KATHOLIEKE
UNIVERSITEIT
LEUVEN

DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

RESEARCH REPORT 0324

INFLUENCE PROPERTIES OF PARTIAL LEAST SQUARES REGRESSION

by
S. SERNEELS
C. CROUX
P. J. VAN ESPEN

D/2003/2376/24

Influence properties of partial least squares regression

Sven Serneels[1] Christophe Croux[2]
Pierre J. Van Espen[1*]
[1]Department of Chemistry, University of Antwerp, Belgium
[2] Department of Applied Economics, K.U. Leuven, Belgium

Abstract

In this paper, we compute the influence function for partial least squares regression. Thereunto, we design two alternative algorithms, according to the PLS algorithm used. One algorithm for the computation of the influence function is based on the Helland PLS algorithm, whilst the other is compatible with SIMPLS.
The calculation of the influence function leads to new influence diagnostic plots for PLS. An alternative to the well known Cook distance plot is proposed, as well as a variant which is sample specific. Moreover, a novel estimate of prediction variance is deduced. The validity of the latter is corroborated by dint of a Monte Carlo simulation.
KEYWORDS: partial least squares regression, influence function, variance estimation, diagnostic plot.

*Correspondence to P. Van Espen, Departement Scheikunde, Universiteit Antwerpen, Universiteitsplein 1, 2610 Antwerpen (Belgium) E-mail: piet.vanespen@ua.ac.be. . Tel.: +328202358; Fax.: +3238202376

1 Introduction

Partial Least Squares (PLS) regression [1] is one of the most widely used chemometrical tools to estimate concentrations from measured spectra. As it is mostly a chemometrical tool, it has hitherto only been granted little attention in the statistical literature. A consequence thereof is that some properties of partial least squares regression have never been investigated. One of these properties is the *influence function* [2], which is of widespread use in the literature on robust and mathematical statistics. Indeed, one can define an estimator to be robust whenever its influence function is bounded, but also for non-robust, so-called *classical* estimators (such as PLS), the influence function has major applicability.
In this paper, the influence function for partial least squares regression is computed, and used as a diagnostic tool to assess the influence of individual calibration samples on prediction. In PLS a calibration stage is required in which a regression vector is being estimated from a calibration matrix, consisting e.g. of spectra of "standards" with known concentrations. Once this stage is completed, the responses of samples can be estimated by means of a single (matrix) multiplication. It is the influence on these predicted responses which will be assessed. Diagnostic plots will be proposed.
In contrast to the ease of which in PLS responses (e.g. concentrations) are estimated, the uncertainties thereof are very hard to assess, and often unknown. Faber *et al.* [3] correctly point out that the most common technique to assess this uncertainty consists of using the regression vector to estimate the responses for a set of samples of which the true response is known, but which have not been used for calibration. Usually, this set of samples is referred to as the *validation set*. Estimated and true responses are then used to compute a so-called *root mean squared error of prediction (RMSEP)* [4], which is then supposed to be a measure of the uncertainty of *all* future predictions made by this model.
The RMSEP is an *average* measure of uncertainty. Methods which allow the estimation of a sample specific prediction error have been proposed [5, 6, 7, 8]. All of these approaches are based on a local linearization of the PLS estimator. However, a variance estimate can also be computed from the influence function. This approach has got many advantages over the existing techniques. Firstly, the variance estimates based on the influence function of the PLS estimator are independent on any model assumption. Moreover, the estimate of variance derived from the influence function requires very little computational effort, contrary to the aforementioned variance estimation techniques which have never become popular due to computational difficulties. Computation of the influence function leads at once to both

variance estimation and diagnosis of influence, a combination heretofore not reported.
In Section 2 we introduce the reader to the notation used throughout this article.
In Section 3 we provide a short introduction to partial least squares regression.
In Section 4 we introduce the reader to the population version of PLS, since this insight is needed for a correct computation of the influence function.
In Section 5 we introduce the reader to the basic concept of the influence function, as well as the results that can be deduced thereof.
In Section 6 we propose algorithms which allow efficient calculation of the influence function, derived from different PLS algorithms.
In Section 7 this leads to a sample specific prediction interval in PLS, as well as to novel diagnostic plots.

2 Notation and definitions

Before we can give an introduction to partial least squares regression, we first need to define the notation used. The calibration matrix X is a matrix of size $n \times p$ in which the rows are n spectra of standard samples, measured at p channels. Matrices will always be denoted in upper case letters. The corresponding n concentrations of these standard samples constitute the response vector $\mathbf{y}$. Vectors will always be denoted by bold-face lower case letters. When we refer to individual columns of matrices, we shall denote these vectors using the corresponding letter. Throughout this work, we will assume both calibration and response matrices to be mean-centred. When calibration is completed, spectra of new samples (unknowns or validation set, if used) are denoted by means of the corresponding Greek letter, i.e. a new spectrum is denoted as $\boldsymbol{\xi}$. The corresponding (mostly unknown) concentrations are consistently denoted as y_ξ. A circumflex accent denotes an estimate, e.g. $\hat{y}_\xi$ for the estimated concentration. Expectated values with respect to a distribution G will be denoted as $\mathrm{E}_G(\cdot)$ Whenever it is necessary to make use of row vectors, they will be represented by lower-case underlined letters. Finally, ^{T} and $\mathrm{IF}(\cdot)$ denote the transposition and the influence function, respectively.

3 Partial least squares regression (PLS)

Partial least squares regression can be seen as a way to estimate a regression vector $\boldsymbol{\beta}$ in a linear model

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{1}$$

In this equation $\boldsymbol{\varepsilon}$ is a constant vector of identically and independently distributed erros with zero expectation and constant variance. PLS is a latent variable regression technique. This means that PLS extracts independent latent variables from the original set of p (often correlated) variables. The regression vector is calculated from these latent variables, hence overcoming difficulties in ordinary least squares such as multicollinearity. In a spectrometric context, one can intuitively see that this is a correct way to proceed, as pointed out by Svante Wold [9]. As heeded in the introduction, this insight has lead to the proposal of an alternative *multivariate latent variable regression model* [10]. However, the choice of the model upon which PLS is based has no effect on the results in this article, and is henceforth disregarded. In PLS, the latent variables are computed in such a way that they contain a maximum of relevant information concerning the relation between X and $\mathbf{y}$. Mathematically, this is expressed by the following objective function [11] in which the hth weighting vector ($\hat{\mathbf{a}}_h$) is defined as:

$$\hat{\mathbf{a}}_h = \underset{\mathbf{a}}{\operatorname{argmax}} \operatorname{cov}(X\mathbf{a}_h, \mathbf{y}) \tag{2a}$$

under the constraints that

$$\| \mathbf{a}_h \| = 1 \quad \text{and} \quad \mathbf{a}_h^T X^T X \mathbf{a}_i = 0 \text{ for } 1 \leq i < h. \tag{2b}$$

This objective is a maximization problem under two constraints, which can be solved by dint of the Langrange multiplier method. All univariate PLS algorithms share the same objective function. However, different algorithms have been proposed to accomplish the same objective in which different scaling conventions are used. E.g. in SIMPLS [12] the convention is to re-scale the estimated weighting and score vectors (i.e. $\mathbf{t}_h = X\hat{\mathbf{a}}_h$) in such a way that the score vectors ultimately carry unit variance. In any algorithm, the first weighting vector must be the dominant eigenvector of the matrix $X^T\mathbf{y}\mathbf{y}^T X$, which will then be or be not scaled, according to the convention imposed. From the second latent variable on, the second constraint becomes important: it requires the following latent variables to be orthogonal (uncorrelated) to the previous ones. Hence, the following weighting vectors will be dominant

eigenvectors of the matrix $X^T\mathbf{y}\mathbf{y}^TX$, multiplied by a projection matrix which projects onto the orthogonal complement of the subspace spanned by the previous score vectors. Hence, before scaling, the hth weighting vector will in general be equal to:

$$\hat{\mathbf{a}}_h = X^T \left(I_n - \sum_{i=1}^{h-1} \frac{\mathbf{t}_i \mathbf{t}_i^T}{\mathbf{t}_i^T \mathbf{t}_i} \right) \mathbf{y}. \tag{3}$$

The first two factors can be seen as a deflation of the datamatrix X. This deflation can either be carried out directly on the datamatrix X or on the vector $X^T\mathbf{y}$ as is the case in both algorithms used throughout this article (the Helland [13] and SIMPLS [12] algorithms). Both algorithms will be explained at the population level in the next section.

4 PLS at the population level

Before we can give an introduction to the definition and calculus of the influence function, we first need a short description of the distinction between PLS at the *population level* and PLS at the *sample level*. Individual experiments are samples taken from a certain population. E.g. the datamatrix X corresponds to a p-variate random vector $\mathbf{x}$ of which n samples are drawn from the population. In chemometrics this discrepancy is most currently disregarded, as the theoretical background is of minor importance to the analytical chemist. However, computation of the influence function requires prior definition of PLS at the population level. Let $(\mathbf{x}, y)$ be centred and distributed with given distribution G, then the objective for PLS is:

$$\mathbf{a}_h(G) = \underset{\mathbf{a}}{\operatorname{argmax}} \, \mathrm{E}_G \left[\mathbf{a}^T \mathbf{x} \, y \right] \tag{4a}$$

under the constraints that

$$\| \, \mathbf{a}_h(G) \, \| = 1 \quad \text{and} \quad \mathbf{a}_h(G)^T S(G) \mathbf{a}_i(G) = 0 \text{ for } 1 \leq i < h. \tag{4b}$$

The only difference to the objective function stated in the previous section is the explicit dependence on the distribution G. As this does not change the maximization problem, the exact solutions of problem (4) are known and can be copied to the population level from the aforementioned algorithms.
Hence, both the Helland [13] (Equations 5) and SIMPLS [12] (Equations 6) algorithms also hold at the population level, if one does not omit the fact that all vectors are population

quantities and thus dependent on the distribution G. The starting values should in both cases be:

$$\begin{aligned} \mathbf{s}(G) &= \mathrm{E}_G[\mathbf{x}y] \\ S(G) &= \mathrm{E}_G\left[\mathbf{x}\mathbf{x}^T\right]. \end{aligned}$$

In the Helland algorithm [13], an additional starting value is needed: $H_0 = \mathbf{0}_{p\times p}$. For the Helland version of the population definition of PLS, the quantities are sequentially defined as:

$$\mathbf{a}_h(G) = (I_p - S(G)H_{h-1}(G))\mathbf{s}(G) \tag{5a}$$

$$\tilde{\mathbf{a}}_h(G) = (I_p - H_{h-1}(G)S(G))\mathbf{a}_h(G) \tag{5b}$$

$$H_h(G) = H_{h-1}(G) + \frac{\tilde{\mathbf{a}}_h(G)\tilde{\mathbf{a}}_h(G)^T}{\tilde{\mathbf{a}}_h(G)^T S(G)\tilde{\mathbf{a}}_h(G)} \tag{5c}$$

$$\boldsymbol{\beta}_h(G) = H_h(G)\mathbf{s}(G) \tag{5d}$$

for any $1 \leq h \leq p$. The Helland algorithm is frequently used as a starting point for any deviation on PLS (e.g. [6, 8]) since it only consists of four equations. However, computationally it is outperformed by Sijmen de Jong's SIMPLS algorithm, which has over the last years steadily become the "standard" PLS algorithm included in commercial packages due to its computational efficiency (less flops and memory are required than in any other PLS algorithm). Hence, our work would not be complete were our approach not applied to SIMPLS.

The population quantities corresponding to SIMPLS are defined as follows:

$$\mathbf{a}_h = \begin{cases} \mathbf{s}(G) & \text{for } h = 1 \\ \left(I_p - \tilde{V}_{h-1}(G)\right)\mathbf{a}_{h-1}(G) & \text{for } h > 1 \end{cases} \tag{6a}$$

$$\mathbf{r}_h(G) = \frac{\mathbf{a}_h(G)}{\sqrt{\mathbf{a}_h(G)^T S(G)\mathbf{a}_h(G)}} \tag{6b}$$

$$\mathbf{p}_h(G) = S(G)\mathbf{r}_h(G) \tag{6c}$$

$$\mathbf{v}_h(G) = \left(I_p - \sum_{i=1}^{h-1} \tilde{V}_i(G) \right) \mathbf{p}_h(G) \tag{6d}$$

$$\tilde{V}_h(G) = \frac{\mathbf{v}_h(G)\mathbf{v}_h(G)^T}{\mathbf{v}_h(G)^T\mathbf{v}_h(G)} \tag{6e}$$

$$\boldsymbol{\beta}_h(G) = R_h(G)R_h(G)^T\mathbf{s}(G) \tag{6f}$$

for $1 \leq h \leq p$. Recall that $R_h(G)$ is a matrix containing $\mathbf{r}_1(G), \ldots, \mathbf{r}_h(G)$ as its columns. Remark that both Equations 5 and 6 hold for *any* given distribution G. The only condition is that the starting quantities $\mathbf{s}(G)$ and $S(G)$ need to exist, which boils down to existence of the second moment. The above equations define the statistical functionals $\mathbf{a}_h, \tilde{\mathbf{a}}_h, H_h, \mathbf{r}_h, \mathbf{p}_h, \mathbf{v}_h, \tilde{V}_h$ and $\boldsymbol{\beta}_h$, all being defined as mappings sending distributions G to vector or matrix valued quantities.

To return to the sample level, the empirical distribution G_n may be plugged in for G into the above expressions to yield the well-known PLS algorithms. The empirical distribution function G_n is a discrete distribution giving mass $1/n$ to each of the n measured data points, and can be shown to converge G. It is therefore the sample-based analogue of G. Starting from

$$\mathbf{s}(G_n) = \mathrm{E}_{G_n}[\mathbf{x}y] = X^t\mathbf{y}/n \quad \text{and} \quad S(G_n) = \mathrm{E}_{G_n}\left[\mathbf{x}\mathbf{x}^T\right] = X^TX/n, \tag{7}$$

one finds all other quantities, now based on the sample, by applying (5) or (6).

Let $\boldsymbol{\xi}$ be a new observation, and denote h the select number of latent variables. Then the functional $\hat{y}_{h,\boldsymbol{\xi}}$ corresponding to the predicted value based on $\boldsymbol{\xi}$ is defined as

$$\hat{y}_{h,\boldsymbol{\xi}}(G) = \boldsymbol{\xi}^T\boldsymbol{\beta}_h(G) \tag{8}$$

for any distribution G. At the sample level this corresponds to predicting a concentration of a (possibly new) sample on the basis of the calibration matrix.

5 The notion of the influence function

The influence function (IF) has been introduced by Hampel [2] in order to theoretically assess the influence that an observation $\mathbf{z}$ has on the value that a statistical functional T takes.

This observation $\mathbf{z}$ may be an observed data point, a potential outlier, ... One supposes that a small fraction ε of the data are placed at the point $\mathbf{z}$, while the other fraction $(1-\varepsilon)$ is coming from the population distribution G. Hence, the distribution becomes:

$$G_\varepsilon = (1-\varepsilon)G + \varepsilon\delta_{\mathbf{z}}, \tag{9}$$

where $\delta_{\mathbf{z}}$ is the a point mass distribution at $\mathbf{z}$. The influence function is then defined as

$$\mathrm{IF}(\mathbf{z}, T, G) = \lim_{\varepsilon \downarrow 0} \frac{T\left[(1-\varepsilon)G + \varepsilon\delta_{\mathbf{z}}\right] - T(G)}{\varepsilon} \tag{10}$$

It can be interpreted as the influence of adding an observation $\mathbf{z}$ to the data on the value of the estimator. If the value of the IF is high, then $\mathbf{z}$ is called an *influential* observation. Hence, the IF can be used to diagnose influential observations. Another use of the influence function is to asses the robustness of an estimator. If the influence function is bounded, the statistical functional is said to be robust. The influence function of the PLS-estimator will turn out to be unbounded, indicating the non-robustness of the classical PLS-procedure. Using robust PLS-procedures, e.g. as in [14, 15], might result in bounded influence functions. Computation of the IF for robust PLS procedures will be presented in a forthcoming article. In this paper we are interested in assessing the influence of an observation on the classical PLS-procedure, hence we will compute the IF for ordinary PLS.

Actual compuation of the IF makes use of derivation of the statistical functions, since (10) yields:

$$\mathrm{IF}(\mathbf{z}, T, G) = \frac{\partial}{\partial\varepsilon} T(G_\varepsilon)\bigg|_{\varepsilon=0}. \tag{11}$$

Is also turn out that the influence function is closely related to the variance of an estimator. It has been shown that [2]:

$$\mathrm{var}(T, G) \approx \frac{\mathrm{E}_G\left[\mathrm{IF}(\mathbf{z}, T, G)^2\right]}{n} \tag{12}$$

where the approximation becomes more precise as the sample size n increases. More information on the use of influence functions can be found in Hampel *et al.* [2].

Let us now proceed to the applicability of the influence function in the PLS-setting. The observation $\mathbf{z}$ will now be a couple $(\underline{x}^T, y)$ containing a spectrum and the corresponding concentration. Often $\mathbf{z}$ will be an observation $(\underline{x_i}^T, y_i)$ from the calibration matrix. The IF is then a measure of the influence of $\mathbf{z}$ on T, where T can be any of the statistical

functionals defined in the previous section. We are mainly interested in the influence that each calibration spectrum has on the regression estimators $\boldsymbol{\beta}_h$ and on the predictions $\hat{y}_{h,\boldsymbol{\xi}}$, but also the IF for the weighting vectors appear as an intermediate result. In this way, we can compute an alternative to the currently used Cook's Distance, which will be mentioned in more detail later.
Moreover, as mentioned before, the influence function allows to estimate the variance of a given estimator, which will be used in Section 5 to compute sample-specific variance estimates.

6 Algorithms for the influence function

6.1 The influence functions for the Helland algorithm

First the influence functions for the starting values of the algorithm should be found. Let $\mathbf{z} = (\mathbf{x}, y)$ be an arbitrary point in the $p+1$-dimensional space. We will make usage of the shorthand notations $\mathrm{IF}(T)$ instead of $\mathrm{IF}(\mathbf{z}, T, G)$ and hence drop the dependence on the distribution G and on $\mathbf{z}$. It is an easy exercise to check that

$$\mathrm{IF}(S) = \mathbf{x}\mathbf{x}^T - S \tag{13}$$

$$\mathrm{IF}(\mathbf{s}) = \mathbf{x}y - \mathbf{s}, \tag{14}$$

where $\mathbf{s}$ and S are shorthand notations for $\mathbf{s}(G)$ and $S(G)$. With these starting values, the influence functions can be computed recursively as follows:

$$\mathrm{IF}(\mathbf{a}_h) = (I_p - SH_{h-1})\,\mathrm{IF}(\mathbf{s}) - (S\,\mathrm{IF}(H_{h-1}) + \mathrm{IF}(S)H_{h-1})\mathbf{s} \tag{15a}$$

$$\mathrm{IF}\,(\tilde{\mathbf{a}}_h) = (I_p - H_{h-1}S)\,\mathrm{IF}(\mathbf{a}_h) - (H_{h-1}\,\mathrm{IF}(S) + \mathrm{IF}(H_{h-1})S)\mathbf{a}_h \tag{15b}$$

$$\begin{aligned}\mathrm{IF}(H_h) &= \mathrm{IF}(H_{h-1}) + \frac{\tilde{\mathbf{a}}_h\,\mathrm{IF}(\tilde{\mathbf{a}}_h)^T + \mathrm{IF}(\tilde{\mathbf{a}}_h)\tilde{\mathbf{a}}_h^T}{\tilde{\mathbf{a}}_h^T S\tilde{\mathbf{a}}_h} \\ &\quad - \frac{\tilde{\mathbf{a}}_h\tilde{\mathbf{a}}_h^T}{(\tilde{\mathbf{a}}_h^T S\tilde{\mathbf{a}}_h)^2}\left(\tilde{\mathbf{a}}_h^T S\,\mathrm{IF}(\tilde{\mathbf{a}}_h) + \tilde{\mathbf{a}}_h^T\,\mathrm{IF}(S)\tilde{\mathbf{a}}_h + \mathrm{IF}(\tilde{\mathbf{a}}_h)^T S\tilde{\mathbf{a}}_h\right)\end{aligned} \tag{15c}$$

$$\mathrm{IF}(\boldsymbol{\beta}_h) = \mathrm{IF}(H_h)\mathbf{s} + H_h\,\mathrm{IF}(\mathbf{s}), \tag{15d}$$

for $1 \leq h \leq p$. Again, short-hand notations $\mathbf{a}_h = \mathbf{a}_h(G), \tilde{\mathbf{a}}_h = \tilde{\mathbf{a}}_h(G), \ldots$ are used.

6.2 The influence functions for the SIMPLS algorithm

It is clear that the starting values for the Helland algorithm also hold in this case. The algorithm continues as

$$\mathrm{IF}(\mathbf{a}_h) = \begin{cases} \mathrm{IF}(\mathbf{s}) & \text{for } h = 1 \\ \mathrm{IF}(\mathbf{a}_{h-1}) - \mathrm{IF}(\tilde{V}_{h-1})\mathbf{a}_{h-1} - \tilde{V}_{h-1}\,\mathrm{IF}(\mathbf{a}_{h-1}) & \text{for } h > 1 \end{cases} \tag{16a}$$

$$\mathrm{IF}(\mathbf{r}_h) = \frac{\mathrm{IF}(\mathbf{a}_h)}{(\mathbf{a}_h^T S \mathbf{a}_h)} - \frac{\mathbf{a}_h}{(\mathbf{a}_h^T S \mathbf{a}_h)^{\frac{3}{2}}} \left(\mathrm{IF}(\mathbf{a}_h)^T S \mathbf{a}_h + \mathbf{a}_h^T\,\mathrm{IF}(S)\mathbf{a}_h\right) \tag{16b}$$

$$\mathrm{IF}(\mathbf{p}_h) = \mathrm{IF}(S)\mathbf{r}_h + S\,\mathrm{IF}(\mathbf{r}_h) \tag{16c}$$

$$\mathrm{IF}(\mathbf{v}_h) = \mathrm{IF}(\mathbf{p}_h) - \sum_{i=1}^{h-1} \left[\tilde{V}_i\,\mathrm{IF}(\mathbf{p}_h) + \mathrm{IF}\left(\tilde{V}_i\right)\mathbf{p}_h\right] \tag{16d}$$

$$\mathrm{IF}\left(\tilde{V}_h\right) = \frac{\mathrm{IF}(\mathbf{v}_h)\mathbf{v}_h^T + \mathbf{v}_h\,\mathrm{IF}(\mathbf{v}_h)^T}{\mathbf{v}_h^T\mathbf{v}_h} - 2\frac{\mathbf{v}_h\mathbf{v}_h^T}{(\mathbf{v}_h^T\mathbf{v}_h)}\,\mathrm{IF}(\mathbf{v}_h)^T\mathbf{v}_h \tag{16e}$$

$$\mathrm{IF}\left(\boldsymbol{\beta}_h\right) = \mathrm{IF}(R_h)R_h^T\mathbf{s} + R_h\,\mathrm{IF}(R_h)^T\mathbf{s} + R_h R_h^T\,\mathrm{IF}(\mathbf{s}), \tag{16f}$$

for any $1 \leq h \leq p$.

6.3 Further comments

A proof of the expressions in (15) and (16) for both algorithms is straightforward by applying the functionals on G_ε, using (11) and standard differentiation rules. The equations are valid at any given step h of the iteration. Furthermore, note that the influence function for the prediction $\hat{y}_{h,\boldsymbol{\xi}}$ is immediately obtained as

$$\mathrm{IF}\left(\hat{y}_{h,\boldsymbol{\xi}}\right) = \xi^T\,\mathrm{IF}\left(\boldsymbol{\beta}\right). \tag{17}$$

Both algorithms lead to identical influence functions for the regression vector and the predictions. This is logical, since both the Helland and SIMPLS algorithm yields the same values for $\boldsymbol{\beta}$ and $\hat{y}_{h,\boldsymbol{\xi}}$. Therefore also the influence functions are identical. One needs to

consider the two algorithms as just two different ways of computing the IF and not as different approximations of the IF, since one is computing in both cases the exact IF. Computationally, the SIMPLS algorithm to compute the IF for the regression estimator outperforms the Helland algorithm.

The expressions found for the IF are valid for any distribution G for which $\mathbf{s}$ and S are existing. In practice, the population distribution G in unknown but can be estimated by the empirical distribution function G_n. Hence, in the practical applications the IF will always be evaluated for G taken to be G_n. This implies that all quantities $\mathbf{s}, S, \mathbf{a}_h, \tilde{\mathbf{a}}_h, H_h, \boldsymbol{\beta}_h, \mathbf{r}_h, \ldots$ appearing in (15) or (16) are taken to be the sample estimates as obtained by plugging G_n in the equations (5) or (6).

7 Applications of the influence function

7.1 Influence diagnosis

Most commonly, the influence of individual samples is shown using the so-called *Cook's squared distance* [4]. Roughly, it measures the change in the regression coefficients if the ith observation is omitted from the data. It is computed as follows:

$$\mathrm{CD}(\mathbf{z}_i) = \frac{1}{p\sigma_e^2} \sum_{j=1}^{n} \left(\hat{y}_j - \hat{y}_j^i\right)^2 \tag{18}$$

In this equation, σ_e^2 is the residual variance and $\hat{y}_j^i$ denotes the predicted concentration for sample j based on a regression vector computed from calibration matrices from which sample i has been deleted. A large value for the Cook distance is an indication that an observation is an influenctial observation or outlier. The Cook distances are illustrated in Figure 1.
It should be clear that the influence function is apt to be a suitable measure for the influence of a sample on prediction. An analogous approach as in the Cook's squared distance leads to a diagnostic plot based on the influence function which is a viable alternative to the existing approaches. The measure of influence of sample i on prediction is the sum of squared influence functions for sample i on the predicted concentrations of all other samples, i.e.:

$$\mathrm{SID}(\mathbf{z}_i) = \frac{1}{n} \sum_{j=1}^{n} \mathrm{IF}\left(\mathbf{z}_i, \hat{y}_j, G_n\right)^2 \tag{19}$$

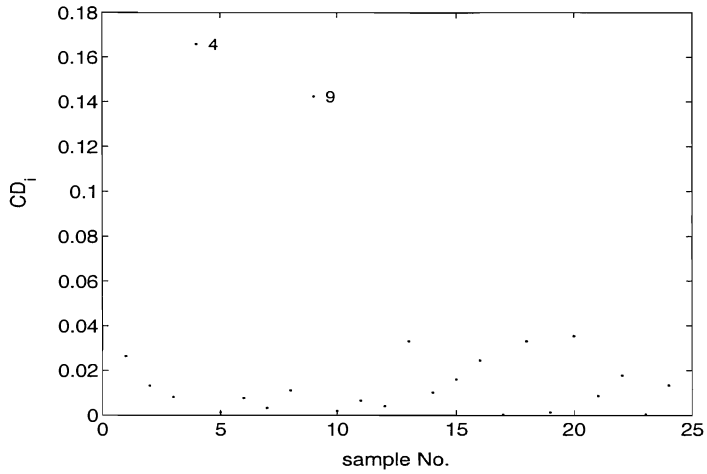

Figure 1: The Cook's Squared Distance for the Fearn data

where $\hat{y}_j$ stands for the statistical functional $\hat{y}_{h,\mathbf{x}_i^T}$. We carried out a comparison of the Cook Distance (CD, Equation 18) and the Squared Influence Diagnostic (SID, Equation 19) for the "Fearn" data [16]. The goal of the analysis by Fearn was to predict the protein content in wheat samples. Thereunto, near-infrared measurements were carried out at six wavelengths. The dataset has thenceforth extensively been used and referred to in the chemometrical literature [7, 6, 17] and references therein.

From Figures 1 and 2 we can see that both the CD and the SID detect sample 4 as a highly influential sample (an outlier). Furthermore, in the CD plot sample 9 is regarded is quite influential (but not as influential as sample 4) whereas in the SID plot the influence of sample 9 is less important, but still outlying with respect to the other sample points.

Using the influence function, it is also possible to establish a measure of the influence of a certain sample on the prediction of the concentration of a new sample, i.e. a sample that was not included in the original datamatrix. In this sense, it can be seen as a sample-specific influence diagnosis (SSID). Although such a sample-specific influence diagnosis makes no sense in the classical multivariate calibration set-up (one will not calculate a new regression vector for each sample to be predicted, based on a datamatrix from which highly influential

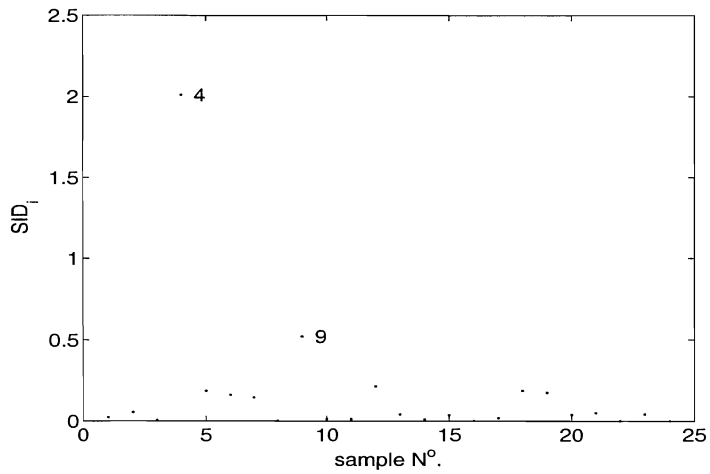

Figure 2: The Squared Influence Diagnostic plot for the Fearn data

samples have been deleted), it may be useful in a semi-local context, where on possesses a myriad of spectra for calibration and one tries to find the ones closest to the new one to be analysed. A too high sample-specific influence would indicate that the calibration spectrum should not be included in the local calibration matrix. We include an example of the SSID. Calibration was in this case done on all but one of the samples from the Fearn dataset; the last sample was predicted and the influence of the first samples on prediction of the latter is plotted in Figure 3. One sees that sample 3 is the most influential for this prediction, but that none of the calibration samples has an extremely high influence on the prediction of the concentration. Note that the SSID plot allows to distinguish between negative and positive influence on prediction.

7.2 Variance estimation

7.2.1 Theory

It has been stated in Section 5 that the influence function leads to an estimate of variance (Equation 12). An estimate of variance for a predicted concentration is found by combining Equations 12 and 17: plugging in the computed influence function for the predicted

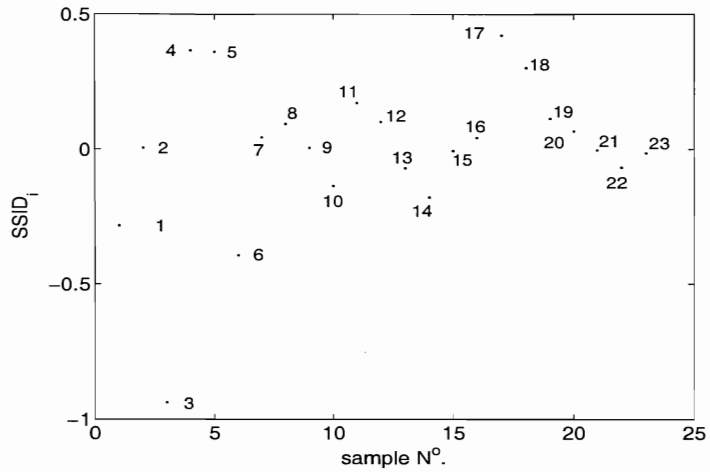

Figure 3: Influence of samples to the prediction of a new sample, Fearn data

concentration. Taking G_n for G in Equation 12 yields

$$\widehat{\text{var}}(T, G_n) = \frac{1}{n^2} \sum_{i=1}^{n} \text{IF}(\mathbf{z}_i, T, G_n)^2. \tag{20}$$

Note that the expected value in (12) reduces to an arithmethic mean when working at the sample level. Having obtained a sample-specific estimate of variance, a sample-specific prediction interval can be computed, but the main drawback of the method proposed here is that up till now no estimate of degrees of freedom can be derived from the influence function. In case of large sample sizes this plays no role, but otherwise we suggest to plug in the cross-validated estimate by Van der Voet [18].

7.2.2 Verification through Monte Carlo simulation

We investigated the correctness of the estimate of variance for the regression vector by means of a Monte Carlo simulation, analogously to the work by Faber [17]. The set-up of the simulation was as follows:

1. Determine the optimal number of latent variables for the mean-centred datamatrices. This was done by venetian blinds cross-validation; four latent variables was considered

to be the optimal number. This corresponds to the number found in the publications cited above.

2. Compute SIMPLS vectors up to the aforementioned number of latent variables; define new datamatrices $\check{X} = T_4 P_4^T$ and $\check{\mathbf{y}} = \check{X}\hat{\boldsymbol{\beta}}_4$. These "new" data are perfectly described by the PLS model.

3. By adding noise to the datamatrices $\check{X}$ and $\check{\mathbf{y}}$, $N_{\text{rep}} = 1001$ new data sets are generated. This noise can be described as follows: $\check{X}_i = \check{X} + E_i$; $\check{\mathbf{y}}_i = \check{\mathbf{y}} + \varepsilon_i$, where the noise matrices E_i and ε_i are filled up with random numbers taken from standard normal distributions with appropriate dimensions.

4. For the first $\left(\check{X}_1, \check{\mathbf{y}}_1\right)$, compute the estimate of variance for the regression vector according to Equation 20. The square roots of its diagonal elements give estimated standard errors.

5. Compute the PLS regression vector for each of the other N_{rep}-1 generated samples $\left(\check{X}_i, \check{\mathbf{y}}_i\right)$.

6. By taking the elementwise standard deviation of these N_{rep}-1 estimated regression vectors "true" (or simulated) standard deviations are obtained and can be compared with the estimated standard error from step 4.

As original datset, we used for this simulation the "NIR Biscuit" data, a dataset first analysed by Osborne *et al.* [19]. The experimental set-up was prediction of (among other analytes) the sucrose content in biscuits by means of near infrared (NIR) spectrometry. The dataset consists of 40 samples and 600 spectral variables are used. It has previously been analysed in [20]. Because the simulation generates a 600×1-vector of standard deviations, the results are summarized by taking their average value. This is an appropriate way to summarize the results since standard deviations on individual components of the regression vector do not differ very much. To have a complete comparison, we also computed the estimate of standard deviation obtained from a local linearization of the PLS estimator. The results are summarized in Table 1. One sees that both the local linearization technique as the IF-based variance estimate yield outcomes close to the "true" standard error.

	$\overline{\text{std}}\left(\hat{\beta}\right)$
IF	0.0395
local linearization	0.0320
MC ("true")	0.0350

Table 1: Average standard deviations for the regression vector, comparison of the "true" deviation with the estimates obtained from the influence function (IF) and a local linearization

8 Summary and conclusions

In this work, we have adopted an approach which is common in the field of robust statistics, but has been shed from the chemometrics community for years. We investigated the usefulness of the influence function in the field of chemometrics, and thus its practical applicability in the analytical laboratory. We concluded that the contribution of the computation of the influence function to the PLS method is of major importance.
The benefits of the influence function are twofold: at first it allows to investigate the influence of individual samples to prediction on a sample specific basis, which may be of importance when the analytical chemist has very large datasets at his disposition and it is not very clear which samples to choose as a calibration matrix (a semi-local approach). Also the detection of very large or zero influences may indicate in the general, non-local context that a certain sample is not suitable to be included in the calibration set. In this sense we proposed an alternative to the current method of influence diagnosis, the Cook Distance.

One should realize that both CD and SID are based on classical, non-robust PLS and are subject to the masking effect. By this it is meant that outliers can bias the estimates in such a way that the diagnostic measures CD or SID, based on these estimates, become non-reliable and can fail to detect the outliers. Such a masking effect can occur when there are clusters of several outliers in the data, but in presence of only few outliers CD or SID are still believed to be effective. In any case, by estimating the population quantities appearing in the expressions for the IF by robust estimates, more resistant measures for detecting influenctial observations could be obtained.

The influence function also provided us with a novel estimate of variance of the PLS-estimator. As computation of the existing estimates of variance is sometimes cumbersome, the influence function approach might be considered as a viable alternative to those methods.

Monte Carlo simulations have corroborated the hypothesis of its correctness. Moreover, if both diagnosis and variance estimation are required – a very likely setting in the analytical laboratory – the approach supposed here is the only one suitable for a joint computation which we wot of.

Acknowledgements

This research was financed by a PhD grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT Vlaanderen).
The authors would also like to thank S. Verboven for her file "`labelen.m`".

References

[1] H. Wold in P.R. Krishnaiah (ed.), Multivariate analysis, Proceedings of an International Symposium, 14-19 June 1965, Dayton, Ohio, Academic Press, NY, 1966, 391-420.

[2] F.R. Hampel, E.M. Ronchetti , P.J. Rousseeuw and W.A. Stahel, Robust Statistics: the approach based on influence functions, Wiley, New York, 1986.

[3] N.M. Faber, X.-H. Song and P.K. Hopke, in press.

[4] D.L. Massart, B.G.M. Vandeginste, L.M.C. Buydens, S. de Jong, P.J. Lewi and J. Smeyers-Verbeke, Handbook of Chemometrics and Qualimetrics: Part A, Elsevier, Amsterdam, The Netherlands, 1997, pp. 203, 282.

[5] A. Phatak, P.M. Reilly and A. Penlidis, Analytica Chimica Acta, 277 (1993), 495-501.

[6] M.C. Denham, J. Chemometr., 11 (1997), 39-52.

[7] N.M. Faber and B.R. Kowalski, J. Chemometr., 11 (1997), 181-238.

[8] S. Serneels, P. Lemberge and P.J. Van Espen, submitted for publication.

[9] S. Wold, Technometrics, 35 (1993), 136-139.

[10] A.J. Burnham, J.F. MacGregor and R. Viveros, Chemometr. Intell. Lab. Syst., 48 (1999), 167-180.

[11] C.J.F. Ter Braak and S. de Jong, J. Chemometr., 12 (1998), 41-54.

[12] S. de Jong, Chemometr. Intell. Lab. Syst., 18 (1993), 251-263.

[13] I.S. Helland, Commun. Stat. – Simul. Comput., 17 (1988), 581-607.

[14] I.N. Wakeling, H.J.H. MacFie, J. Chemometr., 6 (1992), 189-198.

[15] J.A. Gil and R. Romera, J. Chemometr., 12 (1998), 365-378.

[16] T. Fearn, Appl. Stat., 32 (1983), 73.

[17] N.M. Faber, Chemometr. Intell. Lab. Syst., 64 (2002), 169-179.

[18] H. van der Voet, J. Chemometr., 13 (1999), 195-208.

[19] B.G. Osborne, T. Fearn, A.R. Miller and S. Douglas, J. Scient. Food Agric., 35 (1984), 99-105.

[20] M. Stone and R.J. Brooks, J. R. Stat. Soc. B, 52 (1990), 237-269.