Universidade de Trás-os-Montes e Alto Douro

Analysis of Extraction Feature Methodologies in Data Sets

Licenciatura em Engenharia Informática

Projeto em Engenharia Informática

Author

Marcelo Queirós Pinto

Advisor

Pedro José de Melo Teixeira Pinto

Co-Advisor

Véronique Medeiros Gomes

Vila Real, 2017

Abstract

The abundance of data generated in hyperspectral images compels more robust methods for the processing of complex processes in real time. The solution involves the development of feature extraction methodologies, allowing the computational load to be reduced, identifying relevant spectral bands to the problem in question and eliminating the remaining ones, keeping almost all the information involved.

With this purpose, two feature extraction methodologies were developed: Competitive Adaptive Reweighted Sampling (CARS) and Successive Projections Algorithm (SPA).

The ending results shown that CARS algorithm allowed selection of more informative variables because all the tests have a higher R^2 and lower RMSE than the tests made by the SPA algorithm. However, the SPA algorithm can select less variables, allowing a better reduction of the computational load, containing as best R^2 and RMSECV only 17 variables

Index

Illustrations Index

1. Introduction

In winemaking, precision and process automation is increasingly important and, therefore, technologies to support the evaluation of grape ripening status represent an asset in the competitive market of wine production.

The hyperspectral image allows the non-intrusive evaluation of oenological parameters related to the grape ripening, providing complex and large data sets that will allow to accurately estimate these parameters. However, the abundance of data generated compels more robust methods for the processing of complex processes in real time. The solution involves the development of feature extraction methodologies, allowing the computational load to be reduced, identifying relevant spectral bands to the problem and eliminating the remaining ones, keeping almost all the information involved.

With this purpose, two feature extraction methodologies were developed: Competitive Adaptive Reweighted Sampling (CARS) and Successive Projections Algorithm (SPA). The proposed methodologies were developed in the Matlab computational platform.

2. Goals

- ✓ N. º 1: Review of some different methodologies for feature extraction and feature selection;
- ✓ N. º 2: Development of the first feature extraction model - CARS (Competitive Adaptive Reweighted Sampling);
- ✓ N. º 3: Development of the second feature extraction model - SPA (Successive Projections Algorithm);
- ✓ N. º 4: Comparison of methodologies developed.

3. Spectral Feature Selection and Feature Extraction Methodologies

3.1. Principal Component Analysis

Principal component analysis (PCA) is an eigenvector-based algorithm, is commonly used as an exploratory method for feature selection in hyperspectral images. PCA may be understood as maximizing the variance of the projection coordinates. In PCA, the original data with high collinearity can be rapidly concentrated into a smaller set of principal component score images (PCs) that are linear transformations of all original variables. The main goal of PCA is to find a group of orthogonal basis for better expressing the original spectral data matrix X. Then PCA breaks apart X into a principal component matrix called the eigenvectors or loadings (F), in which the dimensions are orthogonal to each other and with maximum variations, and the corresponding scaling coefficient matrix called the scores (S) as the following equation:

$$X = SF + e_a,$$

where X is an n x k matrix of spectral data, S is an n x f matrix of score values for all the spectra, and F is an f x k matrix of eigenvectors. e_a is the residual spectra matrix, while n is the number of samples, k is the number of wavelengths, and f is the number of principal components.

The score values in S determine the importance of original variables in the principal components. Therefore, the variables having high score values can be considered as the effective wavelengths that represent the most variance and contribution of data set [5].

3.2. Uninformative Variable Elimination

UVE is a method of wavelength selection method based on regression coefficients of PLS regression model. It eliminates wavelengths that provide no or little information to the established regression model by setting a threshold. The main steps of UVE are summarized as follows [7]:

Step 1: A PLS regression model is established based on spectral data X and their corresponding reference parameter value y.

Step 2: The regression coefficient of PLS model for each

wavelength is calculated.

Step 3: The reliability of each wavelength is analysed by the following formula:

$$C_i = \frac{\text{mean}(b_i)}{s(b_i)},$$

where mean (bi) and s(bi) are the mean value and standard deviation of the regression coefficient (b) of wavelength (i) from PLS, respectively. The larger the reliability C is, the more important the corresponding wavelength becomes. Once a cutoff value (threshold) is established, the variable with reliability below the threshold will be eliminated, and the higher ones will be retained. Advantages of UVE include its simple procedure, fast execution, convenience, and high accuracy. UVE can eliminate the variables that have no more informative variables for modeling than noise. Besides, employing variables selected by UVE to establish models can avoid the risk of overfitting and improve the accuracy of predicting. However, UVE is not suitable for selecting wavelengths directly because the number of wavelength that UVE gets is still large. Therefore, a further selection for the most important variables is critical [5].

3.3. Successive Projections Algorithm

SPA uses the vector projection analysis for finding the minimum of redundant information contained variable group. At the same time, it improves the speed and rate of model by reducing the number of variables [2].

The basic principle of SPA is that it first begins with one wavelength, and then incorporates another one at each iteration until a specially defined number (N) of wavelengths is finished. The aim of this method is to choose a set of wavelengths, which are the most representative, for solving collinearity problems. Several steps of SPA are needed as described below, assuming that the first wavelength k (0) and number N are known [5]:

Step 0: Before the first iteration ($n = 1$), let $x_j = j$th column of $X_{cal}; j = 1, \ldots, J$.

Step 1: Let S be the set of wavelengths, which have not been selected yet. That is, $S = \{j$ such that $1 \le j \le J$ and $j \notin \{k(0), \ldots, k(n-1)\}$.

Step 2: Calculate the projection of x_j on the subspace orthogonal to $x_{k(n-1)}$ as

$$Px_j = x_j - \left(x_j^T x_{k(n-1)}\right) x_{k(n-1)} \left(x_{k(n-1)}^T\right) \left(x_{k(n-1)}^T x_{k(n-1)}\right)^{-1} \quad (8)$$

Step 3 : Let $k(n) = \arg\left(\max ||Px_j||, j \in S\right)$, for all $j \in S$, (9)

where P is the projection operator.

Step 4: Let $x_j = Px_j, j \in S$.

Step 5: Let $n = n + 1$. If $n < N$, then go back to Step 1.

End: The resulting wavelengths are $\{k(n); n = 0, \ldots, N-1\}$.

3.4. Competitive Adaptive Reweighted Sampling

Another random search method for wavelength selection is competitive adaptive reweighted sampling (CARS), which is based on the simple but effective principle "survival of the fittest" [5].

Generally speaking, there are six steps in each sampling run of CARS: (1) k samples are selected by applying Monte Carlo strategy; (2) a calibration model, taking an example of PLS, is built using the selected k samples and the regression coefficient of each variable is produced; (3) exponentially decreasing function is used to remove the wavelengths with relatively small absolute regression coefficients in a stepwise and efficient way; (4) adaptive reweighted sampling (ARS) is applied to realize a further competitive selection of wavelengths, which is similar to the ''survival of the fittest'' principle in Darwin's Evolution Theory; (5) cross-validation is utilized to evaluate the performance of each subset; and (6) the subset with the lowest RMSECV value is selected as the feature subset.

3.5. Artificial Neural Networks

ANN has been widely used for machine learning and pattern recognition. Generally speaking, ANN consists of three layers: an input layer, an output layer, and a hidden layer. The nodes of input layer represent the spectral responses of each wavelength from full spectrum. Usually, a small number of nodes on hidden layer might be set due to the great number of wavelengths. However, the volume of output layer depends on the practical needs for prediction or classification.

The importance of each variable (wavelength) for the ANN model can be calculated by the following equation [8]:

$$M = \frac{\sum_{j=1}^{n_H} \left[\left(\frac{|I|_{p_j}}{\sum_{k=1}^{n_p} |I|_{p_{j,k}}} \right) |O|_j \right]}{\sum_{i=1}^{n_p} \left(\sum_{j=1}^{n_H} \left[\left(\frac{|I|_{p_{i,j}}}{\sum_{k=1}^{n_p} |I|_{p_{i,j,k}}} \right) |O|_j \right] \right)}$$

where M is the index measuring the importance of input variable, np is the number of input variables, nH is the number of hidden layer nodes, [I]pj is the absolute value of the hidden layer weight corresponding to the pth input variable and the jth hidden layer, and [I]j is the absolute value of the output layer weight corresponding to the jth hidden layer. The index M value is calculated for each input node and then normalized into the 0–1 range. The higher the M value, the more important the node (wavelength) is for the classification or prediction model [5].

3.6. Genetic Algorithms

Genetic algorithms (GA) assume that better results, achieved from experimental conditions, will prevail over the worst ones. At the same time, an improvement can be obtained by some sort of recombination together with some random changes. From this perspective, experimental conditions are considered as the genome, of which genes are the variables taking part in the process. Then the fitness of each experimental condition is measured by an optimized response. Generally speaking, GA made up of five elemental steps: (1) coding of all variables; (2) initiation of population; (3) evaluation of the responses; (4) reproductions; and (5) mutations. Steps 3–5 would be alternated until a termination criterion is achieved. The criterion can be either based on a lack of

improvement in the response or simply on a maximum number of generations or on the total time allowed for the elaboration [9].

Due to the randomness nature of GA, the selected wavelengths might be different during different implementations. Therefore, it is necessary to execute GA programs repeatedly to determine the initial wavelength candidates, which are under the assumption that the common wavelengths selected by different runs of GA have great importance in accounting for the targets of interest [5].

4. Results and Discussion

4.1. Methodologies operation

4.1.1. CARS

Figure 4 shows the flowchart of the CARS algorithm to describe its operation. In the algorithm code, each step of the flowchart is easily identified.

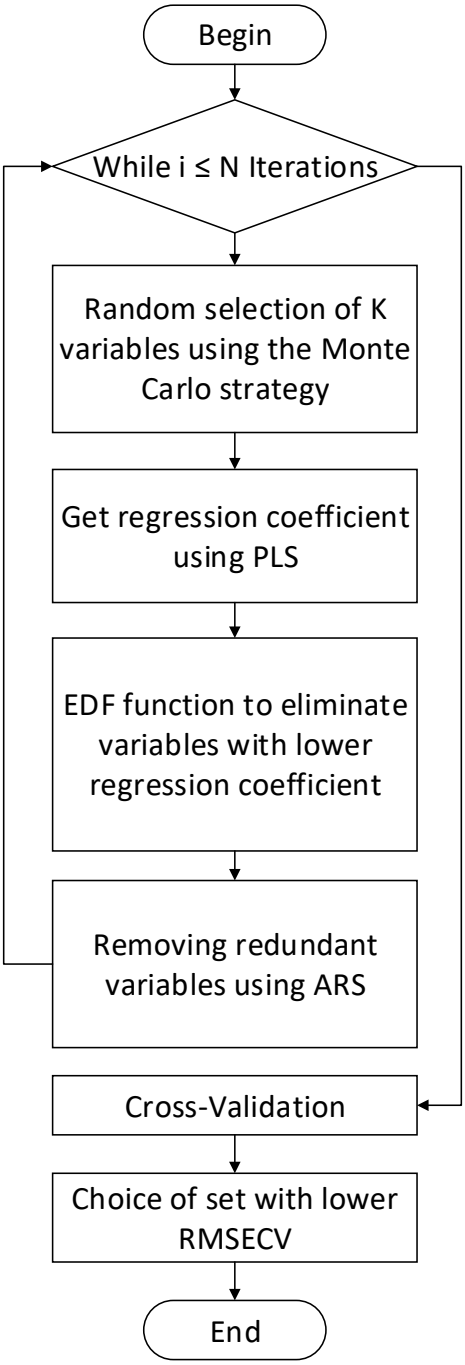

Figure 1: Flowchart of CARS algorithm

4.1.2. SPA

Figure 5 shows the flowchart of the SPA algorithm to describe its operation. In the algorithm code, each step of the flowchart is easily identified.

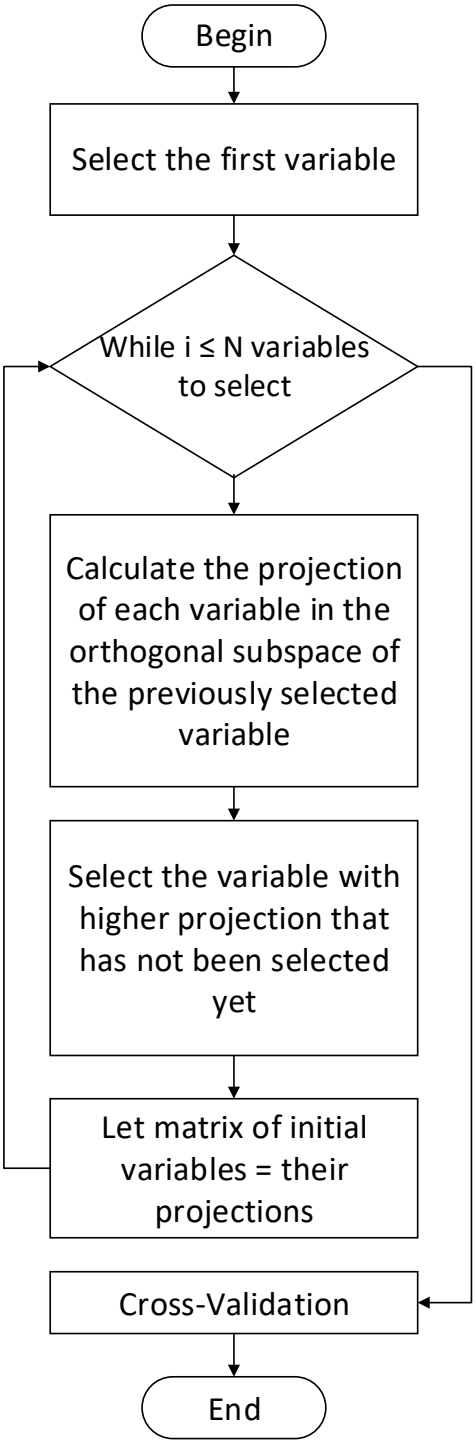

Figure 2: Flowchart of SPA algorithm

4.2. Performed tests

In the CARS methodology were made 1000 tests (implementing BootStrap) with the purpose of eliminating the randomness of possible cases and finding the best results.

The SPA methodology requires that the first variable be given to it, then, more tests were performed with the purpose of giving all possible variables as first for each possible number of variables.

Unlike the CARS methodology, the SPA methodology allows a N (variable number) minimal and a N maximal, and the methodology shows all the results for all N's between the N minimal and N maximal. Tests were performed between N minimal=2 and N maximal =100, so the number of tests performed were (100 - 2 + 1) x 1040 possible variables = 102960 in SPA.

The tests were performed only on the Brix oenological parameter.

4.3. Matlab Files and Input/Output Variables

This section explains attached files and the inputs/outputs for SPA and CARS methodologies.

The files ScriptCARS.m and ScriptSPA.m are the files that execute the algorithms and where the following input variables are defined:

- **A (set to 20)**: the maximal principle to extract;
- **Fold (set to 7)**: the group number for cross validation;
- **Method (set to 'center')**: pre-treatment method;

Just in case of CARS:

- **Num (set to 200)**: the number of Monte Carlo Sampling runs;
- **BootStrap (set to 1000):** number of tests;

Just in case of SPA:

- **Nmin (set to 2):** minimum variables number to select;
- **Nmax (set to 100):** maximum variables number to select.

The files ScriptCARS.m and ScriptSPA.m call the algorithms in files CARSproject.m and SPAproject.m respectively. Other files were used by the algorithms as functions and they are in the same folder.

The output variables contains all RMSE and all R^2 results for cross-validation, the selected variables, minimum RMSECV and maximum R^2, variables number, latent variable number, selected iteration and time. The struct for CARS and SPA is called

Result and for BootStrap is called ResultBootStrap. In the BootStrap results, besides iteration selected (minimum RMSECV) and the other cited outputs, also contains the iteration with less R^2 and the iteration with less number variables, which are useful for future analysis.

4.4. Influence of number of Monte Carlo sampling runs on performance of CARS

To know the influence of number of Monte Carlo sampling runs on the performance of CARS, the following five cases were taken into consideration in which the number of sampling runs was individually set as 50, 100, 200, 500 and 1000 times. Results of statistical box-plots are shown in Figure 3. There is no obvious evidence that the number of Monte Carlo sampling runs has any significant influence on the performance of CARS. Therefore, the number of Monte Carlo sampling runs was set to 200 times.

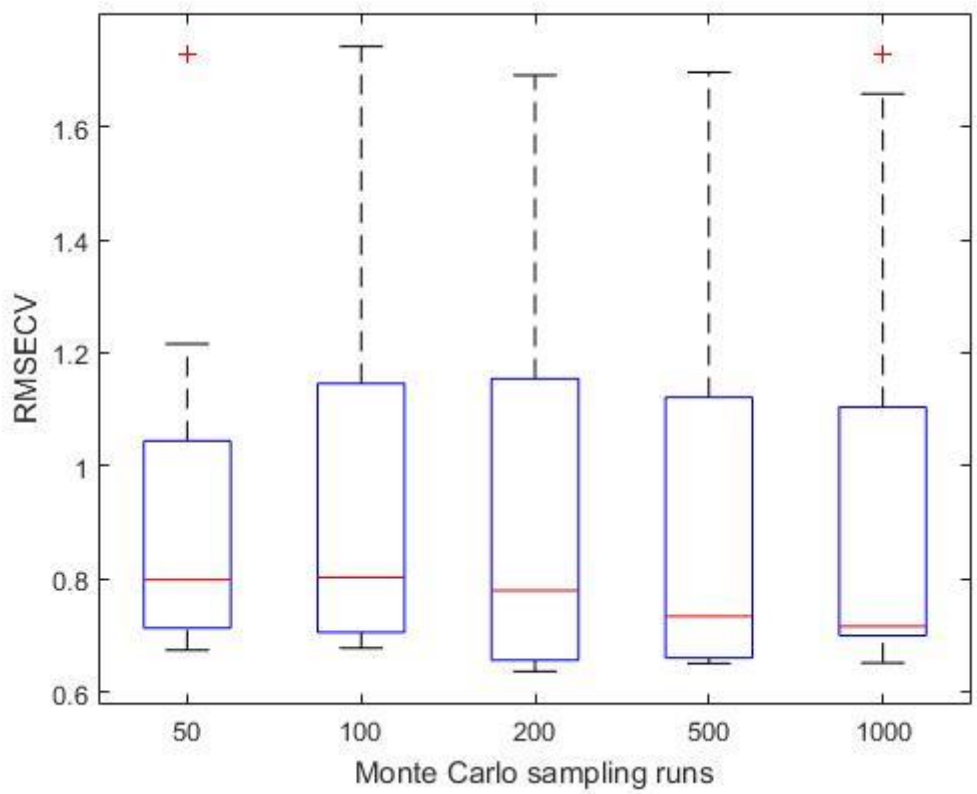

Figure 3: Box-plots to the number of Monte Carlo Sampling runs of CARS set as 50, 100, 200, 500 and 1000 and the corresponding RMSECV

4.5. Variable selection frequencies of tests realized with CARS

Figure 4 shows the frequencies of variables number selected by CARS. The maximum was 143 and the minimum was 44, the average was 83 selected variables.

For more BootSrap iterations, the selected variables number may probably drop more. For these results, 1000 tests of CARS were performed.

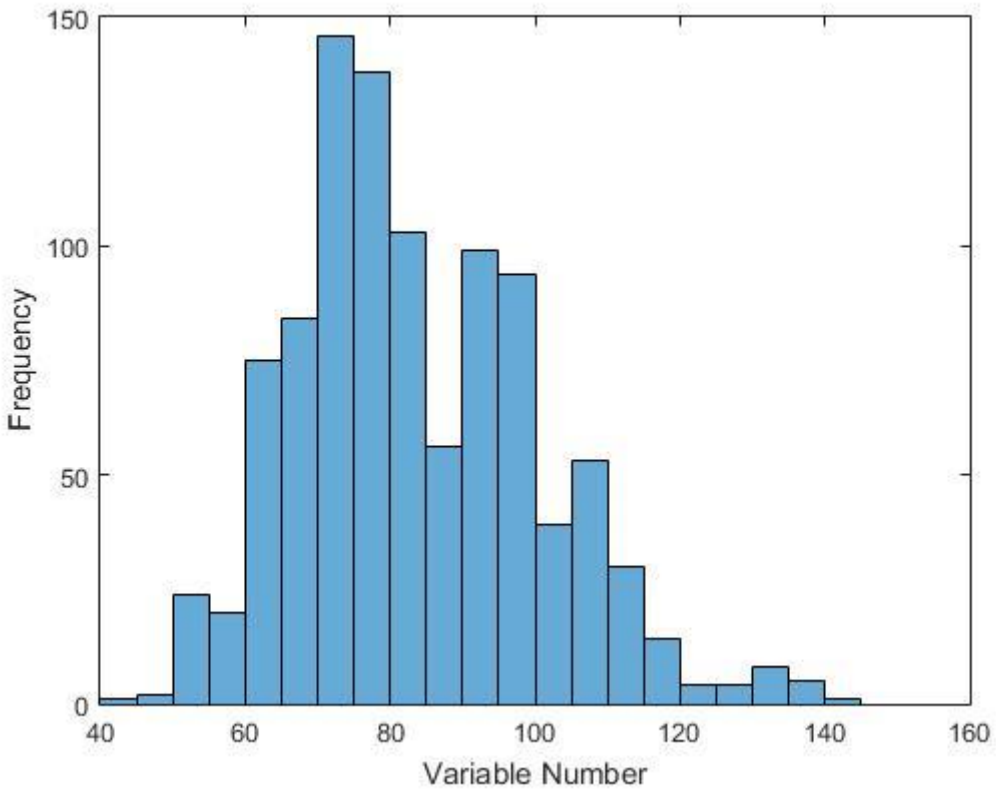

Figure 4: Variable selection frequencies of tests realized with CARS

4.6. RMSECV analysis in tests of CARS algorithm

Figure 5 shows the RMSECV for each CARS test and corresponding Variable Number Selected. The result with the lowest RMSECV was 0.58200 for 77 selected variables and is represented in red. However, other results are interesting as the result painted in green, with 44 selected variables and RMSECV=0.66083.

It was verified that all CARS results reach good levels of RMSECV.

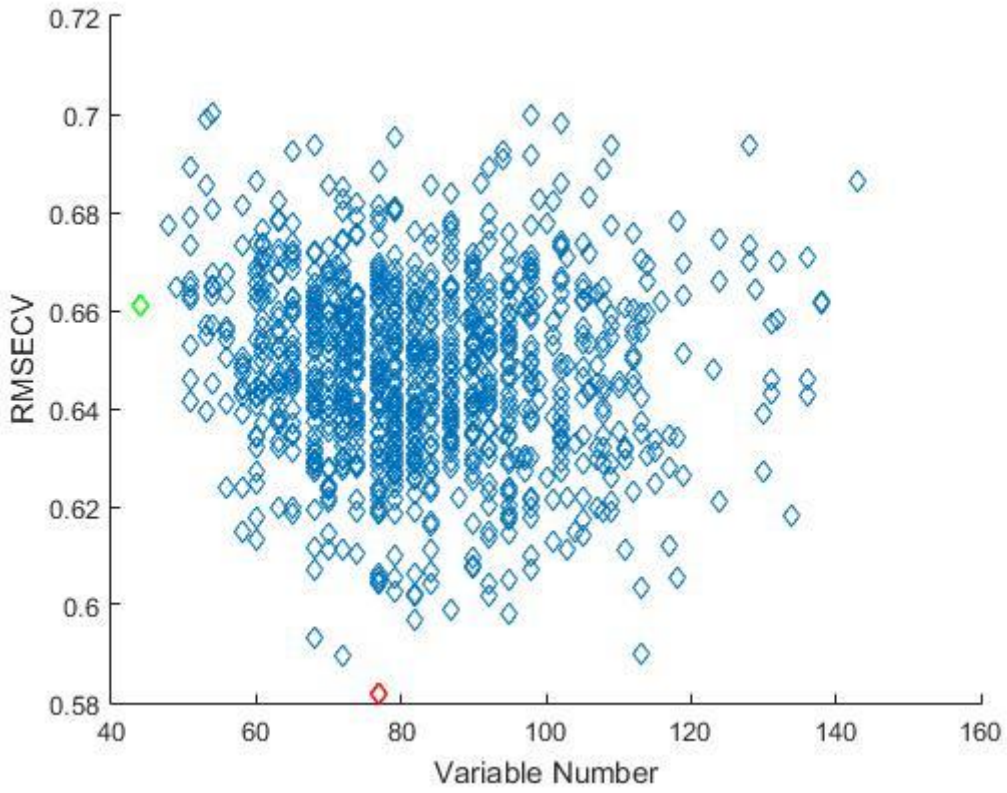

Figure 5: RMSECV for each CARS test and corresponding Variable Number

4.7. RMSECV analysis in tests of SPA algorithm

Figure 6 shows the RMSECV for each SPA test and corresponding Variable Number Selected. The result with the lowest RMSECV was 1.09199 for 17 selected variables and is represented in red.

In comparison with the results of the CARS algorithm, the RMSECV results of the SPA algorithm are significantly worse.

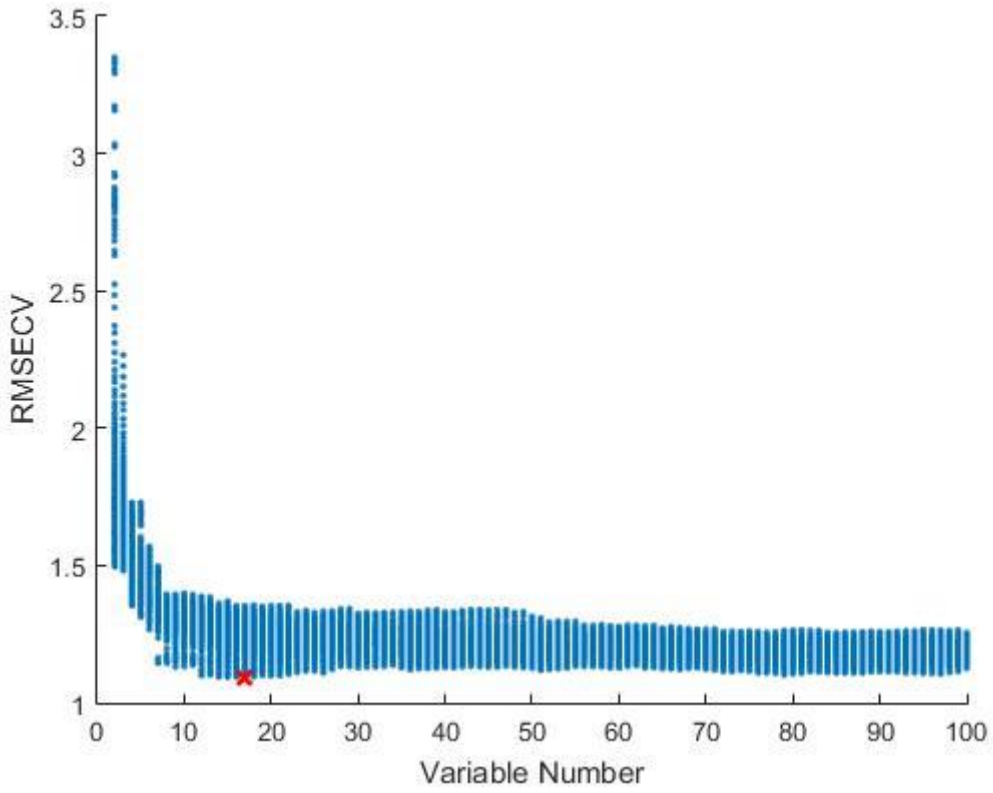

Figure 6: RMSECV for each SPA test and corresponding Variable Number

4.8. RMSECV analysis in tests of CARS and SPA algorithms

Figure 7 shows the comparison of R^2 of both algorithms.

The blue circle surrounds the result of the SPA algorithm with higher R^2 and lower RMSECV, and has 17 variables. The red circle surrounds the result of the CARS algorithm with higher R^2 and lower RMSECV, and has 77 variables. The green circle surrounds the result of the CARS algorithm with lower number variables and interesting results of R^2 and RMSECV. The results are shown below.

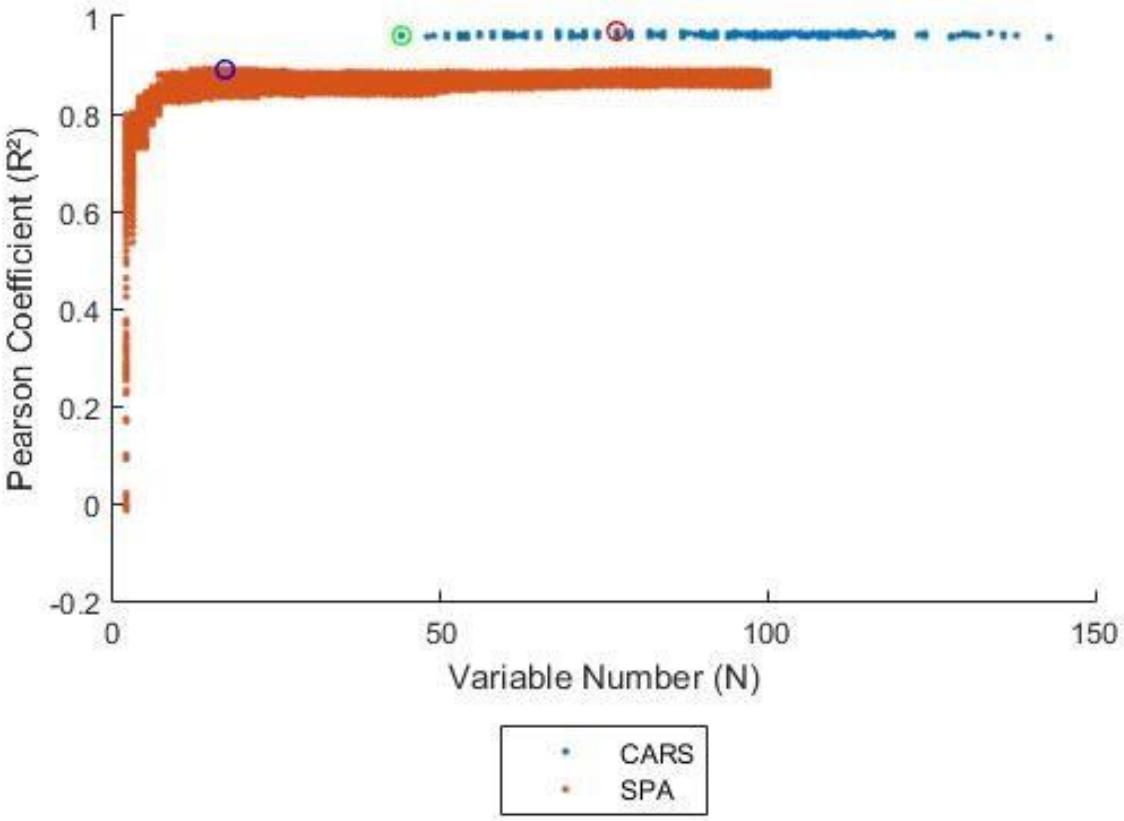

Figure 7: R^2 for all tests of CARS and SPA and corresponding Variable Number

.

Result with higher R^2 and RMSECV of SPA:

R^2: 0.892213116961390

RMSECV: 1.091992688414810

Variable number: 17

Result with higher R^2 and RMSECV of CARS:

R^2: 0.969381724276501

RMSECV: 0.582005566876979

Variable number: 77

Interesting R^2 and RMSECV of CARS and lower variable number of CARS:

R^2: 0,960526797493914

RMSECV: 0,660827268040320

Variable number: 44

5. Conclusion

The CARS algorithm allowed selection of more informative variables because all the tests have a higher R^2 and lower RMSE than the tests performed by the SPA algorithm. However, the SPA algorithm can select less variables, allowing a better reduction of the computational load, containing as best R^2 and RMSECV only 17 variables.

The conclusions are that in cases where the CARS methodology does not obtain a sufficient dimensional reduction, the SPA methodology is a good option. In the other cases, the CARS methodology is more beneficial because it selects more informative data.

6. References

[1] Tang, G., Huang, Y., Tian, K., Song, X., Yan, H., Hu, J., Min, S. (2014). A new spectral variable selection pattern using competitive adaptive reweighted sampling combined with successive projections algorithm. The Analyst, 139(19), 4894. doi:10.1039/c4an00837e

[2] Araújo, M. C., Saldanha, T. C., Galvão, R. K., Yoneyama, T., Chame, H. C., & Visani, V. (2001). The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. Chemometrics and Intelligent Laboratory Systems, 57(2), 65-73. doi:10.1016/s0169-7439(01)00119-8

[3] Goudarzi, N., & Goodarzi, M. (2010). Application of successive projections algorithm (SPA) as a variable selection in a QSPR study to predict the octanol/water partition coefficients (Kow) of some halogenated organic compounds. Analytical Methods, 2(6), 758. doi:10.1039/b9ay00170k

[4] Wu, D., He, Y., Shi, J., & Feng, S. (2009). Exploring Near and Midinfrared Spectroscopy to Predict Trace Iron and Zinc Contents in Powdered Milk. Journal of Agricultural and Food Chemistry, 57(5), 1697-1704. doi:10.1021/jf8030343

[5] Dai, Q., Cheng, J., Sun, D., & Zeng, X. (2014). Advances in Feature Selection Methods for Hyperspectral Image Processing in Food Industry Applications: A Review. Critical Reviews in Food Science and Nutrition, 55(10), 1368-1382. doi:10.1080/10408398.2013.871692

[6] Bin, J., Ai, F., Fan, W., Zhou, J., Li, X., Tang, W., & Liang, Y. (2016). An efficient variable selection method based on variable permutation and model population analysis for multivariate calibration of NIR spectra. Chemometrics and Intelligent Laboratory Systems,158, 1-13. doi:10.1016/j.chemolab.2016.08.006

[7] Centner, V., Massart, DL., Noord, O., Jong, S, Vandeginste, B. M. and Sterna, C. (1996). Elimination of uninformative variables for multivariate calibration. Anal. Chem. 68: 3851–3858

[8] Vemuri, V. (1988). Artificial neural networks. IEEE Comput. Soc. Technol. Ser.. 26:224–233

[9] Leardi, R. and Gonzalez, A. L. (1998). Genetic algorithms applied to feature selection in PLS regression:how and when to use them. Chemom. Intell. Lab. Syst. 41:195–207