Soil Property Prediction: An Extreme Learning Machine Approach

Dina Masri
Wei Lee Woon
Zeyar Aung

Technical Report DNA #2015-02

May 2015

Data & Network Analytics Research Group (DNA)
Electrical Engineering and Computer Science,
Masdar Institute of Science and Technology,
PO Box 54224, Abu Dhabi, UAE.

Soil Property Prediction: An Extreme Learning Machine Approach

Dina Masri, Wei Lee Woon, and Zeyar Aung

Department of Electrical Engineering and Computer Science
Masdar Institute of Science and Technology, Abu Dhabi, UAE.
{dmasri,wwoon,zaung}@masdar.ac.ae

Abstract. In this paper, we propose a method for predicting functional properties of soil samples from a number of measurable spatial and spectral features of those samples. Our method is based on Savitzky-Golay filter for preprocessing and a relatively recent evolution of single hidden-layer feed-forward network (SLFN) learning technique called extreme learning machine (ELM) for prediction. We tested our method with Africa Soil Property Prediction dataset, and observed that the results were promising.

Keywords: Soil property, Prediction, Neural network, Extreme learning machine (ELM), Kernel-based ELM

1 Introduction

Computational prediction of soil properties is an important task in modern agricultural and environmental studies. It allows us to perform low-cost analysis on the measurable soil features in order to forecast the soil's functional properties like primary productivity, nutrient and water retention, and resistance to soil erosion. These properties are important for planning sustainable agricultural intensification and natural resources management. The best available low cost analysis can be done using diffuse reflectance infrared spectroscopy measurements and geo-referencing of soil samples. Using spectroscopy, the amount of light absorbed by a soil sample is measured at different wavelengths to provide an infrared spectrum of the soil sample.

In this work, we propose a method to first preprocess the data using Savitzky-Golay filter, and then build an effective predictive model using a newly emerging single hidden-layer feed-forward neural network learning (SLFN) technique called "extreme learning machine (ELM)". We tested our method with "Africa Soil Property Prediction Challenge" dataset [9], and observed that the results were promising with low prediction error rates and low standard deviations of errors.

2 Problem Definition

Our objective is to predict of 5 target soil "functional properties" from the 16 spatial and the 3,566 spectral features which are relatively easy to measure. (Note: we follow the same objective as in the Africa Soil Property Prediction Challenge [9], which was sponsored by Africa Soil Information Service [1].)

The explanatory input variables are the spectral and spatial features of the soil sample. The spectral features consists of a range of 3,578 Near-infrared (NIR) absorbance measurements at different wavelengths ranging from (599.76–7497.96 cm^{-1}). (Note: among them, 12 CO_2 spectral features ranging between 2352.76 and 2379.76 cm^{-1} are removed as suggested by the experts since this area of the spectrum picks up atmospheric CO_2 absorption features that are not due to the soil sample [9]. After this removal, 3,566 spectral features remain.) The spatial features are from environmental and remote sensing data sources as well as the depth from which the soil sample is taken.

Table 1. Variables of soil samples.

Variable Name	Description
16 Spatial Variables	
BSAN	Near-infrared average long-term Black Sky Albedo measurement
BSAS	Shortwave average long-term Black Sky Albedo measurement
BSAV	Visible average long-term Black Sky Albedo measurement
CTI	Compound topographic index
ELEV	Elevation
EVI	Average long-term enhanced vegetation index
LSTD	Average long-term land surface day time temperature
LSTN	Average long-term land surface day night time temperature
REF1	Average long-term Reflectance measurement for blue
REF2	Average long-term Reflectance measurement for red
REF3	Average long-term Reflectance measurement for near-infrared
REF7	Average long-term Reflectance measurement for mid-infrared
RELI	Topographic relief
TMAP	Mean annual precipitation (rainfall)
TMFI	Modified Fournier index of rainfall
Depth	Depth of soil sample ("Topsoil" for 0–20 cm depth and "Subsoil" for 20–50 cm depth)
3,578 Spectral Variables	
m599.76–m7497.96	Mid-infrared absorbance measurements at different wavelengths (599.76–7497.96 cm^{-1}) [Note: 12 of them are removed later.]
5 Target Variables (Soil Functional Properties)	
CA	Mehlich-3 extractable Calcium
P	Mehlich-3 extractable Phosphorus
pH	pH value
SOC	Soil organic carbon
Sand	Sand content

The five target variables are (1) Mehlich-3 extractable Calcium (Ca), (2) Mehlich-3 extractable Phosphorus (P), (3) pH value (pH), (4) Soil organic carbon (SOC), and (5) Sand content (Sand). A summary of the variables are presented in Table 1.

The dataset we used was the one provided in the Africa Soil Property Prediction Challenge [9]. It contains 1,158 training instances (soil samples) and 728 test instances. We did not use the test instances in our study because the values of their target variable were not publicly available. It should be also noted that geographical clustering was observed in the raw dataset [10], which was due to the spatially stratified multi-level sampling design that was used during the assembling phase of the data. For that reason, the data were randomized (shuffled) before the training and cross validation processes.

3 Proposed Method

3.1 Data Preprocessing

As mentioned in Section 2, the training dataset is of very high dimensionality (3,582 input features) with respect to the number of training instances (1,158). This calls for preprocessing of the data before entering it to the predictive model training process. Since the spectral features comprise 3,566 NIR variables, our main focus is to preprocess them and reduce their number.

According to Rinnan et al. [15], preprocessing of NIR spectral data is an integral part of the any predictive modeling involving this type of data. It is essential to remove any physical phenomena in the NIR spectra in order to retrieve the most important information within. In general, the spectra is highly influenced by non-linearities caused by light scatter of the sample under study. There are two main methods used for NIR preprocessing [15], namely, scatter correction and spectral derivatives. The first method involves statistical analysis such as multiplicative scatter correction. The second method, which is the simplest, involves smoothing the spectra then performing a derivative (usually of first or second order) to decrease the signal to noise ratio.

One of the most famous smoothing filters used for spectral derivative preprocessing is called Savitzky-Golay smoothing filter [16]. This filter basically fits the NIR spectra to a polynomial, then average out the resulting fitted model. It uses a windowed data with odd number of points to fit the polynomial. Up until this step, only the measurement noises are smoothed out, as shown in Figure 1, where a sample spectral raw data is plotted with its Savitzky-Golay smoothed version plotted over it. To change the NIR data from absolute values to relative rate of change values, a first order derivative (differencing) is performed after the smoothing step. The resulting differenced spectral data is presented in Figure 2. It can be noticed in the figure that some parts of the spectra, like the area in the red box, are of no use and most probably carry no information. Removing such areas of the spectra serves as a means of discarding irrelevant features, thus achieving dimensionality reduction.

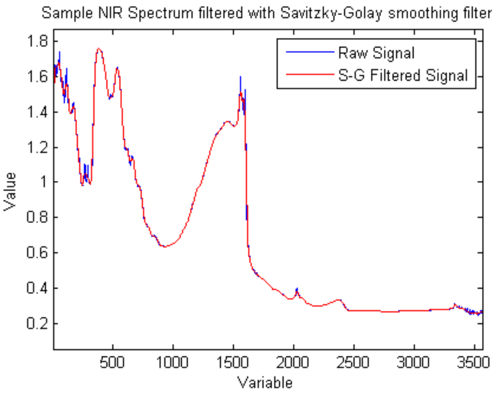

Fig. 1. Savitzky-Golay smoothed NIR sample spectra.

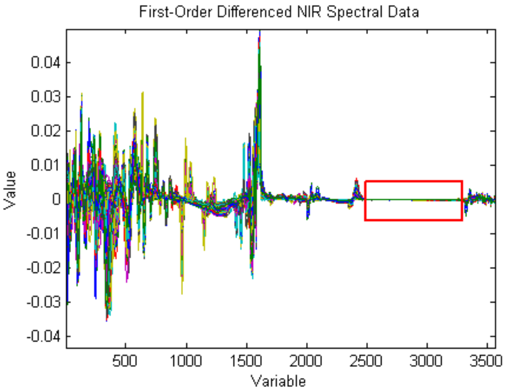

Fig. 2. First order differenced NIR training data.

3.2 Prediction Using Extreme Learning Machine (ELM)

Neural network has proven to provide effective performances in various applications over the years. Single hidden layer feed-forward network (SLFN) is a powerful category of neural network. A SLFN with any continuous bounded and nonlinear activation function can form decision regions with arbitrary shapes in multi-dimensional cases. That means an SLFN can approximate any continuous function and implement any classification application [5]. To overcome the slowness of learning in neural networks and long iterative process for parameter tuning, based on the findings in [5], Huang et al. [7] proposed a new evolutionary learning technique called extreme learning machine (ELM). It can train any SLFN with exceptional speed (thousands of times faster than the traditional technique) and extreme generalization abilities.

These improvements in ELM are based on the observation that the mapping parameters between the input and the hidden layer are not correlated to the output, so they are set randomly at the beginning of the training process. Moreover, this learning technique tends to result in the smallest training errors as well as the smallest norm of weights. That means not only it is targeted at the minimization of error, but also on the minimization of the norm of the tuning parameters. This specific property is in accordance with Bartlett's theory [2] that the generalization ability of any feed-forward neural network is much better with smaller weights or tuning parameters.

Given a training set $X = [(\mathbf{x}_i, \mathbf{y}_i) \,|\, \mathbf{x}_i \in \mathbb{R}^n, \mathbf{y}_i \in \mathbb{R}^m, i = 1, \ldots, N]$, any non-zero activation function $g(\mathbf{x})$, and the number of hidden nodes (neurons) L, the ELM algorithm can be summarized in the following three steps [7].

1. Assign arbitrary (randomly selected) input weight $\mathbf{w}_i$ and bias $b_i, i = 1, \ldots, L$.
2. Calculate the hidden layer output matrix H.
3. Calculate the output weight β, according to the equation:

$$\beta = H^{\dagger} Y \tag{1}$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse of matrix of SLFN's hidden layer output matrix H, which is in turn defined as:

$$H_{(N \times L)} = \begin{pmatrix} g(\mathbf{w}_1.\mathbf{x}_1 + b_1) & ... & g(\mathbf{w}_L.\mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1.\mathbf{x}_N + b_1) & ... & g(\mathbf{w}_L.\mathbf{x}_N + b_L) \end{pmatrix}$$

and

$$\beta_{(L \times m)} = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{pmatrix} \quad , \; Y_{(L \times m)} = \begin{pmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_L^T \end{pmatrix}$$

H is constant throughout the training since the input mapping is done randomly and is fixed. So, training an SLFN using ELM can simply be done by finding a least-squares solution $\widehat{\beta}$ of the linear system $H\beta = Y$, and this solution is equivalent to $\widehat{\beta} = H^{\dagger} Y$. This learning technique is not iterative and the solution is unique. Thus, the risk of local minima does not exist.

To improve the generalization performance robustness, a regularization parameter (C) can be added to Equation 1 [6]. Now, the new solution is:

$$\widehat{\beta} = (\frac{I}{C} + H^{\dagger} H)^{-1} H^{\dagger} H \tag{2}$$

ELM can also be extended to kernel learning [6] because it can use any type of feature mapping (between the hidden layer and the output) including kernels. Here, the output function of ELM becomes:

$$f(\mathbf{x}) = h(\mathbf{x}) H^{\dagger} (\frac{I}{C} + H^{\dagger} H)^{-1} Y = \begin{pmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{pmatrix}^T (\frac{I}{C} + \mathit{\Omega}_{ELM})^{-1} Y \tag{3}$$

where $\Omega_{ELM} = HH^{\dagger} : \Omega_{ELM}[i,j] = h(\mathbf{x}_i) \cdot h(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$, with Ω_{ELM} being the kernel matrix. $K(\mathbf{x}_i, \mathbf{x}_j)$ can be any kernel function like Gaussian, polynomial, linear, etc. Here, the feature mapping function $h(\mathbf{x})$ is not needed to be known by the users, instead its kernel is given. Moreover, unlike the basic ELM where the feature mapping from the hidden layer to the output $h(\mathbf{x})$ is done using any activation function, here the number of hidden nodes (L) is not needed to be known either.

4 Results and Discussions

In order to implement the ELM-based prediction model to solve the soil property prediction task at hand, we used the codes [8] provided by the same research group who developed ELM. Two major ELM algorithms were tested: (1) basic ELM and (2) kernel-based ELM.

For the basic ELM, the Sigmoid activation function was used since it is the most prominent. (It was concluded in [6] that the training time spent by ELM with Sigmoid additive nodes increases very slowly when the number of training data increases.) We varied two main parameters, namely, (1) the number of nodes (neurons) in the hidden layer, L and (2) the ELM regularization parameter, C.

For the kernel-based ELM, the Gaussian kernel was chosen. We tuned two main parameters, namely, (1) the Gaussian kernel parameter, γ and (2) the ELM regularization parameter, C.

For each unique parameter setting, ten-fold cross validation (CV) was performed on the training set of 1,158 instances from the Africa Soil Property Prediction Challenge [9] dataset. For the basic Sigmoid-based ELM, the parameter values tested are: $L = \{10, 50, 100, 200, 500, 700, 1000\}$ and $C = \{0.001, 0.01, 0.1, 1, 10, 50, 100, 1000, 5000, 10000\}$, thus 70 different CV trials were performed. For the Gaussian kernel-based ELM, the following values were used: $C = \{0.001, 0.01, 0.1, 0.2, 0.5, 1, 5, 10, 20, 50, 100, 1000, 10000\}$ and $\gamma = \{0.001, 0.01, 0.1, 0.2, 0.4, 0.8, 1, 5, 10, 50, 100, 1000, 10000\}$, requiring 169 different CV trials.

The results were evaluated using the mean column-wise root mean squared error (MCRMSE) metric, where the errors of each ten-fold CV trials were averaged over $T = 115$ or 116 test instances across $m = 5$ target variables. y_{ij} and $\hat{y}_{ij}$ stand for the actual and the predicted values respectively.

$$\text{MCRMSE} = \frac{1}{m} \sum_{j=1}^{m} \sqrt{\frac{1}{T} \sum_{i=1}^{T} (y_{ij} - \hat{y}_{ij})} \qquad (4)$$

Moreover, we used the mean cross validation standard deviation (MCVSTD) metric, where the average standard deviation of each ten-fold CV test was measured by averaging over all the target variables for each parameter change in the prediction models.

The results of the ten-fold CV tests over the different parameter values are depicted in Figures 3 and 4. A Matlab 3-D plot was created for each of the basic Sigmoid-based and the Gaussian kernel-based models showing the effect of the regularization parameter as well as the model specific parameter (i.e., L and γ, respectively). Tables 3 and 2 show samples of the best CMRMSE values obtained for the both models given specific parameter combinations, as well as the values of MCVSTD.

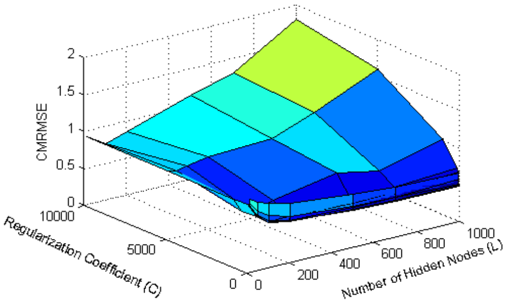

Fig. 3. Basic Sigmoid-based ELM's results for different values of C and L.

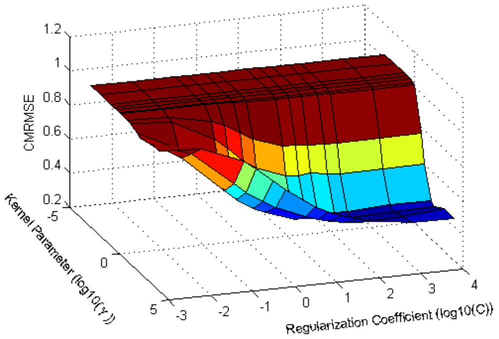

Fig. 4. Gaussian kernel-based ELM's results for different values of C and γ.

In Figure 3, it can be observed that there is no strong dependency of the performance on the combination of (C, L) in the basic Sigmoid-based ELM. As long as L is relatively high, the model performs well. (This conclusion was also obtained in [6].) On the contrary, for the Gaussian kernel-based ELM, Figure 4 shows that the model performance is much more sensitive on the combination of (C, γ).

Regarding the prediction accuracy, under the provided tuning of parameters, the Gaussian kernel-based model performed better with the CMRMSE value as low as 0.3937 with 0.1529 standard deviation of the ten-fold CV. The average

Table 2. Sample results obtained by different combinations of (L, C) parameters for basic Sigmoid-based ELM. The minimum (best) values are highlighted.

L	C	**CMRMSE**	**MCVSTD**
1000	0.01	0.5077	0.1786
1000	0.10	**0.4724**	**0.1342**
1000	1.00	0.4985	0.1419
1000	10.00	0.5792	0.2490
700	0.01	0.5331	0.1675
700	0.10	0.4899	0.1586
700	1.00	0.4987	0.1494
700	10.00	0.5435	0.1402
700	50.00	0.5967	0.1640
500	0.01	0.5533	0.2202
500	0.10	0.5014	0.1511
500	1.00	0.5149	0.1385
500	10.00	0.5438	0.1615
500	50.00	0.5893	0.2212
500	100.00	0.6162	0.2677
200	0.10	0.5720	0.1855
200	1.00	0.5660	0.1952
200	10.00	0.5703	0.1529
200	100.00	0.6035	0.1541

Table 3. Sample results obtained by different combinations of (γ, C) parameters for Gaussian kernel-based ELM. The minimum (best) values are highlighted.

γ	C	**CMRMSE**	**MCVSTD**
10000	1000	0.4430	0.1637
1000	1000	0.4265	0.1294
1000	100	0.4185	0.1356
1000	50	0.4276	0.1696
1000	20	0.4432	0.1433
100	10000	0.4033	0.1528
100	1000	0.4017	**0.1285**
100	100	**0.3937**	0.1529
100	50	0.4073	0.1542
100	20	0.4199	0.1476
100	10	0.4214	0.1534
100	5	0.4143	0.1842
50	10000	0.4219	0.1376
50	1000	0.4234	0.1421
50	100	0.4129	0.1296
50	50	0.4116	0.1444
50	20	0.4299	0.2140
50	10	0.4291	0.2060
50	5	0.4293	0.1969

value of CMRMSE in the table is 0.4199. This beats the leaderboard top result in Africa Soil Property Prediction Challenge [11], which was 0.4689. However, we acknowledge that this is not an accurate comparison because our result is just a ten-fold CV error on the training data, but not on the test data used in the competition (because their actual target variable values are not publicly available). Nonetheless, our results do show the potential effectiveness for our proposed predictive model.

5 Related Work

Different machine learning algorithms have been tried and tested to solve the problem of soil property prediction [9]. These comprise multi-layer neural networks, Bayesian additive regression trees (BART), support vector machine (SVM), ridge regression, lasso regression, elastic net lasso, gradient boosting regressor, and many others [10]. The BART model was provided as an example with a benchmark CMRMSE of 0.56551. However, to our best knowledge, no one has tried ELM to solve this problem before.

ELM as a learning technique has a number of advantages over other state-of-the-art classification and regression algorithms. The main reasons are its generalization ability, unique solutions with minimized training error, and the ability to theoretically model any function or decision boarder no matter how complex it is. For example, as mentioned in [6], compared to two widely-used variants of support vector machine for regression, namely, least-square support vector machine (LS-SVM) and proximal support vector machine (P-SVM), ELM is subject to fewer and milder optimization constraints. Moreover, SVM sometimes may provide sub-optimal solutions, unlike ELM where it has a unique solution. ELM exhibits both better scalability and generalization performance.

Different integration attempts between SVM and the concept of ELM has been proposed in the literature [13, 4], where the concept of randomized feature spaces for SVM algorithms was introduced in order to enhance SVM's generalization performance. The basic idea is to use ELM to compute a kernel of the first layer of a SLFN, which in turn is used train the SVM.

ELM was also compared to another newly emerged neural network-based learning algorithm called deep learning or deep networks [3]. Deep learning outperforms all traditional classification/regression methods like multi-layer neural networks, SLFNs, and SVMs for big data analysis. However, the training process of deep learning is very slow. In [3], a new structure of ELM was proposed to resemble the deep learning process, but with a much faster learning speed. This structure consists of a multi-layer ELM (ML-ELM) which performs layer-by-layer unsupervised learning like deep learning does. This new learning algorithm is much faster than the existing deep learning techniques while maintaining a comparable predictive performance.

ELM was extended into an online sequential learning algorithm in [14]. The proposed algorithm can learn data one-by-one or by chunks with variable sizes. In [12], ELM was used in an application to recognize human actions with incremental learning using a very minimal number of video frames at a high speed.

ELM is still a newly emerging technique that needs a lot of exploration and enhancements, and its concepts are applicable to various learning techniques. In short, it is like a gold mine to dig into.

6 Conclusion

In this paper, we have proposed a predictive modeling technique for predicting the functional properties of soil samples based on their spatial and spectral features. First, the spectral part of the data was pre-processed using a smoothing step with Savitzky-Golay filter, followed by a first order differencing step to get rid of the physical effects and emphasize on the spectral properties. Then, a relatively new and very promising SLFN algorithm called extreme learning machine (EML) was used for predictive modeling of the soil properties. We have tried two variations of EML, namely, basic Sigmoid-based EML and Gaussian kernel-based EML. When tested on the Africa Soil Property Prediction Challenge dataset, both methods offer good prediction results with low prediction error rates and low standard deviations of errors. Therefore, we believe that our proposed method can be practically useful in many agricultural and environmental applications, which have to deal with the soil's functional properties.

References

1. AfSIS: Africa soil information service (2014), http://africasoils.net/
2. Bartlett, P.L.: The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE Transactions on Information Theory 44(2), 525–536 (1998)
3. Cambria, E., Huang, G.B., Kasun, L.L.C., et al.: Extreme learning machines [trends & controversies]. IEEE Intelligent Systems 28(6), 30–59 (2013)
4. Frénay, B., Verleysen, M.: Using SVMs with randomised feature spaces: An extreme learning approach. In: Proceedings of the 2010 European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. pp. 315–320 (2010)
5. Huang, G.B., Chen, Y.Q., Babri, H.A.: Classification ability of single hidden layer feedforward neural networks. IEEE Transactions on Neural Networks 11(3), 799–801 (2000)
6. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42(2), 513–529 (2012)
7. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: A new learning scheme of feedforward neural networks. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks. vol. 2, pp. 985–990. IEEE (2004)
8. Huang, G.B., et al.: ELM: H2O R Interface. Nanyang Technological University, Singapore (2004), http://www.ntu.edu.sg/home/egbhuang/elm_codes.html
9. Kaggle: Africa soil property prediction challenge (2014), https://www.kaggle.com/c/afsis-soil-properties/
10. Kaggle: Africa soil property prediction challenge - forums (2014), https://www.kaggle.com/c/afsis-soil-properties/forums
11. Kaggle: Africa soil property prediction challenge - leaderboard (2014), https://www.kaggle.com/c/afsis-soil-properties/leaderboard
12. Liang, N.Y., Huang, G.B., Saratchandran, P., Sundararajan, N.: A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Transactions on Neural Networks 17(6), 1411–1423 (2006)
13. Liu, Q., He, Q., Shi, Z.: Extreme support vector machine classifier. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol. 5012, pp. 222–233. Springer (2008)
14. Minhas, R., Mohammed, A.A., Wu, Q.M.J.: Incremental learning in human action recognition based on snippets. IEEE Transactions on Circuits and Systems for Video Technology 22(11), 1529–1541 (2012)
15. Rinnan, Å., van den Berg, F., Engelsen, S.B.: Review of the most common preprocessing techniques for near-infrared spectra. TrAC Trends in Analytical Chemistry 28(10), 1201–1222 (2009)
16. Savitzky, A., Golay, M.J.E.: Smoothing and differentiation of data by simplified least squares procedures. Analytical Chemistry 36(8), 1627–1639 (1964)