CHAPTER 3
Chemometrics and Multivariate Data Analysis

3.1 Introduction

“Chemometrics is a chemical discipline that uses mathematics, statistics and formal logic (a) to design or select optimal experimental procedures (b) to provide maximum relevant chemical information by analyzing chemical data and (c) to obtain knowledge about chemical systems.”


Chemometrics is used to solve problems involving large amounts of data. Within process analysis and monitoring, chemical analysis, sensory analysis, spectroscopy, molecular modeling and many other fields, large data-tables are obtained which require be analyzing and visualizing in order to be able to properly realize the problem. This chapter presents the Chemometrics and Multivariate data analysis methods used in this thesis. This will provide insight into the most popular methods for exploring data, building calibration and regression models and for building classification models. These methods include regression techniques like principal component regression (PCR), partial least square regression (PLSR), Support vector machine regression (SVM R) and locally weighted regression (LWR), and pattern recognition techniques like principal component analysis (PCA), partial least square discriminant analysis (PLS DA), hierarchical cluster analysis (HCA) and Kohonen self organizing map (Kohonen SOM)

The multivariate techniques help to explore the large data sets by decomposing the complex data into simpler structures; thus improving the interpretation and extraction of the available information. The application of Chemometrics, in recent decades, has evolved due to the development of new sophisticated instruments, algorithms and faster
computers. In this study, the multivariate data analysis is based on near infrared spectroscopic measurements that are obtained by fast, non-destructive instruments making the techniques suitable for application at-line or on-line.

The two main purposes for applying Chemometrics in this project are quantitative and qualitative. The purpose of quantitative analysis is to predict a certain parameter (e.g. percentage of adulteration) using regression models. In the qualitative analysis, the data are decomposed into a few factors enabling the study of patterns in the data, the trends, the outliers etc.

3.2 Spectral pre-processing

Near infrared spectra often contain out-of-range values, impossible data combinations, missing values etc. which may be caused by scatter effects, chemical interferences or instrument drift. Analyzing data without careful screening may complicate the data analysis and interpretation; can produce misleading results. Hence mathematical pre-processing may be employed. The pre-processing reduces or eliminates the impact of the non-relevant spectral information and often leads to simpler and more robust regression models by the better interpretation of the data.

In this project, various pre-processing tools and methods are used. These include filtering such as smoothing (Savitzky – Golay), first and second derivatives; normalization such as multiplicative scatter correction (MSC mean); scaling and centering such as autoscale, mean centre etc. The combinations of different pre-processing techniques are also employed.

3.2.1 Smoothing (Savitzky – Golay filtering)

The Savitzky – Golay smoothing filter is a type of filter first described in 1964 by Abraham Savitzky and Marcel J.E. Golay [Savitzky & Golay 1964]
The algorithm essentially performs a local polynomial regression (of degrees k) on a series of values (of at least k+1 points) to determine the smoothed value for the each point in the spectrum. These polynomials are used to smooth the data. The algorithm needs the selection of both the filter width or window size and the degree or order of polynomial. Smoothing will be high if the window size is large and the polynomial order is low. Indeed the Savitzky – Golay method can be seen as a generalization of averaging data, since averaging a sub range of data corresponds to using a Savitzky – Golay polynomial of order zero.

Smoothing is a low pass filter used for eliminating high frequency noise from samples. Smoothing assumes that variables in the adjacent columns of the data matrix are related to each other and contain identical information that can be averaged together to reduce noise without significant loss of the signal of interest.

3.2.2 Derivatives (Savitzky – Golay)

Derivatives of spectral data are a common method to remove or suppress constant background signals and to enhance the visual resolution and spectral features [Hruschka 1992]. Background signals and global baseline variations are low-frequency phenomena. Derivatives are interpreted as high-pass filters and frequency-dependent scaling. These are often used when smooth and broad (low frequency) features like baseline contain interferences and sharp and narrow (high frequency) features contain the signal of interest. Derivatives remove irrelevant baseline signal from samples by taking the derivative of the measured responses relative to the variable number or the relevant axis scale such as wavelength, wave number etc. This method is useful only when the variables are collinear (strongly related to each other) and the adjacent variables contain similar correlated signals.
The simplest form of derivative is a **first derivative** (point-difference), in which each point (variable) in a sample is subtracted from its immediate neighboring point (variable). This subtraction eliminates the signal which is the same between the two points and retains only the part of the signal which is different. Taking the first derivative on an entire sample effectively removes any flat baseline (offset) from the sample and suppresses the lower-frequency signals.

The **second derivative** is calculated by repeating the process as in first derivative which will further amplify higher-frequency features such as noise. The second derivative is the slope of the first derivative. It is more identical to the original spectrum in some ways, having peaks in more or less the same places, although they are inverted in direction. It is a measure of the curvature in the original spectrum at each point (variable) i.e. it improves the spectral solution by decreasing the peak width and increasing the number of peaks [Naes et al. 2002].

Since the derivatives tend to amplify higher frequencies, there is a chance for noise (high frequency signal) to appear. For this reason, the derivative transformation is performed along with Savitzky – Golay smoothing. This greatly improves the utility of derivatized data. As in smoothing, the Savitzky – Golay derivatization algorithm also requires the proper selection of window size, the polynomial order and the derivative order.

### 3.2.3 Multiplicative scatter correction (MSC)

The MSC technique [Martens et al.1983; Geladi et al.1985] originally developed for NIR data is a normalization that attempt the removal of amplification and offset (baseline) effects from the spectra. This correction is obtained by regression of a measured spectrum against an ideal or reference spectrum and then correcting the measured
spectrum using the slope of this fit. (Often the reference spectrum is the mean or average of the calibration data set.)

MSC (mean) performs the following procedure: Each measured spectrum is regressed in the data set-mean spectrum (ideal); the effect of scatter is responsible for variations along a straight line while deviations from the line correspond to the absorption by the sample components. The MSC model for each individual spectrum is

\[ X_{ik} = a_i + b_i \bar{x}_{k} + e_{ik} \quad (i = 1, \ldots, N; k = 1, \ldots, K) \]

where \( i \) is the sample number and \( k \) is the wavelength number. The constant \( a_i \) represents the ‘common shift’ and is related to proportional additive effect while \( b_i \) represents the ‘common amplification’ and is related to multiplicative effect for sample \( i \).

The mean

\[ \bar{x}_{k} = \frac{1}{N} \sum_{i=1}^{N} X_{ik} \]

is the average over samples at the \( k^{th} \) wavelength; \( e_{ik} \) represents the residuals and is related to the chemical information. The corrected spectrum is calculated using the equations:

\[ X_{ij} (MSC) = \left( \frac{x_{ij} - a_i}{b_i} \right) \quad (j = 1, \ldots, P) \]

The \( a_i \) and \( b_i \) coefficients are unknown and must be estimated individually for each sample using all or a subset of the \( k \) spectral measurements.

### 3.2.4 Variable centering and scaling

Centering [Eigenvector documentation 2013] is defined as

\[ X_c = X - \bar{X} \]
where $X_c$ represents the centered data; $\pi$ is a vector representing the reference point (often the mean of the data) for each variable. $I$ is a column vector of ones and $X$ is the data matrix.

Interpretation of loading and samples from models developed on centered data is done with respect to the reference point.

**Mean Center:** Mean centering calculates the mean of each column and subtracts this from the column. Also, after mean centering, each row of the mean centered data includes only how that row differs from the average sample in the original data matrix.

Scaling [Eigenvector documentation 2013] is performed by

$$X_s = XS$$

where $X_s$ is a matrix of the scaled data; $S$ is a diagonal matrix of the scaling factors; $X$ is the data matrix.

**Autoscale:** This preprocessing method makes use of mean centering followed by dividing each column (variable) by the standard deviation (SD) of the column. The diagonal of the matrix $S$ is equal to the inverse of the SD for each column. Thus each column of $X_s$ has a mean of zero and a SD of one.

In most cases, centering and scaling will be performed as the last methods if there are a number of preprocessing methods. i.e. Other preprocessing methods, if being used, are usually performed prior to a centering or scaling method.

### 3.3 Quantitative analysis – Regression analysis

Quantitative analysis refers to analysis in which the concentration or amount of an analyte may be estimated and expressed as a numerical value in proper units [IUPAC 1995]. Quantitative analysis requires the identification (qualification) of the analyte for which numerical estimates are given.
Quantitative (regression) analysis develop regression models which attempt to predict a quantity based on measurements of responses (X-block) and corresponding quantities (Y-block) on known samples. The Y-block may contain a physical quantity which is directly related to the measurements in the X-block, or it may be a value which is indirectly related to the measured X-block values. In the latter case, the resulting model is considered an “inferential” model.

A regression analysis typically consists of two steps:-

1. Calibration Step

   The characteristics of a method are investigated and a robust model is created that can explain the behavior present in the data. The model is in the form

   \[ Y = f(X) \]

   where \( Y \) is the dependent variable and \( X \) is the independent variable.

   The parameters in the model are termed as regression coefficients or sensitivities.

2. Prediction or Test Step

   The independent variables (X) are collected for a number of samples and are used to predict the dependent variables (Y).

   To make the calculations easier, the data is often pre-processed. The first method to be considered is the linear regression method such as principal component regression (PCR) and partial least square regression (PLSR).

3.3.1 Linear regression methods

   PCR and PLSR are two linear (multivariate) regression methods often applied for relating multivariate data X (spectra or wavelengths) to a reference parameter Y (concentration of a certain analyte) [Geladi & Kowalski 1986]. The overall purpose is to predict the reference value in future samples and to interpret the relationship between the two data sets. The multivariate regression models can be explained as
$Y = Xb + E$

where $b$ is the regression coefficients vector obtained by the calibration model and $E$ denotes the residuals containing model errors.

### 3.3.1.1 Principal component regression (PCR):

If $T$ is a matrix of ‘SCORES’ (Samples), $X$ is a DATA matrix and $P$ is a matrix of LOADINGS (variables) [Geladi & Kowalski 1986], then

$$T = XP (= TP^TP)$$

The $T$- matrix containing the score vectors are substituted into the equation

$Y = Xb + E$, we have

$$Y = Tb + E$$

The solution to the above equation (i.e. estimating $b$) is

$$b = (T^T)^{-1}T^TY$$

(The superscript $T$ denotes the ‘transpose’ of the matrix).

The Variables in the $X$ matrix are replaced by new ones (i.e. correlated data into new variables $T$) with better properties (orthogonality i.e. mutually perpendicular uncorrelated linear components) and which also span the multi-dimensionality of $X$. With PCR, the inversion of the $T^T$ $T$ matrix should cause no problem since the scores are mutually orthogonal [Geladi & Kowalski 1986]. In others words, the orthogonality of the scores or components and the exclusion of scores containing less information (with small eigenvalues) remove the problem of collinearity (high correlation between the variables), thus making PCR one of the preferred methods for handling spectral data sets.

Though PCR solves the inverse matrix problem (known as collinearity) and has the ability to eliminate the lesser principal components (PCs) so as to reduce error in the model, the model still has a problem: if the relevant underlying effects are small in comparison with some irrelevant ones, then they may not appear among the first few
principal components. So, a component selection problem remains – the first few principal components cannot just be included, as these may serve to degrade the performance of the model. Instead, all components have to be extracted and determine whether adding each one of these improves the model. This is a complex problem.

Partial least squares regression (PLSR) solves the problem. The algorithm used examines both X and Y data and extracts components (factors) which are directly relevant to both set of variables. The components are extracted in decreasing order of relevance so, to form a model now what to do is extract the correct number of factors or components to model-relevant underlying effects.

3.3.1.2 Partial least square regression (PLSR): PLSR [Martens & Naes 1989] is a multivariate regression method which is commonly used in quantitative spectroscopy to correlate spectroscopic data (independent variable X) with related chemical/physical data/dependent variables Y). This method is employed to predict the dependent variables Y from a set of independent variables X when the number of independent variables (wavelengths in NIR spectroscopy) is large compared to the number of samples. In short, PLS estimates the principal components or PLS factors as projections of the original X variables to relate X and through regression [Wold et al.2001]. PLSR is based on NIPALS algorithm (see details later in PCA).

The linear regression model is defined as

\[ Y = Xb + E \]

where b is the regression coefficients that are determined during calibration and E is the residual matrix.

b is calculated using the equation

\[ b = W (P^T W)^{-1} Q^T \]
A detailed description on PLS regression modeling using the NIPALS algorithm is given in literature [Esbensen 2000].

Two set of X loading that are calculated in PLS calibration are the loadings P and the loading weights or PLS weights W. The P-loadings explain the relation between the X data matrix and its scores T; the loading weights W relate the matrices Y and X through regression. Also, the set of Y-loadings Q containing the regression coefficients connect the Y-variables to T scores. These are also used to estimate b.

Unlike PCR, PLSR utilizes the Y-variance to decompose the X data matrix resulting in an optimal regression for prediction purposes and via fewer principal components. In PLSR, the X and Y spaces are modeled interdependent [Geladi & Kowalski 1986; Esbensen et al. 1994]

PLSR consists of outer relations (X and Y-blocks individually) and an inner relation linking both blocks:

Outer relation for the X-block:

\[
X = \sum t_h P_h^T + E = \sum t_h P_h^T + E
\]

\(t_h\) are the scores vector and \(P_h^T\) are the loadings vector for the X-block.

Outer relation for the Y-block:

\[
Y = \sum U q_h^T + F = \sum U q_h^T + F
\]

\(U_h\) are the scores for the Y block and \(q_h^T\) are the loadings for the Y-block.

The U-score vectors derived from the Y- matrixes are the starting points for the t-score vectors in decompositions of the X - matrix [Esbensen et al. 1994].

The inner relation is evaluated by plotting Y-block scores (u) against X-block scores (t) for each component.
3.3.2 Non-linear regression methods.

Generally the NIR spectral data contain both linear and nonlinear information. It is important to mention that situations exist where the non-linearities, even when there is multivariate linearity, are important enough to favour a non-linear approach [Naes et al. 1990]. In certain situations, the Y-variable is not only a linear function of both X-variables simultaneously, but also a non-linear function of the first principal component of the X-variables. In such cases, a good non-linear model based on a small number of principal components can give a more stable and reliable calibration equation than a linear function of a larger number of principal components incorporated in order to account for non-linearities. Hence, even non-linearities of the simple type may be better handled by non-linear regression methods than the regular linear PCR/PLSR approach.

The non-linear regression method namely locally weighted regression (LWR) and support vector machine regression (SVM R) are commonly employed to extract non-linear information from spectroscopic data.

3.3.2.1 Locally weighted regression (LWR): LWR [Naes et al.1990] calculates a single locally weighted regression models using the given number of principal components to predict a dependent variable Y. i.e. adulteration percent from a set of independent variables X i.e. spectral data. LWR model is useful for performing predictions when the dependent variable, Y has a non-linear relationship with the measured independent variables, X.

In many practical applications in chemistry, the calibration data matrix X is often highly collinear. Hence, the raw X data must first be compressed (dimension-reduction) before LWR can be applied. The steps for performing the algorithm for LWR are as given below:
1) The number of principal components (A) and the number of samples (C) to be used in each local calibration set is decided.

2) A dimension-reduction technique (PCA) on the full calibration set is performed. As usual, let $T$ be the score matrix and $P$ be the loading matrix corresponding to the first $A$ components.

3) For each predictions sample, the scores $t^T = x^T p$ is computed where $x$ is the predication spectrum centered by the same constant as used to centre the calibration matrix $X$.

4) The calibration samples $C$ that are close to the prediction sample are found out. Closeness is determined by using distance measure in the $A$-dimensional principal component score space. Distances between the $A$-dimensional score vector $t$ and all the rows of $T = XP$ are calculated and used to find the $C$ samples that are closest.

5) The least squares criterion is employed to estimate the unknown parameters in the linear regression model (i.e. global PCR/PCR/PLSR).

$$Y_C = 1 q_0 + T_C q$$

where $Y_C$ is a vector of Y-values for the $C$ local samples, $1$ is a vector of ones, $q_0$ is the intercept term, $T_C$ denotes the score matrix for the $C$ samples closest to the prediction sample and $q$ is the vector of regression coefficients for the $A$ scores.

6) The Y-value for the prediction sample is predicted using the estimated regression coefficients,

$$Y = q_0 + t^T q$$

### 3.3.2.2 Support vector machine regression (SVM R): “The support vector method can be applied to the case of regression, maintaining all the main features that characterize the
maximal margin algorithm: a non-linear learning is learned by a linear learning machine in a kernel-induced feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space.”

— Cristianini and Shawe - Taylor (2000)

In SVM, the basic idea is to map the data $X$ into a high dimensional feature space $F$ via a non-linear mapping and to do linear regression in this space [Vapnik 1995].

Using mathematical notation, the linear model (in the feature space) $f(x, \omega)$ is defined as

$$f(x, \omega) = \sum_{j=1}^{m} \omega_j g_j(x) + b$$

where $g_j(x)$, $j = 1, \ldots, m$ is a set of non-linear, transformations and $b$ is the ‘bias’ term. Often the data are assumed to be zero mean (achieved by preprocessing), so the bias term is omitted.

An overview of the basic ideas underlying support vector machines (SVM) for regression and function estimation has been given in literature [Smola & Scholkopf 1998].

SVM regression utilizes a new type of loss function $L(y, f(x, \omega))$ (called $\epsilon$-insensitive loss function) proposed by Vapnik. This function measures the quality of estimation.

$$L_\epsilon(y, f(x, \omega)) = \begin{cases} 0 & \text{if } |y - f(x, \omega)| \leq \epsilon \\ |y - f(x, \omega)| - \epsilon & \text{Otherwise} \end{cases}$$

The empirical risk is:

$$R_{emp}(\omega) = \frac{1}{n} \sum_{i=1}^{n} L_\epsilon(y_i, f(x_i, \omega))$$
SVM regression performs linear regression in the high-dimension feature space using the \( \varepsilon \)-insensitive loss and, at the same time, tries to reduce model complexity by minimizing \( \| \omega \|^2 \). This can be explained by introducing (non-negative) slack variables \( \xi_i \), \( \xi^*_i \), \( i = 1, \ldots, n \), to measure the deviation of training samples outside the \( \varepsilon \)-insensitive loss. Thus SVM regression is expressed as minimization of the following functional.

\[
\min \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{n} (\xi_i + \xi^*_i)
\]

\[
\text{s.t. } \begin{cases} 
y_i - f(x_i, \omega) \leq \varepsilon + \xi_i^* \\
f(x_i, \omega) - y_i \leq \varepsilon + \xi_i \\
\xi_i, \xi^*_i \geq 0, \quad i = 1, \ldots, n
\end{cases}
\]

This optimization problem can be transformed into the dual problem and its solution is given by

\[
f(x) = \sum_{i=1}^{n_{sv}} (\alpha_i - \alpha^*_i) K(x_i, x)
\]

\[
\text{s.t. } 0 \leq \alpha_i, \alpha_i^* \leq C
\]

\[
0 \leq \alpha_i \leq \mathcal{U}
\]

where \( n_{sv} \) is the number of support vectors (svs) and the kernel function

\[
K(x_i, x) = \sum_{j=1}^{m} E_j(x_i) E_j(x)
\]

It is well known that SVM generalization performance (estimation accuracy) depends on a good setting of the hyper parameters \( C \) (cost), \( \varepsilon \) (epsilon) and kernel parameter \( \gamma \) (gamma). Existing software implementations of SVM regression usually treat SVM parameters as user-defined inputs. Proper selection of a particular kernel type (in this study, radial basis function, RBF, is adopted) and kernel function parameters is usually based on application-domain knowledge and also should reflect distribution of input (X) values of the training data.
Parameter $C$ (penalty factor) determines the trade off between the model complexity (flatness) and the degree to which deviations greater than $\varepsilon$ are tolerated in optimization formulation. If $C$ is too large, then the aim is to minimize the empirical risk only, without considering model complexity part in the optimization formulation.

Parameter $\varepsilon$ controls the width of the $\varepsilon$–insensitive zone used to fit the training data. The $\varepsilon$ - values can affect the number of support vectors used to construct the regression function. The larger the $\varepsilon$, the fewer the support vectors selected. Bigger $\varepsilon$ - values give rise to more ‘flat’ estimates. Hence both $C$ and $\varepsilon$ - values affect model complexity.

3.3.3 Calibration and Validation

Calibration involves connecting one (or more) sets of variables together. Usually, one set (block) is a series of physical measurements such as some spectra or molecular descriptors; the other set contains one or more parameters such as the concentrations of a number of components or a biological activity. We can predict the concentration of a compound in a mixture spectrum or the properties of a material from its known structural parameters by means of calibration techniques. In its simplest form, calibration is a form of regression.

Unquestionably one of the most important aspects of all calibration methods is model validation [Brereton 2003]. There is a large literature on how to decide what model to adapt, which requires an appreciation of model validation, experiment design and how to measure errors. Most methods aim to guide the investigator as to how many principal components to retain.

In multivariate data analysis, the purpose of validation is to ensure that a reliable model is obtained i.e. the model is valid for future new samples and is not a more expression of trends in the present samples only. In practice validation is utilized to check
whether the proper model is applied to determine the optimum number of principal components (PCs) to detect the outliers and to obtain reliable estimates of model parameters such as errors.

3.3.3.1 Determination of optimum number of principal components

The determination of the number of significant components is an important step in regression. One method is to place all the available data in one data set and perform cross-validation, whereas another commonly used method is the Test-set validation. In this project, cross-validation or test-set validation is used depending on the data set.

**Cross-validation:** Cross-validation [Wold 1978] is used when the number of samples in a data set is limited and all samples, therefore, are needed for both calibration and validation. For calibration, the simplest type is leave-one-out (LOO-) or full cross-validation. In this technique, one sample or a group of samples (segment) is left out of the calibration data set; this sample or segment of sample is used for validating the model built on the remaining samples in the calibration set. After this, another sample or a segment of samples is left out and is used for validating the model based on all other samples in the calibration set. This procedure is continued until all samples or segments of samples have been left out exactly one at a time. The average of the validation results is calculated. The final model is based on the performance of all models. The measure of model performance is usually given by the prediction errors. The prediction is estimated as the root mean square error of calibration (RMSEC) or root mean square error of cross-validation (RMSECV) which is calculated by comparing the predicted value with the reference values as:

\[
RMSECV = \sqrt{\frac{\sum_{i=1}^{n} (y_{\text{pred}} - y_{\text{ref}})^2}{n}}
\]
where $y_{\text{pred}}$ is the predicted value, $y_{\text{ref}}$ is the reference (measure or true) value and $n$ is the total number of samples. The lower the RMSECV/RMSEC value, the better will be the model performance and hence calibration.

The minimum number of significant components that contain maximum information and minimum model errors can also be observed by plotting the RMSECV/RMSEC values against the number of principal components (PCs) or latent variables (LVs). The lowest RMSECV/RMSEC value (at which the curve flattens) indicates the optimum number of significant components (often called factors) to be used in the model.

**Test-set Validation:** Test-set validation is employed when the data set is sufficiently large to be divided into two subsets: a calibration set that is used for calibration of the model and a validation set which is used to test the prediction ability i.e. to estimate the prediction error. Since different samples are used for calibration and validation, test-set validation is a better test of the model than cross-validation.

The validity of the model is tested by applying the built-in model to these new independent samples (test set) and evaluating the explained variation, spectral residuals scores, prediction error etc. The predication error, namely root mean square error of prediction (RMSEP), is typically used to evaluate the results. It (defined to be equal to RMSECV) is expressed as

$$\text{RMSEP} = \sqrt{\frac{\sum_{i=1}^{n} (y_{\text{pred}}-y_{\text{ref}})^2}{n}}$$

where the symbols have the usual meaning. The minimum RMSEP value is an indication of good prediction ability and better, reliable performance of the model. The validation results of the model can be displayed in ‘predicted versus measured’ plots where
the y-values as predicted by the model are plotted against the originally measured y-values for the validation samples.

3.3.3.2 Building of robust model using preprocessed data:

Calibration and quantitative analysis are performed on the preprocessed spectroscopic data (X) using regression techniques. The X-matrix is the product of two smaller matrices called score and loading matrices.

The score matrix (T) contains information about the objects or samples. Each object is described in terms of its projections on to the PCs instead of the original variables.

The loading matrix (P) contains information about the variables; it is composed of a few vectors or PCs that are obtained as linear combinations of the original X-variables.

$$X = T \mathbf{P}^T + E$$

The information not included in these matrices remains as “unexplained X-variance” in a residual or ‘error’ matrix E.

The principal components (PCs) among many others have two interesting properties.

- They are extracted according to the decreasing order of importance. The first PC always contains more information than the second, the second more than the third and so on…

- They are orthogonal to each other. Hence there is absolutely no correlation (i.e. no problem of collinearity) between the information contained in different PCs.

In multivariate data analysis, the user can decide how many PCs should be extracted (i.e. the number of significant components or dimensionality of the model). Each new PC extracted or added increases further the amount of information (percentage of variance) explained by the model. However, usually the first four or five PCs explain more
than 90% (confidence limit) of the X-variance. There is not a simple or unique criterion to determine how many PCs to extract. From a theoretical point of view, it is possible to use ‘cross-validation’ techniques to decide the minimum number of PCs to include. From a practical point of view, it doesn’t matter to extract more PCs if the user has no way to interpret the results.

The root mean square error of cross-validation (RMSECV) values are calculated for each factor using ‘leave-one-out’ cross validation (in this project) to determine the optimal number of factors to be included in the calibration model. Thus the model is built by cross-validation method during calibration development. The optimum number of principal factors is selected employing the cancellation of one sample or a segment of samples at a time (leave-one-out cross-validation). This can be done by plotting the number of PCs against the RMSECV values (low value provides good results) and from this, the optimum number of factors is selected [Naes et al. 2002; Martens & Naes 1989] for the robust regression model.

The best model thus selected is used to determine the concentration (the parameter of interest) of the samples in the independent prediction/test set. The relative performance of the established model is accessed by RMSEC (root mean square error of calibration), RMSECV and R² (multiple coefficient of determination or regression coefficient [Wang et al. 2006]).

The predictive ability of the model is evaluated from RMSEP (root mean square error of prediction [Divya & Mishra 2007]. The lower the RMSEP value, the higher the degree of accuracy of the prediction results provided by the calibration model. The correlation coefficient R² is the intensity measure of the correlation between the ‘measured’ values and values ‘predicted’ by the model. This may range from 0 to 1. The closer the value to +1, the higher is the correlation between the data. Also, a plot of the
‘measured’ values of concentration against the ‘predicted’ values of concentrations reveals the accountability of the model.

Another way to extract information from the data, graphically, is by plotting the ‘score’ and ‘loading’ matrices obtained.

**2D and 3D scores Plot:** These plots represent the relative positions of the objects or samples in the space (two-dimensional or three-dimensional) of the principal components. Cluster of objects and single object with particular behavior (outlier) can be easily identified. Moreover, the objects’ position in the plots may serve to interpret the PCs. The first PC explains the maximum amount of variation and therefore (when there are clusters of objects) to distinguish among them. In this context, the PC can be interpreted as a compendium of the distinctive features of the objects in these clusters.

**2D and 3D loading Plot:** These plots represent the original variables in the space of the principal components. It should be remembered that the PCs are obtained as linear combinations of the original X-variables. The loading of a single variable indicates how much this variable participates in defining the PC. The squares of the loadings indicate their percentage in the PC. Variables contributing very little to the PCs have small loading values and are plotted around the centre of the plot. On the other hand, variables contributing to the most are plotted around the boarders of the plot.

Loading plot allow to test the “homogeneity” of the contributions of the X-variable to the model.

It should be noted that “Quantitative analysis” is often used in comparison (or contrast) with “Qualitative analysis” which seeks information about the identity or form of substances present. For example, an unknown solid sample is given to a chemist. He/She will use “qualitative” methods (e.g. IR spectroscopy, NMR etc.) to identify the compounds
present and then quantitative techniques to determine the quantity of each compound in the sample.

Qualitative analysis involves determining the compositions of an unknown compound or mixture. It also involves identifying what materials are present in a sample (quality). This analysis can also analyze a mixture to determine the precise percentage composition of the sample in terms of elements, radicals and compounds.

3.4 Qualitative analysis and Classification – Pattern Recognition Techniques.

One of the first and the most publicized success stories in chemometrics is pattern recognition. Much chemistry involves using data to determine patterns. There are several groups of methods for chemical pattern recognition.

Pattern recognition assigns some sort of output value (or label) to a given input value (or instance) according to some specific algorithm. An example of pattern recognition is classification, which attempt to assign each input value to one of a given set of classes (for example, determine whether a given email is spam or non-spam). Pattern recognition is generally categorized according to the type of learning procedure used to generate the output value: supervised and unsupervised learning.

Supervised learning assumes that a training set (a set of training data) has been provided, consisting of a set of instances (the price of input data for which an output value is generated) that have been properly labeled by hand with the correct output. A learning procedure then, generates a ‘model’ that attempts to satisfy two objectives (sometimes conflicting): perform as well as possible on the training data, and generalize as well as possible to new data.

Unsupervised learning assumes training data that has not been hand-labeled, and attempts to find inherent pattern in the data that can be utilized to determine the correct output value for new data inputs.
3.4.1 Supervised pattern recognition

There are a large number of techniques for supervised pattern recognition (mostly aimed at classification). These are used to assign samples to a number of groups (or classes). These techniques include principal component analysis (PCA), discriminant analyses like partial least square discriminant analysis (PLS DA) etc.

3.4.1.1 Principal component analysis (PCA): Principal component analysis, in this project, is used to reduce the dimension of the data into a few numbers of significant components, to examine the possible clustering of samples and, to observe the presence of outliers if any.

PCA [Pearson 1991; Hotelling 1933; Wold et al. 1987] is the most fundamental chemometric algorithm widely used to transform a set of highly correlated data (multivariate data) into a new set of uncorrelated (orthogonal) components called principal components (PCs) [Naes et al.2002] using a least square approach. Each PC (new variable) is a linear combination of the original measurements. The PCs lie along the direction of maximum variance in the data set. This projection of data is continued by composing additional (orthogonal) principal components until all significant structures of the data are described. A set of n spectra (e.g. near infrared spectra of n samples) can be expressed as a data matrix X (n × p) containing n values of absorbance at each of the p wavelengths. The data matrix is the product of two low-dimensional matrices namely score matrix (T) and transposed loading matrix (P^T) [Otto 1997].

$$X = TP^T + E$$

E is the residual matrix that contains the unsystematic variations or noises. The scores (often referred to as samples in rows and PCs in columns) are the new values of the spectra in the co-ordinate system defined by PCs and the loadings or eigenvectors (often
referred to as variables in rows and PCs in columns) are the link between the wavelengths of the X matrix and the principal components.

An important feature of PCA is the graphic interface i.e. plots of the scores and loadings. Two-dimensional/three-dimensional scatter plots of the scores depict the covariance between samples, providing a data overview. Clusters of the objects as well as outliers and hence the pattern are easily identified in the score-plots. These help the experimenter or analyst to explore the expected and unexpected trends in the data. The loadings-plot explains the importance of the original variables for each PC. This line plot helps to deduce the qualitative difference that give rise to the cluster or trends in the data.

3.4.1.2 Partial least square discriminant analysis (PLS DA): PLS DA is a modified form of PLS regression [Stahele & Wold 1987] applied in classification and discrimination problems. This supervised analysis build one model covering many classes [Eriksson et al. 2004] i.e. This is multivariate projection technique that makes a relation (by building a model) between the dependent Y-variables and independent X-variables. The Y-block variables consists of a set of binary indicator variables (one for each class) representing the class membership.

A ‘dummy’ Y-matrix is constructed from the grouping variables consisting of zeros and ones. The matrix consists of as many columns as there are classes (to be modeled) and an observation sample has the value ‘1’ if it is ‘in-class’ and ‘0’ if it is ‘not in-class’. The principle of PLS is to determine the factor in the input matrix (X) that contain maximum relevant variations in the input variables and at the same time have maximal correlation with the target values (Y) giving less weight to the irrelevant variations or noise. It is well-known that PLS maximizes the covariance between the X and Y-matrices. This discriminant analysis approach projects the variables and factors in the multivariate space that discriminate the established classes in the calibration data set. The optimal numbers of
latent variables (principal components) are estimated by using cross validation or test set validation. The components extracted are not the principal components having maximal variance, but they are components (orthogonal to each other) having maximal correlation with the dependent (y) variables.

The calculation of ‘scores’ for the data does not simply involve a projection on to the loadings because there exists a unique relationship between weights and loadings in a PLS model

\[ T_{\text{new}} = X_{\text{new}} P \]

Given new data \( X_{\text{new}} \), the scores for these new samples are obtained using.

\[ T_{\text{new}} = X_{\text{new}} W (P^T W)^{-1} \]

where \( W \) is the weight matrix of the model and \( P \) is the loading matrix.

The other aspects of the PLS DA model such as predictions for all of the samples including the unknowns, for each of the classes may be explored graphically (for example, Y predicted versus sample plot). The overall sensitivity (number of samples predicted as ‘in the class’ divided by number actually in the class) and specificity (number of samples predicted as ‘not in the class’ divided by actual number not in the class) can be accessed by calibration and cross-validation results. Sensitivity and specificity can also be viewed in a graphical form which is referred to as a Receiver Operating Characteristics curve (ROC curve).

A detailed treatment of the PLS DA technique is given in literature [Barker & Rayens 2003]. In addition, an example of its use is given by literature [Musumarra 2004].

3.4.2 Unsupervised pattern recognition (Cluster analysis)

Often data sets comprise of samples that form several different groups or “classes”. The differences in classes are due to variations in sample preparation, variations in chemical composition (aromatic, aliphatic, carbonyl etc.) or variations in process state.
Classification of new unknown samples can be accomplished by using unsupervised methods. Methods that attempt to identify groups or classes without using pre-established class memberships are known as unsupervised pattern recognition (cluster analysis).

Each sample, in cluster analysis, is initially supposed to be a lone cluster. Then the distances between the samples are calculated using one of the many ways like Euclidean distance, Mahalanobis distance etc. and the samples with the smallest distance are obtained and linked together. This procedure is repeated and those samples with the next closest distance are found and linked. The final results are displayed as a connection dendrogram. The cluster can be obtained from the dendrogram by putting a vertical line over it and moving it on the distance scale. For a given distance, the number of cluster is the number of lines in the dendrogram that cut the vertical line.

In Euclidean distance, the distance $x_{ij}$ between samples $x_i$ and $x_j$ is given as:

$$x_{ij} = \sqrt{(x_i - x_j)(x_i - x_j)^T} \quad \text{(in matrix format)}$$

The distance is expressed as the square root of the sum of squared differences between the samples i.e. $\sqrt{\sum (x_i - x_j)^2}$. Prior to calculating the distance, a data preprocessing is often applied.

In Mahalanobis distance (based on principal component scores) the distance is estimated in matrix terms as:

$$x_{ij} = \sqrt{(t_i - t_j)C^{-1}(t_i - t_j)^T}$$

where $C$ is the variance – covariance matrix of the variable in which the elements (eigenvalues) represent the covariance between any two variables. $t_i$ and $t_j$ are the scores of the samples $x_i$ and $x_j$. A Mahalanobis distance accounts for the fact that variation is much
larger in some directions than in others. This means that distance on some directions is more significant than distance in other directions.

There are also several other similar distance measures (such as Manhattan distance) in many literatures.

The most common approach to link the sample or objects is the agglomerative clustering. As in the case of distance measures, there are several approaches. The main procedure is to recalculate the numerical similarity values between the new group and the remaining samples. The main ways of doing this are nearest neighbor (single linkage) furthest neighbor (complete linking) and average linkage.

3.4.2.1 Hierarchical cluster analysis: In this project, hierarchical clustering [Massart et al. 1997] is used for grouping of objects. It is based on the idea of objects being more related to nearby objects than to objects farther away. These algorithms provide an extensive hierarchy of clusters that merge with each other at certain distances; they do not provide a single partitioning of the data set. The distance matrix is calculated using Mahalanobis distance. From the distance matrices, the dendrogram are created using the K-nearest neighbor or KNN.

3.4.2.2 Kohonen’s self organizing map (Kohonen SOM): A self organizing map (SOM) is an artificial neural network analytical tool that uses unsupervised learning to map objects from a high-dimensional data input space to a low-dimensional (usually two-dimensional) output grid (map). The learning algorithms are designed to present the formation of clusters of similar objects (i.e. samples) at adjacent positions in the grid. SOM was invented by Professor Teuvo Kohonen in the 1980 and is also called Kohonen map [Kohonen 1981].

SOMs run in two phases: Training and mapping. In the training phase, map is built; network is organized using a competitive process (called vector quantization); it is trained
using large number of inputs. In the mapping phase, new vectors are quickly given a location on the coverage map; easily classifying or categorizing the new data.

A SOM consists of elements or components called neurons (notes). Each neuron is associated with a weight vector (a weight for each variable in the data set) of the same dimension as the input data set and a location in the map space. Usually the neurons are arranged in a rectangular or hexagonal grid. The basic process in SOM algorithm is given below:

- Initialize each neuron’s weight.
- Choose an input vector randomly from the training data and present it the SOM.
- Traverse each neuron in the map. For this, use Euclidean distance formula $(d_{ij} = \| x_k - w_{ij} ||)$ to find the similarity between the input vector and the weight vector of the neuron in the map (b) Track the winning neuron $W_{winner} (w_{ij}; d_{ij} = \min (d_{\text{min}})$ that produces the smallest distance (the winning neuron is called best matching unit, BMU)
- Update each neuron in the neighborhood of BMU according to the rule
  $$W_{ij} = W_{ij} + \alpha N_c (W_{\text{winner}}, w_{ij}) \| x_k - w_{ij} ||$$
  where $x_k$ be the n-dimensional training patterns; $w_{ij}$ be the neuron in position (i, j); $0 \leq \alpha \leq 1$ be the learning rate and $N_c$ be the neighborhood function, $N_c$ assumes values in (0,1) and is high for neurons that are close in the output map and small (or 0) for neurons far away.
- Repeat the process for enough iterations (cycles) for convergence. (To guarantee convergence and stability of the map, the learning rate and neighborhood radius are decreased in each iteration, thus converging to zero).

It is also common to use the U-matrix (Unified Distance Matrix). The U-matrix [Ultsch 2003] provides an important information in the form of distances between neuron’s of the SOM grid. In this method, a matrix of distances between the weight
vectors of neighboring neurons of the two-dimensional SOM is computed. The U-matrix value of a given neuron is the average distance between that neuron and its closest neighbors. The U-matrix provides a visual inspection of the trained SOM.

3.5 Conclusion

The fast and ongoing evolution of near infrared spectroscopic applications in research and industry would have been difficult without the simultaneous progress of chemometric evaluation/analytical tools. This chapter is written to introduce the chemometric analytical techniques used in this project and also to show the recent trends in multi-variate data analysis. The important part of chemometrics is the multivariate data analysis which is essential for quantitative as well as qualitative assays based on near infrared spectroscopy. The main advantage of chemometric data analysis is the possibility of projecting multivariate data into few dimensions in a graphical interface. i.e. chemometrics can handle large datasets easily and deal efficiently with real-world multivariate data, taking advantage over the previously-feared collinearity of spectroscopic data.