A Comparison of Methods for Estimating Prediction Intervals in NIR Spectroscopy: Size Matters

Remco R. Bouckaert*, Eibe Frank, Geoff Holmes, Dale Fletcher

Computer Science Department, Waikato University, Private Bag 3155, Hamilton, New Zealand

Abstract

In this article we demonstrate that, when evaluating a method for determining prediction intervals, interval size matters more than coverage because the latter can be fixed at a chosen confidence level with good reliability. To achieve the desired coverage, we employ a simple non-parametric method to compute prediction intervals by calibrating estimated prediction errors, and we extend the basic method with a continuum correction to deal with small data sets.

In our experiments on a collection of several NIR data sets, we evaluate several existing methods of computing prediction intervals for partial least-squares (PLS) regression. Our results show that, when coverage is fixed at a chosen confidence level, and the number of PLS components is selected to minimize squared error of point estimates, interval estimation based on the classic ordinary least-squares method produces the narrowest intervals, outperforming the U-deviation method and linearisation, regardless of the confidence level that is chosen.

Keywords: NIR, prediction interval, PLS regression, experimental design

1. Introduction

In [1], Zhang and Garcia-Munoz made a case for the importance of considering prediction intervals when performing analysis of NIR data using PLS, and several methods were compared. However, when evaluating the quality of prediction intervals, [1] only considered observed coverage for a given user-specified confidence level, that is, the proportion of spectra in which the observed target value (i.e. end point) is within the prediction interval. For example, when a 95% prediction interval is desired, the criterion for acceptability of an interval

*Corresponding author

Email addresses: remco@cs.waikato.ac.nz (Remco R. Bouckaert), eibe@cs.waikato.ac.nz (Eibe Frank), geoff@cs.waikato.ac.nz (Geoff Holmes), dale@cs.waikato.ac.nz (Dale Fletcher)

estimator is that for *at least* 95% of all spectra the target value is within the predicted interval. Based on this criterion alone, it was found that several methods produce satisfactory intervals.

However, the size of the interval estimates is obviously also of significant concern in real-world applications and this criterion was not considered in the evaluation in [1]. In this article, we evaluate interval estimators based on interval size. It is clear that achieving a desired coverage level is crucial, but, assuming it can be obtained reliably, methods can be compared based on the size of the intervals produced. We present a simple non-parametric method for calibrating prediction intervals to achieve a desired level of coverage. Based on this simple method, a specific coverage level can be achieved reliably on the data we investigated, and interval size then becomes the primary issue of concern.

In our experimental set up, we do not use a single train/test split to train and evaluate an interval estimator because this method does not provide any information about the sensitivity of the performance estimates regarding the particular training and test sets that were chosen. Also, single train/test splits make replication of results, an important issue in any scientific undertaking, hard to accomplish since a slightly different split can result in significantly different outcomes [2]. Instead, we use repeated cross-validation in all our experiments, which makes it possible to test for statistically significant differences in observed performance estimates and provides a much higher replicability than single train/test splits.

In the following section, we describe the theory of some common methods used for PLS-based prediction intervals for NIR. Section 3 describes experimental design, including data sets and details of the experiments. Section 4 contains the results of the experiments and discusses the implications, ending in a short concluding summary.

2. Theory

In this article, we treat NIR analysis as a regression problem. We assume that the NIR machine produces a spectrum that (after appropriate filtering and smoothing) can be represented by a vector X, and we are interested in determining a quantity of interest y, such as the freezing temperature of fuel or the nitrogen content in soil samples. To build a predictive model, we gather a set of n training samples (X_i, y_i), $i \in [1...n]$, where $\mathbf{x} = \{X_1, \ldots, X_n\}$ are the vectors representing the spectra, and $\mathbf{y} = \{y_1, \ldots, y_n\}$ are the accompanying target values. The regression problem consists of predicting a value y^* for a new spectrum X^*.

2.1. Prediction intervals for PLS regression

In ordinary least-squares regression, we predict y^* by finding the regression coefficient vector β that fits

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

such that the squared length of the residuals vector $|\epsilon|^2 = \sum_{i=1}^{n} \epsilon_i^2$ is minimized. Here, and in the remainder of the article, we assume that $\mathbf{y}$ and $\mathbf{X}$ are centered so that their mean value is zero. The β that minimizes the squared error on the training data is given by $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

Minimizing the squared error on the training data does not necessarily yield the best linear regression model because and ordinary least-squares regression can over-fit the training data. The error on new data is what is important. In NIR analysis in particular, it is common to filter the spectra first by performing dimensionality reduction using partial least squares (PLS), which applies a linear transformation of $\mathbf{X}$ based on a weight matrix $\mathbf{R}$ that is chosen to minimize the least squares error (see [3] for further technical details). In PLS regression, the coefficient vector used for prediction is then estimated as follows:

$$\hat{\beta} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{X}^T\mathbf{y}$$

Note that PLS regression is equivalent to ordinary least-squares regression when all PLS components are used in $\mathbf{R}$. In practice, it is common to choose an appropriate number of components by measuring predictive performance on validation data (e.g. using cross-validation) to combat over-fitting.

The focus on this article is on interval estimates rather than point estimates. The literature contains several methods for computing prediction intervals using PLS regression (see [1]). We now briefly review the ones we evaluate in our empirical comparison.

2.1.1. Ordinary least squares

After filtering through PLS, the original n training vectors $\mathbf{X}$ are mapped into n filtered vectors $\mathbf{R}$. We can apply least squares regression on the resulting data set $\mathbf{R}$, yielding the above expression for the coefficient vector. In the same manner, the ordinary least-squares (OLS) method for computing prediction intervals can be applied to the PLS-filtered data. Prediction intervals are then estimated as

$$\{\hat{y}^* - t_{\alpha/2,n-df}\, s, \hat{y}^* + t_{\alpha/2,n-df}\, s\} \tag{1}$$

where $\hat{y}^*$ is the OLS prediction for spectrum X^*, α is the significance level for the interval, $t_{.,.}$ is the Student-t distribution, n is the number of spectra in the training set, and df represents the degrees of freedom.

In this expression, we can choose α based on the desired confidence level, and, for ordinary least squares, the value of df is the number of coefficients. Then, s is estimated as

$$s = \sigma\sqrt{1 + h_0 + \frac{1}{n}}$$

where σ the standard error on the training data, based on the df value, and

$$h_0 = X^{*T}(\mathbf{X}\mathbf{X}^T)^{-1}X^*$$

2.1.2. U-deviation

The U-deviation method is an empirically obtained formula used in Unscrambler, a popular chemometrics software package. We refer to the manual [4] (page 342) for a detailed description of the method. Suppose we want to predict the value for spectrum X^*. Let $h_0 = X^{*T}(\mathbf{X}\mathbf{X}^T)^{-1}X^*$. The standard deviation is estimated using

$$s = \sqrt{\frac{\hat{\sigma}_y}{2}\Big(h_0 + \frac{\sigma_{X^*}}{\sigma_{\mathbf{X}}} + \frac{1}{n}\Big)}$$

where $\hat{\sigma}_y$ is the residual variance of the prediction y on the training data, $\sigma_{\mathbf{X}}$ is the average variance in the training data and σ_{X^*} is the average variance in the sample spectrum.

It has been found that this approach can result in intervals that are smaller than is desirable, leading to lower than expected coverage. A version that uses a correction factor for the number of PLS components performs better [5, 6]

$$s = \sqrt{\hat{\sigma}_{test}\Big(1 - \frac{A+1}{n}\Big)\Big(h_0 + \frac{\sigma_{X^*}}{\sigma_{X,test}} + \frac{1}{n}\Big)}$$

where A is the number of PLS components, $\hat{\sigma}_{X^*}$ the variance on training data, and $\sigma_{X,test}$ on test data. This is the version we use in our empirical evaluation. To obtain the interval, s is substituted in Equation (1).

2.1.3. Linearization

Because the PLS-filtered data may contain non-linear components, it can be argued that β is not a linear function of $\mathbf{y}$ and a better approximation is to use a Taylor expansion around the data $(\mathbf{X}_0, \mathbf{y}_0)$. For pragmatic reasons, only the first term in the Taylor expansion is used, and β is estimated as

$$\hat{\beta}(\mathbf{y}) = \hat{\beta}(\mathbf{y}_0) + \mathbf{J}(\mathbf{y} - \mathbf{y}_0)$$

where $\mathbf{J}$ is the Jacobian matrix of the elements of $\hat{\beta}$ wrt the elements of $\mathbf{y}$. We refer to [7, 1] for details and references. To calculate the prediction error, the variance can be approximated by taking the variance on both sides of the above equation, giving $var(\hat{\beta}) \approx \mathbf{J}\mathbf{J}^T\sigma^2$, and the prediction interval can be obtained with $s = \sigma\sqrt{\mathbf{J}\mathbf{J}^T}$ substituted in Equation (1).

One of the drawbacks is the extra computational effort required to perform this Taylor expansion. In fact, the expansion took too long to make repeated cross-validation experiments feasible. Denham [8] designed a method that is computationally more efficient than the standard Taylor expansion, but unfortunately it was still too computationally expensive for our experiments. However, Serneel et al. [7] designed an algorithm that efficiently calculates the Jacobian matrix that is required and this is what we used in our experiments.

2.2. Prediction Interval by Error Estimation

Above, we briefly reviewed some methods for computing prediction intervals in the case of PLS regression, which are all based on the assumption that the underlying distribution of errors is normal. Under this assumption, the prediction

interval can be estimated using Eq (1). An essential component for achieving reliable coverage is the number of degrees of freedom *df*. In [1] various methods were examined, for example, the naive approach which sets *df* equal to the number of PLS components, and the so-called pseudo degrees of freedom, which incorporate the model fit and predictive error. It was found that the so-called generalized degrees of freedom (GDF), proposed in [9], produces acceptable results. GDF requires choosing a method for a weighted estimate of the variance, requires performing a number of iterations, and assumes normally distributed error.

We propose and evaluate a much simpler alternative approach to achieve acceptable coverage. Given some estimate of prediction error, such as the value of s in Section 2.1.1, there is a very simple, non-parametric way to determine prediction intervals empirically. This is done by using the distribution of observed prediction errors to estimate a scaling parameter that can then be used to rescale predicted errors so that they yield prediction intervals with a specified coverage.

In addition to avoiding the need for assumptions regarding the distribution of errors, a further advantage of this approach is that the technique is independent of the regression method applied and it works for PLS regression in the same way as for any other regression method, for instance Gaussian processes or neural networks.

We now describe the details of the specific method we apply. For every spectrum X_i $(1 \leq i \leq n)$ in the calibration data, we produce a prediction and measure the observed error e_i, which is the absolute value of the difference between the predicted value and the actual target value in the data. We also calculate the predicted error p_i, which could, for example, be s from above. The value $\alpha_i = e_i/p_i$ represents the factor required to scale the predicted error to the actual error. In order to guarantee a certain coverage (i.e. prediction intervals such that P percent of the actual target values are in the prediction interval) we need to find the value α such that P percent of the α_i are lower and the remainder higher than α. This value is found by sorting the α_i, denoted as α_i^s. Then, out of the n sorted α values we take $\alpha = \alpha^s_{Pn/100}$ as the multiplication factor for the error. Thus, the prediction interval for a spectrum with prediction m and predicted error p is $[m - \alpha p, m + \alpha p]$.

However, for spectra with target values close to the extremes of the target range such a prediction interval can occasionally exceed the target range by a large margin, resulting in unnecessarily large intervals. So, during training, the range of the target is measured. Let c_{min} be the lowest target value observed and c_{max} be the highest. Then we calculate the actual prediction interval using

$$[\max(c_{min}, m - \alpha p), \min(c_{max}, m + \alpha p)]$$

Note that we can not use the training data, used to train the predictive model, to obtain the error values e_i. Independent calibration data must be used instead. Hence, to obtain the actual errors e_i and the predicted errors, we use k-fold cross validation, splitting the full training set into k equally sized subsets,

and make predictions on each one of these subsets based on a predictor trained on the remaining subsets.

In practice, we observed that actual coverage after calibration can in some cases be marginally lower than the pre-specified confidence level.This is due to the following effect. With small data sets, the index $Pn/100$ from which we obtain $\alpha = \alpha^s_{Pn/100}$ can deviate from the desired value due to rounding errors (since $Pn/100$ is not always an integer). To address this problem, we apply a continuum correction. Let $f = Pn/100 - \lfloor Pn/100 \rfloor$. We can apply a continuum correction by calculating α as follows

$$\alpha = (1 - f)\alpha^s_{Pn/100} + f\alpha^s_{Pn/100+1}$$

Note that the calibration method for interval estimation we just described does not depend on any particular prediction method. In fact, different methods can be used to obtain point estimates to calculate the e_i on the one hand, and the predicted error values p_i on the other hand.

3. Experimental design

To evaluate the different interval estimation methods discussed above, we performed a large number of experiments with real-world NIR data. All results were obtained using m times k-fold cross validation. Here, the data set is randomly split into k approximately equal parts and every part is used as test set for a model trained on the remaining $k - 1$ parts. This process is repeated m times and performance measures (like mean squared error) are averaged over all test sets. Performing 10 times 10-fold cross validation balances accuracy, replicability and computational effort [2]. Note that it generates the same number of samples as Monte Carlo cross-validation [10] with 100 repetitions.

3.1. Data description

Experiments across multiple domains are necessary to establish reliable results. To this end, we considered four NIR data sets in our experiments:

- *Diesel data*: Publicly available data[1] of NIR spectra of diesel fuels along with various properties of those fuels. Table 1 provides an overview of the various properties. There are a total of 784 spectra, but not all properties (i.e. target values) were measured for all spectra. The second column in Table 1 shows the number of spectra for which a measurement is available. For each property we created a data set containing only those spectra for which the property was measured.

- *Corn data*: The corn data set[1] is a very small data set of just 80 spectra with measurements of moisture, oil, protein and starch content. Each spectrum has 700 values. Three different NIR instruments were used, but only spectra of the first instrument were used in the experiments.

[1] Available from `http://software.eigenvector.com/Data/index.html`.

Table 1: Data description.

Property	# spectra	Description
Diesel data		
BP50	395	boiling point at 50% recovery
CN	381	cetane number (like octane number only for diesel)
FLASH	395	flash point temperature of the fuel
FREEZE	395	freezing temperature of the fuel
TOTAL	395	total aromatics, mass %
VISC	395	viscosity, cSt, at 40 degree C
Corn data		
moisture	80	moisture content of corn
oil	80	oil content of corn
protein	80	protein content of corn
starch	80	starch content of corn
Grass data		
carbon	141	carbon content of grass
FERT	141	fertilizer level of grass
nitrogen	141	nitrogen content of grass
sulfur	141	sulfur content of grass
Soil data		
Lactic	255	lactic acid content
Storig	414	soil data
SS	895	soil data
OMD	1010	soil data

- *Grass data*: The grass data set from a 1998 competition contains 141 spectra with powdered (dry ground) grass samples for which carbon, nitrogen, and sulfur content were determined, and level of fertilization (0, 50, 250 and 500 ppm of nitrogen) was recorded.[2] There are multiple measurements for carbon, nitrogen and sulfur content, and only the last of the measurements was used in this experiment. The others were very close. Each spectrum consists of 1050 recorded values.

- *Soil data*: Soil sample spectra with targets called lactic, storig, SS, and OMD were produced with an NIR machine that outputs 700 values per spectrum. This spectrum was Savitzky-Golay smoothed with a window size of 15 and down-sampled (every 4th wavenumber), resulting in a 171 value spectrum. All spectra were measured independently from each other, and data set sizes vary from 255 to 1010 spectra (see Table 1 for details).

[2]See `http://kerouac.pharm.uky.edu/asrg/cnirs/shootout1998/shootout1998.html` for details.

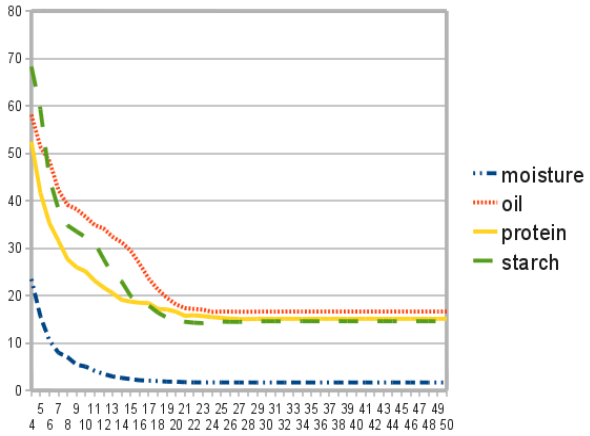

Figure 1: Root relative squared error (on y-axis) for 4 to 50 PLS components (on x-axis) with 10 times 10 fold cross validation (i.e. based on 100 data samples) for corn data.

3.2. Tuning PLS

In a first preliminary experiment, an appropriate number of PLS components for each group of data sets was determined by applying PLS regression with 10 times 10-fold cross-validation, with the number of components varying between 4 and 50. The final number was selected by taking the number of components with the minimal root relative squared error (RRSE), averaged across the data sets concerned. The RRSE is based on the root mean squared error, but scaled with the average error obtained by predicting the mean of the target value. This measure is useful because it shows how much the predictor improves on predicting the average target value in the training data, and is expressed as percentage. An RRSE larger than 100% indicates that it would be better to simply use the average, while RRSE values close to 0% indicate the predictor performs well.

Figures 1 to 4 shows the observed RRSE for various amounts of PLS components, calculated as the average over 10 times 10 fold cross validation for the four data sets.

3.3. Performance of PLS-based interval prediction methods

Having selected parameters that yield satisfactory point estimation performance, we can now consider the performance of the different interval estimators. First, we evaluate the PLS-based methods regarding the size of their prediction intervals: U-deviation, ordinary least squares, and linearisation using Serneel's methods. Bootstrapping [11] and linearisation through Phatak's and Denham's methods [12] turned out to be too slow to be practical for evaluation on a larger scale, so these methods are not considered.

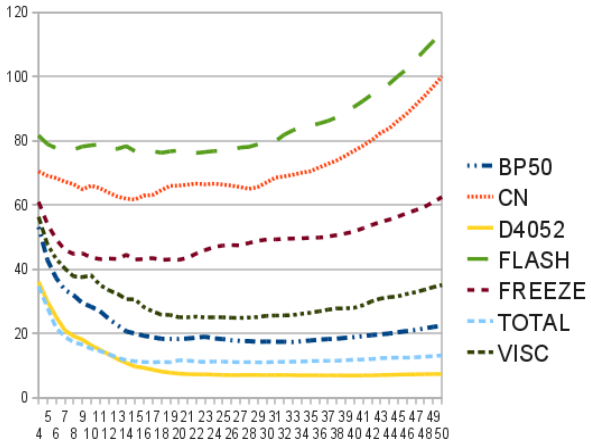

Figure 2: As Figure 1 for diesel data.

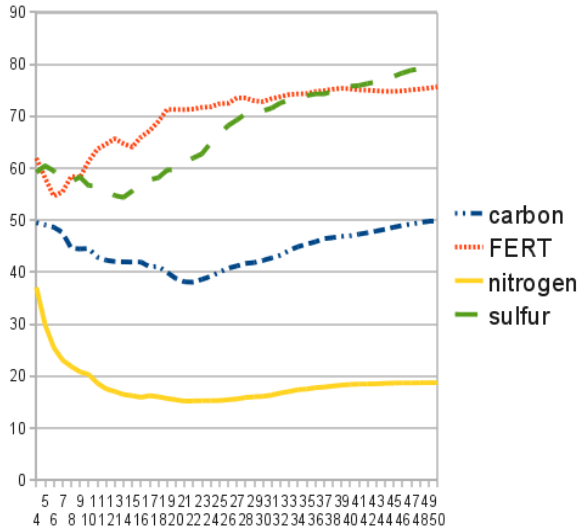

Figure 3: As Figure 1 for grass data.

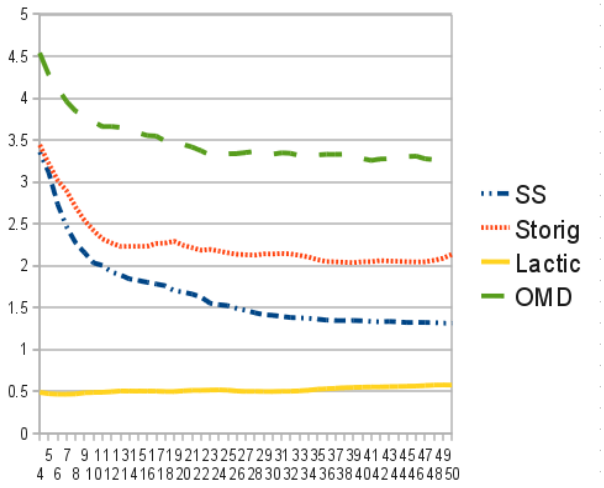

Figure 4: As Figure 1 for soil data.

To fairly compare interval size, we fixed the coverage at a pre-specified level. More specifically, we scaled the predicted intervals to achieve exactly 95% coverage of the test cases in a 10-fold cross validation experiment. This is possible because all methods considered first calculate an estimate of error $s_i$which they then scale into a prediction interval based on the Student-t distribution. The key observation is that this scaling factor is constant per training set. We can thus replace it by a scaling factor that is empirically determined to achieve the desired coverage level. Then, given that coverage levels are empirically fixed to achieve the desired level, we can fairly compare interval size.

More specifically, we consider the standard deviation s_i to be an error estimate for property y_i and pool all error estimates for a 10-fold cross-validation into a single set. Then, a multiplier α is calculated such that exactly 95% of the time $\alpha s_i > e_i$ where e_i is the observed residual of PLS regression. This multiplier is easily calculated by sorting e_i/s_i, giving a sorted sequence $q_1, \ldots, q_n$ and setting $\alpha = q_{\lceil 0.95n \rceil}$. Based on this multiplier, we can then calculate the average size of the prediction intervals as $\sum_{i=1}^{n} \alpha s_i$. We repeated the process ten times with different randomizations of the data—i.e. we performed 10-times 10-fold cross-validation—so that we can calculate both mean interval size and variance.

4. Results and discussion

We now discuss the experimental results obtained, first comparing the different PLS-based interval prediction methods based on their prediction sizes before moving on to evaluating the coverage of prediction intervals when adjusted by our calibration method.

4.1. Performance of PLS-based interval prediction methods

Table 2 shows average size for coverage fixed at 80%, 90% and 95%, for U-deviation, OLS, and linearisation using Serneel's method. We considered Denham's method for linearisation as well, but do not include results. Due to its computational complexity, on average, it took a week to perform the experiment on a single data set and only a few results were obtained. For those results that were obtained the outcomes were very close to those of OLS.

The main feature of the table is that OLS intervals tend to be smaller than intervals produced with the other methods. Furthermore, Serneels' method tends to result in large variability of the interval size. We suspect some numeric instability in the algorithm, since theoretically the outcomes should be very similar to those obtained using Denham's method, which were far less variable. Note that it is possible that this problem would not show up in a single train/test split: on examination we found that for some splits the method behaves quite well, but with 10-times 10-fold cross validation a large number of splits show very large error estimates. Note that, we used the original Matlab implementation from [7] for Serneels' method, which ensures this is not an implementation issue.

Also noteworthy is the size of intervals produced with the U-deviation method. It has been known that this method tends to produce intervals with lower than desirable coverage (see, e.g. [1]). Correspondingly, by fixing the coverage to a desired level, the interval size is larger than that of the OLS method for most instances, sometimes statistically significantly so at the 5% significance level (e.g. for diesel flash at 90% coverage). Hence, we can confirm there are problems with the method's performance compared to OLS prediction intervals.

4.2. Coverage

Where the previous experiment explored interval sizes, we now consider how reliably we can obtain desired coverage level using the error calibration method described in Section 2.2.[3] We consider all the data sets and set the desired coverage to 80%, 90% and 95%. Table 3 shows the observed coverage averaged over 10 times 10 fold cross validation. Let us consider the data sets one by one.

For the diesel data sets we used 20 fold cross validation for calibration, and the coverage is very close to the desired 90% and 95% levels and the only measurements below the desired level (total diesel for 95% coverage) is within a quarter of a percent of the desired level. Note that there are 395 instances, so a quarter of a percent equals one instance, in other words, on average less than one instance is not covered. The coverage for 80% desired coverage is very close to 80%, only exceeding the 80% level by a few percentage points. However, coverage never substantially decreases below the desired target coverage.

For the corn data sets, we found that coverage was considerably elevated when doing 10 fold cross validation for calibrating interval sized. This can be explained by the fact that corn data only contains 80 samples. Training takes

[3]The method was implemented in the Weka software [13] and is available from the first author on request.

Table 2: Size of intervals (normalized by range of target value) at 80%, 90% and 95% coverage for U-deviation method, standard linear regression, and linearization averaged over 10 times 10-fold cross-validation.

Property	Method	80%			90%			95%		
BP50	U-deviation	7.743	±	0.189	9.842	±	0.380	12.288	±	0.331
	OLS	7.025	±	0.220	9.693	±	0.268	12.524	±	0.320
	Linearized	8.316	±	1.203	10.736	±	1.737	13.723	±	2.338
CN	U-deviation	22.838	±	0.475	31.105	±	0.522	39.200	±	1.026
	OLS	20.610	±	0.301	27.116	±	0.620	37.153	±	0.987
	Linearized	36.151	±	43.166	47.250	±	58.035	61.034	±	71.782
FLASH	U-deviation	22.868	±	0.803	32.014	±	0.569	42.925	±	1.449
	OLS	20.560	±	0.585	28.903	±	0.885	38.282	±	0.710
	Linearized	32.649	±	26.177	46.076	±	39.073	56.531	±	43.861
FREEZE	U-deviation	16.353	±	0.342	22.984	±	0.818	29.279	±	0.586
	OLS	14.934	±	0.218	20.549	±	0.657	28.411	±	0.684
	Linearized	15.396	±	0.305	20.132	±	0.584	26.337	±	1.141
TOTAL	U-deviation	4.238	±	0.149	5.557	±	0.176	7.084	±	0.244
	OLS	3.858	±	0.087	5.297	±	0.137	6.436	±	0.222
	Linearized	4.129	±	0.463	5.444	±	0.634	6.889	±	0.802
VISC	U-deviation	11.242	±	0.202	15.403	±	0.462	19.942	±	0.648
	OLS	10.151	±	0.273	14.415	±	0.352	18.689	±	0.644
	Linearized	16.976	±	13.968	23.560	±	19.473	30.150	±	25.325
oil	U-deviation	10.250	±	0.992	13.106	±	0.653	15.323	±	1.166
	OLS	10.106	±	0.663	12.482	±	0.812	14.732	±	0.876
	Linearized	23.969	±	15.844	34.251	±	23.954	40.067	±	26.502
protein	U-deviation	9.130	±	0.922	11.762	±	1.531	15.079	±	1.789
	OLS	9.332	±	0.943	12.359	±	1.191	16.019	±	2.016
	Linearized	25.172	±	44.538	36.571	±	64.716	46.614	±	77.544
starch	U-deviation	8.255	±	1.132	12.017	±	1.517	15.059	±	0.987
	OLS	8.431	±	1.126	12.304	±	1.704	14.679	±	1.210
	Linearized	34.494	±	45.731	51.323	±	71.235	61.849	±	81.516
carbon	U-deviation	17.070	±	0.648	22.922	±	1.098	27.872	±	1.425
	OLS	16.230	±	0.476	20.974	±	1.022	27.968	±	2.343
	Linearized	20.987	±	3.873	26.637	±	4.811	33.761	±	5.999
nitrogen	U-deviation	9.925	±	0.435	13.108	±	0.558	15.196	±	0.588
	OLS	9.643	±	0.435	13.186	±	0.486	16.545	±	0.941
	Linearized	12.890	±	3.275	18.378	±	5.243	23.638	±	8.259
sulfur	U-deviation	28.716	±	1.599	40.335	±	1.756	49.652	±	2.403
	OLS	27.470	±	1.022	39.720	±	1.692	48.872	±	2.052
	Linearized	35.830	±	12.470	52.095	±	16.648	68.856	±	25.618
FERT	U-deviation	53.817	±	2.703	75.934	±	2.595	92.027	±	3.129
	OLS	50.902	±	1.361	69.797	±	1.923	83.339	±	2.430
	Linearized	50.821	±	0.934	69.681	±	2.043	82.719	±	2.438
Lactic	U-deviation	44.022	±	0.763	58.377	±	0.949	74.085	±	1.772
	OLS	36.873	±	1.065	51.110	±	0.951	62.604	±	1.417
	Linearized	36.873	±	1.029	51.142	±	0.857	62.619	±	1.633
Storig	U-deviation	13.738	±	0.331	21.472	±	0.557	27.535	±	0.762
	OLS	10.430	±	0.324	13.964	±	0.209	17.010	±	0.494
	Linearized	11.631	±	0.909	15.675	±	1.371	19.672	±	1.405
SS	U-deviation	8.140	±	0.087	10.869	±	0.223	13.378	±	0.232
	OLS	7.339	±	0.095	9.946	±	0.166	12.202	±	0.241
	Linearized	9.037	±	2.199	12.385	±	3.181	15.387	±	3.871
OMD	U-deviation	13.342	±	0.178	17.714	±	0.288	21.731	±	0.407
	OLS	12.401	±	0.105	17.018	±	0.224	21.693	±	0.305
	Linearized	86.365	±	101.433	120.270	±	140.579	152.178	±	177.830
Total	U-deviation	0		0	1					
	OLS	10		9	7					
	Linearized	1		2	3					

place on 90% of the data in a 10 fold cross validation experiment, which leaves just 72 instances. For the 10 fold cross validation for calibration, this means just 90% of 72, or just 65 instances are left. When increasing the number of folds, we observed that coverage gets closer to the desired levels. Table 3 shows results for 72 fold cross validation for calibration, which is the same as leave one out cross validation. Coverage is very close to the desired levels in all cases, considering one instance equals 1.25% of cases.

The grass data sets also showed slightly elevated coverage at 10 fold cross validation, which can be explained by the size of the data set. Table 3 shows results for 30 fold cross validation and again we found that coverage got close to desired coverage when increasing the number of folds, but never became substantially lower than desired coverage.

For the soil data, which has a considerable number of instances in the data sets, we found coverage very close to desired levels at 10 fold cross validation. Furthermore, the variance in the estimates shows a decreasing trend with increasing data set sizes. This experiment shows that the error estimation method produces acceptable coverage for prediction intervals, with increasingly accurate coverage when increasing computational effort and decreasing variance in coverage with increasing data set sizes.

5. Conclusions

A main goal of this article was to emphasize that interval size is an important consideration when evaluating methods for interval estimation: good coverage is a necessary requirement, but not a sufficient one. As a case study, we have evaluated several different methods for PLS-based interval estimation, and considered interval size at a fixed coverage level. We also showed that a simple non-parametric method can be used to obtain acceptable coverage levels. We can summarize our empirical findings and recommendations as follows:

- The U-deviation method for PLS-based prediction intervals yields larger intervals than OLS-based interval prediction. Therefore it is recommended to use OLS-based interval predictions rather than the U-deviation method.
- Interval prediction based on linearisation produces intervals of similar size compared to OLS. However, in our experiments we found that the computationally efficient method of Serneel is numerical unstable and should be used with caution. Other linearisation methods we have considered appear computationally too expensive to be practical.
- Simple non-parametric interval estimation by calibrating error estimates produces intervals with coverage close to the desired level of confidence. We recommend it as a pragmatic method for interval estimation.

We would like to stress that careful experimental design is necessary for evaluating the performance of interval estimators: tuning methods to a few

Table 3: Coverage of prediction intervals for PLS-based linear regression with the proposed error calibration method. The desired coverage is 80%, 90% and 95%. Average and standard deviation over 10 times 10 fold cross validation.

Diesel Data	80% Coverage			90% Coverage			95% Coverage		
BP50	81.79	±	6.10	90.52	±	4.97	95.03	±	3.65
CN	80.29	±	6.78	90.53	±	4.56	94.86	±	3.70
FLASH	80.93	±	6.52	90.20	±	4.94	95.13	±	3.62
FREEZE	80.38	±	6.43	90.70	±	4.98	95.26	±	3.96
TOTAL	80.64	±	5.90	90.05	±	4.49	94.78	±	3.28
VISC	80.07	±	6.85	90.04	±	4.80	95.14	±	3.56
Corn Data	80% Coverage			90% Coverage			95% Coverage		
moisture	82.00	±	13.80	91.63	±	10.21	95.25	±	7.70
oil	82.75	±	13.96	92.38	±	10.50	95.50	±	7.22
protein	79.63	±	15.86	90.75	±	10.15	95.38	±	7.25
starch	81.75	±	13.35	91.00	±	11.11	95.88	±	6.89
Grass Data	80% Coverage			90% Coverage			95% Coverage		
carbon	80.78	±	10.80	90.85	±	8.23	95.47	±	5.92
FERT	81.29	±	11.41	90.54	±	9.08	94.87	±	6.66
nitrogen	80.85	±	11.29	90.69	±	7.88	95.02	±	5.59
sulfur	80.83	±	11.02	89.95	±	8.60	94.77	±	6.46
Soil Data	80% Coverage			90% Coverage			95% Coverage		
Lactic	81.00	±	9.34	89.97	±	7.34	94.43	±	5.16
Storig	80.52	±	6.54	90.85	±	4.67	95.68	±	3.27
SS	80.15	±	3.91	90.28	±	3.42	95.10	±	2.47
OMD	80.20	±	4.17	90.02	±	3.16	95.09	±	2.39

data sets, perhaps based on a single train/test split in each case, can yield misleading results.

One way to get more reliable results is by averaging over repeated cross-validation estimates. By performing 10-times 10-fold cross-validation we found that Serneel's linearisation method can suffer from numerical instability. Applying this evaluation procedure balances accuracy of estimates, replicability of results, and computational effort.

It is also important to consider a broad range of data sets. Unfortunately, there are very few publicly available NIR data sets, mainly due to commercial sensitivities, which can make replication of results by other practitioners cumbersome or even impossible.

In future work, we aim to explore generalized regression methods that can deal with non-linearities in spectral data. The PLS component graph may give an indication that we are dealing with non-linearity of the data by showing an increase in observed error with an increase in the number of components after some minimum. Preliminary results indicate that Gaussian process regression yields promising performance compared to OLS regression, both in terms of the quality of the point estimates obtained and in terms of the quality of the corresponding prediction intervals.

Acknowledgments

Lin Zhang was helpful in providing hints and Matlab code for the implementation of methods presented in [1].

References

[1] L. Zhang, S. Garcia-Munoz, Chemom. Intell. Lab. Syst. 97 (2009) 152–158.
[2] R.R. Bouckaert, in: C.E. Brodley (Ed.), ICML, volume 69 of ACM International Conference Proceeding Series, ACM, 2004.
[3] S. de Jong, Chemom. Intell. Lab. Syst. 18 (1993) 251–263.
[4] T. Camo A/S, The unscrambler users guide, 1994.
[6] S.D. Vries, C.J.T. Braak, Chemom. Intell. Lab. Syst. 30 (1995) 239–245.
[5] M. Hy, K. Steen, H. Martens, Chemom. Intell. Lab. Syst. 44 (1998) 123–133.
[7] S. Serneels, P. Lemberge, P.V. Espen, J. Chemometrics 18 (2004) 76–80.
[8] M. Denham, J. Chemometrics 11 (1997) 39–52.
[9] J. Ye, Journal of the American Statistical Association 93 (1998) 120–131.
[10] Q.S. Xu, Y.Z. Liang, Chemom. Intell. Lab. Syst. 56 (2001) 1–11.
[11] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.
[12] A. Phatak, P. Reilly, A. Penlidis, Anal. Chim. Acta 277 (1993) 495–501.
[13] R.R. Bouckaert, E. Frank, M.A. Hall, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, J. Mach. Learn. Res. 11 (2010) 2533–2541.