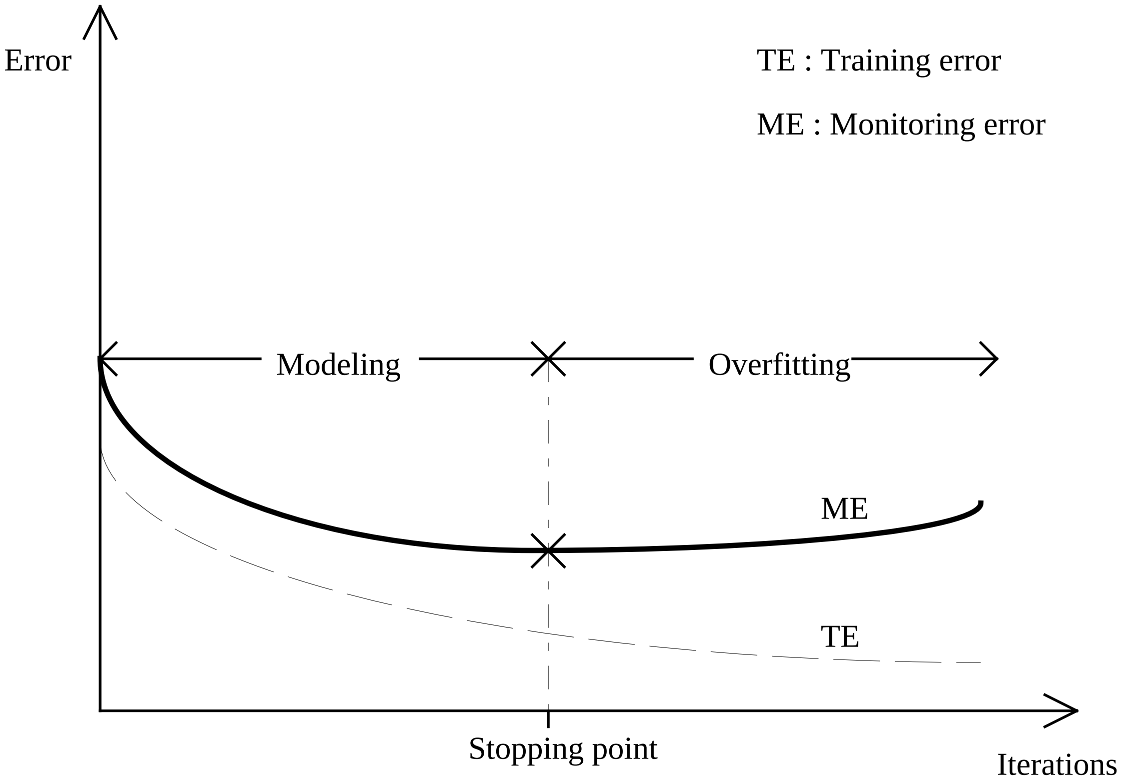
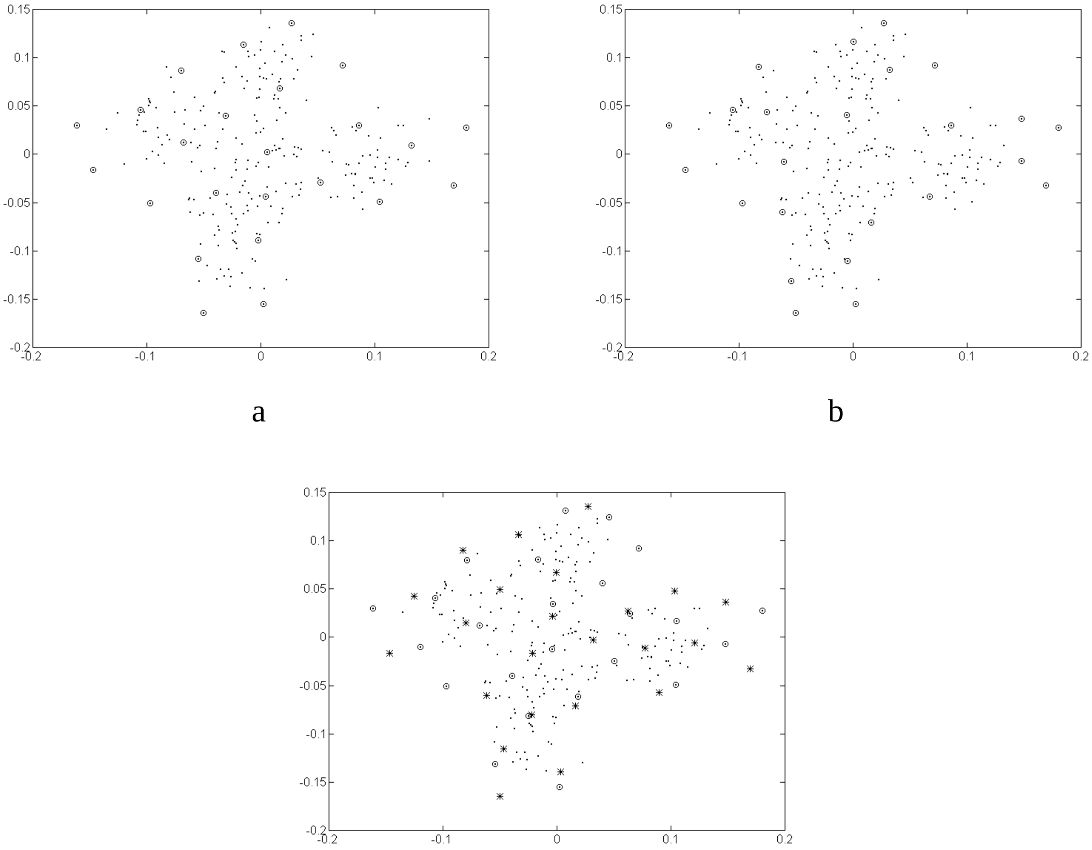


Figure 10

APPENDIX V.4.



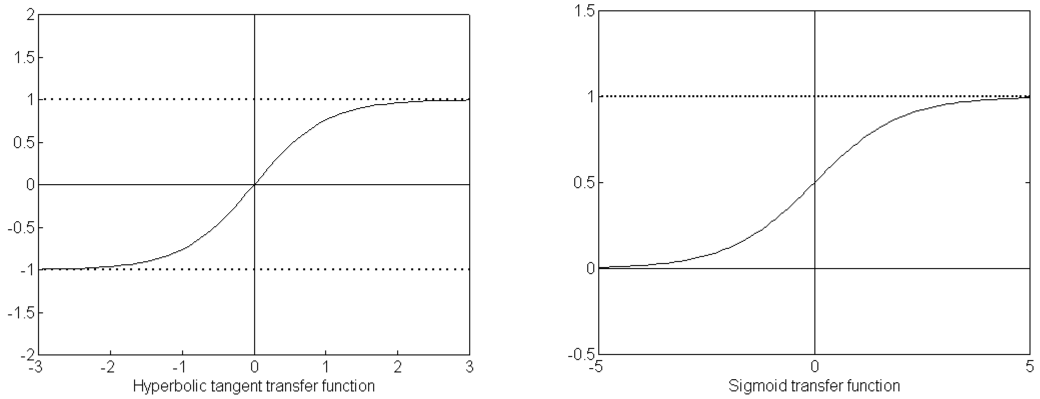
a

b

c

Figure 11

APPENDIX V.4.



a

b

Figure 12

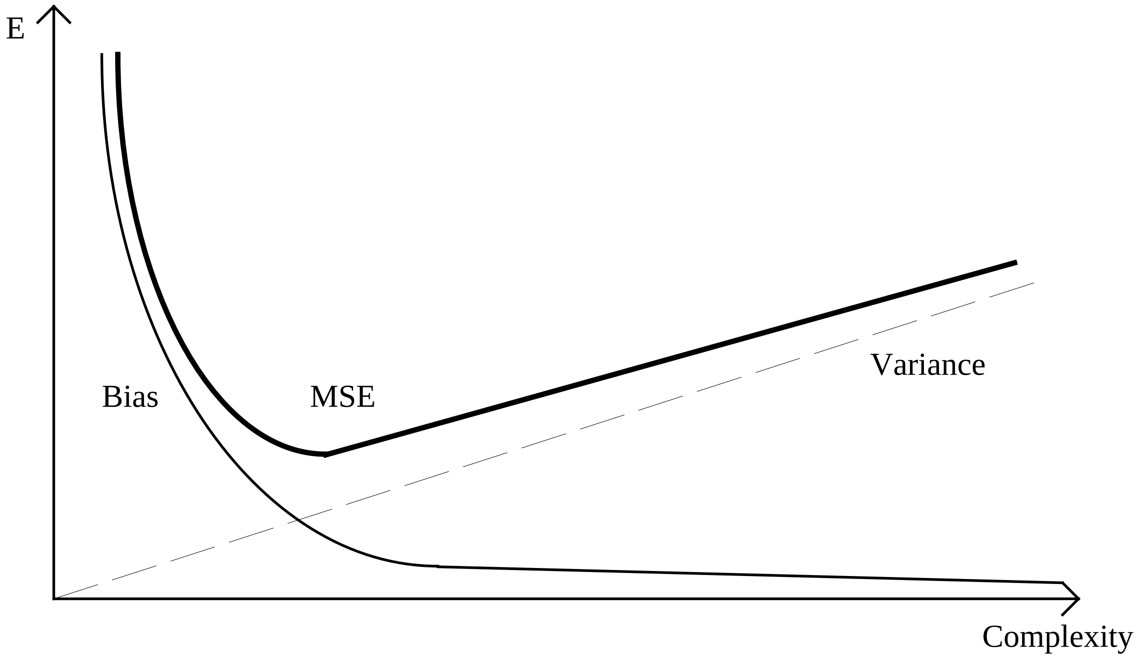
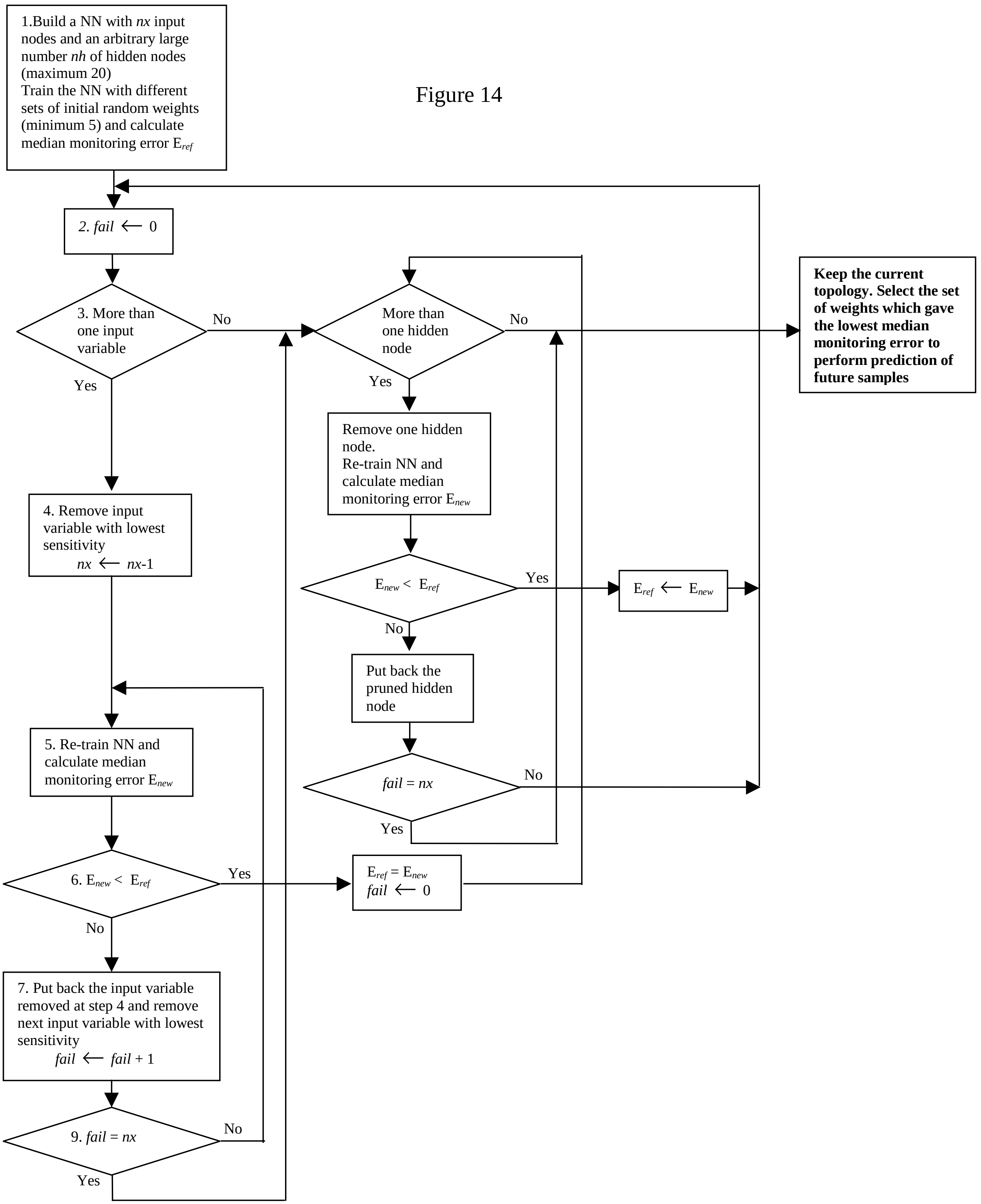


Figure 13

APPENDIX V.4.

Figure 14


1.Build a NN with *nx* input nodes and an arbitrary large number *nh* of hidden nodes (maximum 20)
Train the NN with different sets of initial random weights (minimum 5) and calculate median monitoring error E_{ref}
2. *fail* ← 0
3. More than one input variable
No
Yes
4. Remove input variable with lowest sensitivity
nx ← *nx*-1
5. Re-train NN and calculate median monitoring error E_{new}
6. $E_{new} < E_{ref}$
Yes
No
7. Put back the input variable removed at step 4 and remove next input variable with lowest sensitivity
fail ← *fail* + 1
9. *fail* = *nx*
No
Yes
More than one hidden node
No
Yes
Remove one hidden node.
Re-train NN and calculate median monitoring error E_{new}
$E_{new} < E_{ref}$
Yes
No
E_{ref} ← E_{new}
Put back the pruned hidden node
fail = *nx*
No
Yes
$E_{ref} = E_{new}$
fail ← 0
Keep the current topology. Select the set of weights which gave the lowest median monitoring error to perform prediction of future samples

APPENDIX V.4.

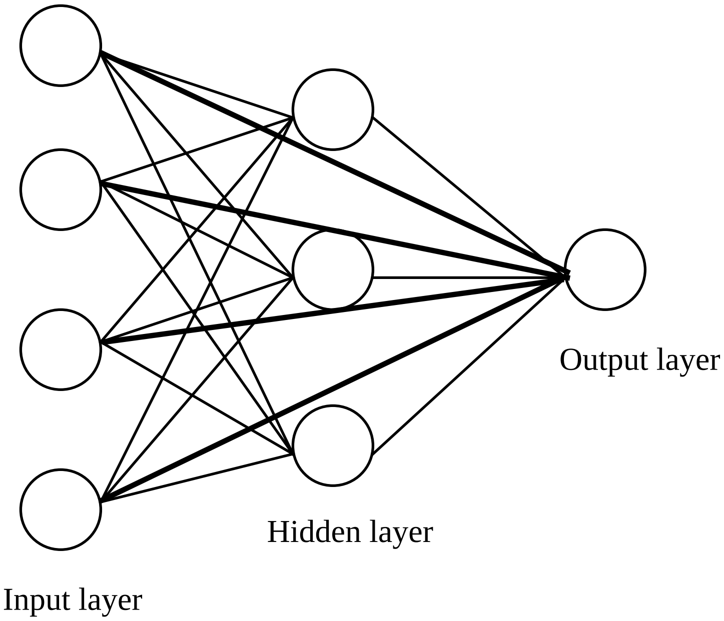


Figure 15

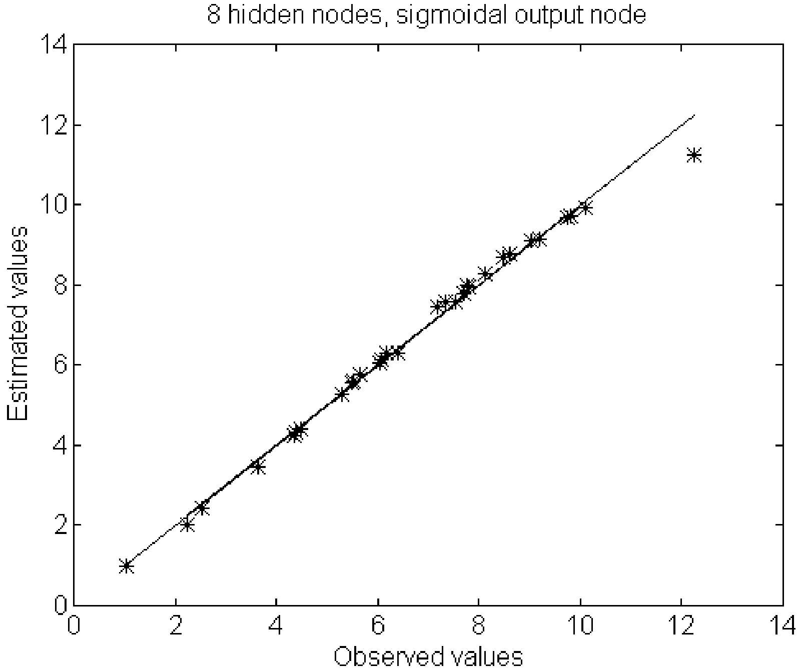

8 hidden nodes, sigmoidal output node
Estimated values
Observed values
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

Figure 16

APPENDIX V.4.

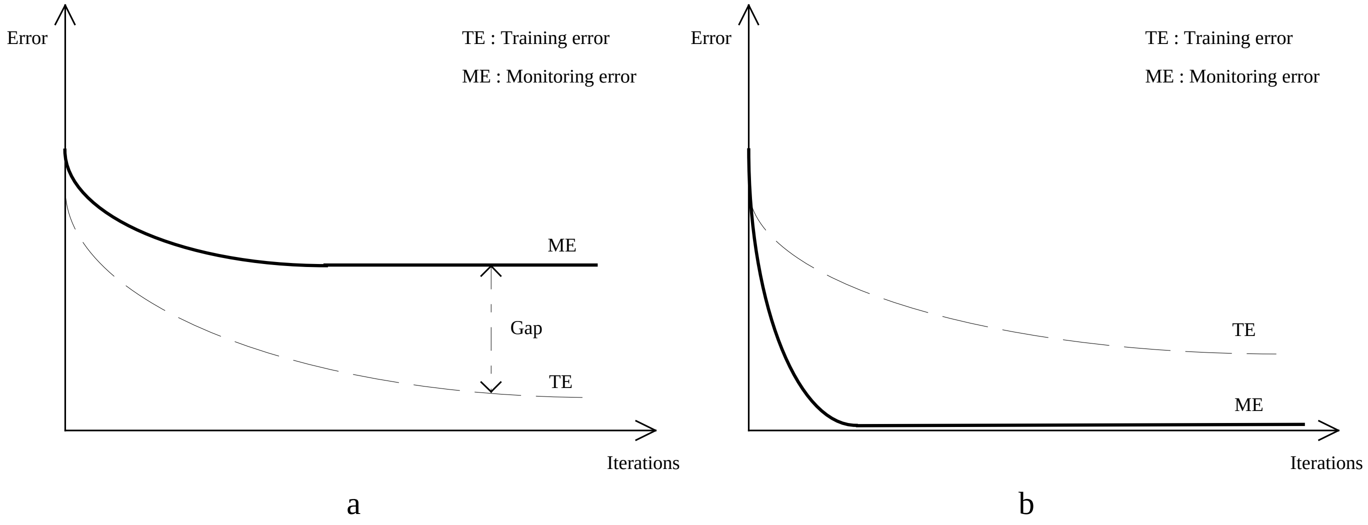
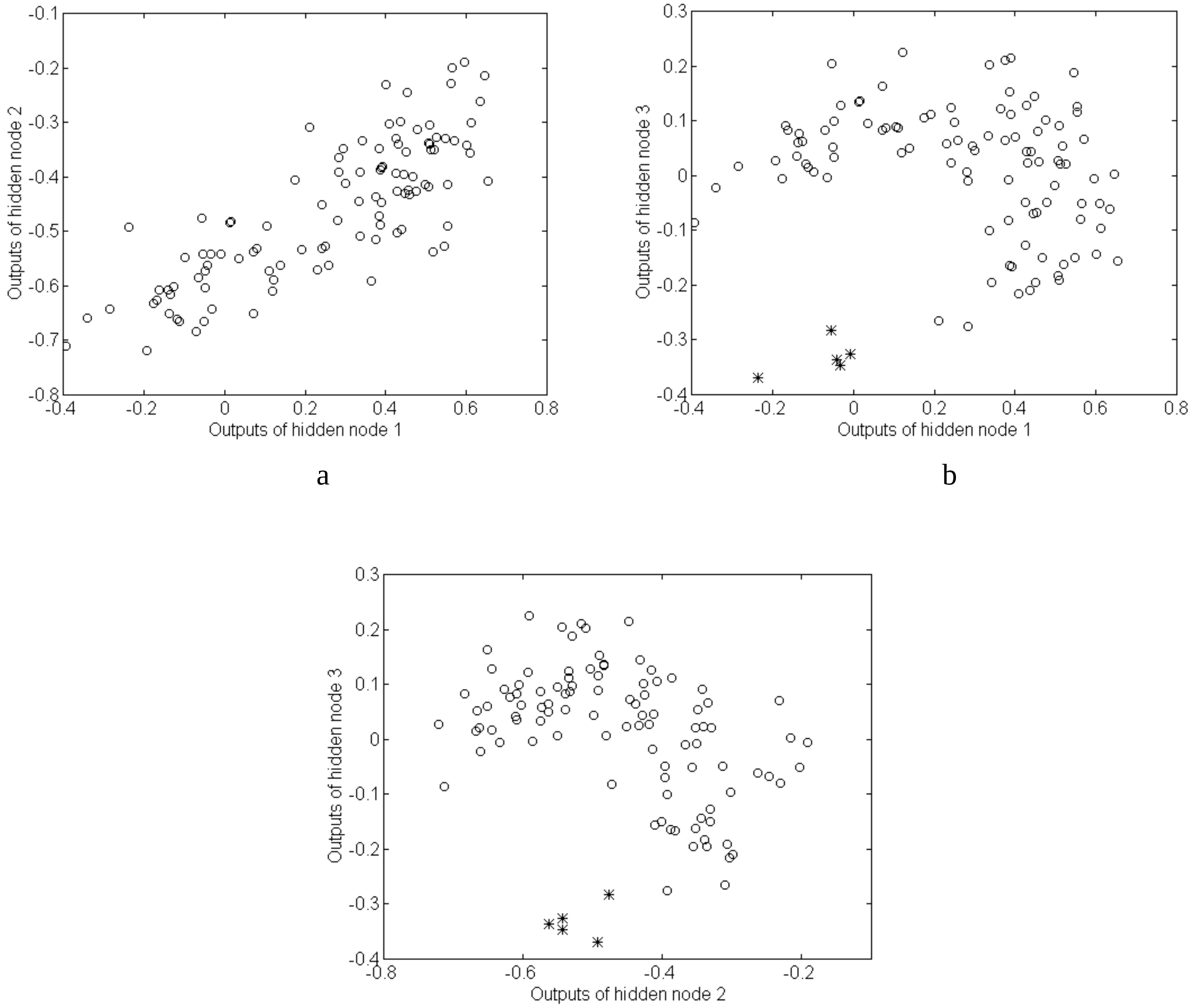


Figure 17

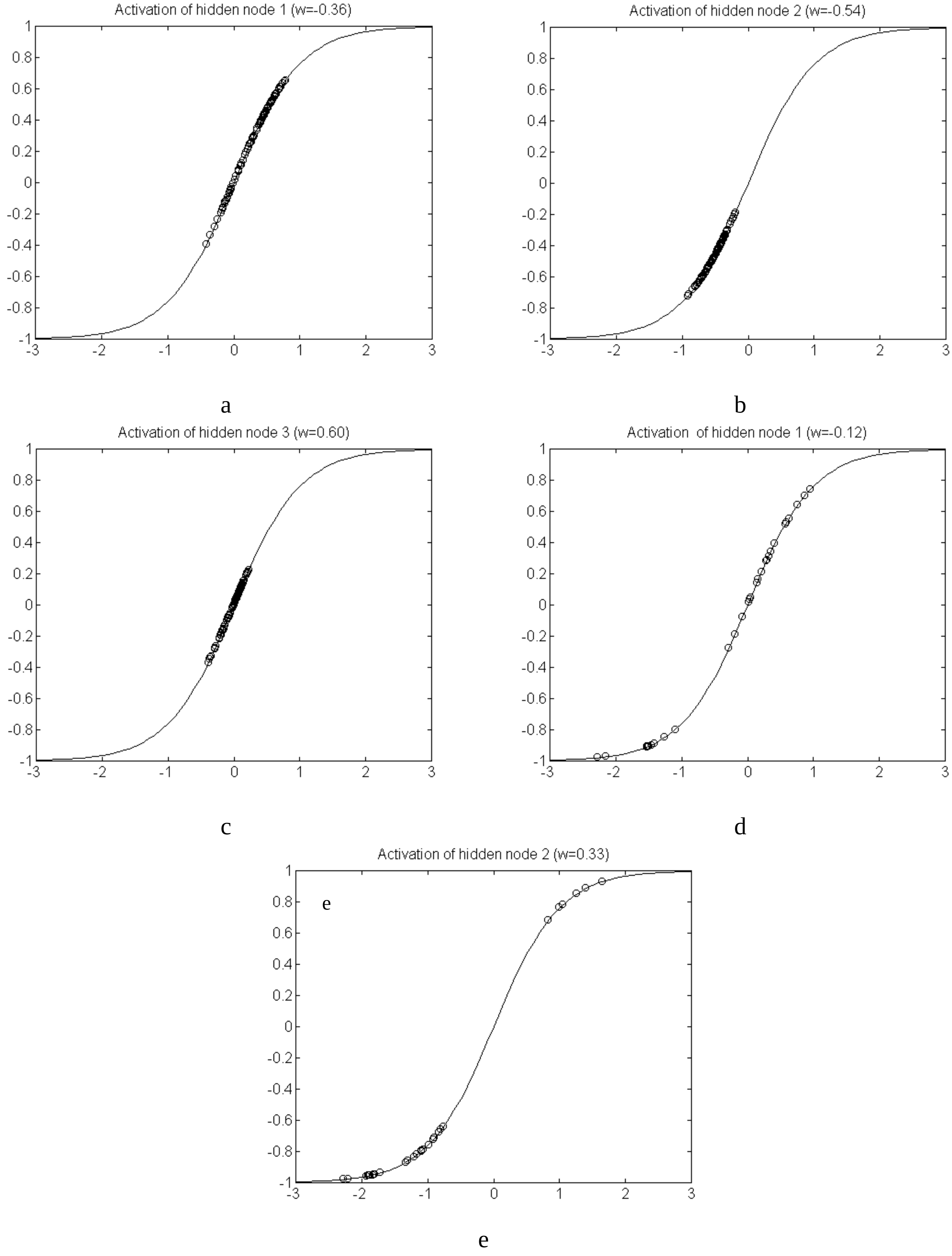


a

b

c

Figure 18



a

b

c

d

e

Figure 19

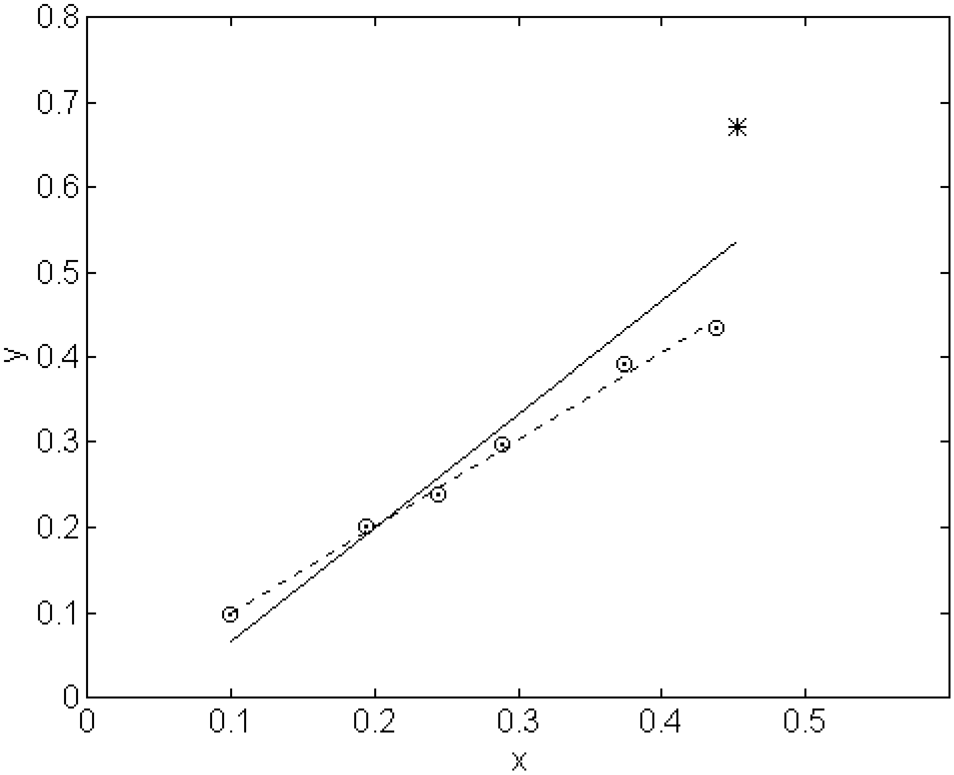


Figure 20