Using of prior information in polynomial multilayered GMDH

Alexandr Kiryanov

Dept. of Computer-Aided Management and Data Processing systems,NTUU KPI, 37 Prospect Peremohy, Kiev,Ukraine

alexandr.kiryanov@infopulse.com.ua

Abstract. *GMDH uses minimum expert's information about searched function. It sorts all cases and selects the best suited according to some external selection criterion. This paper considers how expert knowledge about presence of certain variables in a function could help GMDH. A new modification of the multilayered polynomial GMDH is proposed that uses this information to get better polynomials. Experiments were carried out to check the work of this new algorithm and showed it efficiency.*

Keywords

Inductive modeling, GMDH, multilayered polynomial algorithm

1 Introduction

One of the main advantages of GMDH is that a domain expert does not take part in a selection of dependencies; the expert gets the final result [1]. At the same time this could be a disadvantage. In some cases the expert can suggest a certain direction of a search during dependency selection or suggest what input variables are in dependency (e.g. the expert knows that an output variable depends on x^2, but he does not know the exact structure of the function and what other variable are in it). Question arose: How can we use prior information about the searched function? Is it possible for the expert to help GMDH by specifying information about variable presence in the dependency?

The expert has a possibility to help GMDH by specifying: "correct" algorithm from the GMDH algorithms family, "correct" selection criterion and "correct" learning/test sample division. Of course, to do this the expert needs to know much about GMDH algorithms and their peculiarities. Let's try to find how GMDH could use information about presence of variables in function.

2 Problem statement

Suppose we have a multidimensional sample that consists of *n* input vectors $X=\{X_1, X_2,..X_m\}$ m and output *Y*. The dependency $Y = f(X)$ is found using the polynomial multilayered GMDH algorithm. The quality of the found function ***f*** should be evaluated by LDUC external criterion. Suppose that information about presence of a monomial $M = X_1^{l_1} \cdot X_2^{l_2} \cdot ... \cdot X_m^{l_m}$ in the function is known (l_1 , l_2 , l_m - are the respective powers of X_i). It is necessary to improve the dependency based on the information about monomial presence in it. Dependency is called improved when its LDUC criterion value is less than previous dependencies criterion value.

3 Problem solving

In order to use the information about the monomial presence in the polynomial, it is necessary to correct the direction of multilayered GMDH. Let's examine the main stages of the algorithm [1,2]:

1. Build partial descriptions of all pair combinations of input variables

   $y_1 = f(x_1,x_2), y_2 = f2(x_1,x_3),\ldots,y_k = f_k(x_{n-1},x_n)$

2. Coefficients in partial descriptors are found by LSM (least square method)

3. Find the external criterion value for every partial descriptor

4. Select certain number of partial descriptors that have the least values of the external criterion

5. If the least value of the external criterion is decreased in comparison with the previous layer then go to 6 else algorithm is terminated and returns partial descriptors with the least external criterion values.

6. Go to the next layer. Found partial descriptors together with the input variables form input data for new partial descriptors: $Z_1 = \varphi(x_1,y_1)$, $Z_2 = \varphi_2\ (x_1,y_2),\ldots,\ Z_l = \varphi_k\ (x_{k-1},y_k)$. Goto2.

It is possible to correct the work direction of the algorithm during 2 stages: 1st, when we build partial descriptions, or 4th, when we select partial descriptors for the next layer. Let's investigate the first case. We can influence the building of a partial description only by changing of the supporting function. Thus, it is necessary to select a certain supporting function based on the current combination of the input data. New problems arose: what supporting functions should be used? How should we select a supporting function? Is it possible to use supporting functions with different numbers of variables? The second case appears to be less problematic and more promising. In GMDH partial descriptors are selected only based on an external criterion value. Let's make a possibility to select polynomials also based on the variables that are in function. A multicriterion problem appears: it is necessary to select a polynomial based on the external criterion value and variables that are in it. Let's consider two main approaches in solving of multicriterion problems for such cases:

1. Consider one criterion as the main and use the second one only if the values of the first criterion are equal

2. Introduce utility function that will include both criteria

Probability of the case, when two values of the external criterion are equal, is minimal. That is why the first approach will cause almost full ignoring of the second criterion. This shortcoming could be overcome by the following: let's consider criterion values equal, if their difference is less than certain value ε . We are faced with another problem: transitivity condition is not met. If partial description y 1 is better than y2 and y2 is better than y 3, then this does not mean that y 1 is better than y3. Thus it might be a case when there is no best partial descriptor.

Now let's investigate the second approach and take the following utility function:

$$Crit = Crit^*(1+ k_{cor} * k_{scale}) \quad . \qquad (1)$$

where *Crit* is value of the external criterion, k_{cor} is a corrective coefficient, k_{scale} is a coefficient that defines influence of k_{cor} on the utility function value.

This utility function is multiplicative. We can change the criteria importance by variation of k_{scale} coefficient. Proposed algorithm of the corrected criterion value determination is the following:

Expert specifies monomial *M* and k_{cor}

k_{cor} is found:

a. Set k_{cor}=1

b. For every monomial in polynomial:

i. k_{cur}= 1- n_{MMcur}/ n_M

ii. If $k_{cur}< k_{cor}$, then $k_{cor}= k_{cur}$

3. Set corrected criterion value to $Crit^*(1+ k_{cor} * k_{scale})$

where *M* is a monomial specified by expert, n_M is a number of variables in monomial *M*, n_{MMcur} is a number of variables from *M* in monomial M_{cur} (e.g. if $M=x1 *x2^2*x3$ and $M_{cur}=x1 *x2^2$ then n_{MMcur} is 3).

Thus, we need to add one more stage in multilayered polynomial GMDH:

1. Build partial descriptions of all pair combinations of input variables:

   $y_1=f(x_1,x_2)$, $y_2 =f_2(x_1,x_3)$,…,$y_k = f_k\{x_{n-1},x_n)$
2. Coefficients in partial descriptors are found by LSM (least square method)
3. Find the external criterion value for every partial descriptor.
4. Criterion value is corrected according to the algorithm above.
5. Select certain number of partial descriptors that have the least values of the external criterion.
6. If the least value of the external criterion is decreased in comparison with the previous layer then go to 6 else algorithm is terminated and returns partial descriptors with the least external criterion values.
7. Go to the next layer. Found partial descriptors together with the input variables form input data for new partial descriptors:

   $Z_1 =\varphi(x_1,y_1)$, $Z_2 = \varphi_2 (x_1,y_2)$,…, $Z_l = \varphi_k (x_{k-1},y_k)$ Goto 2.

4 The results of experiments

The following 10 polynomials were got for experiments:

Tab.1. Polynomials for experiments

*x0+5 *x1 *x3+10 *x2 *x4*
*10+4 *x6+4 *x2 *x3 *x4*
*4*x0*x1+5*x2*x3+1*
*7*x4+3*x3+2*x0+6*x1*
*1*x1+2*x2*x3+x4*
x0+x1+x2+x3
*x5+x6*x1+x2*x3*x4*
*x0*x2+x2*x3+x3*x4*
*1+x1*x3+x3+x4*
*2+x0*x1*x2*x3*x4*

The following monomials were specified for the corresponding polynomials (first monomial for the first polynomial and so on):

Tab.2. Experts monomials

x1*x3
x6
x0*x1
x4
x2*x3
x3
x2 *x3 *x4
x2*x3
x3
x2 *x3 *x4

A noise was introduced into every variable and output function value. During experiments the following parameters were changed: noise (100%, 10%, 1%, 0.1% and 0%), number of input and output vectors (10, 20, 40, 50, 70) and k_{scale} (0.1, 0.5, 1, 10, 1000, 10^{300}). Mean numbers of improved, not changed and worsen polynomials were determined. Polynomial is considered improved when the external criterion value has been reduced after monomial tip introducing (the same approach to determine not changed and worsen polynomials). Number of runs was calculated using the following formula [3]:

$$N = t_{\alpha}^{2}\frac{DX}{\varepsilon^{2}} + 1$$

, where *DX* - dispersion. At first 50 runs were done to find the *DX* estimation and then found *DX* was used in formula.

Tab.3. Mean numbers of improved, not changed and worsen polynomials

What is changed	Value	Improved %	Not changed %	Worsen %
k_{scale}	0.1	7	86	6
	0.5	21	62	17
	1	27	51	21
	10	40	32	28
	1000	41	30	29
	10300	42	30	28
Noise	100%	34	33	33
	10%	30	49	21
	1%	29	53	18
	0.1%	28	53	18
	0%	28	54	18
Num. vectors	10	29	49	23
	20	30	49	21
	40	30	48	22
	50	30	48	22
	70	30	48	22

Analysis of the above table shows that the most efficient k_{scale} values are equal to and greater than 10. Number of vectors almost do not influence on the result (only when we have 10 vectors we have very bad result). The same is with the noise - only 100% noise significantly worse the result.

After changing all parameters it was found that the best combination is the following: $k_{scale}=10$, 1% of noise and number of vectors is 40. With such parameters values number of improved polynomials is 39%, not changed - 38% and worsen - 24%.

5 Conclusion

A new modification of the multilayered polynomial GMDH was proposed. This modification uses information about presence of the certain monomial in dependency in order to improve the polynomial. Experiments were conducted and showed that algorithm is really efficient. Further work could be concentrated on the usage of the suggested criterion correction algorithm for other GMDH algorithms. Also it is suggested improving algorithm to use information about presence of several monomials in polynomial.

References

[1] Madala H.R., Ivakhnenko A.G.: Inductive Learning Algorithms for Complex Systems Modeling.- CRC Press, 1994. - 368 p

[2] Zaychenko Yu. P.: Basics of intellectual systems design. - Kyiv: "Slovo", 2004. - 352 p (in Ukrainian).

[3] Tomashevskii V. M. Systems modelling. - K.: BHV, 2007. - 352 p (in Ukrainian).