

Multistage combinatorial GMDH algorithm for parallel processing of high-dimensional data

O.A. Koshulko¹, A.I. Koshulko¹

¹Glushkov Institute of Cybernetics of NAS of Ukraine

koshulko@opengmdh.org

Abstract. *Combinatorial algorithm of the Group method of data handling (GMDH) is a powerful tool for building noise resistant multidimensional mathematical models with optimal complexity. Full combinatorial search is a preferable type of model search, but it can not be applied with high-dimensional objects because of high computational complexity. However there is a way to decrease the computational complexity by suggesting a reasonable procedure of partial combinatorial search. For this purpose we compare simple complexity limitations of GMDH models with a procedure called Multistage combinatorial algorithm.*

Keywords

Combinatorial GMDH, parallel processing, high-dimensional data.

1 Introduction

Combinatorial algorithm of the Group method of data handling (GMDH) [1, 2] builds noise resistant multidimensional mathematical models with optimal complexity. We consider combinatorial algorithm especially effective for structural identification of black-box objects and time series forecasting.

Full combinatorial search among all model structures in a class is a preferable type of data analysis that unfortunately can be rarely applied with high-dimensional objects because of high computational complexity.

Combinatorial algorithm uses Kolmogorov-Gabor polynomial as a base class of models

$$y = a_0 + \sum_{i_1=1}^k a_{i_1} x_{i_1} + \sum_{i_1=1}^k \sum_{i_2=i_1}^k a_{i_1 i_2} x_{i_1} x_{i_2} + \dots + \sum_{i_1=1}^k \sum_{i_2=i_1}^k \dots \sum_{i_P=i_{P-1}}^k a_{i_1 i_2 \dots i_P} x_{i_1} x_{i_2} \dots x_{i_P}, \quad (1)$$

where a – model parameters, k - data dimensions, P – power of polynomial. The number of parameters of full polynomial is maximum model complexity n . n defines the number of partial models generated by full combinatorial search

$$q_n = \sum_{i=1}^n C_i^n = 2^n - 1; \quad (2)$$

q_n growth twice with every additional parameter. This makes computational time unacceptable for objects represented by high-dimensional datasets. Despite our implementation of full combinatorial search algorithm is capable of parallel processing the maximum model complexity remains quite limited (see Tab.1).

Obviously computational complexity can be reduced by a reasonable procedure of partial combinatorial search. An example of such a procedure called Multistage selective combinatorial algorithm can be found in chapter 3 of [2]. A drawback of any partial search is that the model with optimal complexity might be missed. In our particular case we require from a new partial search algorithm to be capable of parallel processing and to minimize a chance of optimal model missing.

Tab. 1. Aproximate processing time for some GMDH problems.

Computer	Complexity n	Processing time
1 core	20	10 sec.
100 cores	30	30 sec.
1000 cores	40	1 hour

A simple way to perform partial search that satisfy our requirements is to use a model generator with gradual structure complication and limit complexity of generated model structures at some level. Such a method is widely used in different implementations of combinatorial algorithms and described also in [2]. For very high-dimensional data this method will frequently miss the optimal model however, it can be improved by a multistage procedure applied at the top of it.

2 Multistage modification

General idea of the proposed multistage procedure consists of the following three steps.

1. Rank variables by importance then generate full structure of base model.
2. Search the optimal model using acceptable portion of base function components.
3. Replace unused components in the initial set with new. Repeat step 3 until all components are considered.

This procedure performs full search among limited number of variables many times. Computational time in this case will be well controlled and even if we add base function components one by one the number of considered models does not exceed

$$q_n \leq (n - m) \cdot 2^m, m < n; , \tag{3}$$

where m - limited number of base function components choused much less than n .

Parallel processing is available in the multistage procedure and at the level of a single stage organized identically to parallelization of full combinatorial search [3]. At every stage of the procedure processors must collect results and then proceed to the next stage.

We checked losses of parallel processing time (inefficiency) of limited complexity and multistage procedures (see Fig.1) relatively to theoretically absolutely efficient time.

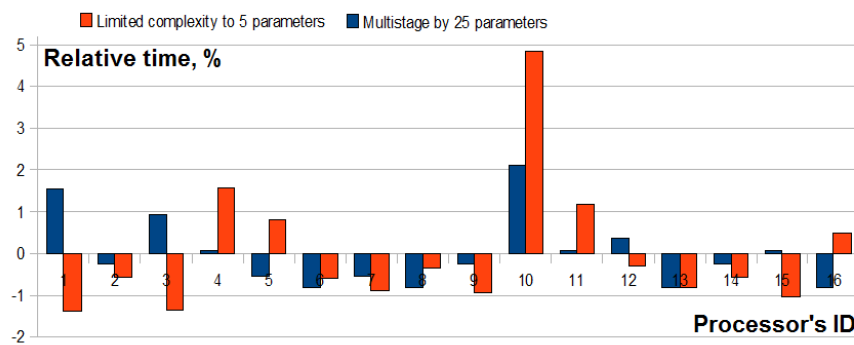


Fig. 1. Relative time loss of different combinatorial procedures. (Multistage - blue; Limited complexity - red.)

Fig.1 shows that Multistage procedure remains very efficient in terms of multi-processor system utilization while with the limited complexity procedure parallel processing is more sensitive to parallelization method drawbacks.

3 Test object

To investigate efficiency of proposed multistage combinatorial search for time series forecasting we use a natural data object - the Top500 Supercomputer's List released semiannually since 1993 [4]. We try to make a one-step forecast of entry-level supercomputer performance measured by HPL benchmark [5].

Input variables are time series of: entry-level system theoretical and peak performance, the number of systems excluded from the Top500, the number of processors of all listed systems, NASDAQ index, enumeration and lags of all variables. Output variable is the entry-level system peak performance.

Different numbers of dimensions are considered starting at the number acceptable for full combinatorial search - 20, then 77 and 129 obtained as deeper lags of the same inputs. We compare results achieved with simple complexity limitations of partial models with multistage combinatorial search and with full search. Computational time in all simulations was acceptable for single-processor computers.

We measure average mistake for 10 simulated historical forecasts performed with different search procedures (see Tab.2). Base function is linear in all experiments that means $n = k - 1$. Criterion of regularity is applied for model selection and best 10 models are always averaged to make a prediction.

Tab. 2. Relative average mistake for 10 simulated forecasts by different procedures.

	20 parameters	77 parameters	129 parameters
Full search	10.6%	–	–
Complexity limited to 3	11.8%	6%	5.7%
Multistage procedure by 20 parameters	10.6%	7.7%	5.2%

In case of 20 parameters multistage procedure is equal to full search because it considers parameters by 20 and finish in one stage. Complexity limitation shows higher average mistake. That means the optimal complexity for the object is higher than 3. None of procedures achieve better results than full combinatorial search and this indicate that we choused acceptable settings for algorithm testing.

In case of 77 parameters full search is not possible however, another two procedures work fast enough. Complexity limitation show better results in this simulation that means multistage consideration by only 20 parameters is not optimal for this dataset. Both partial search procedures show better than initial full search results that make them reasonable.

In case of 129 parameters multistage procedure shows best results on forecasting our testing object and can be recommended for high-dimensional dataset analysis.

4 Conclusion

The results of comparison of different algorithm limitations show that the Multistage combinatorial algorithm improves analysis of high-dimensional objects. Multistage procedure shows better result than complexity limitation approach when we increase the number of considered dimensions to some value. A drawback here is that this value is uncertain and depends a lot on input data set. The multistage procedure is suitable for highly efficient parallel processing and well exploits existing parallelization methodology.

References

- [1] Madala H.R., Ivakhnenko A.G.: Inductive Learning Algorithms for Complex Systems Modeling. CRC Press, 368 p., 1994.
- [2] Ivakhnenko A.G., Stepashko V.S.: Noise-immunity of modeling. Kiev: Naukova dumka, 216 p., 1985. (in Russian)
- [3] O.Koshulko A.Koshulko. Adaptive parallel implementation of the combinatorial GMDH algorithm. In proceedings of International Workshop on Inductive Modelling, Prague, 2007.
- [4] Top500 List – <http://www.top500.org>
- [5] HPL Benchmark – <http://www.netlib.org/benchmark/hpl/>