

# Does Memetic Approach Improve Global Induction of Regression and Model Trees?

Marcin Czajkowski and Marek Kretowski

Faculty of Computer Science, Bialystok University of Technology,  
Wiejska 45a, 15-351 Bialystok, Poland  
{m.czajkowski,m.kretowski}@pb.edu.pl

**Abstract.** Memetic algorithms are popular approaches to improve pure evolutionary methods. But were and when in the system the local search should be applied and does it really speed up evolutionary search is a still an open question. In this paper we investigate the influence of the memetic extensions on globally induced regression and model trees. These evolutionary induced trees in contrast to the typical top-down approaches globally search for the best tree structure, tests at internal nodes and models at the leaves. Specialized genetic operators together with local greedy search extensions allow to the efficient tree evolution. Fitness function is based on the Bayesian information criterion and mitigate the over-fitting problem. The proposed method is experimentally validated on synthetical and real-life datasets and preliminary results show that to some extent memetic approach successfully improve evolutionary induction.

**Keywords:** data mining, evolutionary algorithms, memetic algorithms, regression trees, model trees, global induction.

## 1 Introduction

The most popular algorithms for decision tree induction are based on top-down greedy search [10]. Top-down induction starts from the root node where locally optimal split (test) is searched according to the given optimality measure. Then, the training data is redirected to newly created nodes and this process is repeated recursively for each node until some stopping-rule is reached. Finally, the post-pruning is applied to improve the generalization power of the predictive model.

Nowadays, many research focus on approaches that evolve decision trees as alternative heuristics to the traditional top-down approach [2]. The main advantage of evolutionary induced trees over greedy search methods is the ability to avoid local optima and search more globally for the best tree structure, tests at internal nodes and models at the leaves. On the other hand the induction of global regression and model trees is much slower. One of the possible solutions to speed up evolutionary approach is a combination of evolutionary algorithms with local search techniques, which is known as Memetic Algorithms [6].

In this paper, we focus on regression and model trees that may be considered as a variant of decision trees, designed to approximate real-valued functions.

Main difference between regression tree and model tree is that, for the latter, constant value in the terminal node is replaced by a regression plane. In our previous works we investigated the global approach to obtain accurate and compact regression [8] and model trees with simple linear regression [4] and multivariate linear regression [5] at the leaves. We also investigated the influence of memetic extensions on the global induction of classification trees [7]. In this paper we would like to apply a similar approach for globally induced regression and model trees.

The rest of the paper is organized as follows. In the next section a memetic induction of regression and model trees is described. Experimental validation of the proposed approach on artificial and real-life data is presented in section 3. In the last section, the paper is concluded and possible future works are sketched.

## 2 Memetic Induction of Regression and Model Trees

In this section we present a combination of evolutionary approach with local search techniques in inducing the regression and model trees. The general structure of proposed solution follows a typical framework of evolutionary algorithms [9] with an unstructured population and a generational selection. New memetic extensions are proposed in 2.2 and 2.4.

### 2.1 Representation

Regression and model trees are represented in their actual form as classical univariate trees (tests in internal nodes are based on a single attribute). Depending on the tree type, each leaf of the tree can contain a mean of dependent variable from training objects (regression trees) or a linear model that is calculated at each terminal node of the model tree using standard regression technique (model trees). Additionally, in every node information about learning vectors associated with the node is stored. This enables the algorithm to perform more efficiently the local structure and tests modifications during applications of genetic operators.

### 2.2 Memetic Initialization

Initial individuals are created by applying the classical top-down algorithm [10]. At first, we learn standard regression tree that has a mean of dependent variable values from training objects at each leaf. The recursive partitioning is finished when all training objects in the node are characterized by the same predicted value (or it varies only slightly, default: 1%) or the number of objects at node is lower than the predefined value (default value: 5). Additionally, user can set the maximum tree depth (default value: 10) to limit initial tree size. Next, if necessary, a linear model is calculated at each terminal node of the model tree.

Traditionally, the initial population should be generated randomly to cover the entire range of possible solutions. Due to the large solution space the exhaustive search may be infeasible. Therefore, while creating initial population we

search for a good trade off between a high degree of heterogeneity and relatively low computation time. To create initial population we propose several memetic strategies which involves employing the locally optimized tests and models on randomly chosen internal nodes and leaves. For all non-terminal nodes one of the four test search strategies is randomly chosen:

- *Least Squares (LS)* function reduces node impurity measured by sum of squares,
- *Least Absolute Deviation (LAD)* function reduces the sum of absolute deviations. It has greater resistance to the influence of outlying values to *LS*,
- *Mean Absolute Error (MAE)* function which is more robust and also less sensitive to outliers to *LS*,
- *dipolar*, where a dipol (a pair of feature vectors) is selected and then a test is constructed which splits this dipole. First instance that constitutes dipol is randomly selected from the node. Rest of the feature vectors are sorted decreasingly according to the difference between dependent variable values to the firstly chosen instance. To find a second instance that constitutes dipol we applied mechanism similar to the ranking linear selection [9].

For the leaves, algorithm finds the locally optimal model that minimizes the sum of squared residuals for each attribute or for randomly chosen one.

### 2.3 Genetic Operators

To maintain genetic diversity, we have proposed two specialized genetic operators corresponding to the classical mutation and cross-over. At each evolutionary iteration one of the operators is applied with a given probability (default probability of selecting mutation equals 0.8 and cross-over 0.2) to each individual. Both operators have influence on the tree structure, tests in non-terminal nodes and models at the leaves. Cross-over solution starts with selecting positions in two affected individuals. In each of two trees one node is chosen randomly. We have proposed three variants of recombination [4] that involve tests, subtrees and branches exchange. Mutation solution starts with randomly choosing the type of node (equal probability to select leaf or internal node). Next, the ranked list of nodes of the selected type is created and a mechanism analogous to ranking linear selection is applied to decide which node will be affected. Depending on the type of node, ranking take into account the location of the internal node (internal nodes in lower parts of the tree are mutated with higher probability) and the absolute error (worse in terms of prediction accuracy leaves and internal nodes are mutated with higher probability). We have proposed several variants of mutation for internal node [4] and for the leaf [5] that involve tests, models and modifications in the tree structure (pruning the internal nodes and expanding the leaves).

### 2.4 Memetic Extensions

To improve the performance of evolutionary process, we propose additional local search components that are built into the mutation-like operator. With the user

defined probability a new test can be built on a random split or can be locally optimized similarly to 2.2. Due to the computational complexity constraints, we calculate optimal test for single, randomly chosen attribute. Different variant of the test mutation involve shifting the splitting threshold at continuous-valued feature which can be locally optimized in the similar way. In case of model trees, memetic extension can be used to search for the linear models at the leaves. With the user defined probability a new, locally optimized linear regression model is calculated on a new or unchanged set of attributes.

In previous research, after performed mutation in internal nodes the models in corresponding leaves were not recalculated because adequate linear models could be found while performing the mutations at the leaves. In this paper we test the influence of this recursive model recalculations as it can also be treated as local optimization.

## 2.5 Fitness Function, Selection and Termination Condition

A fitness function is one of the most important and sensitive element in the design of the evolutionary algorithm. It measures how good a single individual is in terms of meeting the problem objective and drives the evolutionary search process. Direct minimization of the prediction error measured on the learning set usually leads to the overfitting problem. In a typical top-down induction of decision trees [10], this problem is partially mitigated by defining a stopping condition and by applying a post-pruning.

In our previous works we used different fitness functions like Akaike's information criterion (*AIC*) [1] and Bayesian information criterion (*BIC*) [11]. In this work we continue to use *BIC* as a fitness function with settings like in [5] but with new assumption. When the sum of squared residuals of the tree equals to zero the original *BIC* fitness is equal infinity therefore no better individual can be found. In our research we continue the search to find the best individual with the lowest complexity.

Ranking linear selection [9] is applied as a selection mechanism. Additionally, in each iteration, single individual with the highest value of fitness function in current population is copied to the next one (*elitist strategy*). Evolution terminates when the fitness of the best individual in the population does not improve during the fixed number of generations. In case of a slow convergence, maximum number of generations is also specified, which allows to limit the computation time.

## 3 Experimental Validation

The proposed memetic approach is evaluated on both artificial and real life datasets. It is compared only to the pure evolutionary versions of our global inducer since in previous work [4] we had a detailed comparison of our solutions with popular counterparts. All results presented in this paper correspond to averages of 10 runs and were obtained by using test sets (when available) or

by 10-fold cross-validation. Root mean squared error (*RMSE*) is given as the prediction error measure of the tested systems. The number of nodes is given as a complexity measure (size) of regression and model trees.

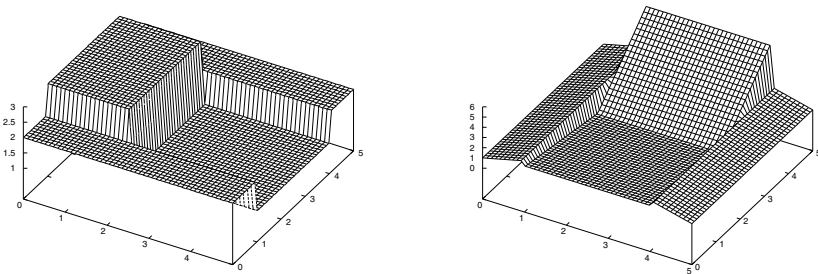
### 3.1 Synthetical Datasets

In the first group of experiments, two simple artificially generated datasets illustrated in figure 1 are analyzed. Both datasets have the same analytically defined decision borders and contain two independent and one dependent feature with 5% noise. Dataset *armchair1* was designed for the regression trees (dependent feature contains only a few distinct values) and *armchair2* for the model trees (dependent variable is modeled as a linear function of single variable). One thousand observations for each dataset were divided into a training set (33.3% of observations) and testing set.

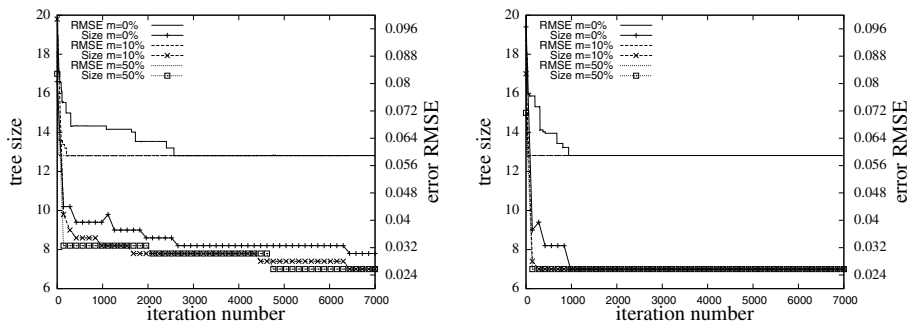
In order to verify the impact of memetic approach on the results, we prepared a series of experiments for global regression trees *GRT* and global model trees *GMT*. Let  $m$  denote the percentage use of local optimizations in the mutation of evolutionary induced trees and equals: 0%, 10% or 50%. The influence of these memetic components on the evolutionary process is illustrated in the figure 2 for *GRT* and in figure 3 for *GMT*. On both figures the *RMSE* and the tree size is shown.

Illustrations on the left side, present the algorithms *GRT* and *GMT* in which after each performed mutation in the internal node corresponding leaves were not recalculated since they could be found during the leaves mutation. In the illustrations on the right, for the algorithms denoted as *GRT<sub>r</sub>* and *GMT<sub>r</sub>*, all the mean values or models in corresponding leaves were recursively recalculated which can also be treated as local optimization 2.4.

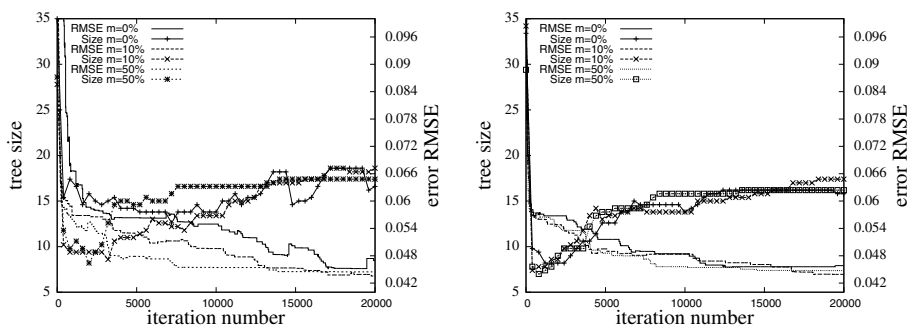
In table 1 we summary the results for the figure 2. All the algorithms managed to find minimum *RMSE* and the optimal tree size which was equal 7. Stronger impact of the memetic approach results in significantly faster algorithm convergence however it also extends the average iteration time. The pure evolutionary algorithm *GRT* managed to find optimal solution but after 28000



**Fig. 1.** Three-dimensional visualization of artificial datasets: *armchair1* - left, *armchair2* - right



**Fig. 2.** The influence of memetic parameter  $m$  on the performance of the algorithm without ( $GRT$  - left), or with ( $GRT_r$  - right) recursive recalculations



**Fig. 3.** The influence of memetic parameter  $m$  on the performance of the algorithm without ( $GMT$  - left), or with ( $GMT_r$  - right) recursive recalculations

iterations where for example  $GRTr$  with memetic impact  $m = 50\%$  need only 100 generations. We can observe that the best performance was achieved for the  $GRTr$  algorithms with local optimization  $m$  equal 10%.

Dataset *armchair2* was more difficult to analyse and none of the  $GMT$  and  $GMTr$  algorithm presented in figure 3 and described in table 2 managed to find the optimal solutions. Similarly to the previous experiment, the algorithms with memetic approach convergence much faster and were able to find good results even after few iterations. The  $GMTr$  with  $m$  equal 50% managed to achieve the highest performance in the terms of  $RMSE$  and the total time.

### 3.2 Real-Life Datasets

In the second series of experiments, two datasets from UCI Machine Learning Repository [3] were analyzed to assess the performance of memetic approach on real-life problems. Table 3 presents characteristics of investigated datasets and obtained results after 5000 performed iterations.

We can observe that for the higher memetic impact, the  $RMSE$  is the smallest but at the cost of the evolution time. Additional research showed that if we run

**Table 1.** Results of the *GRT* and *GRT<sub>r</sub>* algorithms for the *armchair1* dataset

Algorithm	<i>GRT</i> <sub>0</sub>	<i>GRT</i> <sub>10</sub>	<i>GRT</i> <sub>50</sub>	<i>GRT<sub>r</sub></i> <sub>0</sub>	<i>GRT<sub>r</sub></i> <sub>10</sub>	<i>GRT<sub>r</sub></i> <sub>50</sub>
performed iterations	28000	6400	4650	970	190	100
average loop time	0.0016	0.0044	0.011	0.0017	0.0045	0.012
total time	44.8	28.2	51.2	1.65	0.855	1.2
RMSE	0.059	0.059	0.059	0.059	0.059	0.059
size	7	7	7	7	7	7

**Table 2.** Results of the *GMT* and *GMT<sub>r</sub>* algorithms for the *armchair2* dataset

Algorithm	<i>GMT</i> <sub>0</sub>	<i>GMT</i> <sub>10</sub>	<i>GMT</i> <sub>50</sub>	<i>GMT<sub>r</sub></i> <sub>0</sub>	<i>GMT<sub>r</sub></i> <sub>10</sub>	<i>GMT<sub>r</sub></i> <sub>50</sub>
performed iterations	20000	20000	20000	20000	20000	20000
average loop time	0.0040	0.0060	0.011	0.0041	0.0063	0.011
total time	80	120	220	82	126	220
RMSE	0.047	0.044	0.045	0.046	0.044	0.045
size	16	18	17	16	17	16

**Table 3.** Results of the *GMT* and *GMT<sub>r</sub>* algorithms for the real-life datasets

Dataset	Alg.	<i>GRT</i> <sub>0</sub>	<i>GRT<sub>r</sub></i> <sub>0</sub>	<i>GRT<sub>r</sub></i> <sub>10</sub>	<i>GRT<sub>r</sub></i> <sub>50</sub>	<i>GMT</i> <sub>0</sub>	<i>GMT<sub>r</sub></i> <sub>0</sub>	<i>GMT<sub>r</sub></i> <sub>10</sub>	<i>GMT<sub>r</sub></i> <sub>50</sub>
Abalone inst: 4177 attr: 7/1	RMSE	2.37	2.34	2.31	2.30	2.25	2.23	2.23	2.23
	size	39	35	35	39	17	15	13	15
	time	52	56	207	414	149	336	521	1240
Kinemaics inst: 8192 attr: 8	RMSE	0.195	0.191	0.186	0.185	0.185	0.179	0.176	0.174
	size	77	109	129	109	59	61	59	81
	time	96	99	719	1429	285	442	1203	2242

the pure evolutionary algorithm for the same amount of time as *GRT<sub>r</sub>*<sub>50</sub> or *GMT<sub>r</sub>*<sub>50</sub> the results would be similar. Therefore, if we consider the time limit, the global trees with small memetic impact ( $m = 10\%$ ) would achieved the highest performance in the terms of *RMSE* and size.

## 4 Conclusion

In the paper the memetic approach for global induction of decision trees was investigated. We have assessed the impact of local optimizations on evolutionary induced regression and model trees. Preliminary experimental results suggest that at some point memetic algorithms successfully improve evolutionary induction. Application of the memetic approach results in significantly faster algorithm convergence however it also extends the average iteration time. Therefore, too much of local optimizations may not really speed up the evolutionary process. Experimental results also suggest that additional recursive model recalculations after performed mutation for corresponding leaves may be a good idea.

Further research to fully understand the influence of the memetic approach for the decision trees is advised. Currently we plan to analyze each local optimization separately to see how it affects three major elements of the tree: structure, test and models at the leaves.

**Acknowledgments.** This work was supported by the grant S/WI/2/08 from Bialystok University of Technology.

## References

1. Akaike, H.: A New Look at Statistical Model Identification. *IEEE Transactions on Automatic Control* 19, 716–723 (1974)
2. Barros, R.C., Basgalupp, M.P., et al.: A Survey of Evolutionary Algorithms for Decision-Tree Induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (2011) (in print)
3. Blake, C., Keogh, E., Merz, C.: UCI Repository of Machine Learning Databases (1998), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
4. Czajkowski, M., Kretowski, M.: Globally Induced Model Trees: An Evolutionary Approach. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) *PPSN XI*. LNCS, vol. 6238, pp. 324–333. Springer, Heidelberg (2010)
5. Czajkowski, M., Kretowski, M.: An Evolutionary Algorithm for Global Induction of Regression Trees with Multivariate Linear Models. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) *ISMIS 2011*. LNCS, vol. 6804, pp. 230–239. Springer, Heidelberg (2011)
6. Gendreau, M., Potvin, J.Y.: *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, vol. 146 (2010)
7. Kretowski, M.: A Memetic Algorithm for Global Induction of Decision Trees. In: Geffert, V., Karhumäki, J., Bertoni, A., Preneel, B., Návrat, P., Bieliková, M. (eds.) *SOFSEM 2008*. LNCS, vol. 4910, pp. 531–540. Springer, Heidelberg (2008)
8. Kretowski, M., Czajkowski, M.: An Evolutionary Algorithm for Global Induction of Regression Trees. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2010*. LNCS, vol. 6114, pp. 157–164. Springer, Heidelberg (2010)
9. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd edn. Springer, Heidelberg (1996)
10. Rokach, L., Maimon, O.: Top-down induction of decision trees classifiers - A survey. *IEEE Transactions on Systems, Man, and Cybernetics - Part C* 35(4), 476–487 (2005)
11. Schwarz, G.: Estimating the Dimension of a Model. *The Annals of Statistics* 6, 461–464 (1978)