JOURNAL OF CHEMOMETRICS, VOL. 11, 157–171 (1997)

BOOTSTRAPPING PRINCIPAL COMPONENT REGRESSION MODELS

R. WEHRENS[1] AND W. E. VAN DER LINDEN[1]
[1] *Department of Chemical Analysis, University of Twente, PO Box 217, NL-7500 AE Enschede, Netherlands*

SUMMARY

Bootstrap methods can be used as an alternative for cross-validation in regression procedures such as principal component regression (PCR). Several bootstrap methods for the estimation of prediction errors and confidence intervals are presented. It is shown that bootstrap error estimates are consistent with cross-validation estimates but exhibit less variability. This makes it easier to select the correct number of latent variables in the model. Using bootstrap confidence intervals for the regression vectors, it is possible to select a subset of the original variables to include in the regression, yielding a more parsimonious model with smaller prediction errors. The methods are illustrated using PCR, but can be applied to all regression models yielding a vector or matrix of regression coefficients. © 1997 by John Wiley & Sons, Ltd.

Journal of Chemometrics, Vol. **11,** 157–171 (1997) (No. of Figures: 8 No. of Tables: 4 No. of Refs: 16)

KEY WORDS variable selection; prediction error estimation; bootstrap; latent variable regression

1. INTRODUCTION

Latent variable methods such as partial least squares regression (PLS) and principal component regression (PCR) are widely used in chemometrics. The selection and validation of an optimal calibration model can be performed in a number of ways. If many data are available, separate calibration and test sets can be used for the construction and validation respectively. In general, however, one would want to use as many data as possible in the calibration to ensure a correct model, especially if the number of data is not very large. In that case, cross-validation methods are widely used. Leave-one-out cross-validation methods in general are unbiased,[1, 2] but they may suffer from a large variability. Grouped cross-validation, in which more samples are left out at the same time, reduces this variance at the expense of introducing bias.

An alternative is provided by bootstrap methods. Originally proposed by Efron in the 1970s,[3] a vast number of papers have appeared, mainly in the statistical literature. Comprehensive overviews can be found in References 4–6. In chemistry, bootstrap methods are mentioned in a few papers only.[7–11]

The bootstrap is intimately related to techniques such as the jack-knife and cross-validation. Indeed, the bootstrap can be seen as a smoothed version of cross-validation with adjustments to correct for bias. However, the bootstrap is more versatile, because not only error estimates can be generated but also confidence intervals. It is even possible to use a double bootstrap to assess the precision of the confidence intervals and so on, although this rapidly becomes unpractical because of the long computing times.

The name ‘bootstrap’ originates from the expression ‘pulling yourself up by your own bootstraps’ and refers to the basic idea of the bootstrap, sampling *with replacement* from the data. In this way a

Correspondence to: R. Wehrens, Department of Analytical Chemistry, Catholic University Nijmegen, Toernooiveld 1, NL-6525 ED Nijmegen, Netherlands.

CCC 0886–9383/97/020157–15 $17.50

Received April 1996
Accepted September 1996

large number of 'bootstrap samples' is generated, each of the same size as the original data set. From each bootstrap sample the statistical parameter of interest is calculated. This yields an ensemble of estimates that is used to obtain a meta-estimate such as the standard error or the width of a confidence interval. For instance, when estimating the error of a mean, we could generate 100 bootstrap samples and calculate the mean for each sample. The standard error of the 100 means is the bootstrap estimate of the standard error of the mean. In this simple case it can be shown analytically that the bootstrap estimate is equal to the usual estimate and no resampling is needed. However, in the case of other statistics, exact formulae are not available and sampling is necessary.

The procedure described above is known as the non-parametric bootstrap. Another approach is the parametric bootstrap, in which knowledge of the distribution of the data can be utilized by sampling from that distribution rather than from the data. In this way, estimates can be generated with the same or even better accuracy than by using textbook formulae and answers can be given in cases where textbook formulae cannot. However, in this paper we will concentrate on the non-parametric bootstrap. As in all non-parametric methods, no assumptions on the distribution of the data are needed, which can be a significant advantage over methods presupposing e.g. a standard normal distribution. The non-parametric bootstrap, therefore, is able to provide estimates in cases where other methods cannot. Moreover, in cases where exact answers are possible, the bootstrap is able to reproduce them.

In the next section a data set obtained with a voltammetric sensor consisting of an array of individually modified electrodes is described. This set is used to illustrate the bootstrap methods explained in the following sections. We are interested in estimates for prediction error and the selection of an optimal model, which encompasses both the selection of an optimal number of factors and variable selection. The bootstrap is applied to PCR, though it can be applied in exactly the same way to any other regression method using latent variables, such as PLS or non-linear PCR or PLS. The paper concludes with some general remarks.

2. EXPERIMENTAL

2.1. Data

We use data that have already been described elsewhere.[12] A voltammetric sensor consisting of an array of individually modified microelectrodes was used. Sixteen electrodes were placed around a larger, central electrode made of iridium. Two of the 16 microelectrodes were coated with gold, two with rhodium, two with platinum and the rest with iridium. During a measurement, for each electrode a voltammogram was recorded in a single sweep. Four electrodes were selected (one for each type of top layer material) and their voltammograms were collected in a data matrix for each measurement. Each voltammogram consisted of 496 data points. After smoothing with a moving average filter, the first derivative was taken, since this representation yielded the best calibration model.[12] Finally, only every eighth point was retained and the data matrix was unfolded, yielding a vector of 244 points for each measurement.

The samples contained nitrobenzene (NB) and the three disubstituted dinitrobenzenes (*ortho*-, *meta*- and *para*-DNB) with concentrations of $4{\cdot}02\times10^{-5}$, $2{\cdot}41\times10^{-4}$ and $5{\cdot}01\times10^{-4}$ mol l^{-1} in a full factorial design (81 samples). Additionally, 15 samples were measured in which one or more of the nitrobenzenes were absent, including a blank. In these samples the concentration of the nitrobenzenes that were present was $2{\cdot}41\times10^{-4}$ mol l^{-1}. This yielded a data matrix of 96×244. All reagents were obtained from Merck and were of proAnalysi grade. The sample containing $4{\cdot}02\times10^{-5}$ mol l^{-1} of all four components is depicted in Figure 1.

The calibration is difficult because of non-linear effects. These are caused by gradual fouling of the

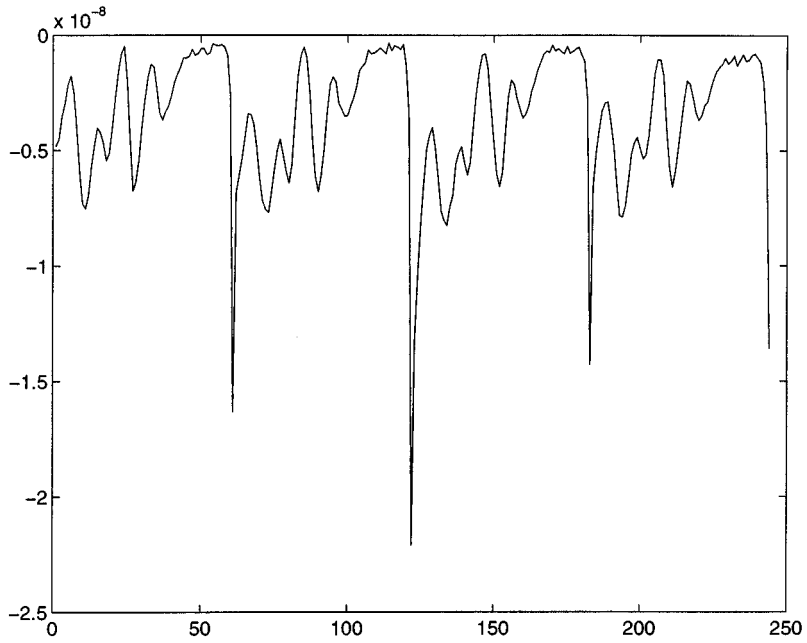

Figure 1. Data vector of sample containing $4{\cdot}02\times10^{-5}$ mol l^{-1} of each component. The first 61 points are from the gold electrode, points 62–122 from the rhodium electrode, points 123–183 from the platinum electrode and the remaining 61 points from the iridium electrode

electrodes, reducing the amount of available metal surface, and electrochemical reactions in which products or reactants react directly with each other. Furthermore, the voltage axis may shift over time, since no true stable reference electrode is present in the sensing device. Non-linear calibration methods such as neural networks are expected to yield better results in this situation, as is found elsewhere.[12] However, PCR and PLS are to some extent capable of modelling non-linearities by including extra latent variables.[13] As the purpose of this paper is to show the possibilities of bootstrap technology in multivariate calibration, we will not discuss these methods any further. Concentrations were standardized to zero mean and unit variance; voltammograms were unscaled. The scores on the first five principal components are plotted against each other in Figure 2.

2.2. Hardware and software

All calculations were performed in Matlab 4.0 on a Silicon Graphics R4400 SC workstation. PCR was performed using the `pcr1` function in the Matlab toolbox by Barry Wise. Computing times were up to 35 min for confidence intervals on regression coefficients (2000 bootstrap samples).

3. NON-PARAMETRIC BOOTSTRAP METHODS

Central to the idea of the non-parametric bootstrap is the notion that the best esimate of an unknown probability distribution F is given by the empirical distribution $\hat{F}$, putting probability $1/n$ on each of the n observed data points. Using a Monte Carlo algorithm, B bootstrap samples are generated by drawing from the empirical distribution *with replacement*. For each bootstrap sample the statistic of interest is calculated and finally, from the B estimates, meta-estimates such as the standard error, bias or percentiles can be calculated. Many modifications to this simple scheme are possible and, indeed,

sometimes necessary, but the overall idea is generally applicable.

In a regression context, two basic paradigms can be used to construct bootstrap samples. The first is called bootstrapping pairs. Each bootstrap sample is constructed by selecting n-tuples (x, y) from the original data set:

$$(\mathbf{X}^*, \mathbf{Y}^*) = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_n}, y_{i_n})\} \tag{1}$$

where (x_i, y_i) denotes the ith (physical) sample. Because sampling is performed with replacement, in almost all bootstrap samples data occur more than once.

The other method is called bootstrapping residuals. There the regression procedure is executed to find estimates for the coefficients and the residuals:

$$\mathbf{Y} = \hat{\beta}\mathbf{X} + \hat{\mathbf{E}} \tag{2}$$

and new bootstrap samples are generated by adding random selections from the (centred) residuals to $\mathbf{Y}$:

$$(\mathbf{X}^*, \mathbf{Y}^*) = \{(x_1, \hat{\beta}x_1 + \varepsilon_{i_1}), (x_2, \hat{\beta}x_2 + \varepsilon_{i_2}), \ldots, (x_n, \hat{\beta}x_n + \varepsilon_{i_n})\} \tag{3}$$

where ε_i is a random selection from $\mathbf{E}$. Here all data appear in each bootstrap sample and the randomly selected residuals are added to the response variables to yield bootstrap samples. This is the method described by Bonate.[7]

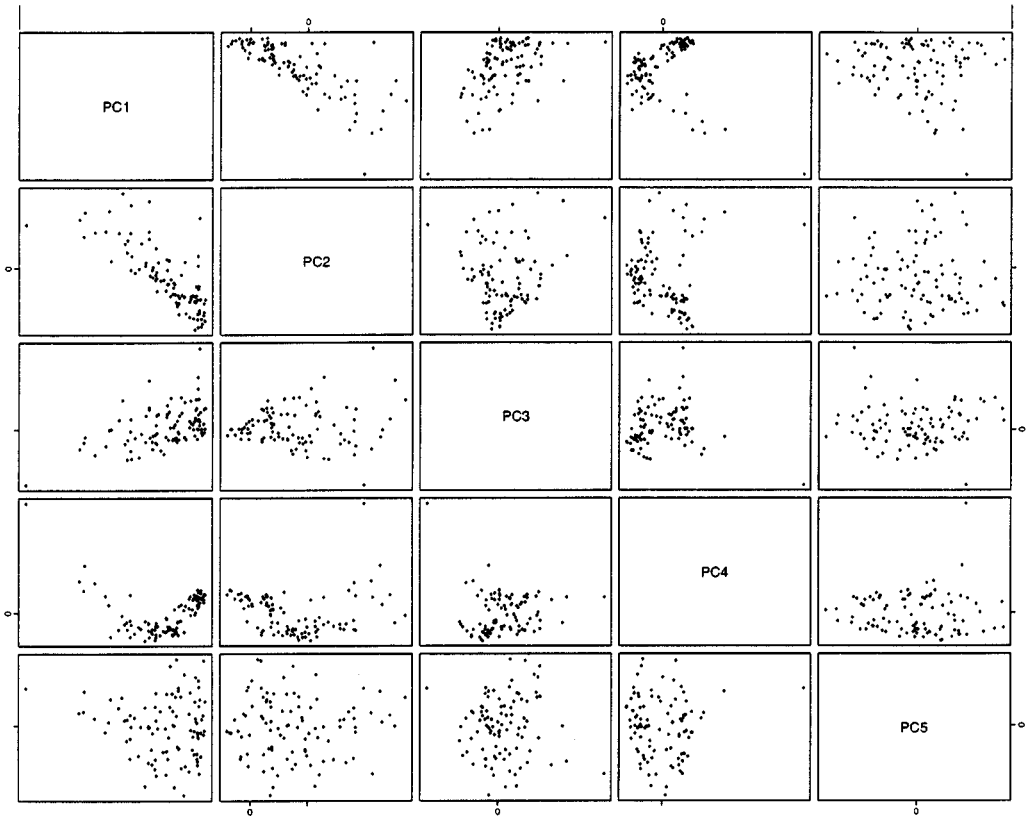

Figure 2. Scores on first five principal components for all 96 samples. The outlier in the PC1 and PC4 directions is the blank: no nitrobenzenes were present in the sample

The difference between the two methods is that in the latter case the *X*-variables are regarded as fixed. One assumes that the basic regression model is correct and that the residuals can be regarded as equal. If this is not the case, for instance when residuals have different variances or when errors are present in the *X*-variables, bootstrapping residuals will yield erroneous results. Bootstrapping pairs, on the other hand, is less sensitive to wrong model assumptions. Furthermore, if the assumptions underlying bootstrapping residuals are met, bootstrapping pairs will yield approximately the same results. In this paper we will concentrate on bootstrapping pairs.

3.1. The number of bootstrap samples

The number of bootstrap samples needed is dependent on the problem. If too few samples are taken, an added variability due to the sampling procedure is present in the results. For point estimates such as the prediction error estimate, 200 samples in most cases should suffice, whereas to construct accurate confidence intervals, in general more bootstrap samples are necessary. A number of at least 1000 samples is advocated in the literature.[5] However, it is wise to check whether the variability decreases significantly when performing more bootstrap replications.

3.2. Prediction error estimates

In regression it is important to obtain an estimate of the prediction error, not only because of the importance of this parameter itself but also because this parameter is often used as a criterion to identify the best model. Cross-validation is widely used. The estimated leave-one-out cross validation error is given by

$$\widehat{\mathrm{err}}_{\mathrm{cv}} = \sqrt{\left(\frac{\Sigma_i (y_i - \hat{\beta}_i x_i)^2}{N}\right)} \tag{4}$$

where $\hat{\beta}_i$ is the model determined without using (x_i, y_i) and N is the number of samples in the data set. In grouped cross-validation, more samples are left out during the model-building phase; these are subsequently used in the evaluation phase so that the summation contains less terms. Other methods can be applied as well, such as Mallow's Cp and the Bayesian information criterion (BIC). These, however, are not very robust; the Cp and BIC statistics require a roughly correct working model to obtain valid estimates.[13] Moreover, they require knowledge of the number of parameters, p, which may not be known for more complicated models.

Two bootstrap methods will be described here: the bias-corrected simple bootstrap[1] and the 0·632 estimator.[1] A generalization of the latter, the 0·632+ estimator,[2] appears to perform exactly equal to the 0·632 estimator for the current data and is explained only briefly.

Bias-corrected simple bootstrap

The simplest bootstrap method generates B bootstrap samples, estimates the model on each (x^*) and applies each model to the original sample ($\hat{F}$) to give B estimates of the prediction error. The average, $\overline{\mathrm{err}}(x^*, \hat{F})$, could be used as an estimate, but is biased downwards because data are used for both construction and evaluation of the model. A correction for this bias is easily found: instead of assessing the prediction error using the original sample, it is also possible to do this using the individual bootstrap samples ($\hat{F}^*$). This, of course, leads to an even smaller estimate, $\overline{\mathrm{err}}(x^*,\hat{F}^*)$, but the difference between the two estimates can be seen as a measure of the 'optimism' that is present

when assessing the prediction error using the same data that were used to build the model:

$$\mathrm{OPT} = \overline{\mathrm{err}}(x^*, \hat{F}) - \overline{err}(x^*, \hat{F}^*) \tag{5}$$

The final estimate consists of the sum of the apparent error rate (where the model is built *and* evaluated with the complete set of data) and the optimism:

$$\overline{\mathrm{err}}_{\mathrm{bcs}} = \mathrm{err}(x, \hat{F}) + \mathrm{OPT} \tag{6}$$

This estimator will hereafter be denoted BCS bootstrap.

The 0·632 estimator

Another approach is the so-called 0·632 estimator. The prediction error is estimated from both those samples that do not contain the datum for which an error is estimated and the complete data set. The final prediction error estimate is obtained by a weighted sum of the two components:

$$\widehat{\mathrm{err}}_{0\cdot632} = 0\cdot368\mathrm{err}(x, \hat{F}) + 0\cdot632\varepsilon_0 \tag{7}$$

where $\mathrm{err}(x, \hat{F})$ is the apparent error rate and ε_0 is the average error obtained from bootstrap data sets not containing the point being predicted. The value 0·632 stems from the fact that it is roughly the probability that a given observation appears in a bootstrap sample of size n. This factor makes the estimator roughly unbiased.

However, in some special cases where overfitting can occur, the 0·632 estimator is biased. Consider the example where one has to predict $y=0$ or 1, both events with probability 0·5; then the expected value for the 0·632 estimator is $0\cdot632\times0\cdot5=0\cdot316$, lower than the true value of 0·5. Another estimator, called 0·632+,[2] is a less biased compromise between $\mathrm{err}(x, \hat{F})$ and ε_0. Instead of using the weights 0·632 and 0·368, they are computed according to the amount of overfitting in the model. Thus the 0·632 estimator is a special case of the 0·632+ estimator. It can be shown that the latter always gives an equal or higher error estimate. In simulation studies the 0·632+ estimator appeared to consistently outperform cross-validation methods.[2] In this work, however, the 0·632 and 0·632+ estimators yielded exactly the same results, so the interested reader should consult the reference for details.

3.3. Confidence intervals

The percentile method

Again the basic bootstrap paradigm is useful in which B bootstrap samples are generated and the parameter of interest is calculated. The simplest method, called the percentile method, sorts the estimates and takes the α, $1-\alpha$ values out of the list. The method has two important advantages: it is transformation-respecting as well as range-preserving. The first property states that an interval estimated from a transformed parameter (e.g. $\log\theta$) yields, after back transformation, the same interval as estimated from the parameter θ itself. The range-preserving property ensures that if e.g. the values for θ are always in the range $[-1, 1]$, the confidence interval will be in the same range. The coverage properties of the percentile method are reasonably good, even in non-symmetric distributions;[5] however, because of the non-parametric nature of the estimation, the intervals in general are a bit too short (i.e. more than 5% of the data will lie outside the 95% interval). Furthermore, the method is afflicted by bias.

The BC_α *method*

A better method is the BC_α method. The name is an acronym for bias-corrected and accelerated and its calculation involves two parameters $\hat{\alpha}$ and $\hat{z}_0$, the acceleration and bias correction respectively. The

interval of intended coverage $1-2\alpha$ is given by

$$(\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{up}}) = (\hat{\theta}^{*(\alpha_1)}, \hat{\theta}^{*(\alpha_2)}) \tag{8}$$

where

$$\alpha_1 = \Psi\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(\alpha)}}{1 - \hat{a}(\hat{z}_0 + z^{(\alpha)})}\right) \tag{9}$$

$$\alpha_2 = \Psi\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(1-\alpha)}}{1 - \hat{a}(\hat{z}_0 + z^{(1-\alpha)})}\right) \tag{10}$$

In these formulae, $z^{(\alpha)}$ is the 100αth percentile point of a standard normal distribution and Ψ is the cumulative normal distribution function, e.g. $\Psi(1{\cdot}645) = 0{\cdot}95$ and $z^{0{\cdot}95} = 1{\cdot}645$. The confidence interval is given by taking the appropriate percentile of the bootstrap distribution θ^*, e.g. if values for α_1 and α_2 are $0{\cdot}110$ and $0{\cdot}985$ respectively, then the confidence points when using 1000 bootstrap replications will be the 110th and 985th ordered values of θ^*. Finally, the acceleration and bias correction are approximated by

$$\hat{z}_0 = \Psi^{-1}\left(\frac{\#\{\hat{\theta}^*(b) < \hat{\theta}\}}{B}\right) \tag{11}$$

$$\hat{a} = \frac{\Sigma_i(\hat{\theta}_{(\cdot)} - \hat{\theta}_i)^3}{6[\Sigma_i(\hat{\theta}_{(\cdot)} - \hat{\theta}_i)^2]^{3/2}} \tag{12}$$

where the argument of the inverse Ψ-function is the fraction of bootstrap replications yielding a value smaller than $\hat{\theta}$, and $\hat{\theta}_{(\cdot)}$ is the mean value of $\hat{\theta}_i$.

The BC_α confidence points have two important advantages: first, they are transformation-respecting; second, they are more accurate than the percentile intervals. A very good approximation to BC_α intervals is given by the ABC method[5] in much shorter computing times.

4. RESULTS AND DISCUSSION

4.1. The number of bootstrap samples

In Table 1 the results of some experiments are reported in which prediction errors and confidence intervals are estimated using the $0{\cdot}632$ estimator and the percentile method respectively. Twenty experiments were performed in each case. For the error predictions the numbers reported represent the ratio of the standard deviation of the twenty estimates and the mean. In all cases the variability was less than 1% of the error prediction. For the upper and lower percentiles for each coefficient a similar ratio can be calculated. The numbers represent the median value over all coefficients, since this ratio tends to be very large for a few insignificant coefficients that are nearly zero. Again we see that the variability decreases with a larger number of bootstrap samples.

In the following subsections, 1000 bootstrap samples are used when estimating prediction errors and 2000 bootstrap samples are used when estimating confidence intervals. The latter estimates are

Table 1. Sampling variability when using different numbers of bootstrap samples. Estimates obtained for *ortho*-dinitrobenzene with four latent variables

	Prediction error	Lower percentile	Upper percentile
200	0·0071	0·073	0·125
500	0·0045	0·049	0·079
1000	0·0031	0·033	0·047
2000	0·0016	0·021	0·033
5000	0·0014	0·008	0·018

less accurate than the error estimates, but it is felt that in the current application this is of minor importance.

4.2. Optimal number of latent variables

First of all we assess the performance of the bootstrap in cases where cross-validation is usual in chemometrics. An example is the estimation of the optimal number of latent variables in a PCR or PLS regression. Usually, the number of variables in which a (local) minimum in the estimated prediction error is reached is selected as being optimal. The prediction error can be estimated by cross-validation, but also by the three bootstrap methods described earlier. In Table 2 the results are given for each of the nitrobenzenes. Calibration was performed for each compound separately. The optimal number of latent variables was located at the first local minimum or, if no local minimum was present, at the point where an extra latent variable contributed least to the reduction of error.

Interestingly, the number of factors selected differs for the different methods. This is partly due to the nature of the data set, which requires extra factors to model the non-linearities. Another reason can be seen in Figure 3. There the estimated prediction error is plotted against the number of latent variables for all methods employed. The same general trends can be found with all methods, but the bootstrap curves are smoother. Therefore local minima are less likely to occur, which influence the selection of the optimal number of factors, e.g. in the case of *ortho*-DNB. Estimating the prediction error with the 0·632 estimator for nine latent variables would yield 0·69, in agreement with the value found with cross-validation. In some cases no local minimum can be discerned, such as in the case of *para*-DNB (0·632 estimator). There the number of latent variables is selected where the error would decrease least when adding another latent variable.

Table 2. Optimal number of latent variables and estimated errors. Errors are in 10^{-4} mol l^{-1}. Methods: leave-one-out cross-validation (CV, equation (4)), simple bias-corrected bootstrap (BCS, equation (6)), and the 0·632 bootstrap estimator (equation (7))

	ortho-DNB		*meta*-DNB		*para*-DNB		NB	
	# LVs	Error	# LVs	Error	# LVs	Error	#LVs	Error
CV	9	0·70	9	0·85	10	0·81	2	1·70
BCS	12	0·63	9	0·85	10	0·75	5	1·72
0·632	12	0·65	9	0·86	10	0·78	5	1·72

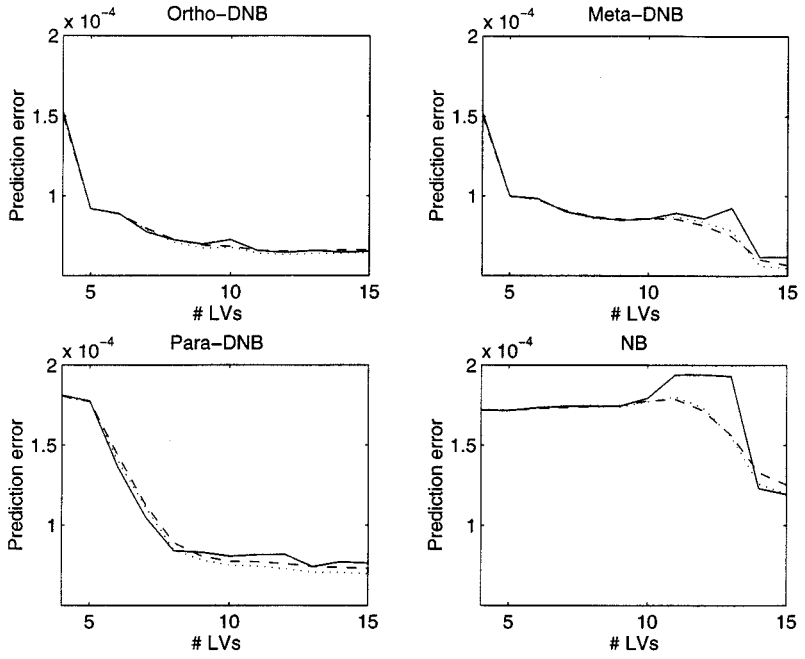

Figure 3. Prediction error estimation for four nitrobenzenes: full line, leave-one-out cross-validation; dotted line, bias-corrected simple bootstrap; broken line, 0·632 bootstrap. Error values are given in 10^{-4} mol l^{-1}

It is interesting to look at the optimism parameter found with the BCS estimator (not shown). All three disubstituted nitrobenzenes show a local minimum in this optimism at nine latent variables, while for *ortho*- and *meta*-DNB another one is seen at five latent variables. This is more or less in agreement with the number of latent variables selected; in the cases of *ortho*- and *meta*-DNB a sharp drop in prediction error is observed on inclusion of the fifth latent variable, whereas this drop is not so sharp in the *para*-DNB case. Therefore this optimism could serve as an extra indicator for the number of significant latent variables.

Cross-validation yields a local minimum at two latent variables in the case of nitrobenzene, whereas at least four would be expected because of the four compounds in the data set. However, the error continues to increase until 14 latent variables are incorporated in the model; then the error suddenly drops. This behaviour can also be seen to some extent in the other compounds. Fourteen factors, on the other hand, seem an unreasonably large number for this data set. The estimated prediction errors for nitrobenzene are very large, especially when one compares the prediction errors with the standard deviation of the nitrobenzene concentrations in the data set: $1{\cdot}89\times10^{-4}$ mol l^{-1}. This behaviour was also found in earlier investigations.[12] The regression is hardly better than estimating the concentration with the mean value.

The BCS and 0·632 estinmators agree almost exactly; the 0·632 estimator has slightly higher error estimates, indicating the slight downward bias present in the BCS estimator. The 0·632+ estimator gives the same estimates as the 0·632 estimator except in cases where overfitting occurs; the correction causes the estimates to be higher in that case. Here no overfitting was apparent in any case and therefore the results of the 0·632+ estimator are not shown. Closer inspection showed that the bootstrap error term ε_0 in all cases was lower than the simple error estimate err(x, $\hat{F}$), causing the variable weight of ε_0 in the 0·632+ estimator to be set to its maximum value of 0·632. Therefore the 0·632 and 0·632+ estimators yield the same estimates.

This is not what one would expect but is in agreement with the results reported earlier,[12] where the data were split into separate training and test sets. There the error in the test set was almost always smaller than the error in the training set. When looking closer at the cross-validation errors, we see that the mean error is heavily influenced by a few large residuals. Taking the median of the cross-validation errors instead of the mean *would* result in values for ε_0 smaller than err(x, $\hat{F}$).

The number of significant factors is equal in all cases, except when only a very shallow local minimum is found with some methods. A possibility to avoid taking too many factors into account is to construct confidence intervals around the prediction error estimates and select the number of factors after which the error does not decrease significantly. This could be done with bootstrap methods, of course, but it will take some computing time.

4.3. Variable selection

Regression methods such as PCR yield a vector of coefficients $\hat{\beta}$ that can be used to predict new data:

$$\hat{\beta} = \hat{\mathbf{P}}\hat{\mathbf{Q}} \tag{13}$$

$$\hat{Y}_{\text{new}} = \hat{\beta} X_{\text{new}} \tag{14}$$

where **P** and **Q** are the principal component X- and Y-loadings respectively. However, in many cases the prediction is hampered by the inclusion of variables that do not contain relevant information. The calibration can be improved by excluding these variables from the data set. One approach was recently described by Garrido Frenich *et al.*[14] They compared the results of the calibration in which variables with coefficients $\hat{\beta}$ lower than a certain cut-off value were removed. By doing so for several cut-off values, they finally arrived at an optimal set.

However, it is not so much the absolute value of a coefficient that determines whether that coefficient (and variable) is significant, but more the variation in that coefficient. We constructed 90% confidence intervals around the coefficients using the percentile and BC_α methods. This was done for several numbers of latent variables. Next, all coefficients for which the confidence interval included zero were removed from the data set and the prediction error if the reduced data set was estimated using the 0·632 estimator. The results are gathered in Tables 3 and 4 for the percentile and BC_α methods respectively. It is clear that the percentile method selects fewer variables than the BC_α method. In this case the predicted confidence intervals for the percentile method appear to be larger than those for the BC_α method. The better coverage properties of the latter method are also reflected by the slightly smaller prediction errors found using the models that utilize the variables selected by the BC_α method. In both the number of selected variables and estimated prediction errors the same trends can be observed, however.

In Figure 4 the significant variables for the four compounds are indicated, as found with the two methods. Negative loadings are indicated in white, insignificant (zero) loadings in grey and positive loadings in black. As can be seen, both methods essentially select the same variables, each compound having a distinct set of significant variables. This, of course, reflects the electrode potentials at which the different electrochemical reactions occur. The significant variables selected by the BC_α method appear to lie somewhat more scattered over the complete range than those selected by the percentile method. The somewhat lower prediction errors resulting from regression using the variables selected with the BC_α method imply that the extra variables do have a beneficial effect in the construction of the calibration model.

The reason for the bad prediction of the nitrobenzene now is clear: only a few variables are significant and these are heavily correlated with significant values for the *ortho*-DNB prediction. Only

Table 3. Number of significant variables as determined with percentile method and prediction error estimates (0·632 estimator) for varying number of latent variables. Error values are given in 10^{-4} mol l^{-1}

	ortho-DNB		*meta*-DNB		*para*-DNB		NB	
# LVs	# vars	Error	# vars	Error	# vars	Error	# vars	Error
4	147	0·80	204	1·01	151	1·63	203	1·68
5	176	0·82	133	0·99	123	0·98	83	1·66
6	117	0·73	113	0·90	73	0·93	48	1·48
7	101	0·95	98	0·70	87	0·94	24	1·73
8	119	0·62	124	0·78	99	0·91	23	1·71
9	132	0·63	121	0·69	124	0·94	21	1·62
10	128	0·63	100	0·58	118	0·73	20	1·67
11	113	0·63	63	0·51	95	0·69	14	1·75
12	104	0·65	51	0·52	88	0·61	21	1·41
13	97	0·66	48	0·52	80	0·62	48	1·21
14	93	0·67	66	0·51	69	0·62	71	1·16
15	87	0·67	81	0·47	68	0·77	87	1·19

a small region around variables 130–150 seems to be reasonably unique for nitrobenzene. For the BC_α method, Figure 5 shows the coefficients and 90% confidence intervals for a model with eight latent variables (*ortho*-DNB). Likewise, Figures 6–8 display the coefficients in models with six, seven and seven latent variables for *meta*-DNB, *para*-DNB and nitrobenzene respectively. These models have the smallest prediction errors as estimated with the 0·632 estimator.

It is clear that the reduction of the number of variables has a positive effect on the quality of the calibration. In all cases the minimum in the prediction error is found with fewer latent variables, yielding an even more parsimonious model. In general, approximately half of the variables do not contain relevant information. This is hardly surprising, since only the jumps in the voltammograms contain relevant information and the variables are highly correlated. This is especially clear in the case of nitrobenzene.

Table 4. Number of significant variables as determined with BC_α method and prediction error estimates (0·632 estimator) for varying number of latent variables. Error values are given in 10^{-4} mol l^{-1}.

	ortho-DNB		*meta*-DNB		*para*-DNB		NB	
# LVs	# vars	Error	# vars	Error	# vars	Error	# vars	Error
4	173	0·81	218	1·04	194	0·84	200	1·68
5	182	0·82	144	0·96	133	0·95	84	1·66
6	123	0·88	136	0·84	161	0·78	58	1·61
7	129	0·75	144	0·87	125	0·69	23	1·36
8	142	0·61	147	0·86	136	0·76	32	1·70
9	171	0·65	134	0·83	165	0·88	50	1·66
10	156	0·64	122	0·75	160	0·71	17	1·52
11	132	0·62	105	0·69	113	0·73	0	—
12	120	0·64	105	0·60	98	0·69	97	1·72
13	115	0·65	87	0·47	87	0·79	89	1·43
14	104	0·66	111	0·62	90	0·72	98	1·20
15	106	0·66	117	0·58	94	0·69	112	1·22

5. CONCLUSIONS AND OUTLOOK

In this paper we have drawn attention to applications of the bootstrap paradigm in chemometrical practice. It is shown that bootstrap methods are not only an alternative to cross-validation but can also be used in other areas such as variable selection. A big advantage of bootstrap methods is that they can be applied to the estimation of any statistic, no matte how complicated the calculations. Furthermore, it is possible to assess the accuracy of bootstrap estimates by jack-knifing or, indeed, a second bootstrap.[5] The main disadvantage of bootstrap methods is the amount of calculation required, but with the ever-increasing speed of hardware, most problems would be easily solvable in a reasonable time.

The procedures described in this paper are also applicable to other regression procedures. The extension to PLS and non-linear PCR and PLS variants is straightforward, since in this case only the coefficient vector and the prediction error were estimated using bootstrap methods. Hastie and

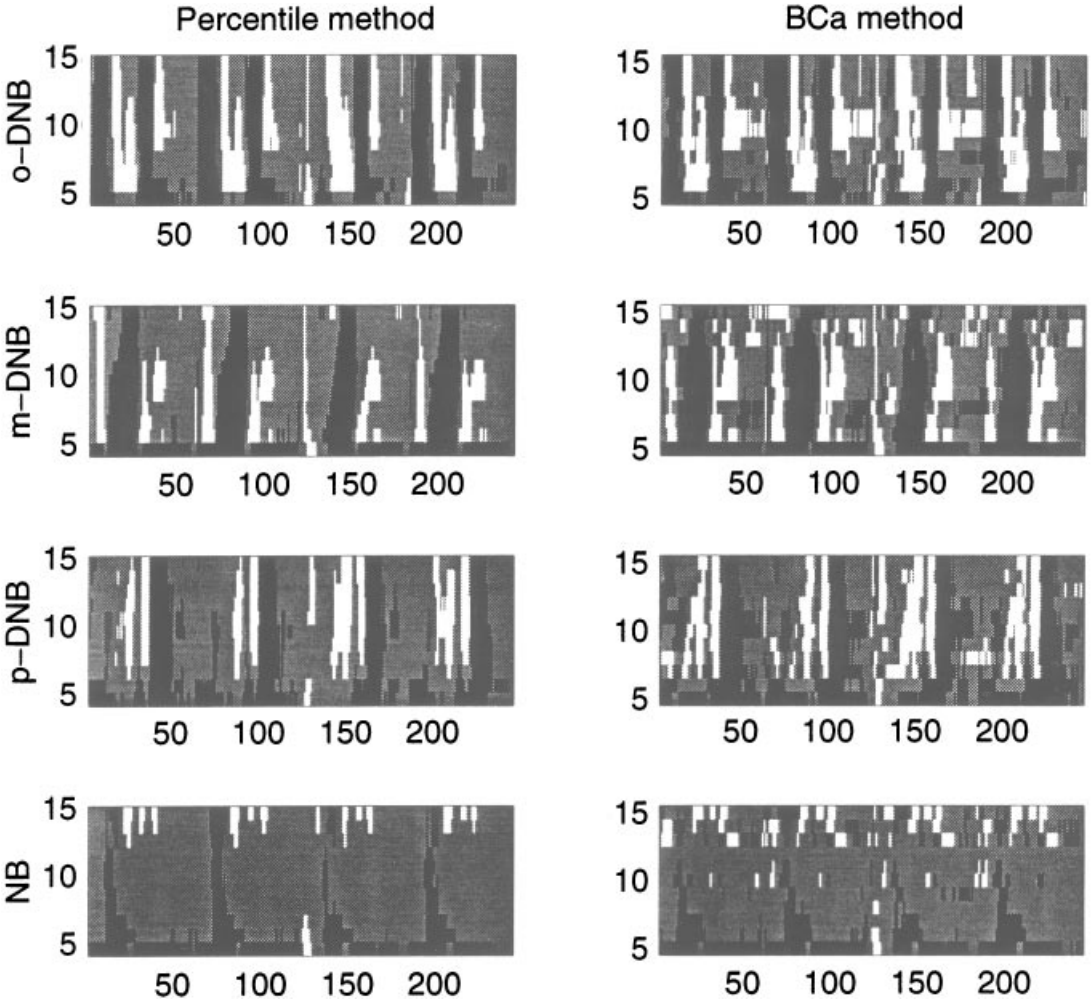

Figure 4. Significant coefficients: vertical axis, number of latent variables; horizontal axis, measured variables; black, significant negative; grey, not significant; white, significant positive

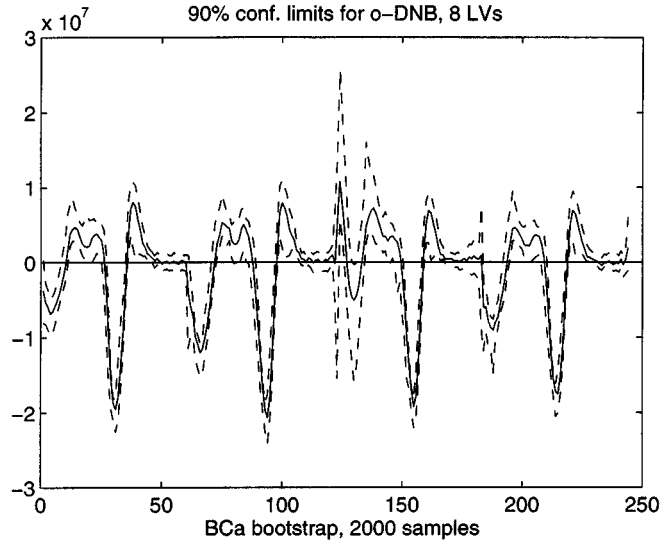

Figure 5. Confidence intervals for *ortho*-DNB coefficients

Tibshirani present an example in which they apply the bootstrap on projection pursuit regression.[15] They are able to show that whereas the first direction of the projection pursuit model is quite stable, the second has a large variability. Therefore only the first should be incorporated in the model. Also, it is interesting to see whether bootstrap methods can be used in the validation of neural networks. Tibshirani[16] uses a normal bootstrap procedure in which a model is repeatedly built on bootstrap samples, similar to the regression examples described in this paper. It is probably not advisable to use bootstrap methods to assess the variability of network weights, since the information is distributed throughout the network. Many different weight conformations may yield the same model.

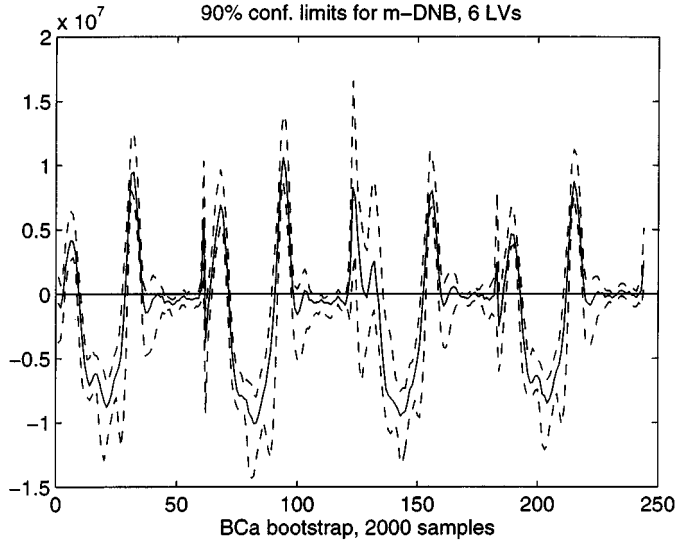

Figure 6. Confidence intervals for *meta*-DNB coefficients

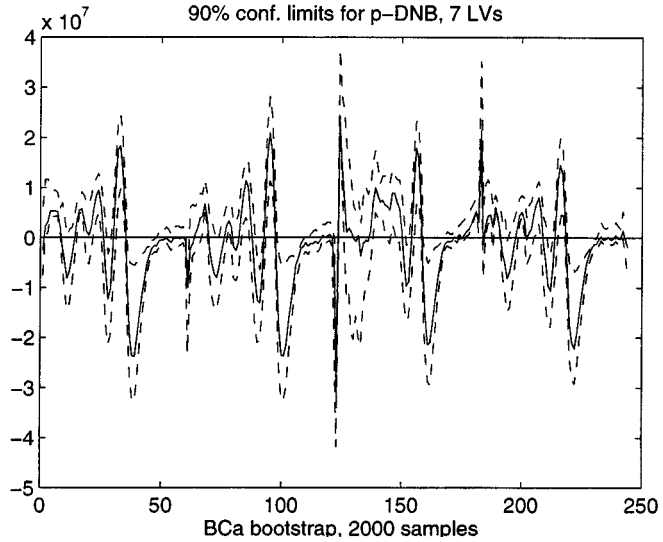

Figure 7. Confidence intervals for *para*-DNB coefficients

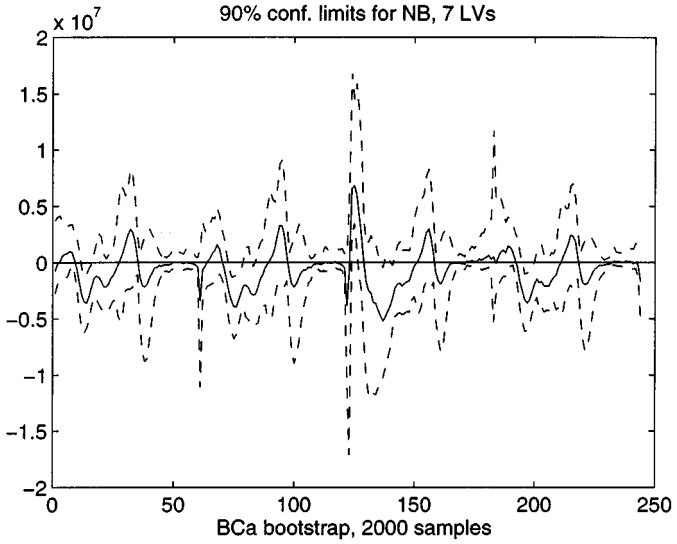

Figure 8. Confidence intervals for NB coefficients

ACKNOWLEDGEMENTS

The authors would like to thank Hein Putter for his expert advice concerning the bootstrap, Eduard Derks for his comments on the manuscript and Jo Simons for the data.

REFERENCES

1. B. Efron, *J. Am. Stat. Assoc.* **78**, 316 (1983).
2. B. Efron and R. Tibshirani, *Tech. Rep. 176*, Department of Statistics, Stanford University (1995).
3. B. Efron, *Ann Stat.* **7**, 1 (1979).

4. B. Efron and R. Tibshirani, *Stat. Sci.* **1**, 54 (1979).
5. B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, New York (1993).
6. C. Léger, D. N. Politis and J. P. Romano, *Technometrics*, **34**, 378 (1992).
7. P. L. Bonate, *Anal. Chem.* **65**, 1367 (1993).
8. R. Henrion, G. Henrion, K. Szukalski, I. Fabian, A. Thiesies and P. Heininger, *Fresenius J. Anal. Chem.* **340**, 1 (1991).
9. P. K. Hopke, C. L. Li, W. Ciszek and S. Landsberger, *Chemometrics Intell. Lab. Syst.* **30**, 69 (1995).
10. T . Roy, *J. Chemometrics*, **8**, 477 (1994).
11. R. A. Lodder and G. M. Hieftje, *Appl. Spectrosc.* **88**, 1500 (1988).
12. R. Wehrens and W. E. Van der Linden, *Anal. Chim. Acta* **334**, 93 (1996).
13. H. Martens and T. Naes, *Multivariate Calibration*, Wiley, New York (1989).
14. A. Garrido Frenich, D. Jouan-Rimbaud, D. L. Massart, S. Kuttatharmmakul, M. Martinez Galera and J. L. Martinez
Vidal, *Analyst*, **120**, 2787 (1995).
15. T. Hastie and R. Tibshirani, *Ann. Stat.* **13**, 502 (1985).
16. R. Tibshirani, *Tech. Rep.*, Department of Statistics, University of Toronto (1995); *Neural Comput.* in press.

