World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:2, No:2, 2008

Ensembling Adaptively Constructed Polynomial Regression Models

Gints Jekabsons

***Abstract*—**The approach of subset selection in polynomial regression model building assumes that the chosen fixed full set of predefined basis functions contains a subset that is sufficient to describe the target relation sufficiently well. However, in most cases the necessary set of basis functions is not known and needs to be guessed – a potentially non-trivial (and long) trial and error process. In our research we consider a potentially more efficient approach – Adaptive Basis Function Construction (ABFC). It lets the model building method itself construct the basis functions necessary for creating a model of arbitrary complexity with adequate predictive performance. However, there are two issues that to some extent plague the methods of both the subset selection and the ABFC, especially when working with relatively small data samples: the selection bias and the selection instability. We try to correct these issues by model post-evaluation using Cross-Validation and model ensembling. To evaluate the proposed method, we empirically compare it to ABFC methods without ensembling, to a widely used method of subset selection, as well as to some other well-known regression modeling methods, using publicly available data sets.

***Keywords*—**Basis function construction, heuristic search, model ensembles, polynomial regression.

I. Introduction

VARIOUS applications in engineering, statistics, computer science, health sciences, and social sciences are concerned with estimating "good" predictive models from available data. In such problems the goal is to estimate unknown dependency (or model) from training data, in order to use this model for predicting future samples. A model describes the relation between a multidimensional input, *x*, and an output, *y*. If the *y* is a continuous variable, the task is called a regression task and the model is called a regression model.

In regression commonly polynomial models are used. Polynomials are very flexible and often used when there is no theoretical model available. However, the regression model to use should be neither too simple (too low number of basis functions in polynomial), causing underfitting, nor too complex (too high number of basis functions), causing overfitting. Otherwise model's ability to generalize to new data will be relatively poor.

To obtain a polynomial regression model, that describes the relations in data sufficiently well, typically the subset selection (also called variable selection) approach [1] is used where the goal is from a fixed full set of predefined basis functions to find the best subset that gives the best predictive performance of the regression model. Usually, the full set of basis functions is equal to the set of basis functions in a predefined "full" model of a maximal allowed complexity (which is usually chosen as a full polynomial of some order).

To find the best subset some kind of search must be performed. Searching through all possible subsets needs exponential runtime and thus is impractical in most cases. Hence heuristic search methods are used. They efficiently traverse the space of subsets, by adding and deleting basis functions and use an evaluation measure that directs the search into areas of increased performance. The typical examples of the search methods are the Forward Selection (also known as Sequential Forward Selection, SFS) and the Backward Elimination (also known as Sequential Backward Selection, SBS) [1], [2].

The approach of subset selection assumes that the chosen *fixed* full set of *predefined* basis functions contains a subset that is sufficient to describe the target relation sufficiently well. However, in most cases the necessary full set is not known and needs to be guessed (e.g., by specifying the order of the "full" model) since it will differ from one data set to another. In many cases that means either a non-trivial (and long) trial and error process or acceptance of a possibly inadequate model.

We consider a different approach than the subset selection – letting the regression model building method itself construct the basis functions necessary for creating the model without restricting oneself to the basis functions of a predefined full model. This is achieved by replacing the standard refinement operators of subset selection, namely the addition and deletion of the basis functions, with other operators that not only allow adding or deleting but also allow changing the basis functions themselves. In this manner all the needed basis functions are adaptively constructed during the heuristic search process, efficiently trading-off the simplicity and predictive performance of the models. Hence we call this approach Adaptive Basis Function Construction (ABFC). The approach allows generating polynomials of arbitrary complexity, does not require the user to predefine any basis functions for model creation, and, in addition, allows using most of the same

Manuscript received January 28, 2008. This work was 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".

G. Jekabsons is with the Institute of Applied Computer Systems, Riga Technical University, Riga, Latvia (e-mail: gintsj@cs.rtu.lv).

heuristic search algorithms and evaluation measures which are used in subset selection methods. Note that the ABFC approach is partially a generalization of the ideas from a rather recently published work [3] where the authors introduce to a polynomial equation induction method called Constrained Induction of Polynomial Equations for Regression (CIPER) which was developed in the context of differential equation discovery, inductive databases, and constraint-based data mining. The CIPER can also be viewed as an instance of the ABFC approach. However, as we already demonstrated in our previous work (e.g., [4]), it has some drawbacks regarding too high sensitivity to local minima and the nesting effect [5]. In [4] we introduced another instance of the ABFC approach – a regression model building method called Floating ABFC (F-ABFC).

However, there are two issues that to some extent plague the methods of both the subset selection as well as the ABFC, especially when working with relatively small data samples: These are the selection bias and the selection instability (see Section III for details).

Both these issues are usually ignored frequently resulting in models of lower predictive performance (and rising criticism to subset selection, e.g., [1], [6]). In our research we try to correct these issues for the F-ABFC by using a collaboration of two techniques: 1) *v*-fold Cross-Validation (CV; e.g., [7]) over the entire search process strictly for selection of one best model from the best models of each iteration (the validation set is not used for model evaluation during the search, instead it is used for post-evaluation after the search process has ended); 2) the *v* models from the *v* CV loops are combined using a simple ensemble technique – model averaging [8].

To evaluate the proposed method "Ensemble of Floating ABFC" (EF-ABFC), we empirically compared its predictive performance to the original F-ABFC, to CIPER, to a widely used instance of subset selection, SFS, to full polynomials, as well as to some other well known regression modeling methods using four different publicly available regression data sets.

In the following section we shortly review the ABFC approach (mainly from the viewpoint of heuristic search) and describe F-ABFC. The EF-ABFC is proposed in the third section. The fourth section deals with the empirical comparisons of the regression modeling methods.

II. Adaptive Basis Function Construction

A polynomial regression model may be defined by a linear summation of basis functions:

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

where a_i are model's parameters; k is the number of basis functions (equal to the number of model's parameters); $f_i(x)$ are the basis functions that generally may be defined as a product of original input variables each raised to some order:

$$f_i(x) = \prod_{j=1}^{d} x_j^{r_{ij}} \quad (2)$$

where d is the number of the original input variables; r_{ij} is the order of the j-th variable in the i-th basis function (a non-negative integer). Note that when all r_j's of a basis function are equal to 0, we have the intercept term. The estimation of parameters a_i of the polynomial regression models is made based on a finite number, n, of training data cases, $(x_{(1)}, y_{(1)})$, $(x_{(2)}, y_{(2)})$, …, $(x_{(n)}, y_{(n)})$, typically using the ordinary least-squares method, OLS.

Summarizing [9]-[11], in order to characterize a heuristic search problem one must define the following: 1) initial state of the search; 2) available state-transition operators; 3) search strategy; 4) evaluation measure; 5) termination condition. Note that in the rest of this paper instead of the term "state-transition operator" we will use the term "(model) refinement operator" (as in [3], [4]), which is somewhat more convenient in the context of regression model building.

In the polynomial regression subset selection approach, typically the *initial states* are models that correspond to the empty subset, the subset with only the intercept term in it, full subset of all the defined basis functions, or a randomly chosen subset; the typical *refinement operators* are addition and deletion of any one basis function; the typical *search strategy* is the hill climbing [11] which in combination with the empty subset initial state and the addition operator becomes SFS but in combination with the full subset initial state and the deletion operator becomes SBS; the classical *evaluation measures* are the statistical significance tests [1], however, currently two other strategies predominate: employment of complexity penalization criteria (e.g., the Akaike's Information Criterion, AIC [1], [6], [12], [13]) and the resampling techniques (e.g., Hold-Out, CV, and Bootstrap [1], [7]); the *termination condition* typically corresponds to finding of state in that none of the refinement operators can lead to a better state.

The main difference between the subset selection approach and the ABFC approach is in the refinement operators used. As already said, in the ABFC approach, the standard refinement operators of subset selection are replaced with other operators that not only allow adding or deleting the basis functions but also allow changing the basis functions themselves (i.e., increasing or decreasing orders of variables).

In Fig. 1, there is shown relation between subset selection and ABFC approaches. Subset selection operates with a string of bits of constant length (column named "Included") where each bit indicates whether a predefined basis function (columns "Function" and "Form") is present ("1") or absent ("0"). ABFC approach on the other hand operates directly with the orders of each input variable in each function as well as creates new functions as necessary (column "Matrix of orders, *r*"). Thus, in ABFC the search operates directly with the dynamically-sized matrix r in (2). This results in an infinite space of candidate regression models and we can

generate polynomials of arbitrary complexity without predefining any basis functions.

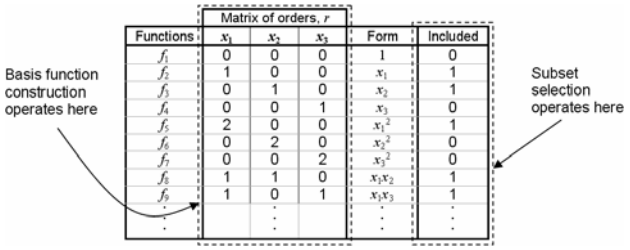

Fig. 1 Relation between subset selection and ABFC

In ABFC, as the state space has become infinite, a natural *initial state* of the search is now the state where the subset of basis functions is empty or, for example, the subset with one function that corresponds to the intercept term already included. In F-ABFC the second variation is used (moreover, the function stays in the model at all times and is not allowed to be modified or deleted).

Using efficient *refinement operators* is vital for the search process to be successful. Here are the five considered refinement operators. *Operator1*: Addition of a new basis function with one of the orders set to one (and all the others set to zero). *Operator2*: Increasing of one of the orders in one of the existing basis functions by one. *Operator3*: Addition of an exact copy of an already existing basis function with one of the orders increased by one. *Operator4*: Decreasing of one of the orders in one of the existing basis functions by one. *Operator5*: Deleting of one of the existing basis functions.

Additionally for each refinement operator, except *Operator5*, special care is taken to prevent basis function duplicates in the resulting model. Also, to maintain generality, the function corresponding to the intercept term is not allowed to be used for construction of other functions (by *Operator3*) [4].

We categorize the listed refinement operators as complication operators (the first three) and simplification operators (the last two). If the search is started from an empty or some small set of functions, the complication operators do the main job – they "grow" the model. The simplification operators on the other hand work as purifiers – they decrease the unnecessarily high orders and delete the unnecessary basis functions.

The initial state and the refinement operators together form state space [11]. In Fig. 2, there is shown a small example of the state space for F-ABFC. Each state represents a set of basis functions which are included in the regression model – the matrix *r*. However, note that connections created by the *Operator3* here are not shown – they would simply be cross-layer connections between the states.

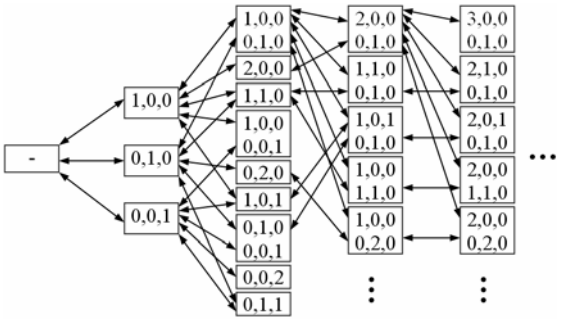

Fig. 2 A small example of state space in F-ABFC when the number of input variables is three. For simplicity the function with all 0's and connections of *Operator3* are omitted

In relation to *search strategies* most of the algorithms that are applicable to subset selection can also be used in ABFC. This is achieved by treating the complication and simplification operators as the addition and deletion operators (correspondingly) of the subset selection approach [4]. However, there are two exceptions. Firstly we can not use the search algorithms that start their search from the full subset (e.g., SBS) as in ABFC there exists no full subset. And secondly, we can not use the strategies that require the state-representing data structures to be of constant length and are not generally biased towards simpler models (e.g., the chromosomes in most Genetic Algorithms). However, with appropriate modifications they might become applicable.

In our research so far we considered only the directly applicable search strategies the simplest of which is the SFS. However, SFS moves only forward, in direction of more complex models, so it would use only the complication operators. Hence the Steepest Descent Hill Climbing [11], Plus-a Take Away-b [14], and Sequential Floating Forward Selection (SFFS) [5] strategies may be considered, all of which allow also the backward moves. F-ABFC's search strategy is based on SFFS (that already showed its advantages in [4], [5]) – hence the name of the method.

As *evaluation measures* the same predominating ones of subset selection can be efficiently used also in ABFC: complexity penalization criteria and resampling techniques. The former in contrast to the latter usually does not require high computational resources, allows one to use all the available data for training, as well as is less noisy (creating less local minima in the state space). The most widely known and used complexity penalization criterion is AIC. In F-ABFC we use its small sample corrected version (AICC) [6], [13] as for problems with small *n* it is suited better than AIC but converges to AIC as *n* becomes large [6], [13]. AICC criterion is defined as follows:

$$AICC = n\log(MSE) + 2k + (2k(k+1))/(n-k-1) \quad (3)$$

where MSE is Mean Squared Error in training data. Note that the best fitting model is that whose AICC value is the lowest. For more details on using AICC in F-ABFC consult [4].

The typical *termination condition*, that is met when the search locates a state in that none of the refinement operators can lead to a better state, is of course a natural choice also in the ABFC.

A more detailed description of the approach, its instance, F-ABFC, as well as a more detailed theoretical comparison to subset selection approach is given in our previous study [4].

III. Ensemble of Models Built by F-ABFC

As already said, there are two issues that plague the methods of both the subset selection as well as the ABFC – the selection bias and the selection instability.

Selection bias occurs when in the search procedure one uses the same data to compute values of models' parameters and also to evaluate the models for selection purposes [15]-[17]. This also includes usage of resampling techniques (e.g., CV) that evaluate a model using validation data set and then put the validation data back into the training set, e.g., like the well-known Prediction Sum of Squares (PRESS) resampling-based criterion [1], [18] does. In addition, it occurs even when performing model evaluation using completely independent validation set – because the search procedure is evaluating so many subsets, it is likely that some of them lead to models that have high accuracy for the validation set but low accuracy for the test set [19]. In any case, the more intensive the search procedure, the larger the selection bias will be.

The other issue, selection instability (also called selection variance), is related to the fact that small perturbations of the data can lead to vastly different subsets of the basis functions, e.g., because of getting stuck in different local minima or because of noisy model evaluation methods [20]-[22].

In EF-ABFC, to deal with the selection bias, *v*-fold CV over the entire search process is used. Note that as an evaluation measure for the search algorithm still the AICC is applied. The validation set, however, is used for a post-evaluation (using MSE in validation data) of the best models of each iteration of the search and for selection of the one final best model. This post-evaluation can detect whether the search process at some iteration might have started to generate overfitted models and select a model of some earlier iteration that is hopefully not (or at least less) overfitted (see Fig. 3). The whole process (including the search from the initial state) is repeated *v* times, each time with different CV fold served as the validation set and all the other folds served as the training set, producing *v* different models.

To deal with the selection instability, the *v* models from the *v* CV loops are combined using a simple ensemble technique – unweighted model averaging [8]. Note that, prior to combining, all the models are re-fitted to the whole training set (without the CV partitioning). This is done to compensate for the smaller training set used during the individual model building.

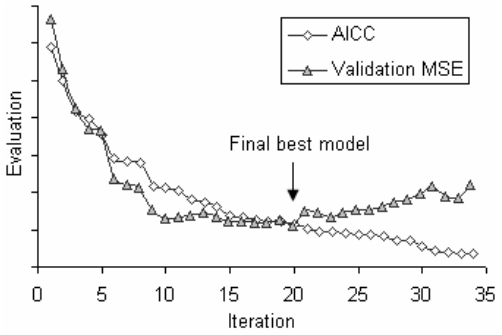

Fig. 3 An example of how less overfitted model is selected using post-evaluation in validation set. Note that here starting from the 35th iteration also the AICC values start to increase, however, this might be too late due to selection bias

The used ensemble building method is similar to Bagging (bootsrap aggregating; [21]) where the training set is bootstrapped (usually to build varied decision trees), and the unweighted average of their estimates is taken.

Model combining by unweighted model averaging consists in taking the average of estimates of all the models:

$$\hat{y}_{comb} = (1/v)\sum_{i=1}^{v} \hat{y}_i \quad (4)$$

where $\hat{y}_i$ is *i*-th individual model and $\hat{y}_{comb}$ is the combined model. For polynomial regression models this simply means summation of all the polynomials and then division of all the parameters of $\hat{y}_{comb}$ (which is also a polynomial) by *v*.

Combining models in this way can have the effect of smoothing out erratic models that overfit the data and gain more stability in the modeling process [8], [15], [21].

See Fig. 4 for an outline of the EF-ABFC modeling process when the number of CV folds *v* is three. However, note that in our experiments we use $v = 10$. This is because too small number of models in ensemble will yield too little diversity hindering the models to correct each others errors, but, on the other hand, using too many models will yield no further improvement [8], [21], [22]. Additionally, too high number of CV folds can yield unreliable validation MSE estimates for the selection of the individual final best models, as the validation sets will be too small.

In recent literature there is ever growing confidence that model ensembles often perform better than individual models and consistently reduce generalisation error [6], [8], [15], [21]-[23]. However, note also that model ensembles are not always the best solutions [22]: if there is too little data, the gains achieved via an ensemble may not compensate for the decrease in accuracy of individual models, each of which now sees an even smaller training set. On the other end, if the data set is sufficiently large, even a single flexible model can be quite adequate. Using large data sets also considerably decreases potential selection bias, so superiority of EF-ABFC over F-ABFC in such situations is expected to diminish.

Another criticism of ensembles is that surely their increased

complexity will lead to overfit and thus, inaccuracy on new data. However, it is known that model ensembles often have effective complexity *less* than their components [15].

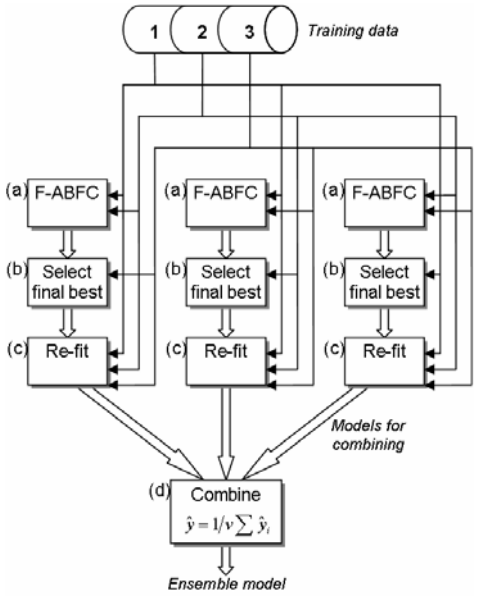

Fig. 4 An outline of the EF-ABFC modeling process when v = 3. (a) search for the best model according to AICC using F-ABFC; (b) select the one final best model according to validation MSE; (c) ret-fit the model (recalculate its parameters) using the whole training data; (d) combine the models.

IV. Empirical Experiments

The main goal of the performed experiments was to compare the EF-ABFC to the original F-ABFC, to CIPER, to a widely used instance of subset selection, SFS, as well as to full polynomials (FP) and to some other well known regression modeling methods – Radial Basis Function Neural Networks (RBFNN), Multi-Layer Perceptrons (MLP) and M5' model trees (MT) [23]. We compared the methods in terms of both, predictive performance of the induced regression models as well as necessary computational resources. The performance of the methods is evaluated on four different regression data sets from the Torgo repository (http://www.liaad.up.pt/~ltorgo/ Regression/DataSets.html).

All the experiments were performed on Pentium 4 2.4GHz computer with Hyper Threading turned on. Note that the time consumption presented in the tables is only a rough measurement as the methods are implemented in different software and with different levels of optimization of calculations. In the experiments we used our in-house software with implementations of EF-ABFC, F-ABFC, CIPER, and SFS (all with the AICC criterion), as well as the full polynomials (of order p). Usage of AICC allows us to compare the use of refinement operators and search strategies without any hindrance because of the different criteria. As an implementation of the original CIPER (with a modification of the Minimum Description Length (MDL) [24] criterion instead of AICC) we used the original software that is publicly available at http://ai.ijs.si/pljubic/ciper/ciper.html, kindly provided by the authors of the original method. In both versions of CIPER we used the default beam width, which is 16. As implementations of RBFNN, MLP, and MT we used the WEKA software [23]. For RBFNN the number of clusters is selected from range [1,40] using 10-fold CV. For MLP and MT we used the default parameters.

In the experiments we estimated predictive performance of the built models on unseen data samples using 10-fold CV and averaged the results. Note that with all the methods, including EF-ABFC, the CV was done as an outer loop over the entire model building process and, in each CV iteration, the set aside test set was not used until the final evaluation of the built model. Here it is important to distinguish between the CV inside the EF-ABFC or RBFNN and the CV used for evaluation of model building methods.

The predictive performance of a model in test data set is measured in terms of Relative Root Mean Squared Error:

$$RRMSE = \sqrt{\sum_i \left(y_{(i)} - \hat{y}_{(i)}\right)^2} \Big/ \sqrt{\sum_i \left(y_{(i)} - \bar{y}\right)^2} \quad (5)$$

where $\hat{y}_{(i)}$ is the corresponding predicted value for the observed value $y_{(i)}$; $\bar{y}$ is the mean of the observed values. The lower the value of RRMSE, the more accurate the model.

The four data sets from the repository are the following: "housing", "housingNOX" (both 506 cases, 13 input variables), "autoPrice" (159 cases, 15 input variables), and "machine-cpu" (209 cases, 6 input variables). They are chosen because of the relatively low number of data cases, which is also common in real practical situations, as well as because of mostly continuous input variables. Note that prior to dividing the data sets into CV folds, the order of the cases was randomized.

Table 1 presents the results of the performed experiments. The results confirm the superiority of EF-ABFC over F-ABFC. With "housing" data set F-ABFC is the second best, with "housingNOX" it is outperformed by CIPER, MT and FP, with "autoPrice" it is outperformed by MT, and with "machine-cpu" it considerably overfits the data outperforming only the FP with p = 3. EF-ABFC, on the other hand, gives overall the best results – it has the overall lowest RRMSE and one of the lowest standard deviations. There is only one exception: with "autoPrice" the MT is slightly better.

The decrease of RRMSE and its standard deviation of EF-ABFC over F-ABFC indicates the success of both post-evaluation using validation set as well as ensembling. This is most pronounced with the "machine-cpu" data set.

In relation to computational performance the EF-ABFC is of course about v times slower than F-ABFC, however, here it is still faster than SFS with p = 4 and CIPER+AICC. The former is outperformed because of the very rapidly growing number of basis functions in respect to p [4]. The latter is outperformed because of its large number of model evaluations which in turn is because of its beam search

algorithm (e.g., [11]). Note that the original CIPER is much faster as its model evaluation criterion has a larger complexity penalty than that of AICC and therefore stops the search much earlier. However such criterion can cause also considerable underfitting as we already showed in [4].

V. Conclusion

In this paper, we proposed an ensembling extension EF-ABFC to the F-ABFC regression model building method. The advantage of the EF-ABFC, compared to F-ABFC, is that most of the time it considerably reduces the effects of both selection bias and selection instability. The disadvantage is the reduced computational efficiency. However, the fact that the *v* models before combining are built completely separately allows for an easy parallelization of the process dividing the execution time by *v*. Additionally the results of the empirical comparisons to the other well-known regression modeling methods for both, F-ABFC and EF-ABFC, seem promising.

References

[1] J. O. Rawlings, *Applied Regression Analysis: A Research Tool*, 2nd ed. CA: Wadsworth & Brooks/Cole, 1998.
[2] D. W. Aha and R. L. Bankert, "A comparative evaluation of sequential feature selection algorithms," *Learning from Data*, D. Fisher, H. J. Lenz, Eds., New York: Springer, 1996, pp. 199-206.
[3] L. Todorovski, P. Ljubic, and S. Dzeroski, "Inducing polynomial equations for regression," *Lecture notes in computer science, Lecture notes in artificial intelligence*, 3201, Berlin: Springer, pp. 441-452, 2004.
[4] G. Jekabsons and J. Lavendels, "Polynomial regression modelling using adaptive construction of basis functions", *IADIS International Conference, Applied Computing 2008*, Algarve, Portugal, 2008, to be published
[5] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119-1125, 1994.
[6] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, 2002.
[7] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Mateo, CA, pp. 1137-1145, 1995.
[8] D. Opitz and R. Maclin, „Popular Ensemble Methods: An Empirical Study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169-198, 1999.
[9] M. L. Ginsberg, *Essentials of Artificial Intelligence*. Morgan Kaufmann, 1993.
[10] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, pp. 245-271, 1997.
[11] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edition. Prentice Hall, Englewood Cliffs, New Jersey 07632, 2002.
[12] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, pp. 716-723, 1974.
[13] C. M. Hurvich and C.-L. Tsai, "Regression and time series model selection in small samples," *Biometrika*, vol. 76, pp. 297-307, 1989.
[14] S. D. Stearns, "On selecting features for pattern classifiers," *Proceedings of the 3rd International Joint Conference on Pattern Recognition, IEEE*, pp. 71–75., 1976.
[15] J. F. Elder IV, "The Generalization Paradox of Ensembles," *Journal of Computational and Graphical Statistics*, vol. 12, pp. 853-864, 2003.
[16] J. Reunanen, "Overfitting in making comparisons between variable selection methods," *Journal of Machine Learning Research*, vol. 3, pp. 371-382, 2003.
[17] J. Loughrey and P. Cunningham, "Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets," *24rth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2004)*, pp. 33-43, 2004.
[18] D. M. Allen, "The prediction sum of squares as a criterion for selection of predictor variables," Tech. Rep. 23, Department of Statistics, University of Kentucky, 1971.
[19] R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, pp. 273-324, 1997.
[20] F. E. Harrell Jr., *Regression Modelling Strategies with Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York: Springer, 2001.
[21] L. Breiman. "Heuristics of instability and stabilization in model selection," *Annals of Statistics*, vol. 24, pp. 2350-2383, 1996.
[22] S. Kotsiantis and P. Pintelas, "Combining Bagging and Boosting," *International Journal of Computational Intelligence*, vol. 1, pp. 324-333., 2004.
[23] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed., SF: Morgan Kaufmann, 2005.
[24] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465-471, 1978.

TABLE I
The Results of the Performed Experiments

	housing		housingNOX		autoPrice		machine-cpu	
Method	RRMSE (SD)*	Time**	RRMSE (SD)*	Time**	RRMSE (SD)*	Time**	RRMSE (SD)*	Time**
FP, $p = 1$	53.66 (9.60)	-	49.54 (4.24)	-	48.98 (10.60)	-	54.76 (28.21)	-
FP, $p = 2$	41.14 (11.97)	-	41.00 (8.54)	-	104124 (161951)	-	48.66 (17.64)	-
FP, $p = 3$	-	-	-	-	-	-	278.01 (248.46)	-
SFS, $p = 1$	53.48 (9.50)	< 0.1	49.42 (4.09)	< 0.1	52.00 (11.36)	< 0.1	54.37 (28.11)	< 0.1
SFS, $p = 2$	41.72 (12.82)	23	37.97 (7.26)	13	53.10 (18.64)	1.3	46.25 (17.19)	0.1
SFS, $p = 3$	38.52 (9.27)	779	38.10 (7.81)	397	50.35 (15.76)	31	35.02 (8.65)	1.5
SFS, $p = 4$	53.63 (23.17)	5222	46.35 (18.58)	2350	57.66 (26.28)	217	92.55 (95.43)	4.7
CIPER	45.21 (18.51)	23	37.36 (5.67)	19	46.62 (16.43)	1.5	41.60 (16.09)	1.0
CIPER+AICC	46.34 (30.01)	1803	39.35 (10.86)	845	47.70 (16.21)	83	64.25 (46.91)	10
F-ABFC	36.91 (8.66)	305	39.73 (7.20)	78	45.36 (16.80)	2.5	108.23 (135.62)	0.9
EF-ABFC	33.82 (7.80)	2084	32.03 (4.94)	802	41.62 (10.31)	21	34.61 (15.94)	5.6
RBFNN	67.26 (6.14)	89	44.60 (4.58)	90	60.48 (10.14)	20	48.23 (16.98)	14
MLP	49.25 (17.87)	4.9	45.37 (9.65)	4.9	48.43 (25.30)	1.9	68.44 (41.27)	0.7
MT	42.77 (12.28)	0.8	36.46 (6.70)	0.4	39.68 (9.38)	0.2	42.10 (16.85)	0.2

*average Relative Root Mean Squared Error (percents) and its standard deviation.
**average elapsed time (seconds) of the modeling method.