Systems Analysis Modelling Simulation
Vol. 43, No. 10, October 2003, pp. 1331-1339

Taylor & Francis
Taylor & Francis Group

ROBUST POLYNOMIAL NEURAL NETWORKS IN QUANTATIVE-STRUCTURE ACTIVITY RELATIONSHIP STUDIES

TETYANA I. AKSYONOVA[a], VLADIMIR V. VOLKOVICH[b] and IGOR V. TETKO[c]

[a]*Laboratory of Applied Nonlinear Analysis, Institute of Applied System Analysis, prosp. Peremogy, 37, 03056, Kyiv, Ukraine;* [b]*Control System Department International Researching-Training Center of Information Technologies, Glushkova 40, 252022, Kyiv, Ukraine;*
[c]*Laboratoire de Neuro-heuristique, Institut de Physiologie, Universite de Lausanne, Rue du Bugnon 7, CH-I005, Lausanne, Switzerland, http://vcclab.org*

(Received 13 August 2001)

This article presents the Robust Polynomial Neural Networks, a self-organizing multilayered iterative GMDH-type algorithm that provides robust linear and nonlinear polynomial regression models. The accuracy of the algorithm is compared to traditional GMDH and the multiple linear regression analysis using artificial and real data sets in quantitative-structure activity relationship studies. The calculated data shows that the proposed method is able to select nonlinear models characterized by a high prediction ability, it is insensible to outliers and irrelevant variables and thus it provides a considerable interest in quantitative-structure activity relationship studies.

Keywords: GMDH-type neural network; Nonlinear regression analysis; Robust estimation; Chemical engineering; Forecasting

1. INTRODUCTION

Many methods can be used to extract knowledge from experimental data and to determine its mathematical description. Multiple linear regression analysis (MLRA) is widely used in quantitative-structure relationship studies (QSAR) because of the rather simple way to interpret the results. The QSAR studies represent an important part of the drug design process and are used to reveal relationships between chemical structure of compounds and their biological activities. The power of MLRA can be significantly increased if it is combined with evolutionary algorithm [1]. Another widespread method in QSAR study, the partial least squares [2] (PLS) represents a generalized regression method based on latent vectors. It is a promising tool to analyze large data sets with highly collinear variables [3]. However, both MLRA and PLS methods are limited to linear regression models. The PLS algorithm is also sensitive to outliers or

*Corresponding author. E-mail: aksenova@pnn.com.ua

ISSN 0232-9298 print: ISSN 1029-4902 online
DOI: 10.1080/02329290290024330

irrelevant variables. Contrary to these methods, the feed-forward artificial neural networks can be used to model complex nonlinear relationships and it is a useful method in drug design studies [4,5]. However, a serious disadvantage of this method is that the dependencies detected between parameters and responses are hidden within neural network structure and therefore the interpretation of calculated results is difficult.

Group Method of Data Handling (GMDH) algorithms represent sorting-out methods that can be used for analysis of complex objects having no definite theory [6,7]. The choice of the appropriate GMDH algorithm depends on the specificity of the problem to be solved. The specific features of the QSAR tasks can be summarized as follows: there is a large number of input variables; some of these variables can be irrelevant and highly correlated. While the GMDH approaches are well suited to solve such problems, the results of MLS method are sensitive to outliers, not stable and, as a rule, cannot be easily interpreted.

In this article we describe Robust Polynomial Neural Network, iterative GMDH type algorithm that provides robust linear and nonlinear modeling in the presence of outliers or/and correlated and irrelative variables. It allows controlling the complexity - number and the maximal power of terms in the models. The algorithm calculates the stable results that can be easily interpreted. Performance of the new approach is compared with GMDH-type Neural Network and MLR algorithm.

2. METHOD

The important feature of the Iterative GMDH algorithm is its ability to identify both linear and nonlinear polynomial models using the same approach. The iterative GMDH-type algorithms can be described following Yurachkovsky [8].

Let us designate $X = \{x_1, x_2, \ldots, x_m\}$ the set of input variables and n the number of observations of vector X of input and y output variables. It is possible to determine the class of models G that is characterized by the following properties:

1. Class G contains structures that are linear according to the parameters. Under "structure" we assume any model with unidentified parameters.
2. There is (and it is known) a transform $g()$ such as $g(f_i, f_j) \in G$, if $f_i, f_j \in G$;
3. Any element of class G is either constant, or one of initial input variables, or it is calculated using transform $g()$ applied to other elements of the class.

In the simplest case, the class G consists of linear functions only. The transformation $g()$ can be defined as $g(f_i, f_j) = af_i + bf_j$ in this case. Such a class contains only a limited number of structures equal to $2^{k+1} - 1$, where k is the number of input variables. If class G contains polynomials of arbitrary degree, the transformation $g()$ can be defined for example as $g(f_i, f_j, f_k) = af_i + bf_jf_k$ or $g(f_i, f_j, f_k) = af_i + bf_jf_k + cf_i^2 + df_j^2$. Such class contains an infinite number of structures.

The purpose of the algorithms is to find a subset of variables $\{x_{i1}, x_{i2}, \ldots, x_{ik}\}$ and a model $y = u_i(x_{i1}, x_{i2}, \ldots, x_{ik}\}$ that minimizes some criterion value (CR). Examples of relevant criteria are

$$\mathrm{AR} = \mathrm{SUM}(Y_{\mathrm{calc}} - Y_{\mathrm{exp}})^2/n \qquad (1)$$

and

$$\mathrm{FPE} = \mathrm{SUM}(Y_{\mathrm{calc}} - Y_{\mathrm{exp}})^2(N+n)/(n-N) \tag{2}$$

where Y_{calc} and y_{exp} are, respectively, the calculated and target values, n is the number of observations in the data set, and N is the number of terms in the model. The first criterion AR (Eq. (1)), known as the regularity criterion, has traditionally been used in the GMDH approaches. This criterion should be calculated using the validation (test) data set in order to provide regularization and to prevent overfitting of the model [6]. However, selection of a representative validation set is usually a complex problem for small data sets. The second criterion (Eq. (2)), known as Final Prediction Error (FPE) criterion [9, 10], is also widely used to select regression and auto-regression models. Depending on the noise the models selected using these criteria can actually be mathematically simpler than the underlining physical model might demand, thereby increasing the prediction ability of the method [7]. It should be noted that other, more complex, criteria such as the Akaike information criterion [11] (AIC) or Schwarz information criterion [12, 13] (SIC) can be used for the selection of unbiased models.

The traditional GMDH algorithms are implemented as iteration procedures. Let us denote $W_0 = X = \{x_1, x_2, \ldots, x_n\}$, $W_0 \subset P^1$, P^s is the set of polynomials of s-power, and assume that transformations $g(f_i, \ldots, f_j)$ is specified as $g(f_i, f_j, f_k) = af_i + bf_jf_k + cf_i^2 + df_j^2$, $g(f_i, f_j, f_k) = af_i + bf_jf_k$ or $g(f_i, f_j) = af_i + bf_j$ for example.

Step 1 All possible models of the form $y = g(w_i, \ldots, w_j)$ are considered, $w_i, \ldots w_j \in W_0$. The coefficients are estimated using method of least squares (MLS). The best models $g_1(w_i, \ldots, w_j), \ldots, g_F(w_i, \ldots, w_j)$ are selected according to criterion CR and they form the set $W_1 = W_0 \cup \{g_1(w_i, \ldots, w_j), \ldots, g_F(w_i, \ldots, w_j)\}$. Notice that $W_1 \subset P^2$ for non-linear transform *g()* considered below.

Step r The models of the form $y = g(w_i, \ldots, w_j)$ is considered for all combinations $w_i, \ldots, w_j \in W_{r-1}$. The coefficients are estimated using MLS. The best models $g_1(w_i, \ldots, w_j), \ldots, g_F(w_i, \ldots, w_j)$ are selected according to criterion CR and they form the set $W_1 = W_0 \cup \{g_1(w_i, \ldots, w_j), \ldots, g_F(w_i, \ldots, w_j)\}$, $W_r \subset P^{r+1}$.

The procedure is terminated if there is no improvement of the criterion value. The superposition of functions *g()* at each iteration increases the amount of term and the power of the resulting polynomial in the nonlinear case. Since the number of iterations can be large, the application of traditional GMDH algorithms generates very complex models that cannot be easily interpreted except in the linear case. The algorithms that generate polynomials of high power could be unstable and sensitive to outliers.

The main objective of the work was to create a stable algorithm for nonlinear model selection such that its results are easily interpreted by users. To achieve this purpose we added some restrictions on the class G used to select the best model. This made it possible to specify maximum degree of polynomials and the number of terms in the equations. Such restrictions made it also possible to incorporate preliminary information and to specify desired properties of the expected solution. The developed algorithm represents a GMDH-type algorithm with the control of the complexity of the model.

The main features of the algorithm can be summarized as follows:

(1) *Fast learning* The transforms with two coefficients only are used, for example $g(f_i, f_j) = af_i + bf_j$ in the linear case. Irrespectively of the power of resulting model and the number of terms the second order matrices are only inverted. This provides fast learning of the algorithm.

(2) *Results in the parametric form* The polynomial structures are coded using vector of simple numbers [8] that provides the presentation of the results in the parametric form. Let us associate the set of input variables $X = \{x_1, x_2, \ldots, x_m\}$ with the corresponding set of simple numbers $V = \{v_1, v_2, \ldots, v_m\}$, $v_i \neq v_j$. Each term $x_i^{q1} \cdots x_j^{q2}$ in the equation is coded as a multiplication of the appropriate powers of simple numbers $v_i^{q1} \cdots v_j^{q2}$ (Gedel's number) i.e.x_i,- is substituted by the corresponding simple number v_i. Thus the polynomial is coded using a vector of Gedel's numbers. Transform g() contains multiplication and summation. To multiply two terms it is enough to multiply their Gedel's numbers. To add the term it is enough to add the appropriate nonzero number to the vector. Because of one-to-one correspondence of the terms of polynomials to their Gedel's numbers this coding scheme can be used to transform the neural net results to the parametric form of equation. Vector of Gedel's numbers is calculated for each intermediate model. This vector can also be used to detect and exclude redundant models.

(3) *Complexity control* Let us denote vector $(power, c)^T$ as a complexity, *power* is the power of the polynomial and c is the number of terms. As it was mentioned above, the number of terms in the equation is given simply by the number of non-zero elements in the vector of their Gedel's numbers. The power of the new model is controlled by the condition that if, for example, $g(w_i, w_j, w_l) = a\, w_i + b\, w_j w_l$, then

$$power(g(w_i, w_j, w_l)) = \max(power(w_i), power(w_j) + power(w_l)),$$

where *power()* designates the power of the polynomial. Both power and number of terms are calculated for each intermediate model. It gives us the possibility to restrict the class of the models under consideration by $power(w_i) < p$ and to search models among the polynomials with power less than p. The maximum complexity is defined by the user or can be automatically selected using a full cross-validation method.

(4) *Twice-hierarchical neural net structure* Twice-hierarchical neural net structure is an important feature of PNN. One of the problems is that power of polynomials increases too fast in the traditional GMDH algorithm. At the step r of iteration procedure one can have models of power $r+1$, $W_r \subset P^{r+1}$. The control of complexity gives us an opportunity to implement the iteration procedure without an increase of the power of polynomials or/and the number of terms. PNN is implemented as a twice-hierarchical neural net. External iterative procedure controls the complexity, i.e. the number of the terms and the power of the polynomials in the intermediate models, and discards models that are out of the specified range. The best models form initial set for the next iterative procedure. This procedure realizes a wide search without the complexity increase. Besides that the twice-hierarchical neural net structure provides the convergence of the coefficients. The models that are calculated as a result of several transformations have the coefficients that are close to the appropriate regression coefficients. This fact was proved mathematically for algorithm with linear transform [14] and it was confirmed by calculating experiments for nonlinear cases [15].

The algorithm with the aforementioned properties was realized by us earlier [16] and we will refer to it as traditional PNN algorithm.

(5) *Robust estimation* In the presence of large errors (outliers) the noise can be described as a mixture of normal distributions. Let us assume that the observation $y_i, i = 1, 2, \ldots, n$ are the independent random variables with distribution determined from the model of large errors

$$P_\delta(\xi_i) = (1 - \delta)\varphi(\xi_i) + \delta h(\xi_i),$$
$$\xi_i = \left(y_i - \sum \beta_i f_{ji} \right)/\sigma,$$

where $\phi(\xi)$ is the normal distribution density $N(0,1)$; $h(\xi)$ δ are thedistribution density and the level of the large errors respectively; σ is the variance. The density function $h(\xi)$ is symmetrical over y-axis and it has heavy tails. In the presence of outliers the multiple regression parameters can be calculated following maximum-likelihood or M-estimation, as a result of functional minimization [17].

$$\min_{\beta_1, \ldots, \beta_m} \sum_{i=1}^{n} \rho\left(y_i - \sum_{j=1}^{m} \beta_i f_{ji} \right)$$

The most known Huber's approach consists of using the function $p()$ that minimizes the variance of estimation

$$\rho(z) = \begin{cases} (1/2)z^2 & \text{if } |z| \le C \\ C|z| - (1/2)C^2 & \text{if } |z| > C, \end{cases}$$

where constant C is determined by the level of the large errors. It is important to mention that the tests of hypotheses for M-regression are not yet elaborated sufficiently even for linear M-regression structure identification [18]. There are also other methods for robust estimations, such as L- or R-estimates [19].

In the current work we have developed the PNN algorithm for nonlinear M-regression model identification. This made it possible to improve the stability of PNN algorithm to large errors. We will refer to the new method as Robust PNN or RPNN.

3. DATA SETS AND RESULTS

The performance of the developed algorithms was demonstrated using examples of artificial and real QSAR data sets.

Analysis of an Artificial Data Set

A first set was generated from a nonlinear model of fourth power $Y = X1*(X5**3) + 10$ that is traditionally used for the testing of GMDH-type algorithms [15]. Random noise and three further random variables *(X2-X4)* were added and 13 observations were generated (n=13; m=5*)*. The comparison of RPNN algorithm was done with the traditional PNN algorithm.

Ten and three observations were used as training and test sets, respectively. The model with excellent approximation and low prediction error were calculated by PNN: $Y1=0.\ 19*X1*X5\ +0.\ 22*(X5**2)+\ X1*(X5**3)$, RMSE = 7. 18 for training set and RMSE = 22.45 for these test set (Table 1). RPNN found the exact model structure $Y1\ =X1*(X5**3)+9.\ 1$ and provided lower prediction error RMSE = 2. 1 for the test set. In order to study the stability of PNN and RPNN algorithms, the values of two and three data cases from the training set were changed to be large errors of the initial model. Results of the PNN algorithm were affected by the outliers (Table II). This method provided low generalization ability for test set. On the contrary, RPNN results for the test set were practically the same as for the noiseless data (Fig. 1, Table II).

Prediction of the Sublimation Enthalpy

To study the stability of RPNN, we developed QSAR models for the sublimation enthalpy of a series of 18 polychlorinated hydrocarbons (PCBs).

The atom and bond-type *E*-state indices (8 parameters) were used as input parameters for the MLRA, PNN and RPNN. The first analysis included 16 PCBs in the training and two molecules in the test set. Statistically significant models with low prediction error of the test molecules RMSE= 1.06 and 0.93 were calculated by

TABLE 1 Values calculated by PNN and **RPNN** method for the training and test sets

Without errors			*With 2 errors*			*With 3 errors*		
Y	*PNN*	*RPNN*	*Y*	*PNN*	*RPNN*	*Y*	*PNN*	*RPNN*
Training set								
38	28	36	38	37	27	38	26	27
520	524	521	1520	1270	513	1520	1251	513
200	193	201	3200	3227	192	3200	3212	192
139	129	137	139	152	128	139	124	128
2571	2564	2569	2571	2652	2562	571	2521	2562
1457	1471	1467	1457	1653	1458	1457	1598	1458
1723	1718	1724	1723	1822	1716	1723	1705	1715
31360	31358	31260	31360	31332	31280	31360	31380	31281
2929	2925	2925	2929	2978	2917	2929	2865	2917
5832	5832	5841	5832	5863	5827	5832	5636	5827
Test set								
6761	6789	6759	6761	7336	6758	6761	7279	6758
43	31	41	43	3059	53	43	3057	54
1011	1019	1009	1011	997	999	1011	966	999

TABLE II RMSE of PNN and **RPNN** methods for training and tests sets

Errors	*Training Set*				*Test Set*	
	PNN		*RPNN*		*PNN*	*RPNN*
	With errors	*Exact*	*With errors*	*Exact*	*Exact*	
0	7.2	7.2	30	30	22	2.1
2	125	987	957	27	2170	11
3	612	994	1030	27	2163	12

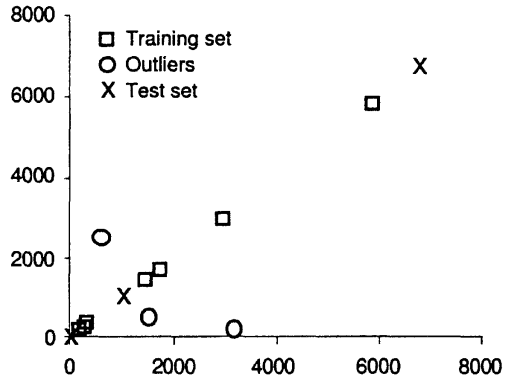

FIGURE 1 Results of the RPNN model calculated using training set contaminated with three large errors.

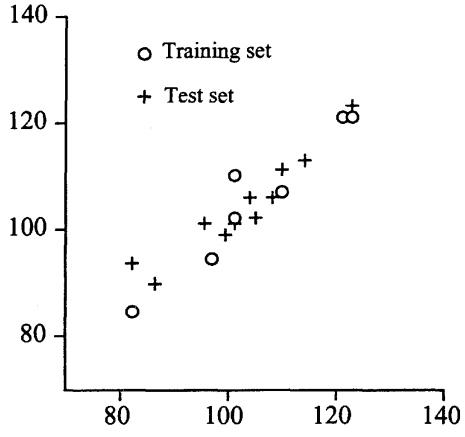

FIGURE 2 Prediction of the sublimation enthalpy of molecules by the RPNN method using seven and eleven molecules as the training and test sets, respectively.

the MLRA and the RPNN methods, respectively. PNN calculated lower result RMSE = 3. 1. However, PNN and RPNN selected nonlinear models of the same structure $y = -0.036x_7^2 + 133x_4 - 5.9x_1$ and $y = -0.023x_7^2 + 132x_4 - 5.9x_1$. MLRA significantly decreased its prediction ability when number of molecules in the training set dropped below 12. On the contrary, the RPNN results were not affected by the number of molecules in the training set. Even if the number of molecules in this set decreased from 16 to 7 molecules, the RPNN still calculated the same regression equations with only slight variations of the regression coefficients. For example, the equation $y = -0.041x_7^2 + 134x_4 - 5.9x_1$ was calculated with seven molecules in the training set. This model provided RMSE=4. 3 for 11 molecules in the test set (Figs. 2 and 3). These results indicated a stability of RPNN method. The results calculated by PNN method were lower compared to RPNN but much better compared to MLRA method.

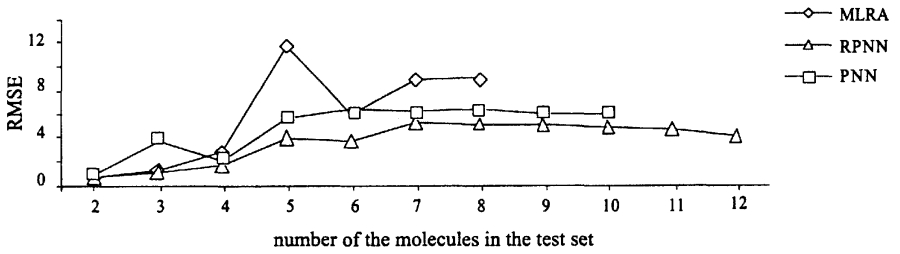

FIGURE 3 Root Mean Squared Error calculated for the test set molecules as a function of the number of molecules in the training set. **MLRA** failed to provide significant models if more than eight molecules were used in the test set.

DISCUSSION

RPNN represents a promising method for applications in environmental and toxicological studies. It provides the model in parametric form as an equation that can be easily interpreted by the users. RPNN is a robust method that can be used even in the presence of outliers in the training set. The use of m-estimates to estimate model parameters represents an important improvement of this algorithm compared to the traditional PNN algorithm. This feature is very important for application in chemistry and in drug design. It is known that chemical data quite often contain a number of large errors. Such errors can appear due to mistakes in molecular coding, representation, experimental design, etc. The RPNN method is able to provide reliable results even for such difficult cases. Moreover, models selected by RPNN are characterized by a high predictive ability even for small data sets. RPNN provides high speed of training compared to the other neural network approaches. Therefore, it can be also applied for large data sets.

Acknowledgments

This study was partially supported by INTAS-INFO 2000-363, NATO HTECH.LG 972304 and the Swiss National Science Foundation SCOPES 7IP 62620 grants.

References

[1] I.V. Tetko, V.Y. Tanchuk and A.I. Luik (1994). Application of an evolutionary algorithm to the structure-activity relationship. In: *Proceedings 3rd Annual Conference on Evolutionary Programming,* pp. 109-119. World Scientific, River Edge, N.J.

[2] H. Kubinyi (1996). Evolutinary variable selection in regression and PLS analyses. *Journal of Chemometrics,* **10,** 119-133.

[3] S. Rannar, F. Lindgren, P. Geladi and S. Wold (1994). A PLS kernel algorithm for data sets with many variables and fewer objects. 1. Theory and algorithm. *Journal of Chemometrics,* 8, 111-125.

[4] J. Zupan and J. Gasteiger (1999). *Neural Networks for Chemistry and Drug Design: An Introduction;* 2nd edn. VCH, Weinheim.

[5] I.V. Tetko, D.J. Livingstone and A.I. Luik (1995). Neural network studies. 1. comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences,* 35, 826 833.

[6] H.R. Madala, A.G. Ivakhnenko (1994). *Inductive Learning Algorithms for Complex Systems Modelling,* CRC Press Inc., Boca Raton.

[7] T.I. Aksenova and Y.P. Yurachkovsky (1988). Characterization of unbiased structure and condition of its J-optimality. *Soviet Journal of Automation and Information Science,* **21,** 24-32.

[8] Y.P. Yurachkovsky (1981). Restoration of polynomial dependencies using self-organization. *Soviet Automatic Control,* **14,** 17-22.
[9] D. Rothman (1968). Letter to the editor. *Technometrics,* **10,** 661-667.
[10] H. Akaike (1969). Fitting autoregressive models for prediction. *Ann. Inst. Statist. Math.,* **21,** 243-247.
[11] H. Akaike (1974). A new look at the statistical model identification. *IEEE Trans. Automatic Control,* **AC-19,** 716-723.
[12] G. Schwarz (1978). Estimating the dimension of a model. *The Annals of Statistics, 6,* 461-464.
[13] A.A. Neath and J.E. Cavanaugh (1997). Regression and time series model selection using variants of the Schwarz information criterion. *Communications in Statistics - Theory and Methods,* **26,** 559-580.
[14] Y.P. Yurachkovsky (1981). Convergence of multilayer algorithms of the group method of data handling. *Soviet Automatic Control,* **14,** 29-35.
[15] A.G. Ivakhnenko and Y.P. Yurachkovsky (1987). Complex system modeling on experimental data. *Radio and Communication.* Moscow, (in Russian)
[16] I.V. Tetko, T.I. Aksenova, V.V. Volkovich, T.N. Kasheva, D.V. Filipov, W.J. Welsh, D.J. Livingstone and A.E.P. Villa (2000). Polynomial neural network for linear and non-linear model selection in quantitative-structure activity relationship studies on the Internet. *SAR QSAR Environ. Res.,* **11,** 263 280
[17] P.J. Huber (1981). *Robust Statistics.* Wiley and Sons Inc., N.-Y.
[18] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W.A. Stahel (1989). *Robust Statistics.* The approach based on influence functions. Wiley and Sons Inc., N.-Y.
[19] W.H. Press, S.A. Teukolsky, W.T., Vetterling and B.P. Flannery (1994). *Numerical Recipes in C,* 2nd edn. Cambridge University Press, New York.