Mathematical Modelling and Analysis
Volume 13 Number 1, 2008, pages 17–27
© 2008 Technika ISSN 1392-6292 print, ISSN 1648-3510 online

A COMPARISON OF HEURISTIC METHODS FOR POLYNOMIAL REGRESSION MODEL INDUCTION

G. JEKABSONS and J. LAVENDELS

Institute of Applied Computer Systems, Riga Technical University

Meza 1/3, LV1048, Riga, Latvia

E-mail: gintsj@cs.rtu.lv;jurisl@cs.rtu.lv

Received September 29, 2007; revised October 24, 2007; published online February 15, 2008

Abstract. We compare four different heuristic methods for polynomial regression model induction. The methods are very different in their approaches. Our main concern in this study is in the differences of candidate model spaces the methods deal with (completely predefined versus non-predefined), as well as search strategies used. We investigate the advantages and disadvantages of the approaches represented by the methods in terms of predictive error, complexity of the induced models and required computational resources. For empirical comparisons, we use twelve test problems.

Key words: Polynomial regression, model selection, heuristic search, state space.

1 Introduction

In regression modelling, to describe the relation between variables, typically a linear model is used. Linear models are very flexible and often used when there is no theoretical model available. A linear model may be defined by a linear summation of basis functions:

$$\hat{y} = \sum_{i=1}^{K} a_i f_i(x)$$

where K is the number of the model's basis functions $f_i(x) = \prod_{j=1}^{D} x_j^{r_{ij}}$, D is the number of the original predictor variables, r_{ij} is the order of the j-th variable in the i-th basis function (a non-negative integer), a_i are model's parameters. Note that when all r_j's of a basis function are equal to 0, we have the intercept term.

The problem is that the model to use should be neither too simple (causing underfitting) nor too complex (causing overfitting). Otherwise, model's ability to generalize to new data will be relatively poor.

A model should be selected based on its generalizability, rather than on its goodness of fit. In general, a data set containing N samples can be fit exactly by an equation with N terms (i.e., polynomial of order $N-1$). Thus, the complexity must be regulated for an induced model to be useful. To generalize well from the known data to new situations, a compromise must be made between the accuracy on training data, possible with complex models, and the robustness on new data, characteristic of simple ones. In addition, reducing model's complexity may help decrease the cost of acquiring data. The goal of model selection is to identify the model, from a set of competing models, which best captures the regularities underlying the process of interest and does not overfit.

When computing power was expensive, analysts were forced to employ rigid models even when the relationships between independent and dependent variables were not clear enough to warrant the necessary prior specifications. Because of modern computers, researchers have increasingly turned to modelling techniques which consider searching through very large sets of candidate models seeking for a model which has the best generalizability, requiring high computation resources.

Yet searching through all the candidate models in the candidate model space in most problems is impractical. Therefore a convenient paradigm for viewing such problems is that of heuristic search with each state in the space specifying a specific candidate model (e.g. a subset of basis functions). One can use heuristic search methods that traverse the space of models, e.g. by adding and deleting the basis functions, and select the best found model.

Usually, the space of polynomial models is chosen such that the order of each polynomial does not exceed previously chosen highest allowed order p, i.e. $r_{ij} = 0, 1, \ldots, p$ and $\sum_{j=1}^{D} r_{ij} \leq p$. The well known classical heuristic method for searching for the best model in such completely predefined spaces is Forward Selection (also known as Sequential Forward Selection, SFS) [1]. Some other methods include also use of some stochastic elements. They are, for example, Random Restart Hill Climbing, Random Mutation Hill Climbing, Simulated Annealing, as well as Genetic Algorithms (e.g. see [6]). In this paper, we will consider Sequential Forward Selection, and Random Mutation Hill Climbing (RMHC) as the former is the most widely known and the latter already has shown a rather reasonable performance in our previous studies [8] as well as in studies of other authors (e.g. [6]).

The above described approach presumes a *fixed* library of basis functions out of which competing models are built (and it requires user to specify p). Thus, modelling reduces to subset selection. Another approach is to work with non-predefined model spaces. One can use such polynomial refinement operators which allow generating polynomials of arbitrary order. The method Constrained Induction of Polynomial Equations for Regression (CIPER) [10] does exactly that. Or one can use another class of heuristic methods which solve the overall problem by solving portions of the problem at a time - blocks of terms at a time, and connect those blocks into a multi-stage model. The Group Method of Data Handling (GMDH) methods [3, 4, 5] fall into this category.

As can be seen, the four methods are very different in their approaches to polynomial model induction. It is not clear which one to use in different situations. This is a very important question also in practical applications of the methods. Our main concern in this study is in the differences of types of candidate model spaces the methods deal with (completely predefined versus non-predefined), as well as search strategies used. Our objective in this study is to investigate the advantages and disadvantages of the approaches represented by the methods in terms of predictive error, complexity of the induced models and required computational resources. For empirical comparisons, we use twelve test problem data sets with various sizes of data sample and numbers of variables paying most attention to situations with relatively short data samples (which is often the case in practical applications) where the partial polynomials outperform the full polynomials the most.

The rest of the paper is organized as follows. Section 2 deals with regression model selection criteria used in the modelling methods to regulate model complexity and to guide the search. Section 3 describes the four methods of polynomial regression model induction. Section 4 describes the performed empirical experiments, presents the results, and draws some conclusions about advantages and disadvantages of the methods (and their represented approaches).

2 Model Selection Criteria

One of the ways to perform model selection is to use complexity penalization criteria: train the models using all available data, measure model complexity (usually by the number of parameters, K) and choose the model which is best according to a function of K, the training error, the number of cases in training data set, N and (perhaps) a prior estimate of error variance, σ^2. Complexity penalization criteria usually do not require high computational resources and allow one to use all the data for training.

Some of the most widely known complexity penalization criteria are Akaike's Information Criterion, AIC [2], small sample corrected Akaike's Information Criterion, AICC [7], and the two-stage code version of the Minimum Description Length criterion (MDL) [9]. In general, one can define such criteria as a sum of deviance of the model and the complexity penalty:

$$CR = deviance + penalty.$$

The *deviance* term is equal to minus twice the log-likelihood. In a regression problem with known σ^2 this is the least-squares measure S divided by σ^2, and if σ^2 is not known, the *deviance* is $N \log(S/N)$. The *penalty* term for AIC is $2K$, for AICC is $2K + (2K(K+1))/(N-K-1)$, and for MDL is $K \log(N)$. The best fitting model is that whose criterion value is the lowest.

Note that the AIC definition with known σ^2 behaves equally to the Predicted Squared Error (PSE) [3] – one of the implemented criteria in Neuroshell2 software [11], which we used for GMDH modelling (see section 3.4).

3 The Methods for Induction of Polynomial Regression Models

The first two considered methods work with completely predefined candidate model spaces. A small example of such space is given in Figure 1. A search method traverses such space step by step seeking for the state (a subset of basis functions included in model) with the best value of the selection criterion, and guided by the so constructed evaluations. The goal is to find as good state as possible, hopefully global minimum – the best model, with as less state transitions as possible, since each transition requires evaluation of a number of models, what includes least-squares estimation of parameters for each model.

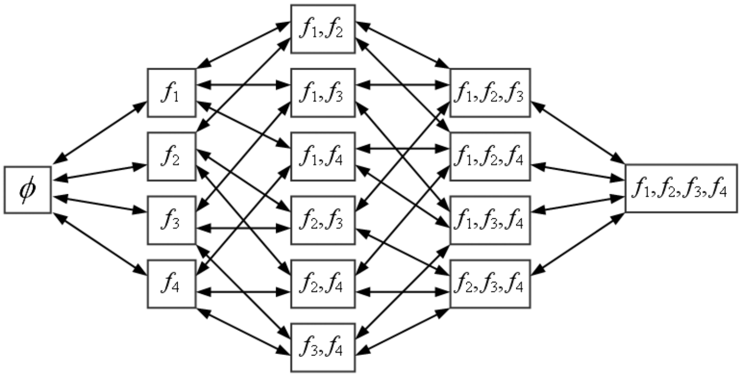

Figure 1. A small example of completely predefined candidate model space.

3.1 Sequential forward selection

SFS is one of the simplest most widely known search methods [1]. It starts with an empty set of selected basis functions and iteratively adds the function leading to the highest performance increase to the set of selected functions, until the performance cannot be enhanced any further by adding a single function.

SFS method is deterministic in the sense that no matter how many times we restart it from the beginning, we will always obtain the same solution. The method usually requires relatively small amount of computational resources, but because of its greedy and deterministic nature, it has a tendency to get trapped in local minima [4, 8].

3.2 Random mutation hill climbing

To escape from local minima, non-deterministic heuristic search methods can be used. Such methods include some use of stochastic elements, for example, restarts of search at random states or random transitions to states that may not be the best greedy choices. Consequently the non-deterministic methods may give different solutions in each restart.

One of the non-deterministic heuristic search methods is RMHC [6]. So far, it has shown rather reasonable performance in our previous studies [8]. RMHC works similarly to SFS except that it can also delete already existing basis functions and that the starting state of the search as well as the next current state of each iteration is chosen randomly. Usually RMHC is run with an outer loop, which restarts the search for a given number of times. Because of the restarts, RMHC is usually overall slower than SFS, but, because of its reduced sensitivity to local minima, usually finds solutions with better values of selection criteria. In our experiments of this study, the number of restarts is equal to 10. However, note that the initial state of RMHC in our experiments is the same as for SFS – the empty subset. One reason for this choice is computational – building smaller models is much faster. The other reason is that in our earlier studies it could be seen that in many real-world applications, where only small data sample was available, often there was only a little or no improvement in generalizability of models found by RMHC with random restarts, but the time consumption was much higher.

In our experiments, we used the implementations of both SFS and RMHC in our in-house software. Selection criterion used in both methods was AICC and, as a rule of thumb, the highest allowed order of polynomials, p, was set as high as possible but so that the number of all basis functions did not exceed the number of data cases.

3.3 Constrained induction of polynomial equations for regression

While the two methods described above can only refine the current model by adding a new basis function to it or deleting an existing one, and require the user to specify the maximal order of the polynomial, CIPER can also add a variable to an already existing basis function [10]. Using this kind of refinement, CIPER can generate polynomials of arbitrary order and the need to predefine the set of basis functions disappears. A small example of CIPER's candidate model space is given in Figure 2.

For polynomial regression model evaluation CIPER uses a slightly modificated the two-stage code MDL criterion [10]:

$$\mathrm{MDL} = N \log(S/N) + L \log(N),$$

where $L = \sum_{i=1}^{K} \sum_{j=1}^{D} r_{ij}$ is the "length" of polynomial, i.e. the sum of the orders of all basis functions in the polynomial.

CIPER employs beam search through the space of candidate models starting with a model which has only the intercept term in it [10]. CIPER, in

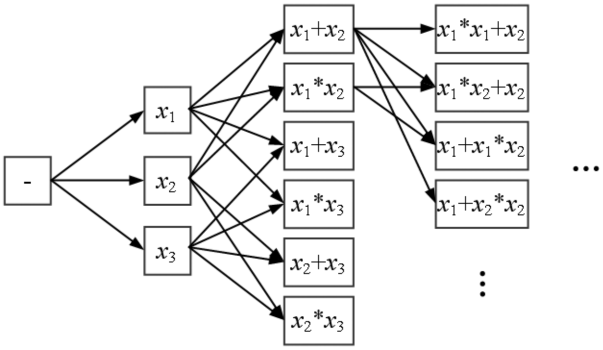

Figure 2. A small example of CIPER's candidate model space (for simplicity, the coefficients of the terms as well as the intercept term are omitted).

common with SFS, is deterministic. The method also usually requires relatively short amount of computational resources. But unlike SFS (and RMHC), CIPER can not move in direction of simpler models – it only can make the models more complex.

CIPER's candidate model space, in comparison with that of SFS and RMHC, has lower branching factor [10], which on the one hand leads to higher probability to get trapped in local minima and on the other hand makes the method faster and permits the use of high beam widths to avoid the entrapment. The default beam width of CIPER is 16 (in our experiments it was also fixed to 16).

CIPER was already empirically compared with linear regression, Stepwise Regression with F-test, Regression Trees, and Model Trees in [10]. It usually outperformed the other methods.

3.4 Group method of data handling

Another method which can be used for polynomial regression model induction is GMDH [4, 5]. The method is mainly referred as a method for self-organizing polynomial (neural) networks. The most widely known of its varieties work exclusively with polynomials and therefore also the result of the method can be written in polynomial form.

The building blocks of GMDH usually are second or third order polynomials (often partial) of two or three input variables. Such building blocks, also called partial descriptions (PDs), like neurons in neural networks, are arranged in layers (see Figure 3). The coefficients of each PD are estimated using the ordinary least-squares method by trying to approximate the original dependent variable y of training data. The exact number of layers, composition and structure of PDs and connections is not set a priori, but is the object of search layer by layer. The number of PDs selected in each layer, F, is preferred to be as large as possible. However, in practice it is known that GMDH methods perform enough well for F equal to the number of original input variables.

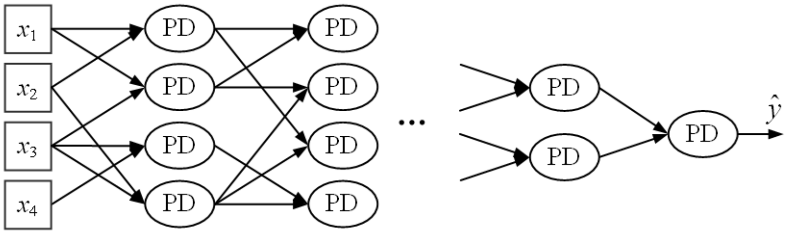

Figure 3. An example of GMDH polynomial network structure.

GMDH is said to be advantageous in handling a relatively big number of variables (including irrelevant ones) and noisy or small data samples [4]. In comparison with such one-step-greedy methods as SFS, RMHC, and CIPER, described above, GMDH grows models by a block of terms at a time – so arguably the local minima are avoided more efficiently [4].

There exists a wide spectrum of GMDH methods, even if we consider only those which work only with polynomial PDs (e.g. see [3, 4, 5, 11]). In our experiments, we used a variety of GMDH implemented in Neuroshell2 tool developed by Ward Systems Group, Inc. [11], which is also similar to the well known Algorithm for Synthesis of Polynomial Networks, ASPN [3]. In our experiments, the following GMDH settings were used: maximal complexity of PD is a full polynomial of third order with three inputs; full exhaustive optimization of each PD (this is the only choice in which we know for sure what the software is doing as all other choices are not fully described in the user's manual); F is equal to the number of original input variables; PDs of each successive layer can take inputs from PDs of immediately preceding layer as well as the original input variables; as a selection criterion, PSE was used as it is one of the most frequently used complexity penalization criteria for GMDH (e.g. [3, 4, 11]).

4 Experimental Comparisons

In our comparisons, we used twelve various publicly available regression data sets which all can be downloaded from the WEKA project website (http://www.cs.waikato.ac.nz/ml/weka/). We chose data sets with relatively small sample sizes, as this is often the case in practical applications. Table 1 presents the basic properties of the data sets.

4.1 Experimental methodology

In all the experiments presented here, we estimated predictive error of the induced models on unseen data samples using 10-fold Cross Validation (CV). In contrast to the three deterministic methods, the RMHC, because of its stochastic nature, was executed 10 times in each CV iteration and the results were averaged.

Table 1. The basic properties of the data sets (number of data cases and input variables) and the maximum order of the polynomials for SFS and RMHC.
* denotes data sets the order of cases was left unchanged as it was already randomized, for all other data sets the order was randomized.

Data set	Cases	Vars	p
autoMpg	392	7	4
autoPrice	159	15	2
bodyfat	252	14	2
bolts	40	7	2
elusage	55	2	8
housing	506	13	2
housingNOX	506	13	2
machine-cpu	209	6	3
pollution*	60	15	1
pwLinear*	200	10	2
servo*	167	4	5
triazines*	186	60	1

The predictive error of a model is measured in terms of relative root mean squared error, RRMSE defined as model's root mean squared error divided by standard deviation of the dependent variable y, both calculated using the unseen examples of the test set.

4.2 Experimental results

We present the results of the experiments in Tables 2 and 3. Table 2 compares the methods in terms of their predictive error and its standard deviation. Table 3 compares the time of the methods needed to induce the models and the complexity of the induced models in terms of number of parameters and number of included original input variables. All experiments were performed on Pentium 4 2.4GHz computer with Hyper Threading turned on.

In terms of predictive error, none of the methods significantly outperformed the other ones. But some tendencies could be observed. In comparison with SFS, the RMHC, despite of its better avoidance of local minima, did not perform as well as was predicted. Apparently, there are two main reasons for this. The first reason is that such traditional model selection criteria as AICC (and many other) are asymptotic – when the available data sample is relatively small, the induced models may overfit the data. The second reason is the RMHC's search intensity – in big candidate model spaces it can find models which have low AICC just by chance (having high predictive errors at the same time). Yet it seems that the SFS still can successfully find reasonably good models just because it gets stuck in some early local minima. For the same reason, it almost always has constructed less complex models than those of RMHC.

The time consumption of RMHC is much higher than that of SFS which is explainable by the 10 iterations of its outer loop of the search.

Table 2. Average RRMSE (%) and its standard deviation of the induced models.

	SFS		**RMHC**		**CIPER**		**GMDH**	
Data set	*Err.*	*(STD)*	*Err.*	*(STD)*	*Err.*	*(STD)*	*Err.*	*(STD)*
autoMpg	37.82	(4.61)	38.12	(6.02)	38.31	(6.50)	35.07	(6.68)
autoPrice	53.10	(18.64)	49.50	(26.97)	46.62	(16.43)	49.71	(20.47)
bodyfat	15.30	(12.86)	16.79	(12.81)	14.71	(13.01)	15.74	(14.18)
bolts	18.83	(41.36)	9.51	(16.19)	7.05	(13.88)	44.72	(98.56)
elusage	43.93	(12.63)	51.87	(24.90)	43.39	(12.00)	43.39	(12.00)
housing	41.72	(12.82)	41.40	(12.50)	45.21	(18.51)	42.50	(13.59)
housingNOX	37.65	(7.06)	37.54	(7.16)	37.46	(5.67)	38.78	(3.57)
machine-cpu	34.57	(9.09)	62.79	(40.41)	41.60	(16.09)	49.15	(21.34)
pollution	73.11	(22.80)	78.68	(21.36)	80.96	(24.59)	103.44	(36.81)
pwLinear	35.45	(6.96)	35.90	(6.66)	60.42	(23.66)	36.04	(7.16)
servo	39.59	(15.38)	43.70	(15.44)	59.24	(14.87)	39.77	(14.82)
triazines	99.92	(15.75)	163.13	(227.68)	96.19	(11.38)	110.16	(33.61)

Table 3. Average running time (in s), average number of basis functions (K) and average number of included variables (d) of the induced models (GMDH time was measured only to a precision of 1 second).

	SFS		**RMHC**		**CIPER**		**GMDH**	
Data set	*Time*	$K(d)$	*Time*	$K(d)$	*Time*	$K(d)$	*Time*	$K(d)$
autoMpg	45.5	26 (7)	373.9	38 (7)	1.2	7 (5)	8	18 (6)
autoPrice	2.6	16 (10)	57.3	34 (14)	1.4	6 (4)	63	15 (7)
bodyfat	1.7	13 (7)	47.2	25 (9)	1.2	6 (3)	10	6 (2)
bolts	0.02	6 (3)	0.3	5 (3)	0.6	6 (4)	2	2 (2)
elusage	0.01	3 (1)	0.3	4 (1)	0.1	3 (1)	<1	3 (1)
housing	49.9	48 (13)	274.7	46 (13)	23.1	14 (10)	114	33 (12)
housingNOX	24.8	35 (12)	174.4	34 (13)	19.1	14 (10)	104	28 (9)
machine-cpu	3.2	24 (6)	22.3	24 (6)	1.1	7 (5)	8	8 (4)
pollution	0.01	8 (7)	0.2	7 (6)	0.7	6 (6)	87	15 (8)
pwLinear	1.0	16 (9)	8.4	18 (9)	2.5	10 (7)	31	17 (7)
servo	3.5	21 (4)	42.6	31 (4)	0.4	6 (4)	<1	31 (4)
triazines	0.7	11 (10)	7.6	12 (11)	11.4	8 (8)	2283	49 (18)

In terms of predictive error, the CIPER performs comparably to SFS and RMHC but works much faster and, what is even more important, it does not require user to set the maximum order of the polynomials. Though, in some of test problems it has elevated predictive error, which may be explained by the fact that the version of CIPER uses a not well studied ad-hoc model selection criterion which seems to penalize too much the models which contain products of input variables. Other reasons may be that firstly the search algorithm of CIPER has low branching factor, and secondly it can only add new basis functions or add new variables to already existing basis functions but can not remove any of them from the model. Both these reasons can cause the algorithm to get stuck in local minima too early. The possibility of removal or use of another criterion would be a possible improvement of the search algorithm and needs some additional study.

The high penalization ratio of CIPER's model selection criterion has also caused it to choose much less complex models as those of any other method.

They also include less input variables.

The performance of GMDH was similar to the other methods however not as good as promised in [5]. In [4] was suggested that in certain situations GMDH can have a higher predictive error because of combining variables in pairs or threes, rather than larger groupings, when forming PDs – "the "reach" of the method is limited when contributions from variables must truly be simultaneous".

Selecting the maximum order of the polynomial for SFS and RMHC is a nontrivial problem, since it can differ from one data set to another and would also be, for practical reasons, guided by computational complexity issues. The non-predefined candidate model spaces of CIPER and GMDH makes the task easier – the methods "choose" the orders themselves and achieve similar results.

5 Conclusion

This paper presents a comparison of four different heuristic methods for polynomial regression model induction. The methods are very different in their approaches. We investigate the advantages and disadvantages of the approaches represented by the methods in terms of predictive error, complexity of the induced models and required computational resources.

In terms of predictive error, overall none of the methods significantly outperformed the other ones. But some tendencies could be observed. The main observation can be summarized in the following: because of the non-predefined candidate model spaces of CIPER and GMDH (in contrast to SFS and RMHC) the user does not need to set the maximal order of the polynomial – the methods "choose" the order themselves and achieve similar predictive performance with less complex models by using less computational resources.

Directions of future research include developments of combinations of these methods with the aim to combine their advantages. For example, it is possible to create GMDH polynomial networks with CIPER PDs, making them more flexible.

Acknowledgement

This work has been partly supported by the European Social Fund within the National Program "Support for the carrying out doctoral study program's and post-doctoral researchers" project "Support for the development of doctoral studies at Riga Technical University".

References

[1] D.W. Aha and R.L. Bankert. A comparative evaluation of sequential feature selection algorithms. In H.J.Lenz D.Fisher(Ed.), *Learning from Data*, volume 4, pp. 199–206. Springer, NY, 1996.

[2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**:716–723, 1974.

[3] R.L. Barron, A.N. Mucciardi, F.J. Cook, J.N. Craig and A.R.Barron. Adaptive Learning Networks: Development and Application in the United States of Algorithms related to GMDH Ch. 2. In S.J. Farlow(Ed.), *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, pp. 25–65. Marcel Dekker, NY, 1984.

[4] J.F. Elder IV and D.E. Brown. Induction and Polynomial Networks. In M.D.Fraser(Ed.), *Network Models for Control and Processing*, pp. 143–198. Portland, OR, Intellect, 2000.

[5] S.J. Farlow. *Self-organizing Methods in Modeling: GMDH Type Algorithms*, volume 54. Marcel Dekker, Inc., NY, 1984.

[6] S. Forrest and M. Mitchell. Relative building-block fitness and the building-block hypothesis. *Foundations of Genetic Algorithms*, **2**:109–126, 1993. D. Whitley (ed.), San Mateo, CA: Morgan Kaufmann

[7] C.M. Hurvich and C.L. Tsai. Regression and time series model selection in small samples. *Biometrika*, **76**:297–307, 1989.

[8] G. Jekabsons, J. Lavendels and V. Sitikov. Model evaluation and selection in multiple nonlinear regression analysis. *Mathemacal Modelling and Analysis*, **12**(1):81–90, 2007.

[9] J. Rissanen. Modeling by shortest data description. *Automatica*, **14**:465–471, 1978.

[10] L. Todorovski, P. Ljubic and S. Dzeroski. Inducing polynomial equations for regression. *Lecture notes in computer science, Lecture notes in artificial intelligence*, **3201**:441–452, 2004.

[11] Ward Systems Group. Inc. *Neuroshell 2. User's Guide*. Frederick, MD, USA, 1998. (http://www.wardsystems.com)