1

Model Population Analysis for Statistical Model Comparison

Hong-Dong Li[1], Yi-Zeng Liang[1] and Qing-Song Xu[2]
[1]*College of Chemistry and Chemical Engineering, Central South University, Changsha,*
[2]*School of Mathematic Sciences, Central South University, Changsha,*
P. R. China

1. Introduction

Model comparison plays a central role in statistical learning and chemometrics. Performances of models need to be assessed using a given criterion based on which models can be compared. To our knowledge, there exist a variety of criteria that can be applied for model assessment, such as Akaike's information criterion (AIC) [1], Bayesian information criterion (BIC) [2], deviance information criterion (DIC),Mallow's Cp statistic, cross validation [3-6] and so on. There is a large body of literature that is devoted to these criteria. With the aid of a chosen criterion, different models can be compared. For example, a model with a smaller AIC or BIC is preferred if AIC or BIC are chosen for model assessment.

In chemometrics, model comparison is usually conducted by validating different models on an independent test set or by using cross validation [4, 5, 7], resulting in a single value, *i.e.* root mean squared error of prediction (RMSEP) or root mean squared error of cross validation (RMSECV). This single metrics is heavily dependent on the selection of the independent test set (RMSEP) or the partition of the training data (RMSECV). Therefore, we have reasons to say that this kind of comparison is lack of statistical assessment and also at the risk of drawing wrong conclusions. We recently proposed model population analysis (MPA) as a general framework for designing chemometrics/bioinformatics methods [8]. MPA has been shown to be promising in outlier detection and variable selection. Here we hypothesize that reliably statistical model comparison could be achieved via the use of model population analysis.

2. Model population analysis

2.1 The framework of model population analysis

Model population analysis has been recently proposed for developing chemometrics methods in our previous work [8]. As is shown in **Figure 1**, MPA works in three steps which are summarized as (1) randomly generating N sub-datasets using Monte Carlo sampling (2) building one sub-model on each sub-dataset and (3) statistically analyzing some interesting output of all the N sub-models.

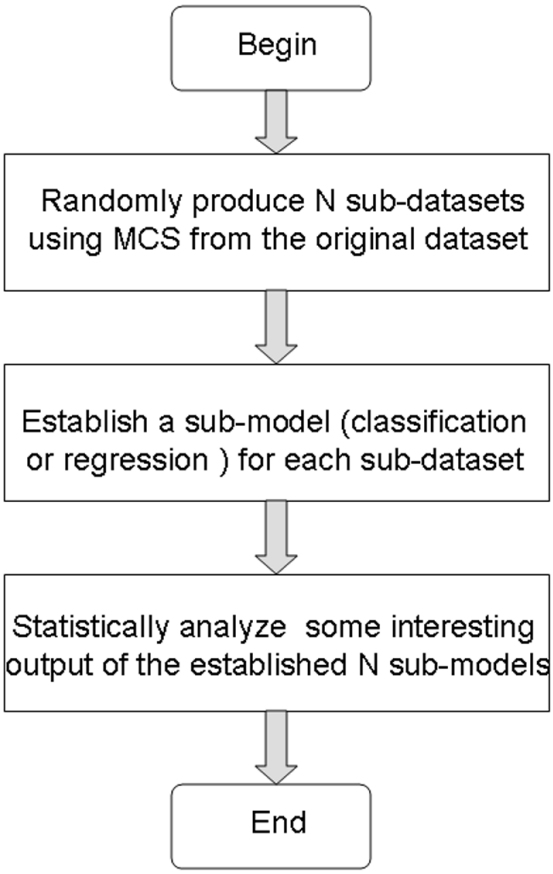


Fig. 1. The schematic of MPA. MCS is the abbreviation of Monte Carlo Sampling.

2.1.1 Monte Carlo sampling for generating a sub-dataset

Sampling plays a key role in statistics which allows us to generate replicate sub-datasets from which an interested unknown parameter could be estimated. For a given dataset (**X**, **y**), it is assumed that the design matrix **X** contains m samples in rows and p variables in columns, the response vector **y** is of size $m \times 1$. The number of Monte Carlo samplings is set to N. In this setting, N sub-datasets can be drawn from N Monte Carlo samplings with or without replacement [9, 10], which are denoted as $(\mathbf{X}_{sub}, \mathbf{y}_{sub})_i$, i = 1, 2, 3, …N.

2.1.2 Establishing a sub-model using each sub-dataset

For each sub-dataset $(\mathbf{X}_{sub}, \mathbf{y}_{sub})_i$, a sub-model can be constructed using a selected method, *e.g.* partial least squares (PLS) [11] or support vector machines (SVM) [12]. Denote the sub-model established as f_i (**X**). Then, all these sub-models can be put into a collection:

$$C = (f_1(X), f_2(X), f_3(X), \ldots, f_N(X)) \quad (1)$$

All these N sub-models are mutually different but have the same goal that is to predict the response value **y**.

2.1.3 Statistically analyzing an interesting output of all the sub-models

The core of model population analysis is statistical analysis of an interesting output, *e.g.* prediction errors or regression coefficients, of all these sub-models. Indeed, it is difficult to give a clear answer on what output should be analyzed and how the analysis should be done. Different designs for the analysis will lead to different algorithms. As proof-of-principle, it was shown in our previous work that the analysis of the distribution of prediction errors is effective in outlier detection [13].

2.2 Insights provided by model population analysis

As described above, Monte Carlo sampling serves as the basics of model population analysis that help generate distributions of interesting parameters one would like to analyze. Looking on the surface, it seems to be very natural and easy to generate distributions using Monte Carlo sampling. However, here we show by examples that the distribution provided by model population analysis can indeed provide very useful information that gives insights into the data under investigation.

2.2.1 Are there any outliers?

Two datasets are fist simulated. The first contains only normal samples, whereas there are 3 outliers in the second dataset, which are shown in Plot A and B of **Figure 2**, respectively. For each dataset, a percentage (70%) of samples are randomly selected to build a linear regression model of which the slope and intercept is recorded. Repeating this procedure 1000 times, we obtain 1000 values for both the slope and intercept. For both datasets, the intercept is plotted against the slope as displayed in Plot C and D, respectively. It can be observed that the joint distribution of the intercept and slope for the normal dataset appears to be multivariate normally distributed. In contrast, this distribution for the dataset with outliers looks quite different, far from a normal distribution. Specifically, the distributions of slopes for both datasets are shown in Plot E and F. These results show that the existence of outliers can greatly influence a regression model, which is reflected by the odd distributions of both slopes and intercepts. In return, a distribution of a model parameter that is far from a normal one would, most likely, indicate some abnormality in the data.

2.2.2 Are there any interfering variables?

In this study, we first simulate a design matrix **X** of size 50 × 10, the response variable **Y** is simulated by multiplying X with a 10-dimensional regression vector. Gussian noises with standard deviation equal to 1 are then added to **Y**. That is to say, all the variables in **X** are "true variables" that collectively predict **Y**. This dataset (**X**, **Y**) is denoted SIMUTRUE. Then another design matrix **F** is simulated of size 50 × 10. Denote the combination of **X** and **F** as **Z**=[**X F**]. This dataset (**Z**, **Y**) is called SIMUINTF, which contains variables that are not predictive of **Y**. For both datasets, we randomly choose 70% samples to first build a regression model which is then used to make predictions on the remaining 30% samples, resulting in a RMSEP value. Repeating this procedure 1000 times, we, for both datasets,

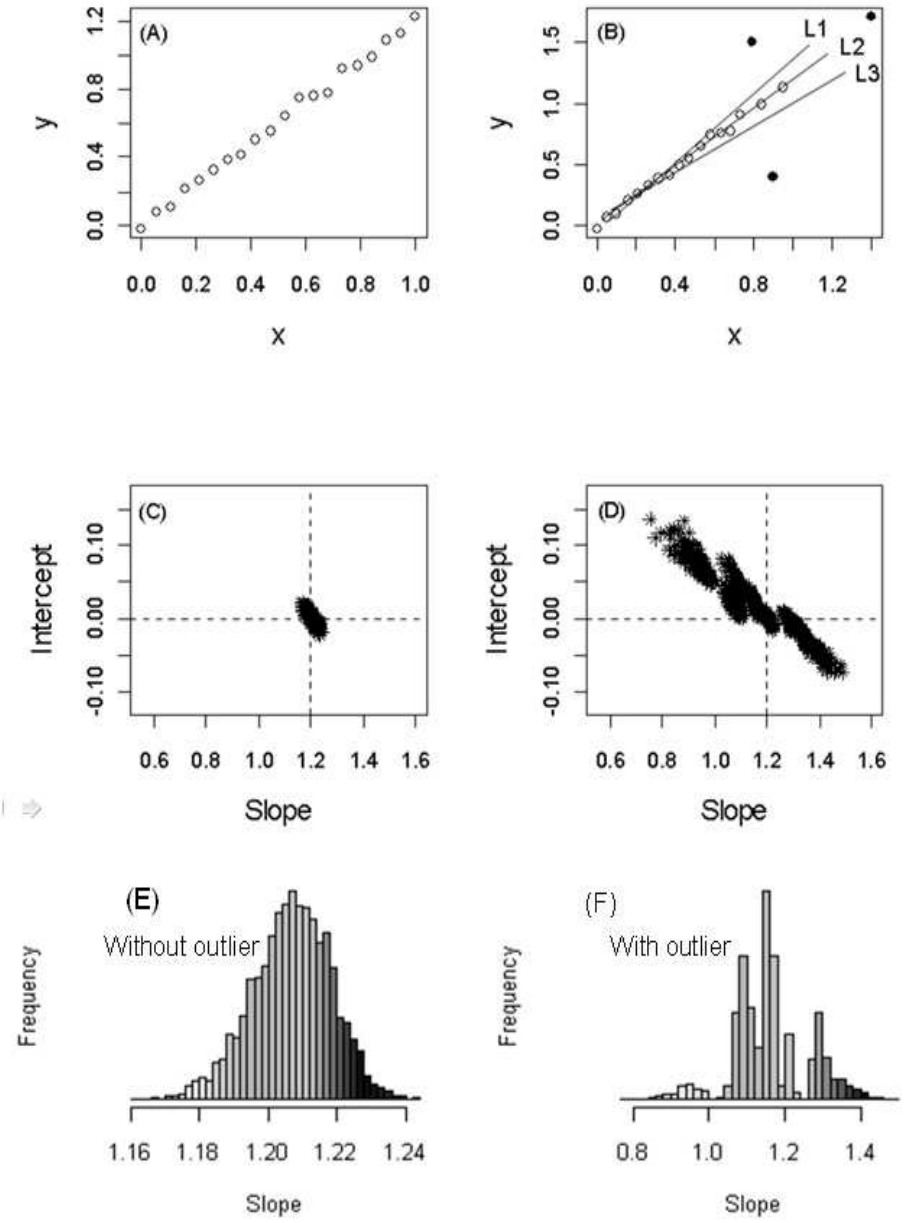


Fig. 2. A simulation study illustrating the use of model population analysis to detect whether a dataset contains outliers. Plot A and Plot B shows the data simulated without and with outliers, respectively. 1000 linear regression models computed using 1000 sub-datasets randomly selected and the slope and intercept are presented in Plot C and D. Specifically, the distribution of slope for these two simulated datasets are displayed in Plot E and Plot F.

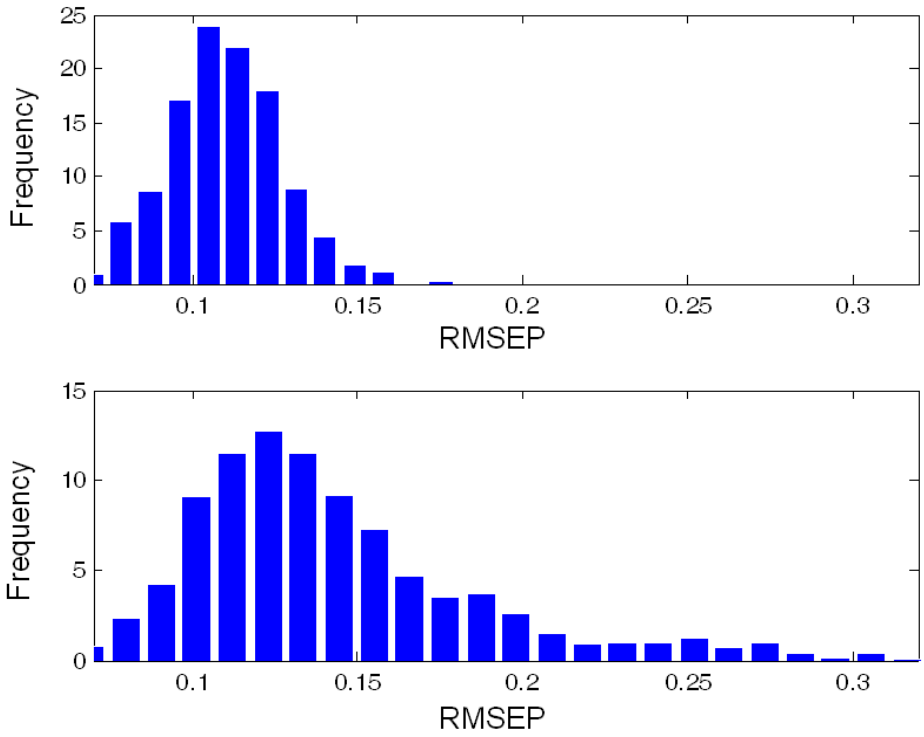


Fig. 3. The distribution of RMSEPs using the variable set that contains only "true variables" (upper panel) and the variable set that includes not only "true variables" but also "interfering variables" (lower panel).

obtain 1000 RMSEP values, of which the distributions are given in **Figure 3**. Clearly, the distribution of RMSEP of the SIMUINTF is right shifted, indicating the existence of variables that are not predictive of Y can degrade the performance of a regression model. We call this kind of variables "interfering variables". Can you tell whether a dataset contains interfering variables for a real world dataset? Curious readers may ask a question like this. Indeed, we can. We can do replicate experiments to estimate the experimental error that could serve as a reference by which it is possible to judge whether interfering variables exist. For example, if a model containing a large number of variables (with true variables included) shows a large prediction error compared to the experimental error, we may predict that interfering variables exist. In this situation, variable selection is encouraged and can greatly improve the performance of a model. Actually, when interfering variables exist, variable selection is a must. Other methods that use latent variables like PCR or PLS cannot work well because latent variables have contributions coming from interfering variables.

2.3 Applications of model population analysis

Using the idea of model population analysis, we have developed algorithms that address the fundamental issues in chemical modeling: outlier detection and variable selection. For

outlier detection, we developed the MC method [13]. For variable selection, we developed subwindow permutation analysis (SPA) [14], noise-incorporated subwindow permutation analysis (NISPA) [15] and margin influence analysis (MIA) [16]. Here, we first give a brief description of these algorithms, aiming at providing examples that could help interested readers to understand how to design an algorithm by borrowing the framework of model population analysis.

As can be seen from **Figure 1**, These MPA-based methods share the first two steps that are (1) generating N sub-datasets and (2) building N sub-models. The third step "statistical analysis of an interesting output of all these N sub-models" is the core of model population analysis that underlines different methods. The key points of these methods as well as another method Monte Carlo uninformative variable elimination (MCUVE) that also implements the idea of MPA are summarized in Table 1. In a word, the distribution from model population analysis contains abundant information that provides insight into the data analyzed and by making full use of these information, effective algorithms can be developed for solving a given problem.

Methods*	What to statistically analyze
MC method	Distribution of prediction errors of each sample
SPA	Distribution of prediction errors before and after each variable is permuted
NISPA	Distribution of prediction errors before and after each variable is permuted with one noise variable as reference
MIA	Distribution of margins of support vector machines sub-models
MCUVE	Distribution of regression coefficients of PLS regression sub-models

*: The MC method, SPA, NISPA, MIA and MCUVE are described in references [13], [14], [15] [16] and [27].

Table 1. Key points of MPA-based methods.

2.4 Model population analysis and bayesian analysis

There exist similarities as well as differences between model population analysis and Bayesian analysis. One important similarity is that both methods consider the parameter of interest not as a single number but a distribution. In model population analysis, we generate distributions by causing variations in samples and/or variables using Monte Carlo sampling [17]. In contrast, in Bayesian analysis the parameter to infer is first assumed to be from a prior distribution and then observed data are used to update this prior distribution to the posterior distribution from which parameter inference can be conducted and predictions can be made [18-20]. The output of Bayesian analysis is a posterior distribution of some interesting parameter. This posterior distribution provides a natural link between Bayesian analysis and model population analysis. Taking Bayesian linear regression (BLR) [20] as an example, the output can be a large number of regression coefficient vectors that are sampled from its posterior distribution. These regression coefficient vectors actually represent a population of sub-models that can be used directly for model population analysis. Our future work will be constructing useful algorithms by borrowing merits of both Bayesian analysis and model population analysis.

2.5 Model population analysis and ensemble learning

Ensemble learning methods, such as bagging[21], boosting [22] and random forests [23], have emerged as very promising strategies for building a predictive model and these methods have found applications in a wide variety of fields. Recently, a new ensemble technique, called feature-subspace aggregating (Feating) [24], was proposed that was shown to have nice performances. The key point of these ensemble methods is aggregating a large number of models built using sub-datasets randomly generated using for example bootstrapping. Then ensemble models make predictions by doing a majority voting for classification or averaging for regression. In our opinion, the basic idea of ensemble learning methods is the same as that in model population analysis. In this sense, ensemble learning methods can also be formulated into the framework of model population analysis.

3. Model population analysis for statistical model comparison

Based on model population analysis, here we propose to perform model comparison by deriving an empirical distribution of the difference of RMSEP or RMSECV between two models (variable sets), followed by testing the null hypothesis that the difference of RMSEP or RMSECV between two models is zero. Without loss of generality, we describe the proposed method by taking the distribution of difference of RMSEP as an example. We assume that the data **X** consists of m samples in row and p variables in column and the target value **Y** is an m-dimensional column vector. Two variable sets, say V_1 and V_2, selected from the p variables, then can be compared using the MPA-based method described below.

First, a percentage, say 80%, from the m samples with variables in V_1 and V_2 is randomly selected to build two regression models using a preselected modeling method such as PLS [11] or support vector machines (SVMs) [12], respectively. Then an RMSEP value can be computed for each model by using the remaining 20% samples as the test set. Denote the two RMSEP values as $RMSEP_1$ and $RMSEP_2$, of which the difference can be calculated as

$$D = RMSEP_1 - RMSEP_2 \tag{2}$$

By repeating this procedure N, say 1000, times, N D values are obtained and collected into a vector **D**. Now, the model comparison can be formulated into a hypothesis test problem as:

Null hypothesis: the mean of **D** is zero.

Alternative hypothesis: the mean of **D** is not zero.

By employing a statistical test method, *e.g.* *t*-test or Mann-Whitney U test [25], a *P* value can be computed for strictly assessing whether the mean of **D** is significantly different from zero ($P<0.05$) or not ($P>0.05$). If $P<0.05$, the sign of the mean of **D** is then used to compare which model (variable set) is of better predictive performance. If $P>0.05$, we say two models have the same predictive ability.

4. Results and discussions

4.1 Comparison of predictive performances of variables subsets

The corn NIR data measured on *mp5* instrument is used to illustrate the use of the proposed method (http://software.eigenvector.com/Data/index.html). This data contain NIR spectra

measured at 700 wavelengths on 80 corn samples. The original NIR spectra are shown in **Figure 4**. The chemical property modeled here is the content of protein. As was demonstrated in a large body of literature [26-30], variable selection can improve the predictive performance of a model. Here we would like to investigate whether the gain in predictive accuracy using variable subsets identified by variable selection methods is significant.

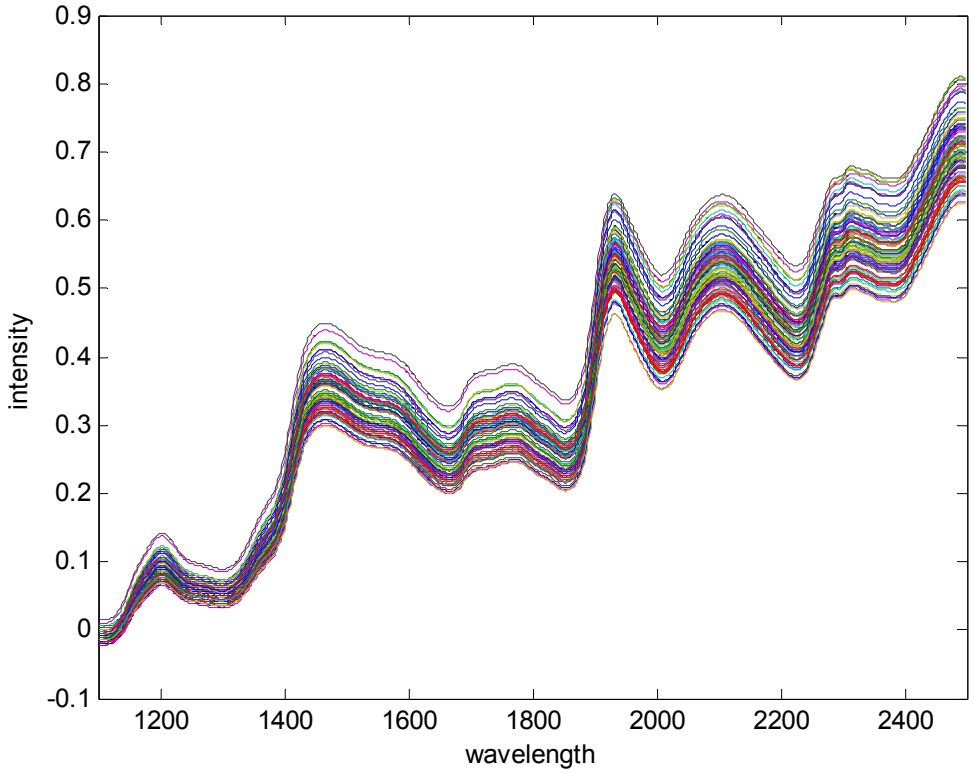


Fig. 4. Original near infrared spectra of corn on the mp5 instrument.

Uninformative variable elimination (UVE) is a widely used method for variable selection in chemometrics [26]. Its extended version, Monte Carlo UVE (MCUVE), was recently proposed [27, 31]. Mimicking the principle of "survival of the fittest" in Darwin's evolution theory, we developed a variable selection method in our previous work, called competitive adaptive reweighted sampling (CARS) [8, 28, 32, 33], which was shown to have the potential to identify an optimal subset of variables that show high predictive performances. The source codes of CARS are freely available at [34, 35].

In this study, MCUVE and CARS is chosen to first identify two variable sets, named V_1 and V_2, respectively. The set of the original 700 variables are denoted as V_0. Before data analysis, each wavelength of the original NIR spectra is standardized to have zero mean and unit variance. Regarding the pretreatment of spectral data, using original spectra, mean-centered

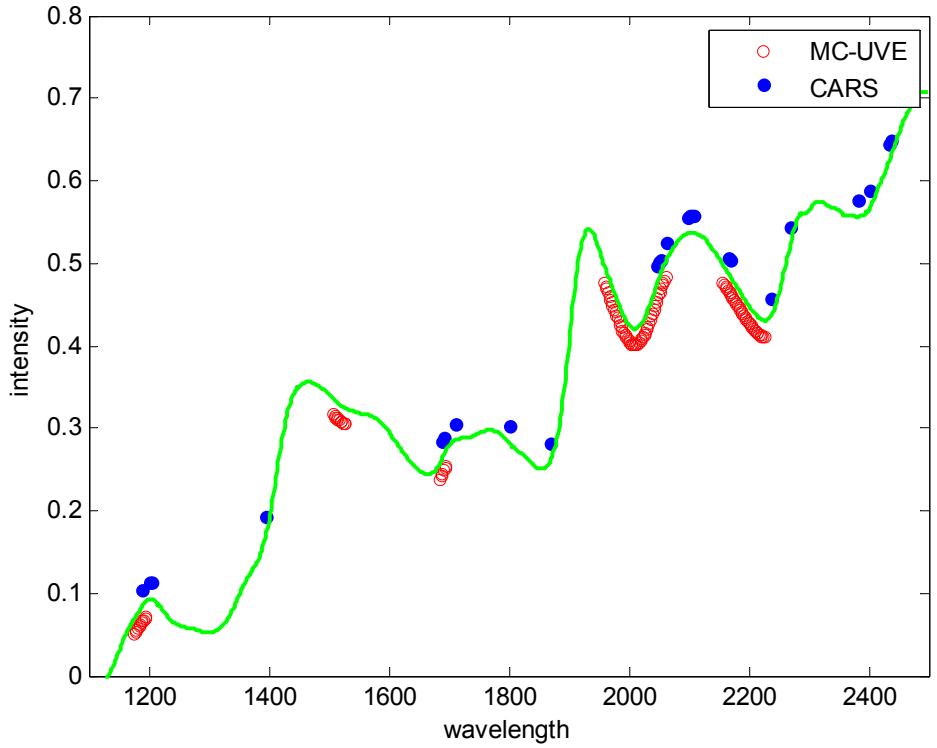


Fig. 5. Comparison of selected wavelengths using MC-UVE (red circles) and CARS (blue dots). The green line denotes the mean of the 80 corn NIR spectra.

spectra or standardized spectra indeed would lead to different results. But the difference is usually not big according to our experience. The reason why we choose standardization is to remove the influence of each wavelength's variance on PLS modeling because the decomposition of spectrum data X using PLS depends on the magnitude of covariance between wavelengths and the target variable Y. The number of PLS components are optimized using 5-fold cross validation. For MCUVE, the number of Monte Carlo simulations is set to 1000 and at each simulation 80% samples are selected randomly to build a calibration model. We use the reliability index (RI) to rank each wavelength and the number of wavelengths (with a maximum 200 wavelengths allowed) is identified using 5-fold cross validation. Using MCUVE, 115 wavelengths in 5 bands (1176-1196nm, 1508-1528nm, 1686-1696nm, 1960-2062nm and 2158-2226nm) are finally selected and shown in **Figure 5** as red circles. For CARS, the number of iterations is set to 50. Using CARS, altogether 28 variables (1188, 1202, 1204, 1396, 1690, 1692, 1710, 1800, 1870, 2048, 2050, 2052, 2064, 2098, 2102, 2104, 2106, 2108, 2166, 2168, 2238, 2270, 2382, 2402, 2434, 2436, 2468 and 2472 nm) are singled out and these variables are also shown in **Figure 5** as blue dots. Intuitively, MCUVE selects 5 wavelength bands while the variables selected by CARS are more diverse and scattered at different regions. In addition, the Pearson correlations variables selected by both methods are shown in **Figure 6**.

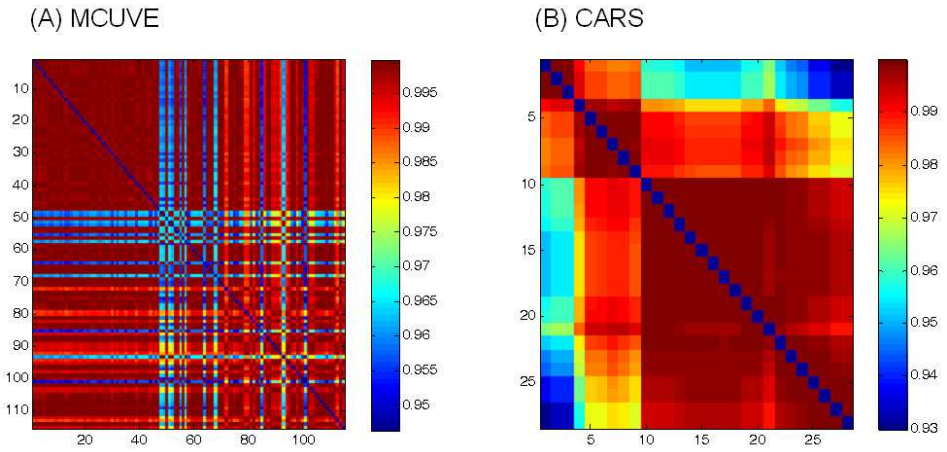


Fig. 6. The Pearson pair-wise correlations of variables selected using MCUVE (115 variables, left) and CARS (28 variables, right).

We choose PLS for building regression models. For the MPA-based method for model comparison, the number of Monte Carlo simulations is set to 1000 and at each simulation 60% samples are randomly selected as training samples and the rest 40% work as test samples. The number of PLS components is chosen based on 5-fold cross validation. In this setting, we first calculated 1000 values of $RMSEP_0$, $RMSEP_1$ and $RMSEP_2$ using V_0, V_1 and V_2, respectively. The distributions of $RMSEP_0$, $RMSEP_1$ and $RMSEP_2$ are shown in **Figure 7**. The mean and standard deviations of these three distributions are 0.169±0.025 (full spectra), 0.147±0.018 (MCUVE) and 0.108±0.015 (CARS). On the whole, both variable selection methods improve the predictive performance in terms of lower prediction errors and smaller standard deviations. Looking closely, the model selected by CARS has smaller standard deviation than that of MCUVE. The reason may be that CARS selected individual wavelengths and these wavelengths display lower correlations (see **Figure 6**) than those wavelength bands selected by MCUVE. The lower correlation results in better model stability which is reflected by smaller standard deviations of prediction errors. Therefore from the perspective of prediction ability, we recommend to adopt methods that select individual wavelengths rather than continuous wavelength bands.

Firstly, we compare the performance of the model selected by MCUVE to the full spectral model. The distribution of D values (MCUVE – Full spectra) is shown in Plot A of **Figure 8**. The mean of D is -0.023 and is shown to be not zero ($P < 0.000001$) using a two-side t test, indicating that MCUVE significantly improves the predictive performance. Of particular note, it can be observed that a percentage (83.1%) of D values are negative and the remaining (16.9%) is positive, which indicates model comparison based on a single split of the data into a training set and a corresponding test set may have the potential risk of drawing a wrong conclusion. In this case, the probability of saying that MCUVE does not improve predictive performances is about 0.169. However, this problem can be solved by the proposed MPA-based method because the model performance is tested on a large number of sub-datasets, rendering the current method potentially useful for reliably

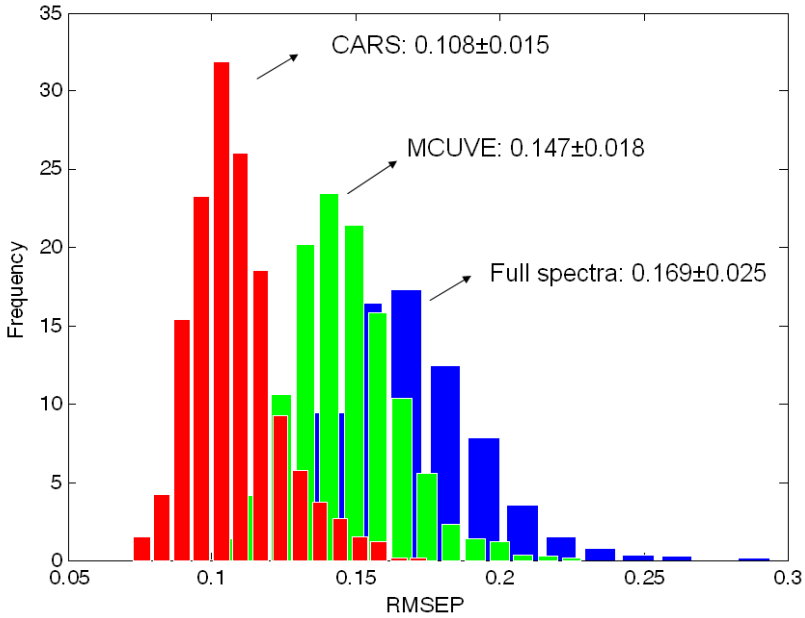


Fig. 7. Distributions of root mean squared errors of prediction (RMSEP) from 1000 test sets (32 samples) randomly selected from the 80 corn samples using full spectra and variables selected by MCUVE and CARS, respectively.

statistical model comparison. With our method, we have evidence showing that the improvement resulting from MCUVE is significant.

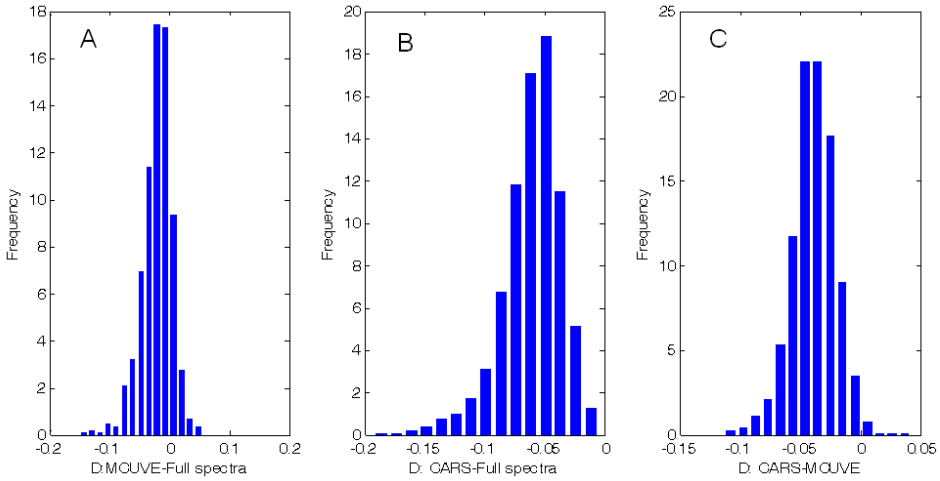


Fig. 8. The distributions of D values. The P values of t test for these three distributions are 8.36×10^{-120}, 0 and 0, respectively.

Further, the performance of the model selected by CARS is compared to the full spectral model. The distribution of D values (CARS – Full spectrum) is shown in Plot B of **Figure 8**. The mean of D is -0.061 which is much smaller that that from MCUVE (-0.023). Using a two-side t test, this mean is shown to be significantly different from zero (P = 0), indicating that the improvement over the full spectral model is significant. Interestingly, it is found that all the D values are negative, which implies the model selected by CARS is highly predictive and there is little evidence to recommend the use of a full spectral model, at least for this dataset.

Finally, we compare the models selected by MCUVE and CARS, respectively. The distribution of D values (CARS – MCUVE) is shown in Plot C of **Figure 8**. The mean of D values is -0.039. Using a two-side t test, this mean is shown to be significantly different from zero (P = 0), indicating that the improvement of CARS over MCUVE is significant. We find that 98.9% of D values are negative and only 1.1% are positive, which suggests that there is a small probability to draw a wrong conclusion that MCUVE performs better than CARS. However, with the help of MPA, this risky conclusion can be avoided, indeed.

Summing up, we have conducted statistical comparison of the full spectral model and the models selected by MCUVE and CARS based on the distribution of D values calculated using RMSEP. Our results show that model comparison based on a single split of the data into a training set and a corresponding test set may result in a wrong conclusion and the proposed MPA approach can avoid drawing such a wrong conclusion thus providing a solution to this problem.

4.2 Comparison of PCR, PLS and an ECR model

In chemometrics, PCR and PLS seem to be the most widely used method for building a calibration model. Recently, we developed a method, called elastic component regression (ECR), which utilizes a tuning parameter $\alpha \in [0,1]$ to supervise the decomposition of X-matrix [36], which falls into the category of continuum regression [37-40]. It is demonstrated theoretically that the elastic component resulting from ECR coincides with principal components of PCA when $\alpha = 0$ and also coincides with PLS components when $\alpha = 1$. In this context, PCR and PLS occupy the two ends of ECR and $\alpha \in (0,1)$ will lead to an infinite number of transitional models which collectively uncover the model path from PCR to PLS. The source codes implementing ECR in MATLAB are freely available at [41]. In this section, we would like to compare the predictive performance of PCR, PLS and an ECR model with $\alpha = 0.5$.

We still use the corn protein data described in Section 4.1. Here we do not consider all the variables but only the 28 wavelengths selected by CARS. For the proposed method, the number of Monte Carlo simulations is set to 1000. At each simulation 60% samples selected randomly are used as training samples and the remaining serve as test samples. The number of latent variables (LVs) for PCR, PLS and ECR ($\alpha = 0.5$) is chosen using 5-fold cross validation.

Figure 9 shows the three distributions of RMSEP computed using PCR, PLS and ECR ($\alpha = 0.5$). The mean and standard deviations of these distributions are 0.1069±0.0140, 0.1028±0.0111 and 0.0764±0.0108, respectively. Obviously, PLS achieves the lowest prediction errors as well as the smallest standard deviations. In contrast, PCR performs the

worst. As a transitional model that is between PCR and PLS, ECR with α = 0.5 achieves the medium level performance.

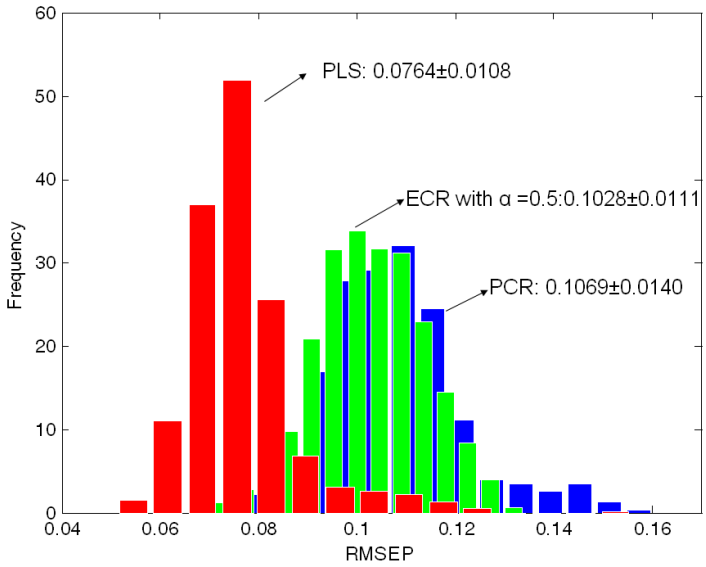


Fig. 9. The distributions of RMSEP from PCR, PLS and an ECR model with α =0.5

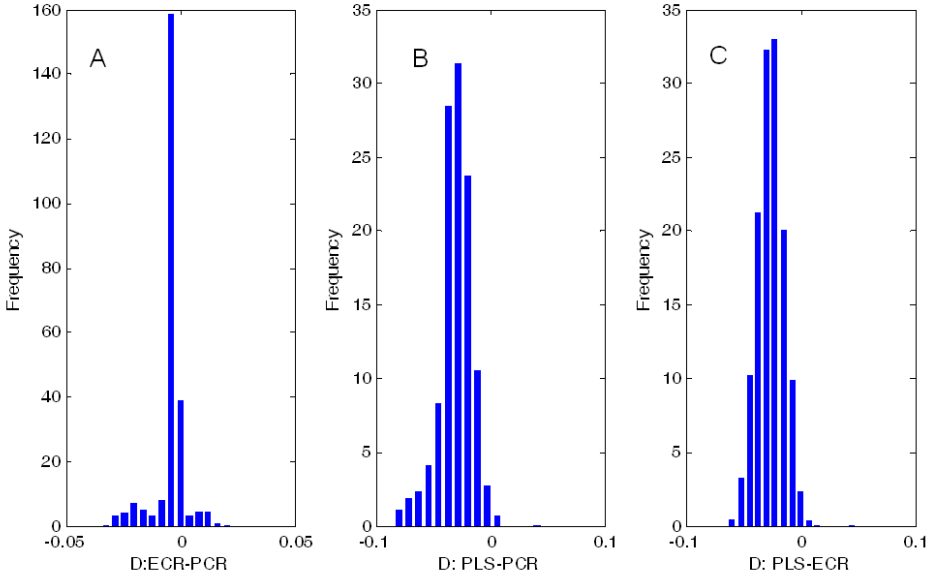


Fig. 10. The distributions of D values. The P values of t test for these three distributions are 0, 0 and 0, respectively.

The distributions of D values are displayed in **Figure 10**. The means of these three distributions are -0.0041 (Plot A), -0.0305 (Plot B) and -0.0264 (Plot C), respectively. Using a two-side t test, it is shown that all these three distributions of D values have a mean value that is significant not zero with P values equal to 0 , 0 and 0 for Plot A, Plot B and Plot C. To conclude, this section provides illustrative examples for the comparison of different modeling methods. Our example demonstrates that PLS (an ECR model associated with α = 1) performs better than PCR (an ECR model associated with α = 0) and a specific transitional ECR model associated with α = 0.5 has the moderate performance.

4.3 Comparison of PLS-LDA models before and after variable selection

Partial least squares-linear discriminant analysis (PLS-LDA) is frequently used in chemometrics and metabolomics/metabonomics for building predictive classification models and/or biomarker discovery [32, 42-45]. With the development of modern high-throughput analytical instruments, the data generated often contains a large number of variables (wavelengths, m/z ratios etc). Most of these variables are not relevant to the problem under investigation. Moreover, a model constructed using this kind of data that contain irrelevant variables would not be likely to have good predictive performance. Variable selection provides a solution to this problem that can help select a small number of informative variables that could be more predictive than an all-variable model.

In the present work, two methods are chosen to conduct variable selection. The first is t-test, which is a simple univariate method that determines whether two samples from normal distributions could have the same mean when standard deviations are unknown but assumed to be equal. The second is subwindow permutation analysis (SPA) which was a model population analysis-based approach proposed in our previous work [14]. The main characteristic of SPA is that it can output a conditional P value by implicitly taking into account synergistic effects among multiple variables. With this conditional P value, important variables or conditionally important variables can be identified. The source codes in Matlab and R are freely available at [46].We apply these two methods on a type 2 diabetes mellitus dataset that contains 90 samples (45 healthy and 45 cases) each of which is characterized by 21 metabolites measured using a GC/MS instrument. Details of this dataset can be found in reference [32].

Using t-test, 13 out of the 21 variables are identified to be significant ($P < 0.01$). For SPA, we use the same setting as described in our previous work [14]. Three variables are selected with the aid of SPA. Let V_0, V_1 and V_2 denote the sets containing all the 21 variables, the 13 variables selected by t-test and the 3 variables selected by SPA, respectively. To run the proposed method, we set the number of Monte Carlo simulations to 1000. At each simulation 70% samples are randomly selected to build a PLS-LDA model with the number of latent variables optimized by 10-fold cross validation. The remaining 30% samples working as test sets on which the misclassification error is computed.

Figure 11 shows the distributions of misclassification errors computed using these three variable sets. The mean and standard deviations of these distributions are 0.065±0.048 (all variables), 0.042±0.037 (t-test) and 0.034±0.034 (SPA), respectively. It can be found that the models using selected variables have lower prediction errors as well as higher stability in terms of smaller standard deviations, indicating that variable selection can improve the

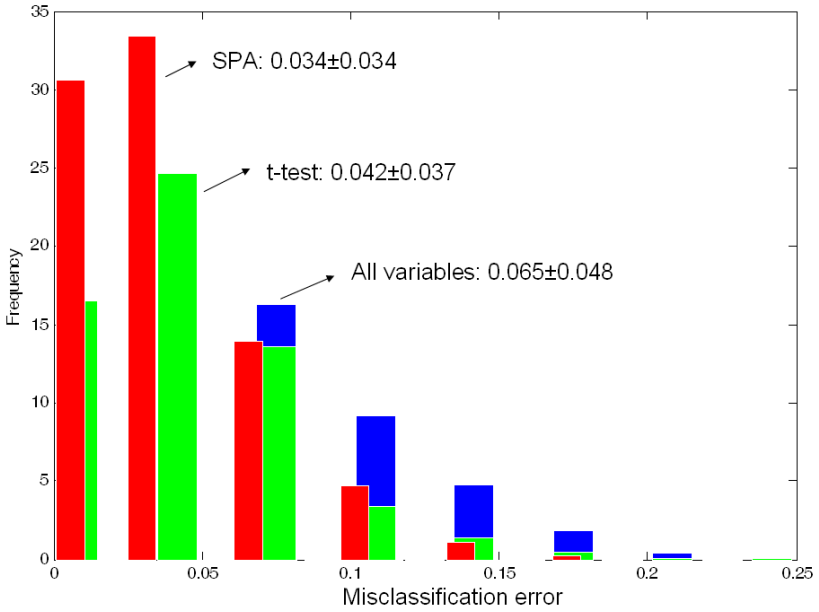


Fig. 11. The distributions of misclassification error on 1000 test sets using all variables and variables selected by t test and SPA, respectively.

performance of a classification model. The reason why SPA performs better than t-test is that synergistic effects among multiple variables are implicitly taken into account in SPA while t-test only considers univariate associations.

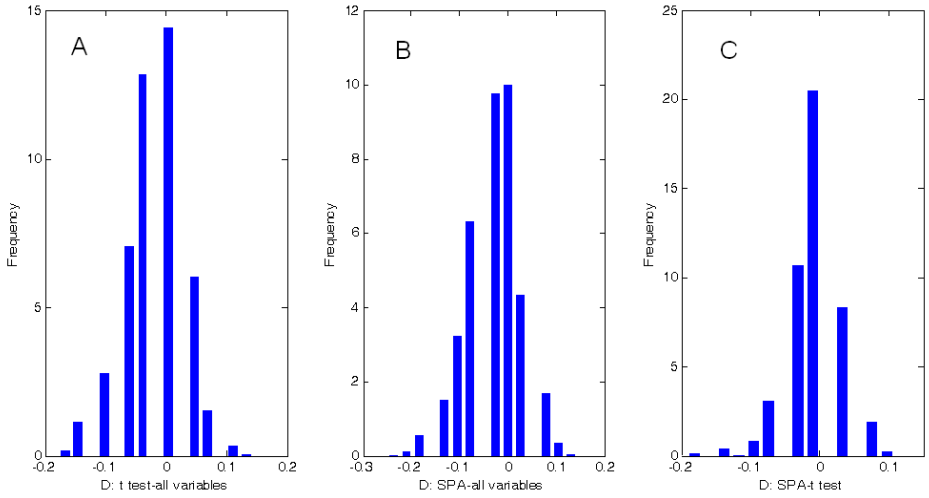


Fig. 12. The distributions of D values. The P values of t test for these three distributions are 1.66×10^{-46}, 1.02×10^{-57} and 1.27×10^{-8}.

We conducted pair-wise comparison of performances of the three variable sets described above. The distribution of D values (t-test – all variables) is shown in Plot A of **Figure 12**. The mean of D is -0.023 and is demonstrated to be significantly not zero (P=0) using a two-side t test, suggesting the improvement of variable selection. In spite of this improvement, we should also notice that a percentage (17.3%) of D values is positive, which again imply that model comparison based on a single split of the data into a training set and a corresponding test set is risky. However, with the aid of this MPA-based approach, it is likely to reliably compare different models in a statistical manner.

The distribution of D values (SPA – all variables) is shown in Plot B of **Figure 12**. The mean of D is -0.031 and is shown to be not zero (P = 0). Also, 171 D values are positive, again indicating the necessity of the use of a population of models for model comparison. In analogy, Plot C in **Figure 12** displays the distributions of D values (SPA-t-test). After applying a two-side t-test, we found that the improvement of SPA over t-test is significant (P= 0). For this distribution, 22.7% D values is positive, indicating that based on a random splitting of the data t-test will have a 22.7% chance to perform better than SPA. However, based on a large scale comparison, the overall performance of SPA is statistically better than t-test.

To conclude, in this section we have compared the performances of the original variable set and variable sets selected using t-test and SPA. We found evidences to support the use of the proposed model population analysis approach for statistical model comparison of different classification models.

5. Conclusions

A model population analysis approach for statistical model comparison is developed in this work. From our case studies, we have found strong evidences that support the use of model population analysis for the comparison of different variable sets or different modeling methods in both regression and classification. P values resulting from the proposed method in combination with the sign of the mean of D values clearly shows whether two models have the same performance or which model is significantly better.

6. Acknowledgements

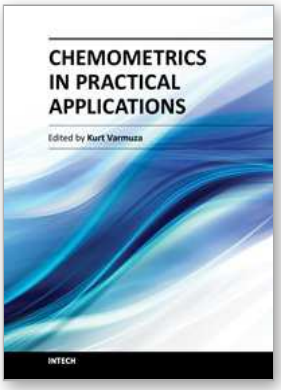
This work was financially supported by the National Nature Foundation Committee of P.R. China (Grants No. 20875104 and No. 21075138) and the Graduate degree thesis Innovation Foundation of Central South University (CX2010B057).

7. References

[1] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control, 19 (1974) 716.
[2] G.E. Schwarz, Estimating the dimension of a model, Annals of Statistics, 6 (1978) 461.
[3] S. Wold, Cross-validatory estimation of the number of components in factor and principal component analysis, Technometrics, 20 (1978) 397.
[4] Q.-S. Xu, Y.-Z. Liang, Monte Carlo cross validation, Chemometr. Intell. Lab., 56 (2001) 1.
[5] P. Filzmoser, B. Liebmann, K. Varmuza, Repeated double cross validation, J Chemometr, 23 (2009) 160.

[6] J. Shao, Linear Model Selection by Cross-Validation, J Am. Stat. Assoc., 88 (1993) 486.
[7] M. Stone, Cross-validatory choice and assessment of statistical predictions, J. R. Stat. Soc. B, 36 (1974) 111.
[8] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, Model population analysis for variable selection, J. Chemometr., 24 (2009) 418.
[9] B. Efron, G. Gong, A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation, Am. Stat., 37 (1983) 36.
[10] B. Efron, R. Tibshirani, An introduction to the bootstrap, Chapman&Hall, (1993).
[11] S. Wold, M. Sjöström, L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemometr. Intell. Lab., 58 (2001) 109.
[12] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, Support vector machines and its applications in chemistry, Chemometr. Intell. Lab., 95 (2009) 188
[13] D.S. Cao, Y.Z. Liang, Q.S. Xu, H.D. Li, X. Chen, A New Strategy of Outlier Detection for QSAR/QSPR, J. Comput. Chem., 31 (2010) 592.
[14] H.-D. Li, M.-M. Zeng, B.-B. Tan, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, Recipe for revealing informative metabolites based on model population analysis, Metabolomics, 6 (2010) 353.
[15] Q. Wang, H.-D. Li, Q.-S. Xu, Y.-Z. Liang, Noise incorporated subwindow permutation analysis for informative gene selection using support vector machines, Analyst, 136 (2011) 1456.
[16] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, B.-B. Tan, B.-C. Deng, C.-C. Lin, Recipe for Uncovering Predictive Genes using Support Vector Machines based on Model Population Analysis, IEEE/ACM T Comput Bi, 8 (2011) 1633.
[17] A.I. Bandos, H.E. Rockette, D. Gur, A permutation test sensitive to differences in areas for comparing ROC curves from a paired design, Statistics in Medicine, 24 (2005) 2873.
[18] Y. Ai-Jun, S. Xin-Yuan, Bayesian variable selection for disease classification using gene expression data, Bioinformatics, 26 (2009) 215.
[19] A. Vehtari, J. Lampinen, Bayesian model assessment and comparison using cross-validation predictive densities, Neural Computation, 14 (2002) 2439.
[20] T. Chen, E. Martin, Bayesian linear regression and variable selection for spectroscopic calibration, Anal. Chim. Acta, 631 (2009) 13.
[21] L. Breiman, Bagging Predictors, Mach. Learn., 24 (1996) 123.
[22] Y. Freund, R. Schapire, Experiments with a new boosting algorithm, Machine Learning: Proceedings of the Thirteenth International Conference, (1996) 148.
[23] L. Breiman, Random Forests, Mach. Learn., 45 (2001) 5.
[24] K. Ting, J. Wells, S. Tan, S. Teng, G. Webb, Feature-subspace aggregating: ensembles for stable andÂ unstable learners, Mach. Learn., 82 (2010) 375.
[25] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Statist., 18 (1947) 50.
[26] V. Centner, D.-L. Massart, O.E. de Noord, S. de Jong, B.M. Vandeginste, C. Sterna, Elimination of Uninformative Variables for Multivariate Calibration, Anal. Chem., 68 (1996) 3851.
[27] W. Cai, Y. Li, X. Shao, A variable selection method based on uninformative variable elimination for multivariate calibration of near-infrared spectra, Chemometr. Intell. Lab., 90 (2008) 188.

[28] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, Key wavelengths screening using competitive adaptive reweighted sampling method for multivariate calibration, Anal. Chim. Acta, 648 (2009) 77.
[29] J.-H. Jiang, R.J. Berry, H.W. Siesler, Y. Ozaki, Wavelength Interval Selection in Multicomponent Spectral Analysis by Moving Window Partial Least-Squares Regression with Applications to Mid-Infrared and Near-Infrared Spectroscopic Data, Anal. Chem., 74 (2002) 3555.
[30] C. Reynes, S. de Souza, R. Sabatier, G. Figueres, B. Vidal, Selection of discriminant wavelength intervals in NIR spectrometry with genetic algorithms, J. Chemometr., 20 (2006) 136.
[31] Q.-J. Han, H.-L. Wu, C.-B. Cai, L. Xu, R.-Q. Yu, An ensemble of Monte Carlo uninformative variable elimination for wavelength selection, Anal. Chim. Acta, 612 (2008) 121.
[32] B.-B. Tan, Y.-Z. Liang, L.-Z. Yi, H.-D. Li, Z.-G. Zhou, X.-Y. Ji, J.-H. Deng, Identification of free fatty acids profiling of type 2 diabetes mellitus and exploring possible biomarkers by GC–MS coupled with chemometrics, Metabolomics, 6 (2009) 219.
[33] W. Fan, H.-D. Li, Y. Shan, H.-Y. Lv, H.-X. Zhang, Y.-Z. Liang, Classification of vinegar samples based on near infrared spectroscopy combined with wavelength selection, Analytical Methods, 3 (2011) 1872.
[34] Source codes of CARS-PLS for variable selection: http://code.google.com/p/carspls/
[35] Source codes of CARS-PLSLDA for variable selection: http://code.google.com/p/cars2009/
[36] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, Uncover the path from PCR to PLS via elastic component regression, Chemometr. Intell. Lab., 104 (2010) 341.
[37] M. Stone, R.J. Brooks, Continuum Regression: Cross-Validated Sequentially Constructed Prediction Embracing Ordinary Least Squares, Partial Least Squares and Principal Components Regression, J. R. Statist. Soc. B, 52 (1990) 237.
[38] A. Bjorkstrom, R. Sundberg, A generalized view on continuum regression, Scand. J. Statist., 26 (1999) 17.
[39] B.M. Wise, N.L. Ricker, Identification of finite impulse response models with continuum regression, J. Chemometr., 7 (1993) 1.
[40] J.H. Kalivas, Cyclic subspace regression with analysis of the hat matrix, Chemometr. Intell. Lab., 45 (1999) 215.
[41] Source codes of Elastic Component Regression: http://code.google.com/p/ecr/
[42] M. Barker, W. Rayens, Partial least squares for discrimination, J. Chemometr., 17 (2003) 166.
[43] J.A. Westerhuis, H.C.J. Hoefsloot, S. Smit, D.J. Vis, A.K. Smilde, E.J.J. van Velzen, J.P.M. van Duijnhoven, F.A. van Dorsten, Assessment of PLSDA cross validation, Metabolomics, 4 (2008) 81.
[44] L.-Z. Yi, J. He, Y.-Z. Liang, D.-L. Yuan, F.-T. Chau, Plasma fatty acid metabolic profiling and biomarkers of type 2 diabetes mellitus based on GC/MS and PLS-LDA, FEBS Letters, 580 (2006) 6837.
[45] J. Trygg, E. Holmes, T.r. Lundstedt, Chemometrics in Metabonomics, Journal of Proteome Research, 6 (2006) 469.
[46] Source codes of Subwindow Permutation Analysis: http://code.google.com/p/spa2010/



Chemometrics in Practical Applications

Edited by Dr. Kurt Varmuza

ISBN 978-953-51-0438-4
Hard cover, 326 pages
Publisher InTech
Published online 23, March, 2012
Published in print edition March, 2012

In the book "Chemometrics in practical applications", various practical applications of chemometric methods in chemistry, biochemistry and chemical technology are presented, and selected chemometric methods are described in tutorial style. The book contains 14 independent chapters and is devoted to filling the gap between textbooks on multivariate data analysis and research journals on chemometrics and chemoinformatics.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821