

Methods of true data mining model selection - with experimental results

Oleksandra Bulgakova¹, Pavel Kordik²

¹*International Research and Training Centre for Information Technologies and Systems of NAS and MES of Ukraine*

Akademik Glushkov Prospect 40, Kyiv, 03680, Ukraine.

²*Dept. of Computer Science and Engineering, Karlovo n.am. 13, 121 35 Praha 2, Czech Republic*

sashabulgakova@list.ru, kordikp@fel.cvut.cz, astrid@irtc.org.ua

Abstract. *This work presents the modeling results of different real noisy data (nalada: humanities, spirals_1 and spirals_2: too complex data, motol_brain: motol hospinal neurosurgery, boshouse: house prices and also artificial data), using intellectual computing – combinatorial group method of data handling (combi GMDH) and Group of adaptive models evolution (GAME) method. All this data, you can find in [1].*

The goal of our work is to get the best possible result on such noisy data and to compare the results of particular methods. Also, we will make to attempt to combine these two methods (Game_Combi_GMDH) to get better prediction.

Keywords

Data mining, prediction, real noisy data, group method of data handling (GMDH), group of adaptive models evolution (GAME), inductive modeling.

1. Introduction

The capability of induction is fundamental for human thinking. It is the next human ability that can be utilized in soft-computing, besides that of learning and generalization. The induction means gathering small pieces of information, combining it, using already collected information in the higher abstraction level to get complex overview of the studied object or process.

Inductive modelling methods utilize the process of induction to construct models of studied systems.

The construction process is highly efficient, it starts from the minimal form and the model grows according to system complexity. At first, the information from most important inputs is analyzed in the subspace of low dimensionality, later the abstracted information is combined to get a global knowledge of the system variables relationship.

Prediction of some data determines future values based on measuring previous values of that data. The goal is to predict unknown future values from available data. There are many methods for prediction time series, ranging from statistical methods to neural networks as typical black box methods.

In this paper we focus on two different methods based on inductive modeling: combi GMDH [2] and Group of adaptive models evolution (GAME) method. Also, we will make to attempt to combine these two methods to get better prediction and less complex structure of predictors.

We compare the performance of the combi GMDH with the performance of the GAME method. The comparison is performed on artificial and real data sets.

The setup of the experiments can be found in the Section 3.2 of this paper.

2. Methods

2.1 Combinatorial group method of data handling

Combi GMDH algorithm uses an input data sample as a matrix containing N levels (points) of observations over a set of M variables. A data sample is divided into two parts. If regularity criterion $CR(s)$ is used, then approximately two-thirds of

observations make up the training subsample N_A , and the remaining part of observations (e.g. every third point with same variance) form the test subsample N_B . The training subsample is used to derive estimates for the coefficients of the polynomial, and the test subsample is used to choose the structure of the optimal model, that is one for which *the regularity criterion* $CR(s)$ takes on a minimal value:

$$CR(s) = \frac{1}{N_B} \sum_{i=1}^{N_B} (y_i - \hat{y}_i(B))^2 \rightarrow \min$$

To obtain a smooth curve of criterion value, which would permit one to formulate the exhaustive-search termination rule, the full exhaustive search is performed on models classed into groups of an equal complexity. The first layer uses the information contained in every column of the sample; that is the search is applied to partial descriptions of the form

$$y = a_0 + a_1 x_i, \quad i = 1, 2, \dots, M$$

Non-linear members can be taken as new input variables in data sampling. The output variable is specified in this algorithm in advance by the experimenter. For each model system, normal equations is solved (by LMS method). At the second layer all models-candidates of the following form are sorted:

$$y = a_0 + a_1 x_1 + a_2 x_j, \quad j = 2, 3, \dots, M$$

The models are evaluated for compliance with the criterion, and the procedure is carried on as long as the criterion minimum will be find. To decrease calculation time recommend to select at some (6-8) layer a set of the best F variables and use only them in further full sorting-out procedure. In this way number of input variables can be significantly increased [2].

A salient feature of the GMDH algorithms is that, when they are presented continuous or noisy input data, they will yield as optimal some simplified *non-physical model*. In the case of discrete or exact data the exhaustive search for compliance with the precision criterion will yield what is called a *physical model*, the simplest of all unbiased models. For noisy or short continuous input data, simplified Shannon non-physical models, received by GMDH algorithms, prove more precise in approximation and for forecasting tasks. GMDH is the only way to get optimal non-physical models. Usage of sorting-out procedure guarantees selection of the best optimal model from all possible.

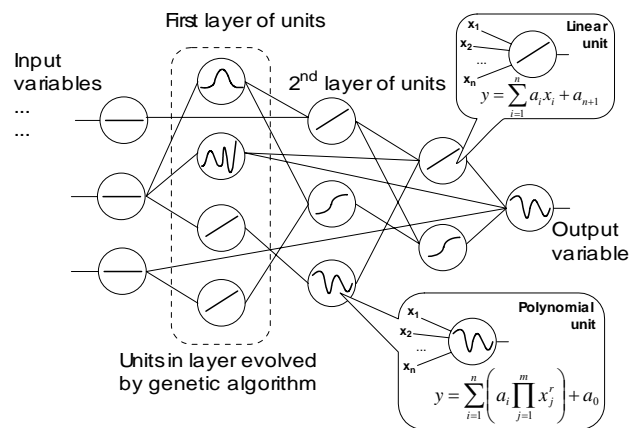


Figure 1. An example of the model produced by the GAME

2.2 Group of adaptive models evolution (game)

Group of Adaptive Models Evolution (GAME) [3] proceeds from the Group Model Data Handling (GMDH) theory [4]. GMDH was designed to automatically generate model of the system in the form of polynomial equations. An example of inductive model created by GAME algorithm is depicted on the Figure 1. Similarly to Multi-Layered Perceptron (MLP) neural networks, GAME units (neurons) are connected in a feedforward network (model). The structure of the model is

evolved by special niching genetic algorithm, layer by layer. Parameters of the model (coefficients of units' transfer functions) are optimized independently [5]. Model can be composed from units of different transfer function type (e.g sigmoid, polynomial, sine, linear, exponential, rational, etc). Units with transfer function performing well on given data set survive the evolution process and form the model. Often, units of several different types survive in one model, making it hybrid.

It is possible to combine the above two algorithms. The most straightforward way to do it is to use the Combi GMDH algorithm in a special linear or polynomial unit of the GAME algorithm.

2.3 GAME with active (combi GMDH) neurons

The GMDH network with active units was introduced in [6]. In the GAME network, units can be of several types (see Figure 1). We have implemented polynomial transfer function unit optimized by Combi GMDH algorithm. In the next section we experiment with homogeneous GAME networks, where just single type of units is enabled - Combi GMDH units.

3. Experiments

At first, we have tested our implementation of the Combi GMDH algorithm on artificial data sets with various levels of noise. The results showed that it is possible to identify true relationship even for medium levels of noise. Then we had experiments with different real data sets – nalada: humanities, spirals_1 and spirals_2: very complex data, motol_brain: motol hospinal neurosurgery and boshouse: house prices.

3.1 Data description

In this section, we want describe data sets, which we used in our experiments and also describe structures of this models:

1. Nalada data (humanities). This data set has 14 input variables and one output. The training data set containing 133 measurements was used for all methods. The data set for testing contained 66 subsequent measurements.
2. Spirals data: too complex data. This data set has 2 input variables and two output. The training data set containing 128 measurements was used for all methods. The data set for testing contained 64 subsequent measurements.
3. Motol brain data (motol hospinal neurosurgery). This data set has 4 input variables and one output. The training data set containing 3333 measurements was used for all methods. The data set for testing contained 1666 subsequent measurements.
4. Boshouse data (house prices). This data set has 12 input variables and one output. The training data set containing 338 measurements was used for all methods. The data set for testing contained 168 subsequent measurements.

3.2 Experimental setup

The differences of model output and the target values were measured as the RMS error, which can be computed as:

$$RMSE = \sqrt{\frac{1}{n} \sum (y - d)^2},$$

where n is number of target values, y is real value from the testing set and d is a predicted value. We used equal division to the training and testing data set. Bellow, we compare average accuracy of models produced by above described algorithms on different data sets. Their abbreviations are explained bellow:

- game – models produced by the GAME algorithm uses polynomial units with randomly generated transfer function

- game_combiGmdh_linear – homogeneous GAME networks with linear units optimized by the Combi GMDH algorithm
- game_combiGmdh_quadratic – homogeneous GAME networks with polynomial units optimized by the layered Combi GMDH algorithm
- combiGmdh_linear – original (Astrid implementation) Combi GMDH algorithm producing linear models
- combiGmdh_quadratic – original (Astrid implementation) Combi GMDH algorithm producing quadratic models

3.3 Results

In the Table 1 you can compare training and testing RMS error of models produced by various algorithms on data described in the section 3.1.

In all experiments where GAME algorithm was used, we build several models, and averaged the resulting RMS error.

Table 1. RMS error

model	data	art	brain	spiral_1
game	training	0,053	0,116	0,496
	testing	0,040	0,108	0,487
game_combiGmdh_lin	training	0,168	0,115	0,501
	testing	0,183	0,108	0,490
game_combiGmdh_qu	training	0,009	0,085	0,499
	testing	0,005	0,082	0,511
combigmdh_lin	training	0,170	0,509	0,500
	testing	0,162	0,517	0,491
combigmdh_qu	training	0,008	0,068	0,495
	testing	0,005	0,070	0,517
model	data	spiral_2	boshouse	nalada
game	training	0,500	5,375	0,139
	testing	0,493	4,735	0,125
game_combiGmdh_lin	training	0,499	4,068	0,138
	testing	0,493	3,399	0,119
game_combiGmdh_qu	training	0,500	3,541	0,142
	testing	0,490	3,089	0,121
combigmdh_lin	training	0,500	4,096	0,138
	testing	0,491	3,649	0,122
combigmdh_qu	training	0,495	3,965	NA
	testing	0,517	3,684	NA

From the Table, it can be seen that GAME+Combi_gmdh wins in most cases (4 times from 6). Also, you can see, that for nalada data set the GMDH quadratic model (construction of model is too time expensive due to the combinatorial explosion). The GAME quadratic model was created because it decomposes the problem and it does not use all input features (arguments) at once.

At the Figure 3, you can see structures of models produced by Game and the game_combiGmdh_qu algorithms on nalada data set. You can notice that the structure of the second model is considerably simpler than first. The reason is that the random transfer functions of the polynomial units in game need to be corrected several times to get optimal bias-variance tradeoff.

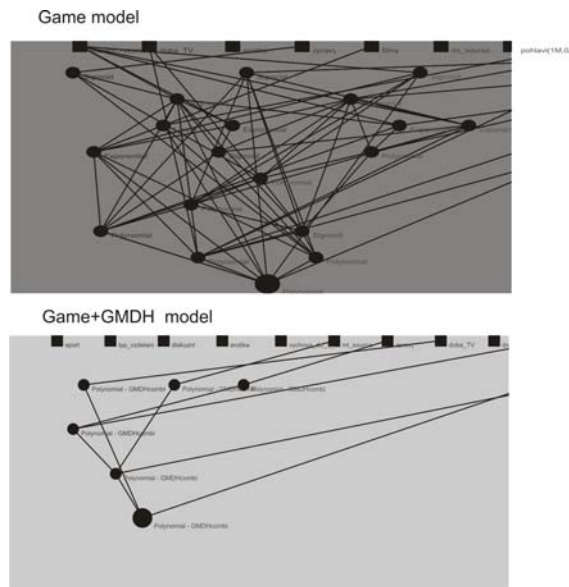


Fig.2: Structures of nalada data for Game (GAME network with radomly initialized polynomial units) and game_combiGmdh_qu (GAME network with polynomial units optimized by Combi algorithm).

4. Conclusion

Prediction of some data determines future values based on measuring previous values of that data. The goal is to predict unknown future values from available data. There are many methods for prediction time series, ranging from statistical methods to neural networks as typical black box methods.

In this paper we focused on two different methods based on inductive modeling: Combi GMDH and Group of adaptive models evolution (GAME) method. We combined these two methods into one algorithm.

We compared the performance of the Combi GMDH with the performance of the GAME containing randomly initialized polynomial units and units optimized by the Combi GMDH. Our results show that Combi GMDH units in GAME network are more efficient than the randomly initialized ones. Final network is simpler with often lower testing RMS error. The structure is also much simpler as demonstrated in Fig 2.

Layered structure of Combi GMDH quadratic units helps to make the search feasible even for medium number of input features. Our future work is to employ efficient heuristic search instead of full state space search used in Combi GMDH.

5. References

- [1] Files with data sets (open source): <http://neuron.felk.cvut.cz/game/data/>.
- [2] GMDH algorithms: describe, application, examples. http://www.gmdh.net/GMDH_com.htmv
- [3] Kordík, P. (2006), 'Fully Automated Knowledge Extraction using Group of Adaptive Models Evolution', PhD thesis, Czech Technical University in Prague, FEE, Dep. of Comp. Sci. and Computers.
- [4] Ivakhnenko, A.G. (1971), 'Polynomial theory of complex systems', IEEE Transactions on Systems, Man, and Cybernetics SMC-1(1), p.364-378.
- [5] Kordík, P.; Kovářík, O. & Šnorek, M. (2007), OPTIMIZATION OF MODELS: LOOKING FOR THE BEST STRATEGY, in 'Proceedings of the 6th EUROSIM Congress on Modelling and Simulation', ARGESIM, Vienna, p. 314-320.
- [6] Ivakhnenko, A.G.; Ivakhnenko, G.A. & Muller, J.A. (1997), 'Self-organization of Neural Networks with Active Neurons', Pattern Recognition and Image Analysis 4(2), 185-196.