Modeling the Transformation of Olive Tree Biomass into Bioethanol with Reg-CO^{2}RBFN

Francisco Charte Ojeda[1], Inmaculada Romero Pulido[2],
Antonio Jesús Rivera Rivas[1(✉)], and Eulogio Castro Galiano[2]

[1] Department of Computer Science, University of Jaén, Jaén, Spain
{fcharte,arivera}@ujaen.es
[2] Department of Chemical, Environmental and Materials Engineering,
University of Jaén, Jaén, Spain
{iromero,ecastro}@ujaen.es

Abstract. Research in renewable energies is a global trend. One remarkable area is the biomass transformation into biotehanol, a fuel that can replace fossil fuels. A key step in this process is the pretreatment stage, where several variables are involved. The experimentation for determining the optimal values of these variables is expensive, therefore it is necessary to model this process. This paper focus on modeling the production of biotehanol from olive tree biomass by data mining methods. Notably, the authors present Reg-CO^{2}RBFN, an adaptation of a cooperative-competitive designing method for radial basis function networks. One of the main drawbacks in this modeling is the low number of instances in the data sets. To compare the results obtained by Reg-CO^{2}RBFN, other well-known data mining regression methods are used to model the transformation process.

Keywords: Regression models · Data mining · Enzymatic hydrolisis · Olive tree biomass

1 Introduction

Nowadays, the interest for renewable energy is increasing [1]. Olive tree biomass (OTB) is an abundant organic residual in Mediterranean countries that can be converted in bioethanol. Due to the benefits of olive oil, olive tree cultivation is expanding worldwide. In these places olive trees are periodically pruned to rejuvenate them, as well as to prevent propagation of diseases. The process for obtaining bioethanol from OTB has the following advantages: large availability of OTB, low cost, CO_2 emissions reduction when this fuel is used, and a decreasing of dependency on energy imports.

The basic steps in the production of bioethanol from OTB are pretreatment, enzymatic hydrolysis and fermentation. One of the main steps is the pretreatment stage [2], whose performance can be affected by several factors such as processing time, temperature and the use of different salts. Therefore, for the

I. Rojas et al. (Eds.): IWANN 2017, Part I, LNCS 10305, pp. 733–744, 2017.
DOI: 10.1007/978-3-319-59153-7_63

experts involved in the field it is important to adequately model or characterize this process. A key challenge in this modeling is the low number of data samples (instances) available in the experimentation, due to the high costs involved in obtaining them.

Data mining methods have been successfully used for regression tasks [3] even in the renewable energy field [4–7]. Together with Multilayer Perceptron (MLPs) [8], Radial Basis Function Networks (RBFNs) are one the best well-known and important Artificial Neural Network (ANN). The efficacy of RBFNs has been proved in many areas, including regression [9]. The authors have developed CO^2RBFN, a cooperative-competitive evolutionary RBFN design method [10] that has been successfully used for regression in the modeling of concentrated photovoltaic modules [5].

In this paper Reg-CO^2RBFN, an adaptation of CO^2RBFN, is applied to the problem of modeling the bioethanol conversion process from OTB. The main objective of this adaptation is facing the training with a low number of instances per data set. With this aim, a supervised clustering algorithm is introduced in the initialization step of the algorithm. Other recognized data mining methods, such as a MLP trained with the backpropagation technique and a Suppport Vector Machine, have been also used to model the same process. The performance of Reg-CO^2RBFN and these other methods will be compared.

This paper is organized as follows: Sect. 2 depicts the bioethanol production process. The Reg-CO^2RBFN method is described in Sect. 3. In Sect. 4 the experimental framework is outlined and the results obtained are presented. The results produced by the proposed method are compared with that obtained by other data mining models, as explained in this section. Finally, the conclusions appear in Sect. 5.

2 Bioethanol Production from Olive Tree Biomass

OTB is a lignocellulose material and it is considered a promising candidate to be transformed into renewable fuels, therefore substituting the well-known fossil fuels [1] at some extent. The pruning of olive trees generates a huge amount of biomass that is dismissed or must be eliminated, for example by burning it. Taking into account environmental considerations, the transformation of OTB into bioethanol implies several advantages. It is a clean way of eliminating organic residuals, and implies a net reduction of CO_2 emissions when this fuel is used. In addition, it reduces the dependency from fuel imports.

The process of converting a lignocellulose material into bioethanol includes three steps: pretreatment, enzymatic hydrolysis and fermentation. Pretreatment is a key step in the whole process and has a direct influence over the hydrolysis step [2]. The objective of this first step is to achieve maximal fermentation yields and rates. The use of $FeCl_3$, a metal salt, has demonstrated its efficacy in the pretreatment phase. This salt concentration, along with the settings for temperature and time, define the input variables of the pretreatment. From the pretreatment experiments, liquids and pretreated solids are obtained and separated by filtration. At this time, the content of glucose and hemicellulosic sugars in both fractions can be determined. Then, the pretreated solids are further submitted to enzymatic hydrolysis under standard conditions, and the enzymatic hydrolysis yield for each experiment (grams of glucose in the hydrolysate per gram of glucose in the pretreated material or in the original material) is evaluated.

This research was carried out in Jaén (Spain), collecting the OTB from its olive tree fields. Due to the costs involved in performing empirical pretreatment analysis, only twenty experiments were conducted. The Box-Behnken methodology was used to equally distribute the values of experiments (instances). For a further explanation see [11].

In summary, the objective is to model the process of transforming a lignocellulose material into bioethanol. In this model the input variables are: duration of the pretreatment (Time), its temperature (Temp) and the molar concentration of $FeCl_3$ (FeCl3). The output variables are the Enzymatic Hydrolysis Yields in Raw Material (EHYRM) and Enzymatic Hydrolysis Yields in Pretreated Material (EHYPM). The low number of the conducted experiments implies a challenge for most learning methods.

3 Reg-CO^{2}RBFN: Adaptation of CO^{2}RBFN for Regression Tasks

In this paper a hybrid evolutionary cooperative-competitive model for designing RBFNs is proposed. This section starts providing a brief introduction to this type of neural networks, in Sect. 3.1. Then, in Sect. 3.2, the detailed description of the proposed model, Reg-CO^{2}RBFN, is addressed.

3.1 Radial Basis Function Networks

RBFNs are an artificial neural network paradigm [12] with remarkable characteristics, such as simple topological structure and universal approximation ability [9]. The topology of an RBFN is composed by three feed-forward connected layers: an input layer with n nodes, a hidden layer with m neurons or RBFs, and an output layer with one node for regression problems.

The neurons in the hidden layer present a radially-symmetric basis activation function, $\phi_i : R^n \rightarrow R$, which can be defined with several shapes, being the Gaussian function (1) the most widely used.

$$\phi_i(\boldsymbol{x}) = e^{-(\|\boldsymbol{x}-\boldsymbol{c_i}\|/d_i)^2} \tag{1}$$

where $\boldsymbol{c_i} \in R^n$ is the center of the basis function ϕ_i, $d_i \in R$ is the width (radius), and $\|\|$ is typically the Euclidean norm on R^n. This expression is the one used in this paper as RBF.

The output neuron implements the weighted sum of RBF outputs in the hidden layer, as can be seen in Eq. 2, where w_i are the weights. Each one of them represents the contribution of one RBF to the output node.

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} w_i \phi_i(\boldsymbol{x}) \qquad (2)$$

The main goal in the RBFN design process consists in determining the centers and widths of the neurons (RBFs) in the hidden layer, as well as the linear output weights connecting the RBFs to the output neuron.

An important paradigm for designing RBFNs is Evolutionary Computation (EC) [13–15]. EC uses natural evolution and stochastic searching to design optimization algorithms. Concretely, EC evolves a population of individuals according to operators as mutation, recombination or selection, and each individual in the population receives a measure proportional to its quality, called fitness. Reviews on EC applied to RBFN design can be found in [16,17].

3.2 The Proposed Method, Reg-CO^{2}RBFN

As mentioned, the presented method is a cooperative-competitive evolutionary proposal for designing RBFNs. In this methodology each individual of the population corresponds to one RBF, and the entire population implements the whole solution. The individuals cooperate towards a definitive solution, but they must also compete for survival.

In this environment, in which the solution depends on the behavior of many components, the fitness of each individual is known as credit assignment. In order to measure the credit assignment of an individual three factors have been proposed: the RBF contribution to the network output, the error in the basis function radius and the degree of overlapping among RBFs.

The application of the operators is determined by a Fuzzy Rule-Based System (FRBS). The inputs of the FRBS are the three parameters used for credit assignment, and the outputs are the operators' application probability.

Algorithm 1. CO2RBFN algorithm main steps.

```
Initialize RBFN

while(Stop condition is not met)                    ▷ Training loop
    Evaluate RBFs
    Apply operators to RBFs
    Substitute the eliminated RBFs
    Select the best RBFs
```

In the adapted version presented in this work, the network initialization step has been changed. The new initialization is based on supervised clustering [18]. Taking into account the low number of instances, the aim is introducing a previous analysis of the data, before the evolutionary phase, that reinforces the learning process. The main steps of Reg-CO^{2}RBFN, detailed below, are shown in Algorithm 1 in pseudocode.

RBFN Initialization. To initialize the RBFN, an adapted version of the k-means algorithm [19] is used. K-means is a clustering that aims to partition the data set into k clusters. Each pattern belongs to a cluster and the cluster prototype is the mean of the patterns that belongs to this cluster. Thus, each RBF center, $\boldsymbol{c}_i$, is assigned to a previously calculated cluster. According to [18] the modification carried out consists in taking into account the input and the output features of the data set during the cluster determination process. The RBF widths, d_i, will be set to the average distance between the centers. Finally, the RBF weights, w_{ij}, are set to zero.

RBFN Training. To adjust the RBF weights the Singular Value Decomposition (SVD) algorithm [20] is used. SVD is a deterministic technique typically used in matrix resolution.

RBF Evaluation. In order to evaluate the fitness or credit assignment of each RBF ϕ_i in the cooperative-competitive environment, three parameters, a_i, e_i, o_i are defined:

- The contribution, a_i, of the RBF ϕ_i, $i = 1 \ldots m$, is determined by considering its maximum weight. An RBF with a low weight will have a low contribution and so on.
- The error measure, e_i, for each RBF ϕ_i, is obtained by calculating the Root Mean Square Error (RMSE) (3) inside its width:

$$e_i = \sqrt{\frac{\sum_{t=1}^{n} (f_t - y_t)^2}{n}} \tag{3}$$

  where f_t is the output of the model (2) for a point inside the width of RBF ϕ_i, y_t is the real output at the same point, and n is the number of points inside the RBF ϕ_i.
- The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i. This parameter is computed by taking into account the fitness sharing methodology [21], whose aim is to maintain the diversity in the population. This factor is expressed as indicated in Eq. 4, where o_{ij} measures the overlapping of the RBFs ϕ_i and ϕ_j, $j = 1 \ldots m$.

$$o_i = \sum_{j=1}^{m} o_{ij} \tag{4}$$

$$o_{ij} = \begin{cases} (1 - \|\phi_i - \phi_j\|/d_i) & if\ \|\phi_i - \phi_j\| < d_i \\ 0 & otherwise \end{cases} \tag{5}$$

Applying Operators to RBFs. Four operators are defined in order to be applied to the RBFs:

- Operator Remove: eliminates an RBF.
- Operator Random Mutation: the number of coordinates to modify of the selected RBF is randomly generated between 1 and 25% of the total number of coordinates. The chosen coordinates are modified in a percentage between 5% and 25% of the old width. The width is adjusted in a similar way to the coordinates.
- Operator Biased Mutation: modifies the width and some coordinates of the center, between 1 and 25% of the total number of coordinates, using local information of the RBF environment. Concretely, the patterns inside the RBF width are determined and the new coordinates are allocated in the average of the corresponding patterns coordinates. This technique follows the clustering methodology [19]. The width is obtained by changing its old values to a random number (between 5% and 25%).
- Operator Null: in this case any operator is applied.

The probability of applying an operator to a given RBF is determined by means of a Mandani-type FRBS [22] which represents expert knowledge in order to obtain a simple and accurate RBFN.

The inputs of this system are the parameters used to define the credit assignment of the RBF, and the outputs represent the probability of applying Remove, Random Mutation, Biased Mutation and Null operators, respectively. Table 1 shows the rule base used to relate the described antecedents and consequents.

Table 1. Fuzzy rule base representing expert knowledge in the design of RBFN

Antecedents				Consequents				Antecedents				Consequents			
	v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}		v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}
R1	L			M-H	M-H	L	L	R6		H		M-H	M-H	L	L
R2	M			M-L	M-H	M-L	M-L	R7			L	L	M-H	M-H	M-H
R3	H			L	M-H	M-H	M-H	R8			M	M-L	M-H	M-L	M-L
R4		L		L	M-H	M-H	M-H	R9			H	M-H	M-H	L	L
R5		M		M-L	M-H	M-L	M-L								

For example, the FRBS promotes that an RBF with a poor credit assignment, low contribution, high error and high overlapping is eliminated. Otherwise the RBF is maintained.

Introduction of New RBFs. In this step, the removed RBFs are substituted by new ones. Each new RBF is located at the center of the area with maximum error. Its width is set to the average of the RBFs. Finally, the weights are set to zero.

Replacement Strategy. In this step the newly generated RBFs are compared with the original ones, and those with the best behavior are incorporated into the new population.

4 Experimentation and Results

As mentioned, applying the Box-Behnken methodology twenty chemical pretreatment experiments have been conducted to equally space the values of the data samples (instances) [11], obtaining twenty different data patterns. These make up the working data set, used to carry out the following modeling experimentation.

4.1 Experimental Framework

The input variables used to train the models are the following:

- Duration of the pretreatment (Time), measured in minutes and with values that oscillates between 0 and 30.
- Temperature of the pretreatment (Temp), which values go from 120 to 180 centigrade degrees.
- Molar concentration of $FeCl_3$ (FeCl3), whose values are in the range 0.050 to 0.275 M.

The considered output variables are the Enzymatic Hydrolysis Yields in Raw Material (EHYRM) and Enzymatic Hydrolysis Yields in Pretreated Material (EHYPM). All the values are summarized in Table 2.

As the number of available patterns is small (only 20), consequently the leave one out validation is applied. This implies that 20 models haven been developed, each one of them using 19 patterns for training and the remainder one for testing.

Two evaluation metrics have been computed to assess the methods performance: RMSE (*Root Mean Square Error*) and the coefficient of determination, R^2. RMSE is defined in Eq. 6, where n is the number of instances, f_t is the output of the model and y_t is the real output for the t-th instance, respectively.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (f_t - y_t)^2}{n}} \tag{6}$$

The quality of the calculated model from the training data is also evaluated with R^2, that obtains the fit between the predicted and the real data (7).

$$R^2 = (\frac{cov(f, y)}{\sigma_f \sigma_y})^2 \tag{7}$$

where $cov(f, y)$ represents the covariance between the model output and the real output, and σ_f and σ_y are the standard deviations for the model output and the real output, respectively.

Table 2. Values from the OTB pretreatment experiments

	Input variables			Output variables	
n	Time	Temp	FeCl3	EHYRM	EHYPM
1	30	120	0.125	10.14	19.71
2	0	140	0.050	12.23	19.92
3	15	160	0.050	19.57	34.61
4	0	140	0.200	11.66	19.43
5	30	140	0.200	19.67	39.94
6	15	140	0.125	14.62	23.17
7	15	140	0.125	15.10	27.34
8	30	160	0.125	28.07	55.80
9	15	120	0.050	11.85	18.85
10	30	140	0.050	11.08	19.81
11	15	120	0.200	8.76	14.63
12	0	160	0.125	18.80	33.69
13	15	140	0.125	14.10	25.70
14	15	160	0.200	36.50	75.54
15	0	120	0.125	12.62	20.07
16	0	180	0.200	25.58	93.16
17	0	160	0.275	28.73	66.87
18	30	180	0.200	20.17	93.55
19	30	160	0.275	38.85	88.71
20	30	180	0.275	11.36	96.15

In order to compare the results obtained for the proposed method, other two data mining methods are used: MLP-BR and NU-SVR. The methods' names and their implementation has been obtained from KEEL [23]. A brief description of this methods follows:

- MLP-BR. This algorithm is a implementation of the well known MLP [8], one of the most popular ANN methods. An MLP consists of multiple layers of interconnected hidden nodes. The hidden nodes receive the inputs from the input layer and calculate outputs which depend on the input and their activation function. The output calculated by the hidden nodes is forwarded to the next layer and so on up to reach the network output layer. As learning technique the model uses back-propagation, the value predicted by the model is compared with the real one, and the committed error is used to adjust the weights connecting the units in each layer. This process continues until a small error is obtained.

MLP-BR [24], the algorithm used in our experiments, is essentially a multilayer perceptron designed to produce a continuous output as prediction, so it is adapted to regression tasks.

- NU-SVR. This implementation follows the *Support Vectorial Machine* (SVM) [25] technique. The training of SVMs consists on solving a quadratic optimization problem to find the maximum separation margin between pattern categories. And adaptation of the basic algorithm, named NU-SVM [26], changes the C parameter by other called NU which is easier to adjust.

  In [27], SVMs can be seen as an universal tool for solving many task such as regression problems. The implementation used in this experimentation, NU-SVR [28], is a regression SVM based on the SMO (*Sequential Minimal Optimization*) [29] algorithm. This algorithm reduces the training process and therefore is more efficient than the traditional learning model.

The values of the configuration parameters for each method are set to the default ones proposed by the respective authors. The number of executions is established to three, as these are non-deterministic methods. The main parameters used for the algorithms are shown in Table 3.

Table 3. Parameter specification for the algorithms employed in the experimentation.

Algorithm	Parameter	Value
Reg-CO^{2}RBFN	Generations of the main loop	100
	Number of RBFs	8
MLP-BP	Hidden_layer	1
	Hidden_nodes	8
	Transfer	Htan
	Eta	0.15
	Alpha	0.10
	Lambda	0.0
NU-SVM	KERNELtype	RBF
	C	100.0
	Eps	0.001
	Degree	1
	Gamma	0.01
	Coef0	0.0

4.2 Results and Analysis

For each output variable of the hydrolysis process, one model is determined by each data mining method. The provided performance indicators are average values from these 20 runs. The results obtained with the training data sets

Table 4. Results obtained from training data sets

Variable	Reg-CO^{2}RBFN		MLP-BP		NU-SVM	
	R^2 ↑	RMSE ↓	R^2 ↑	RMSE ↓	R^2 ↑	RMSE ↓
EHYRM	**0.8846**	**2.8440**	0.6807	5.4435	0.6622	6.6978
EHYPM	**0.9919**	**2.5868**	0.9072	10.7019	0.8532	18.2879

are shown in Table 4. For each method, the RMSE and the R^2 coefficient are calculated. The best results are highlighted in bold. For RMSE lower values are better, whereas for R^2 higher values are better.

In summary, that the Reg-CO^{2}RBFN method outperforms the remaining methods, both in RMSE error and R^2 coefficient for all output variables, can be concluded from Table 4. By observing the RMSE metric, that MLP-BP and NU-SVM are clearly outperformed by Reg-CO^{2}RBFN for both variables can be seen, since MLP-BP and NU-SVM perform quite bad. These results can be explained by the low number of instances in the training data sets. Regarding the R^2 coefficient, Reg-CO^{2}RBFN obtain remarkable values, specially for the EHYPM variable.

The RMSE obtained by the models from the tests data sets are shown in Table 5. As before, best results are highlighted in bold.

Table 5. Results obtained from test data sets

Variable	Reg-CO^{2}RBFN	MLP-BP	NU-SVM
EHYRM	**6.5544**	7.1525	7.8550
EHYPM	**7.1723**	9.8838	22.5404

For the test data sets, it must be noticed that Reg-CO^{2}RBFN outperforms the other methods, both for the EHYRM and specially for the EHYPM variable, whereas Nu-SVM achieve very bad results, possibly attributable to the low number of instances available in the data sets as noted before.

5 Conclusions

The chemical research described in this work, carried out in Jaén, has allowed to obtain bioethanol from olive tree biomass. This study represents another contribution inside the renewable energy field. There are various parameters or variables involved in the process of obtaining bioethanol. In addition, the experimentation is expensive and also time consuming. For this reasons, it is important to automatically obtain a model able to explain the process.

In this paper three data mining methods have been used to modeling the process: Reg-CO^{2}RBFN, an RBFN designing technique developed by the authors, an MLP and an SVM. The main problem for the data mining methods is the low number of experiments (instances) available due to the high cost previously mentioned.

The data mining methods have been trained and have produced a model for each output variable. The results obtained show that Reg-CO^{2}RBFN has a good behavior in spite of the low training data available, and outperforms the remaining data mining methods for the output variables in the study.

As future work a multi-target regression version of Reg-CO^{2}RBFN will be studied.

Acknowledgments. This work is partially supported by the Spanish Ministry of Science and Technology under project TIN2015-68454-R.

References

1. Behera, S., Arora, R., Nandhagopal, N., Kumar, S.: Importance of chemical pretreatment for bioconversion of lignocellulosic biomass. Renew. Sustain. Eng. Rev. **36**, 91–106 (2014). http://dx.doi.org/10.1016/j.rser.2014.04.047
2. Ravindran, R., Jaiswal, A.K.: A comprehensive review on pre-treatment strategy for lignocellulosic food industry waste: challenges and opportunities. Bioresour. Technol. **199**, 92–102 (2016). http://dx.doi.org/10.1016/j.biortech.2015.07.106
3. Maimon, O., Rokach, L.: The Data Mining and Knowledge Discovery Handbook, 2nd edn. Springer, Heidelberg (2010)
4. Saavedra-Moreno, B., Salcedo-Sanz, S., Paniagua-Tineo, A., Prieto, L., Portilla-Figueras, A.: Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms. Renew. Eng. **36**(11), 2838–2844 (2011)
5. Rivera, A., Garca-Domingo, B., del Jesus, M., Aguilera, J.: Characterization of concentrating photovoltaic modules by cooperative competitive radial basis function networks. Expert Syst. Appl. **40**(5), 1599–1608 (2013). http://dx.doi.org/10.1016/j.eswa.2012.09.016
6. García-Domingo, B., Carmona, C., Rivera-Rivas, A., Jesus, M.D., Aguilera, J.: A differential evolution proposal for estimating the maximum power delivered by CPV modules under real outdoor conditions. Expert Syst. Appl. **42**(13), 5452–5462 (2015)
7. Kusiak, A., Zheng, H., Song, Z.: Short-term prediction of wind farm power: a data mining approach. IEEE Trans. Eng. Convers. **24**(1), 125–136 (2009)
8. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Technical report, DTIC Document (1985)
9. Park, J., Sandberg, I.: Universal approximation using radial-basis function networks. Neural Comput. **3**, 246–257 (1991)
10. Pérez-Godoy, M., Rivera, A., del Jesus, M., Berlanga, F.: CO^{2}RBFN: an evolutionary cooperative-competitive RBFN design algorithm for classification problems. Soft. Comput. **14**(9), 953–971 (2010)
11. López-Linares, J., Romero, I., Moya, M., Cara, C., Ruiz, E., Castro, E.: Pretreatment of olive tree biomass with FeCl3 prior enzymatic hydrolysis. Bioresour. Technol. **128**, 180–187 (2013). doi:10.1016/j.biortech.2012.10.076
12. Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex Syst. **2**, 321–355 (1988)

13. Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
14. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
15. Bäck, T., Hammel, U., Schwefel, H.: Evolutionary computation: comments on the history and current state. IEEE Trans. Evol. Comput. **1**(1), 3–17 (1997)
16. Harpham, C., Dawson, C., Brown, M.: A review of genetic algorithms applied to training radial basis function networks. Neural Comput. Appl. **13**, 193–201 (2004)
17. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE Trans. Syst. Man Cybern. B **35**(5), 928–947 (2005)
18. Runkler, T.A., Bezdek, J.C.: Alternating cluster estimation: a new tool for clustering and function approximation. IEEE Trans. Fuzzy Syst. **7**(4), 377–393 (1999). doi:10.1109/91.784198
19. Hartigan, J., Wong, M.: Algorithm as 136: a k-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)
20. Widrow, B., Lehr, M.: 30 years of adaptive neural networks: perceptron, madaline and backpropagation. Proc. IEEE **78**(9), 1415–1442 (1990)
21. Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms, pp. 41–49 (1987)
22. Mandani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man Mach. Stud. **7**(1), 1–13 (1975)
23. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Logic Soft Comput. **17**, 255–287 (2011)
24. Rojas, R.: Neural Networks. Springer, Heidelberg (1996). doi:10.1007/978-3-642-61068-4
25. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory - COLT 1992, pp. 144–152. ACM Press, New York (1992). doi:10.1145/130385.130401
26. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Comput. **12**(5), 1207–1245 (2000). doi:10.1162/089976600300015565
27. Vapnik, V., Vapnik, V., Golowich, S.E., Smola, A.: Support vector method for function approximation, regression estimation, and signal processing. Adv. Neural Inf. Process. Syst. **9**, 281–287 (1996)
28. Fan, R., Chen, P., Lin, C.: Working set selection using the second order information for training SVM. J. Mach. Learn. Res. **6**, 1889–1918 (2005)
29. Platt, J.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines (1998)