Talanta 94 (2012) 301–307

Contents lists available at SciVerse ScienceDirect

Talanta

journal homepage: www.elsevier.com/locate/talanta

An improved ensemble partial least squares for analysis of near-infrared spectra

Yong Hu [a,*], Silong Peng [a], Jiangtao Peng [a], Jiping Wei [b]

[a] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, PR China
[b] Food Research Institute of Tianshili Group, Tianjin, 300410, PR China

ARTICLE INFO

Article history:
Received 24 November 2011
Received in revised form 19 March 2012
Accepted 21 March 2012
Available online 10 April 2012

Keywords:
Partial least squares
Bagging
Boosting
Creating diversity
Quantitative analysis

ABSTRACT

Traditional ensemble regression algorithms such as BAgging Partial Least Squares (BAPLS) and BOosting Partial Least Squares (BOPLS) do not perform very well in the data set that is relatively small or contaminated by random noise. To make the method robust and improve its prediction ability, inspired from bias–variance–covariance decomposition, we propose an improved ensemble partial least squares method based on the diversity. The new method is applied to quantitative analysis of Near InfraRed (NIR) data sets. A comparative study between the proposed method and other previous methods including BAPLS and BOPLS on two NIR data sets is provided. Experimental results show that the proposed method can achieve better performance than other methods.

1. Introduction

Partial least squares (PLS) is one of the most widely used multivariate calibration methods. It can handle the situations where the number of variables considerably exceeds the number of observations and there is collinearity within the data such as near-infrared (NIR) spectroscopic data [1]. However, when the calibration set is relatively small or contaminated by random noise, over-fitting may occur and cause the PLS model unstable [2]. Therefore, a great efforts have been made to improve the performance of PLS. These works include spectral pretreatment techniques, variable selection methods and different robust strategies, such as multiplicative scattering correction (MSC) [3], orthogonal signal correction (OSC) [4], wavelet transform (WT) [5], partial robust M-regression (PRM) [6] and multiblock partial least squares (MB-PLS) [7], etc. In addition, an alternative approach to improve the performance of PLS is ensemble learning strategy, such as Bagging and Boosting.

Bagging is proposed by Breiman [8]. In BAgging Partial Least Squares (BAPLS), each individual PLS model is trained independently, using randomly chosen training samples via a bootstrap technique. The trained individual PLS models are aggregated to make a collective decision by taking the average of the ensemble PLS models, but BAPLS is not very effective when the calibration samples are insufficient. In Boosting strategy, multiple models are developed by using the calibration subsets selected from the whole calibration set according to the distribution of the sample weights obtained under certain hypothesis [9]. Recently, the Boosting procedure is introduced into PLS: Zhang et al. [10] propose a BOosting Partial Least Squares (BOPLS) by combining a series of PLS models. To improve the robustness and prediction ability of the Boosting algorithm, a robust step is added to weaken the effect of the outliers on the model [11], several algorithms such as GentleAdaboost and BrownBoost have been developed [12]. Yu and co-workers [13] add a robust step to boost PLS for QSAR study of angiotensin II antagonists. Lutz et al. [14] propose a robust least squares (L2) Boosting for linear regression. The ensemble strategy has been proved to be an efficient way to improve the stability of the prediction [15]. However, Boosting does not perform very well when given insufficient data [16], which will stop the Boosting method from learning an effective ensemble model [17].

Inspired from bias–variance–covariance decomposition, we propose a new ensemble PLS method called creating diversity partial least squares (CDPLS) to improve the robustness of the PLS model. In this method, an ensemble model is generated iteratively, and is trained on the original calibration samples combined with some virtual samples generated in each successive iteration. In fact, BAPLS and BOPLS generate diverse models by sub-sampling and re-weighting the existed calibration samples respectively, if the calibration samples are not enough, which will limit the amount of ensemble diversity with BAPLS and BOPLS. The proposed method CDPLS directly constructs diverse models with virtual samples which are produced by original calibration samples, and this can increase the amount of ensemble diversity when the calibration samples are not enough.

* Corresponding author. Tel.: +86 10 62520293.
E-mail address: huyong821204@163.com (Y. Hu).

0039-9140/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.talanta.2012.03.047

2. Theory and algorithm

2.1. Bias–variance–covariance decomposition

Let $z = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be the data set, where x_i is the input sample and y_i is the output concentration, N is the number of the input samples, with each element drawn from a random variable z defined over an unknown distribution $p(x, y)$. We will use a parameterized estimator $f(x; \mathbf{w})$ with parameter $\mathbf{w}$ to approximate the correct mapping from input to output, so $f(x; \mathbf{w})$ can be seen as a random variable related with parameter $\mathbf{w}$. For the purpose, we would like to find the set of parameter $\mathbf{w}$ that minimize the following formulation:

$$e(f) = \int (f(x; \mathbf{w}) - y)^2 dp(x, y) \tag{1}$$

where $e(f)$ is the expected mean squared error for the random variable $f(x; \mathbf{w})$. Unfortunately we do not have access to the true distribution $p(x, y)$, so we approximate this integral with a summation over the data set z:

$$e(f) \approx \frac{1}{N} \sum_{n=1}^{N} (f(x_n; \mathbf{w}) - y_n)^2, \quad (x_n, y_n) \in z \tag{2}$$

In place of the integral notation in Eq. (1), we use $E\{(f(x; \mathbf{w}) - y)^2\}$ to stand for the expectation of the random variable $(f(x; \mathbf{w}) - y)^2$ related with parameter $\mathbf{w}$, additionally we will omit the input and parameter vectors, so where it is unambiguous, we use f instead of $f(x; \mathbf{w})$. The bias–variance decomposition from Eq. (1) as follows [18]:

$$\begin{aligned} e(f) = E\{(f - y)^2\} &= (E\{f\} - y)^2 + E\{(f - E\{f\})^2\} \\ &= \text{bias}(f)^2 + \text{variance}(f) \end{aligned} \tag{3}$$

For the ensemble state: we have a collection of estimator: $f_1, \ldots, f_M$, each f_i has its own parameter vector $\mathbf{w}_i$, and M is the total number of estimators. The general ensemble model (GEM) is to take a non-uniformly weighted average:

$$\bar{f}(x; \mathbf{w}_1, \ldots, \mathbf{w}_M) = \sum_{i=1}^{M} \alpha_i f_i(x; \mathbf{w}_i), \quad \sum_{i=1}^{M} \alpha_i = 1 \tag{4}$$

we use $\bar{f}(x; \mathbf{w}_1, \ldots, \mathbf{w}_M)$ instead of $f(x; \mathbf{w})$, omit the input x and parameter vectors $\mathbf{w}_i (i = 1, \ldots, M)$, we will have a bias–variance decomposition like Eq. (3)

$$\begin{aligned} e(\bar{f}) = E\{(\bar{f} - y)^2\} &= (E\{\bar{f}\} - y)^2 + E\{(\bar{f} - E\{\bar{f}\})^2\} \\ &= \text{bias}(\bar{f})^2 + \text{variance}(\bar{f}) \end{aligned} \tag{5}$$

For convenience, we restrict the analysis to the uniform case ($\alpha_i = 1/M$) and define three concepts as follows:

$$\overline{\text{bias}} = \frac{1}{M} \sum_i (E\{f_i\} - y) \tag{6}$$

$$\overline{\text{variance}} = \frac{1}{M} \sum_i E\{(f_i - E\{f_i\})^2\} \tag{7}$$

$$\overline{\text{covariance}} = \frac{1}{M(M-1)} \sum_i \sum_{j \neq i} E\{(f_i - E\{f_i\})(f_j - E\{f_j\})\} \tag{8}$$

where $E\{f_i\}$ is the expected value of ith model $f_i(x; \mathbf{w})$, we then have the bias–variance–covariance decomposition [18] as follows:

$$E\{(\bar{f} - y)^2\} = \overline{\text{bias}}^2 + \frac{1}{M} \overline{\text{variance}} + \left(1 - \frac{1}{M}\right) \overline{\text{covariance}} \tag{9}$$

2.2. Construction of virtual samples

We declare that the sample is a feature vector only represented as a spectrum in the next discussion. We generate virtual samples based on Synthetic Minority Over-sampling TEchnique (SMOTE) method [19]. Details of this method are as follows: the original samples are over-sampled by introducing synthetic samples along the line segments joining any k nearest neighbors of each sample. Depending upon the required amount of over-sampling, neighbors from the k nearest neighbors are randomly chosen, and our implementation currently uses five nearest neighbors. For instance, if the needed amount of over-sampling is 200%, then only two neighbors from the five nearest neighbors are chosen and only one sample is generated in the direction of each. In this paper, we use 100% for the amount of over-sampling. Synthetic samples are generated in the following way: each sample can be seen as a feature vector, we take the difference between the sample (feature vector) under consideration and its nearest k neighbors, multiply this difference by a random number between 0 and 1, and add it to the feature vector. An example is given below: consider a sample (6, 4) and let (4, 3) be one of its k-nearest neighbors, we get the difference $(-2, -1)$, the virtual samples will be generated like this: $x_{\text{vir}} = (6, 4) + \alpha \cdot (-2, -1)$, where α is a random number between 0 and 1.

2.3. The new ensemble partial least squares algorithm based on creating diversity

We need to provide metrics to measure the quality and reliability of a prediction. In the calculations, the root mean squared error of calibration (RMSEC) and the root mean squared error of prediction (RMSEP) are used as an evaluation criterion as follows:

$$\text{RMSEC} = \sqrt{\frac{1}{N_1} \sum_{i=1}^{N_1} (y_{cal,i} - \hat{y}_{cal,i})^2} \tag{10}$$

$$\text{RMSEP} = \sqrt{\frac{1}{N_2} \sum_{i=1}^{N_2} (y_{pred,i} - \hat{y}_{pred,i})^2} \tag{11}$$

in which $y_{\text{cal,i}}$ and $\hat{y}_{cal,i}$ are the measured and PLS fitted values for the ith calibration sample, respectively; $y_{\text{pred,i}}$ and $\hat{y}_{pred,i}$ are the measured and PLS predicted values for the ith prediction sample respectively; N_1 and N_2 are the number of the calibration samples and the prediction samples, respectively.

From Eq. (9), we can see that the generalization error of an ensemble model not only depends on the bias and variance of the individual estimators, but also depends on the covariance between the individuals, so we should train ensemble members separately, and find some way to capture the diversity of each model so as to reduce the covariance. We propose a new method based on creating diversity to solve this problem. The proposed method ensures diversity on an arbitrarily large set with additional virtual samples. Virtual samples and the interchange concentration, which can deliver the data set more disturbance, more comprehensive coverage of the potential changes that may be encountered in the data. We interchange concentrations so that two adjacent individual models are inconsistent as much as possible, the inconsistency makes the ensemble model more effective, which will improve the generalization ability of the model based on bias–variance–covariance decomposition. The basic idea is as follows:

Given a data set $z = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, where x_i is the calibration sample and y_i is the concentration corresponding to x_i. We initialize the ensemble PLS model which is trained on the given calibration set and denote it as f_{in}. Meanwhile we compute

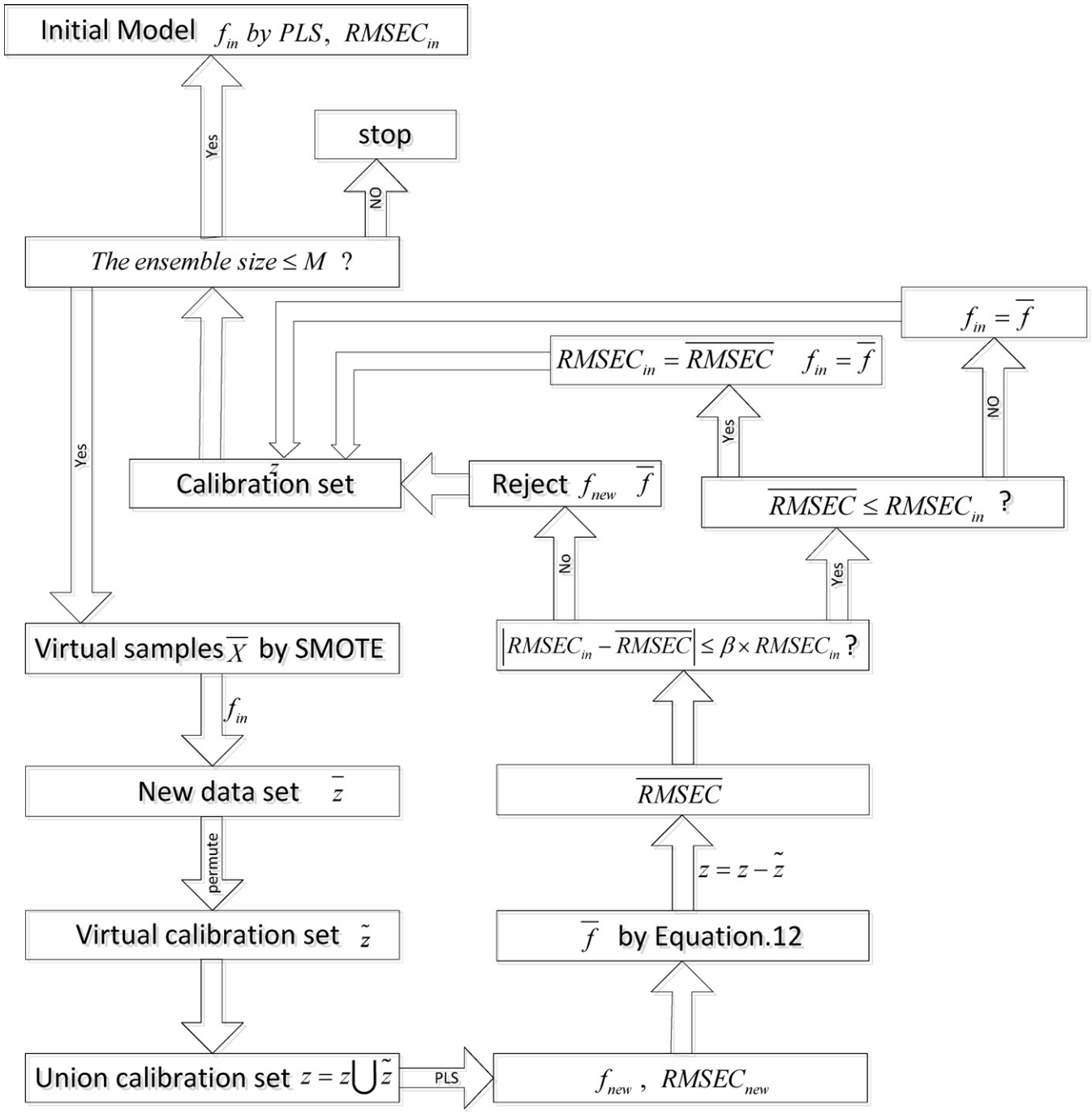

Fig. 1. The framework of the proposed method creating diversity partial least squares.

the initialized root mean squared error of calibration ($RMSEC_{in}$) with model f_{in} based on the given calibration set. In the first iteration, we generate N virtual samples $\overline{X} = \{\overline{x}_1, \ldots, \overline{x}_N)\}$ based on SMOTE method from the calibration set, and then we obtain the "virtual concentration" $\overline{Y} = \{\overline{y}_1, \ldots, \overline{y}_N\}$ with model f_{in} based on the virtual samples $\overline{X}$. We denote the new data set as $\overline{z} = \{(\overline{x}_1, \overline{y}_1), \ldots, (\overline{x}_N, \overline{y}_N)\}$. For any two adjacent elements in $\overline{Y}$, we interchange them to create diversity. We denote the new set as $\widetilde{Y} = \{\widetilde{y}_1, \ldots, \widetilde{y}_N\}$ (two adjacent points in $\overline{Y}$ are permuted), hence we get the diversity data set $\widetilde{z} = \{(\overline{x}_1, \widetilde{y}_1), \ldots, (\overline{x}_N, \widetilde{y}_N)\}$. We call this new data set $\widetilde{z}$ as "virtual calibration set". We then obtain a new PLS model f_{new} trained on the union of z and $\widetilde{z}$. In order to compare the performance between f_{in} and f_{new}, we only compute the root mean squared error of calibration ($RMSEC_{new}$) with f_{new} corresponding to the original data set z. In fact, there are some differences between the individual PLS model f_{in} and f_{new}: at least, they have different concentration results on the virtual samples ($\widetilde{Y}$ is a permutation of $\overline{Y}$, they are different). Meanwhile we compute the normalized weight vector $W(\omega_1, \omega_2)$ for each model:

$$\omega_i = \frac{\hat{\omega}_i}{\hat{\omega}_1 + \hat{\omega}_2} \quad , \quad i = 1, 2$$

where $\hat{\omega}_1 = \frac{1}{RMSEC_{in}}$, $\hat{\omega}_2 = \frac{1}{RMSEC_{new}}$. The new ensemble model is as follows:

$$\overline{f} = \omega_1 f_{in} + \omega_2 f_{new} \tag{12}$$

and then we compute the root mean squared error of calibration ($\overline{\text{RMSEC}}$) with $\bar{f}$ based on z (the original calibration data set again). In order to maintain the training accuracy to some extent, we accept the new model f_{new} according to the following inequality equation:

$$|\text{RMSEC}_{\text{in}} - \overline{\text{RMSEC}}| \le \beta \times \text{RMSEC}_{\text{in}} \quad (13)$$

where β is a tradeoff between the training accuracy and the creating diversity: if β is too small, it requires that $\overline{\text{RMSEC}}$ is close to RMSEC_{in}, we tend to pursue accuracy, while rejecting many diverse models in this situation; If β is too large, it means we can obtain many diverse models, while we cannot guarantee the accuracy of these models, maybe this will further affect the accuracy of the final ensemble model. In short, we need to create diverse models, at the same time we should guarantee the accuracy of the training models to some extent: if Eq. (13) holds, we accept f_{new} and replace f_{in} with $\bar{f}$, in this case, if $\overline{\text{RMSEC}}$ is smaller than RMSEC_{in}, we replace RMSEC_{in} with $\overline{\text{RMSEC}}$. If Eq. (13) does not hold, we will reject f_{new} and then find the next model with new "virtual calibration set" based on SMOTE method. All these processes are repeated until we reach the desired ensemble size M. The framework of the proposed method begins with calibration set z, and it is shown in Fig. 1.

Specific steps of the proposed method creating diversity partial least squares are described in Algorithm 1.

Algorithm 1. Creating diversity partial least squares algorithm

Given N calibration samples $z = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, basic learning algorithm PLS, desired ensemble size M, threshold β to tune the tradeoff between the calibration accuracy and the creating diversity. The final ensemble model is $\bar{f}$.

1: $i = 1$.
2: $f_{\text{in}} = f_i = \text{PLS}(z)$, compute RMSEC_{in} on z with f_{in}.
3: Initialize ensemble $\bar{f} = f_{\text{in}}$.
4: While $i \le M$.
5: Generate N virtual samples $\overline{X}$ based on the SMOTE method according to original samples X, get the"virtual response" $\overline{Y}$ based on the ensemble model $\bar{f}$, and replace $\overline{Y}$ with $\widetilde{Y}$ to obtain the "virtual calibration set" $\widetilde{z}$.
6: $z = z \bigcup \widetilde{z}$, $f_{\text{new}} = f_{i+1} = \text{PLS}(z)$, obtain $\text{RMSEC}_{\text{new}}$ on z with f_{new}, updated ensemble model $\bar{f}$ according to Eq. (12).
7: $z = z - \widetilde{z}$, compute the $\overline{\text{RMSEC}}$ on z with $\bar{f}$.
8: If $|\text{RMSEC}_{\text{in}} - \overline{\text{RMSEC}}| \le \beta \times \text{RMSEC}_{\text{in}}$, $\overline{\text{RMSEC}} \le \text{RMSEC}_{\text{in}}$, we accept f_{new}, replace f_{in} with $\bar{f}$ and replace RMSEC_{in} with $\overline{\text{RMSEC}}$;
If $|\overline{\text{RMSEC}_{\text{in}}} - \overline{\text{RMSEC}}| \le \beta \times \text{RMSEC}_{\text{in}}$ and $\overline{\text{RMSEC}} > \text{RMSEC}_{\text{in}}$, we accept f_{new} and replace f_{in} with $\bar{f}$, not replace RMSEC_{in} with $\overline{\text{RMSEC}}$;
If $|\text{RMSEC}_{\text{in}} - \overline{\text{RMSEC}}| > \beta \times \text{RMSEC}_{\text{in}}$, reject f_{new};
set $i = i + 1$, go back to step 4.

3. Experimental

3.1. Data sets

For the purpose of testing and comparing different calibration methods, two NIR data sets are used which are presented respectively by the Software Shootout at the IDRC (1998) (International Diffuse Reflectance Conference in Chambersburg) (http://www.models.life.ku.dk) and the IDRC (2002) (http://www.eigenvector.com/data/tablets/index.html). Data set 1 contains NIR spectra of 141 fescue grass powdered samples (Fig. 2) with specified carbon contents ranging from 29% to 41%, including 1050 variables from 400 to 2498 nm. These samples are arbitrarily divided into three sets: 71 samples are used for calibration set, 35 samples are used for validation set and the remaining 35 samples are used for prediction set. Data set 2 contains NIR spectra of 654 pharmaceutical tablet samples from the first instrument with specified hardness ranging from 13.8 to 23.5, including 650 variables from 600 to 1898 nm. These samples have been divided into three parts in the original record: calibration set consists of 154 samples (Fig. 3 for 50 of 154), validation set consists of 40 samples and prediction set consists of 460 samples.

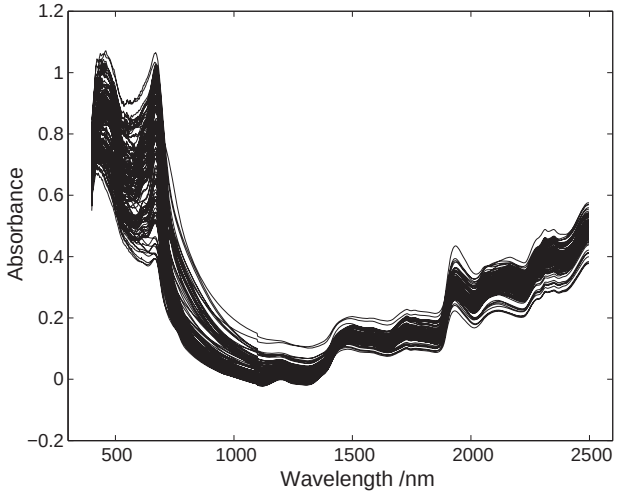

Fig. 2. Spectra of date set 1: fescue grass powdered samples.

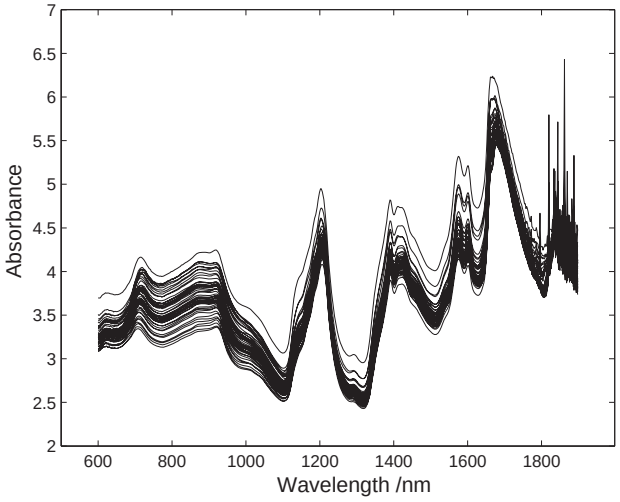

Fig. 3. Spectra of date set 2 (calibration set): 50 pharmaceutical tablet samples.

3.2. Calculations

We use the validation set to optimize the parameters in the proposed method in the following section. The number of latent variables for constructing each PLS model is determined by leave-one-out cross validation (LOOCV) [20]. We compare the proposed method creating diversity partial least squares with the standard partial least squares method and some other ensemble methods including BAPLS and BOPLS based on RMSEP from Eq. (11).

4. Results and discussions

4.1. Number of calibration samples

The number of the calibration samples in CDPLS will affect the prediction result. Thus, a proper number of calibration samples must be investigated prior to development of the diversity model. In this study, the number of the calibration samples N for the two data sets is investigated by increasing from 10 to the maximum samples with a step of 5. For each N, the ensemble model is developed and the model is then used to predict the validation set. As

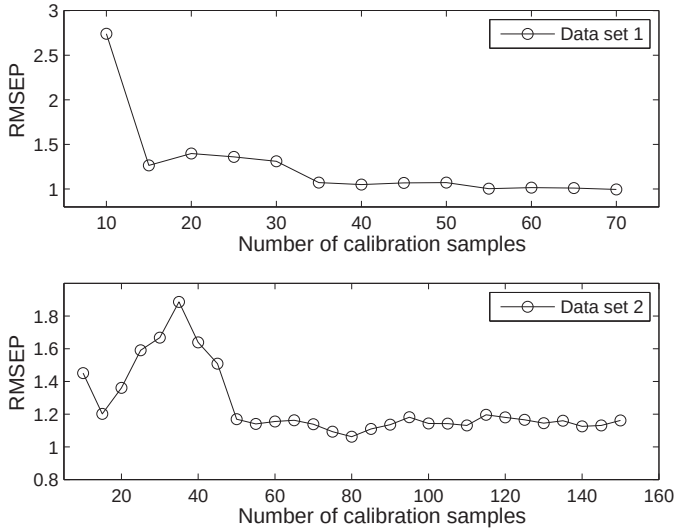

Fig. 4. Variation of RMSEP with the number of calibration samples N used in the calibration subset for data set 1 (above) and data set 2 (below).

shown in Fig. 4, for the data set 1, RMSEP is large when a small N is used, and then decreases with the increase of N. When N reaches 50, the variation of RMSEP tends to be stable although there is a few fluctuations. For data set 2, although there is a few fluctuations in the beginning, the variation of RMSEP tends to be stable at last.

In fact, from Fig. 4, we can see that more samples are not necessary to some extent when using our method CDPLS. In order to evaluate the proposed algorithm CDPLS on more small calibration samples case, we provide a detail analysis and construct two experiments on data set 1, where 35 and 50 samples are chosen as calibration samples, respectively. The algorithm is repeated 20 times and the prediction results are compared for different number of calibration samples. All these are shown in Fig. 5: although the number of calibration samples increases (from 35 to 50), the RMESP does not significantly reduce, and the prediction results on the small data set (35 calibration samples) are comparable with that on the large data set (50 calibration samples). From the pervious analysis, we assign N to 50 and 100 for the two data sets, respectively.

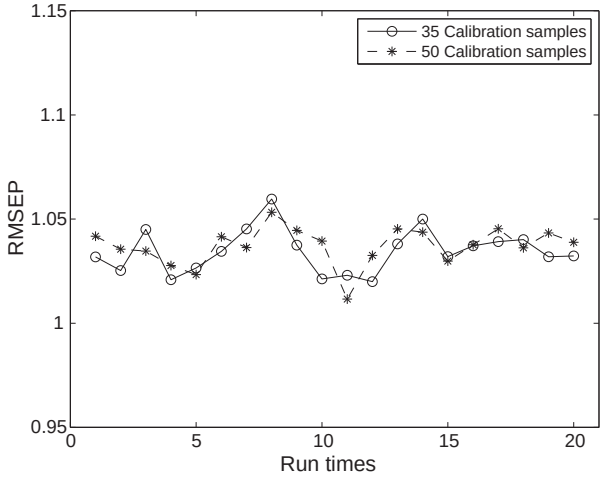

Fig. 5. RMSEP obtained by different number of calibration samples in 20 runs of prediction for data set 1.

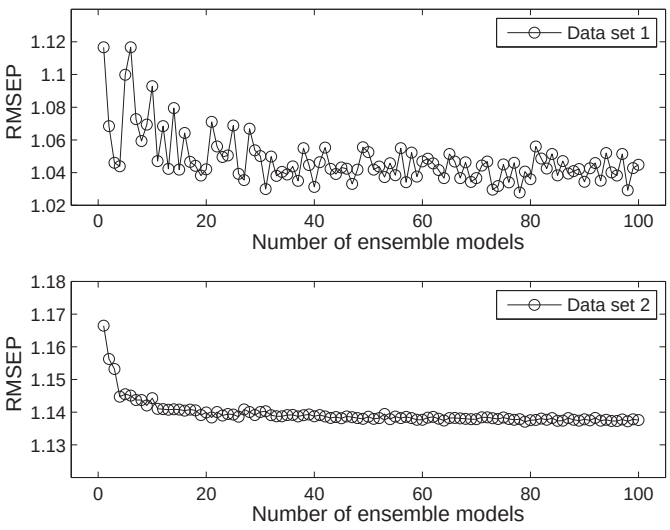

Fig. 6. Variation of RMSEP with the number of ensemble models M used in the calibration subset for data set 1 (above) and data set 2 (below).

4.2. Number of ensemble models

The ensemble size M is another important parameter to affect CDPLS. In this study, for the two data sets, M is investigated by increasing from 1 to 100 with a step of 1. For each M, the ensemble model is developed and the model is then used to predict the validation set. As shown in Fig. 6, for the two data sets, RMSEP is large when a small M is used, and then decreases with the increase of M. The variation of RMSEP tends to be stable at last. Considering the speed of calculation, we assign M to 40 and 20 for the two data sets respectively.

4.3. The effect of parameter β for the ensemble models

Parameter β is a tradeoff between the calibration accuracy and the creating diversity. In this study, β is investigated by increasing from 0.001 to 0.1 with a step of 0.001 for the two data sets. For each β, the ensemble model is developed and the model is then used to predict the validation set. As shown in Fig. 7, for the two data sets, RMSEP is large when a small β is used, then RMSEP decreases with

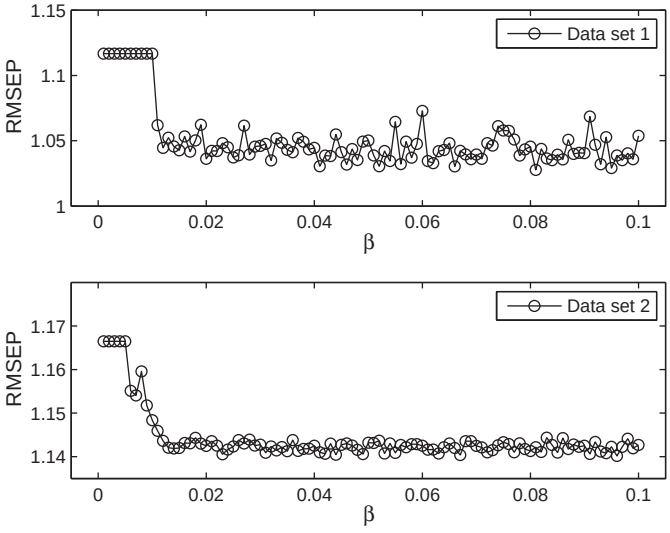

Fig. 7. Variation of RMSEP with parameter β used in the calibration subset for data set 1 (above) and data set 2 (below).

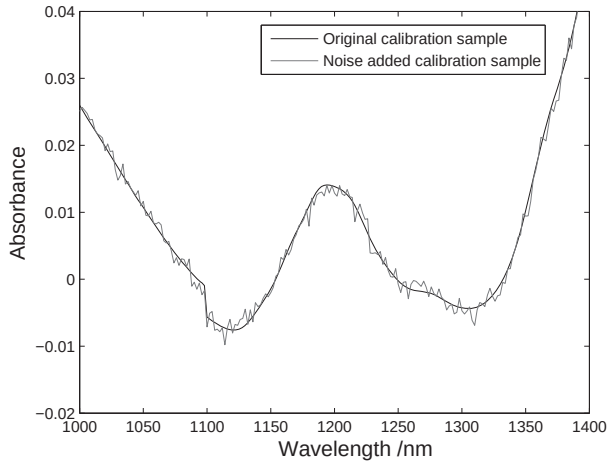

Fig. 8. Spectra for the first calibration sample of data set 1 between 1000 nm and 1400 nm: original sample (black) and noise added sample (gray).

the increase of β, and the variation of RMSEP tends to be stable at last, we assign β to 0.02 for both data sets in the next discussion.

4.4. The effect of noise for the ensemble models

To evaluate the impact of noise on the proposed algorithm CDPLS, we add a gaussian noise to the calibration samples of data set 1 (spectra for the first sample between 1000 nm and 1400 nm show in Fig. 8).

With above optimal parameters on data set 1, the algorithm is repeated 20 times and the prediction results are compared on the two calibration sets: original samples and noise added samples. All these are shown in Fig. 9.

Compared to the original calibration samples, RMESP corresponding to the contaminated calibration samples does not significantly increase, and sometimes is better than that on the original calibration set. This experiment shows that our method CDPLS is not sensitive to random noise to some extent.

4.5. Prediction results

With the optimal parameters, a CDPLS model is developed with the calibration samples and used for the prediction samples. The

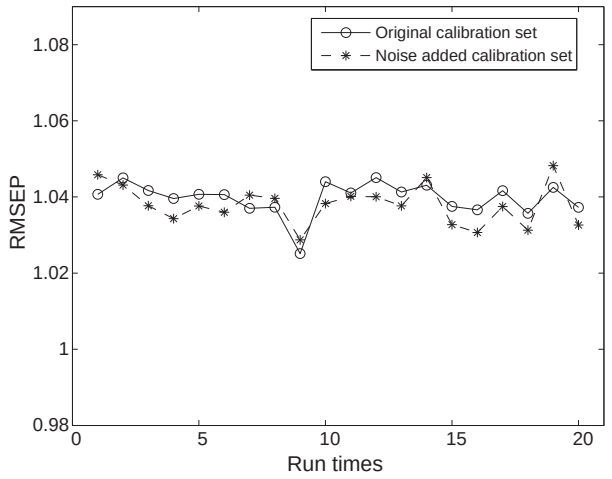

Fig. 9. RMSEP obtained by original calibration set and noise added calibration set in 20 runs of prediction for data set 1.

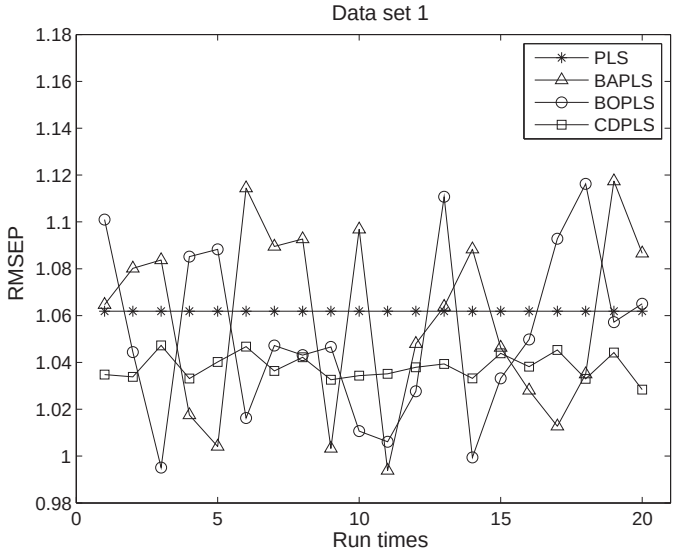

Fig. 10. RMSEP obtained by PLS, BAPLS, BOPLS, CDPLS in 20 runs of prediction for data set 1.

algorithm is repeated 20 times and the prediction results are compared with PLS, BAPLS and BOPLS models (with the same calibration data set, the same prediction data set and the same parameters). All these are shown in Figs. 10 and 11, respectively for the two data sets. It is clear that RMSEP is all the same with the standard PLS because it is developed with a single model, while the other three ensemble models show some fluctuations, and the RMSEP of the proposed CDPLS method is almost always smaller than that of PLS in the 20 runs. Meanwhile, the fluctuations of CDPLS is smaller than those of the BAPLS and BOPLS. BAPLS and BOPLS provide diverse models by sub-sampling and re-weighting the existed calibration samples respectively, although these will generate some diverse individual models, they cannot guarantee the accuracy of each individual model. While for our CDPLS method, we introduce a tradeoff parameter β to tune the training accuracy and the creating diversity. Especially for the data set 2 which contains 154 calibration samples and 460 prediction samples, the fluctuations of the RMSEP with BOPLS method are significant (Fig. 11), and the result of this method is worse than that of the other three methods (PLS, BAPLS,

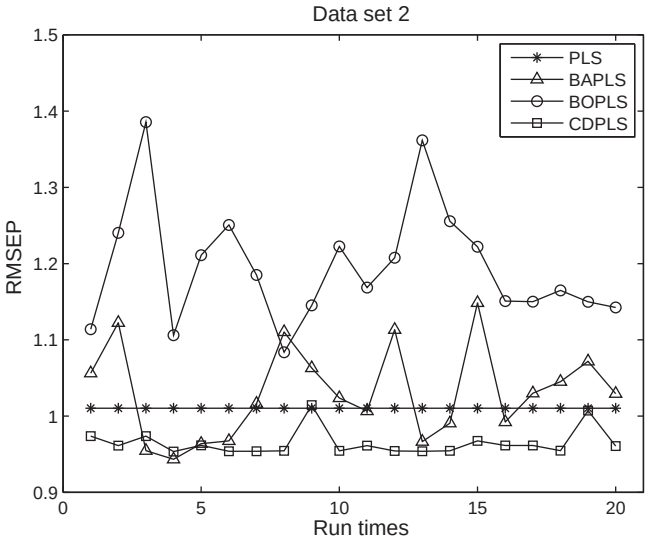

Fig. 11. RMSEP obtained by PLS, BAPLS, BOPLS, CDPLS in 20 runs of prediction for data set 2.

and CDPLS), which may be because the number of calibration samples is less than that of the prediction samples, and the models are only constructed based on the calibration samples, this may easily lead to over-fitting with BOPLS, but our CDPLS method solves this problem to some extent (Fig. 11): we construct models based on the original calibration samples and the "virtual calibration samples", which expands the scope of the original samples. Meanwhile, because we interchange elements of "virtual concentration", this will ensure that each constructed model has some differences with others. Based on the principle of bias–variance–covariance decomposition, the final model which is the weighted average of the single models has better generalization ability. In short, all of the above experiments for the two data sets indicate that both the stability and the accuracy of CDPLS method are much better than that of PLS and BAPLS including BOPLS method.

5. Conclusions

Based on bias–variance–covariance decomposition, we proposed a new algorithm CDPLS for building a robust ensemble model in multivariate calibration of NIR spectra. Experimental results on two NIR data sets demonstrate the efficiency of the proposed algorithm.

Acknowledgments

The research is supported in part by the National Natural Science Foundation of China (60972126), the Joint Funds of the National Natural Science Foundation of China (U0935002/L05), the Beijing Municipal Natural Science Foundation (4102060) and the State Key Program of National Natural Science of China (61032007).

References

[1] S. Navea, R. Tauler, A. de Juan, Anal. Biochem. 336 (2005) 231–242.
[2] Y.K. Li, X.G. Shao, W.S. Cai, Talanta 72 (2007) 217–222.
[3] P. Geladi, D. MacDougall, H. Martens, Appl. Spectrosc. 39 (1985) 491–500.
[4] S. Wold, H. Antti, F. Lindgren, J. Ohman, Chemometr. Intell. Lab. Syst. 44 (1998) 175–185.
[5] D. Chen, B. Hu, X.G. Shao, Q.D. Su, Analyst 129 (2004) 664–669.
[6] S. Serneels, C. Croux, P. Filzmoser, P.J. Van Espen, Chemometr. Intell. Lab. Syst. 79 (2005) 55–64.
[7] M. Jing, W.S. Cai, X.G. Shao, Chemometr. Intell. Lab. Syst. 100 (2010) 22–27.
[8] L. Breiman, Mach. Learn. J. 24 (1996) 123–140.
[9] K. Varmuza, P. He, K.T. Fang, J. Data Sci. 1 (2003) 391–404.
[10] M.H. Zhang, Q.S. Xu, D.L. Massart, Anal. Chem. 77 (2005) 1423–1431.
[11] X.G. Shao, X.H. Bian, W.S. Cai, Anal. Chim. Acta 666 (2010) 32–37.
[12] J.H. Friedman, T. Hastie, R. Tibshirani, Ann. Stat. 28 (2000) 337–407.
[13] Y.P. Zhou, C.B. Cai, S. Huan, J.H. Jiang, H.L. Wu, G.L. Shen, R.Q. Yu, Anal. Chim. Acta 593 (2007) 68–74.
[14] R.W. Lutz, M. Kalisch, P. Buhlmann, Comput. Stat. Data Anal. 52 (2008) 3331–3341.
[15] D. Chen, W.S. Cai, X.G. Shao, Anal. Chim. Acta 598 (2007) 19–26.
[16] R.E. Schapire, Proceedings of the Tenth International Conference on Algorithmic Learning Theory, 1999, pp. 13–25.
[17] T. Dietterich, First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, Springer-Verlag, 2000, 1–15.
[18] G. Brown, J.L. Wyatt, J. Mach. Learn. Res. 6 (2005) 1621–1650.
[19] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, J. Artif. Intell. Res. 16 (2002) 321–357.
[20] X. Liu, W.C. Lu, S.L. Jin, Y.W. Li, N.Y. Chen, Chemometr. Intell. Lab. Syst. 82 (2006) 8–14.