Model Selection and Overfitting in Genetic Programming: Empirical Study

Jan Žegklitz
Czech Technical University in Prague
Technická 2, Prague 6, Czech Republic
zegkljan@fel.cvut.cz

Petr Pošík
Czech Technical University in Prague
Technická 2, Prague 6, Czech Republic
petr.posik@fel.cvut.cz

ABSTRACT

Genetic Programming has been very successful in solving a large area of problems but its use as a machine learning algorithm has been limited so far. One of the reasons is the problem of overfitting which cannot be solved or suppresed as easily as in more traditional approaches. Another problem, closely related to overfitting, is the selection of the final model from the population.

In this article we present our research that addresses both problems: overfitting and model selection. We compare several ways of dealing with ovefitting, based on Random Sampling Technique (RST) and on using a validation set, all with an emphasis on model selection. We subject each approach to a thorough testing on artificial and real–world datasets and compare them with the standard approach, which uses the full training data, as a baseline.

CCS Concepts

•**Computing methodologies** → **Genetic programming;** *Supervised learning;*

Keywords

machine learning; genetic programming; grammatical evolution; overfitting

1. INTRODUCTION

Recent research related to Genetic Programming (GP) as a Machine Learning (ML) algorithm has been focused (among other aspects) on the issue of overfitting, i.e. a condition when a model is fit to the training data too closely that it caputres insignificant deviations or noise rather than the general trend, leading to poor performance on unseen cases.

Bloat is a phenomenon in GP which can be described as an uncontrolled growth of the program size with a very small or no impact on the fitness. Several succesful bloat control techniques were developed (e.g. [9] and [11]). The problem of overfitting was often put into correlation with bloat. This was led by the ideas that bloated models are more likely to fit the noise rather than the short models. However, it was shown [12] that even in a bloat-free envirnoment overfitting can still occur.

One of the overfitting prevention technique is Backwarding (BW) [10], which uses a validation set to select the best individuals. Another validation set based technique is Validation Start (VS) [4]. A technique called Random Subset Selection or Random Sampling Technique was previously used for the speedup of the GP run [3] and for reducing overfitting [8]. This technique was then further explored in [4, 6], yielding RST 1/1, which uses only a single-element subset of training data changing every generation, and RI N%, which uses RST 1/1-like scheme in N% of generations and the whole training set in the other generations. These methods appeared to be successful both in reducing the runtime and overfitting.

All the above mentioned techniques are described in more detail in [13].

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07…$15.00

DOI: http://dx.doi.org/10.1145/2739482.2764678

2. EXPERIMENTAL EVALUATION

Since there are several approaches of overfitting control, we decided to compare all of them on several datasets, including both classification and regression tasks, both artificial and real-world ones. In addition to the already mentioned approaches (STD, BW, VS, RST 1/1, RI 60%) we added several approaches of our own, based on these ones:

VRST 1/1, VRI 60%. Identical to RST 1/1 and RI 60% respectively, but the fitness evaluation datapoints are drawn from a subset of the training data, but the best-so-far model is selected according to the remaining subset of the training data (a validation set).

RST R, VRST R. RST R is identical to RST 1/1, except that not only the element selected for fitness evaluation is selected randomly, the number of such elements is selected randomly too. VRST R is then the extension of RST R with a validation set in the same manner as in VRST 1/1 and VRI 60%.

2.1 Datasets

For evaluation we used six datasets – four artificial and two real-world, which were taken from the UCI repository [1]. Three of the four artificial datasets were binary classification datasets generated by scripts taken from the MATLAB Central File Exchange [7]: the Two Spirals (TS), Cluster in Cluster (CIC) and Halfkernel (HK) datasets. The

fourth dataset is a regression dataset (called Sphere – SPH) defined as $f(\mathbf{x}) = \sum_{i=0}^{30} x_i^2 + noise$. One of the real-world datasets is a regression dataset Forest Fires (FF) [2], the other one is a binary classification dataset Wisconsin Diagnostic Breast Cancer (WDBC).

3. RESULTS

The full version of the results can be seen in [13]. In our experiments, no approach was significantly better than the standard approach. This result is important since it contradicts previous research [4, 6, 5].

Using a validation set (VRST 1/1, VRI 60%, VRST R) did not result in significantly worse performance than the corresponding non-validation variants, however, it is not significantly better either. The reason might be the tradeoff between overfitting control and giving the algorithm enough information to learn.

In all the experiments there was no case of the (V)RST R being significantly worse than (V)RST 1/1 or (V)RI 60%. On CIC and WDBC datasets the (V)RST R was significantly better than both (V)RI 60% and (V)RST 1/1 and on FF dataset VRST R was significantly better than all other RST-based approaches. This might suggest that using random-sized subsets might be more beneficial than using either only a single-element subsets or switching between the full set and single-element subset. The reason for this might be that when using a single-element subset the number of fitness cases (meaning the part of the data the solutions are to predict) the algorithm can encounter is much smaller than in the case of random-sized subsets, causing lower variability of the actual training data.

4. CONCLUSIONS

In this article we revised the issue of overfitting in GP. We discussed the ways the data are handled and based on two patterns (validation set and random sampling) we proposed two new approaches: RST with random-sized subsets and using a validation set in RST-based techniques, including the combination of both. We have carried out a series of experiments with all the presented approaches on six datasets – artificial and real-world, classification and regression.

The good performance of standard approach, contradictory to the previous research, suggests that the performance is data dependent and therefore no general conclusion can be made.

Using a validation set did not bring any significant change in performance, but we tested only one division ratio and different setups could have some impact on the performance of such methods (or not).

Random-sized subsets performed well with respect to the other two RST-based methods. However, this could be data dependent too, and further investigation is also needed. Another aspect of this method is the distribution of the subset size – we used a uniform distribution but other distributions, e.g. favoring smaller subsets, could prove more beneficial.

5. ACKNOWLEDGMENTS

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS14/194/OHK3/3T/13.

Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme „Projects of Large Infrastructure for Research, Development, and Innovations“ (LM2010005), is greatly appreciated.

6. REFERENCES

[1] K. Bache and M. Lichman. UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml.

[2] P. Cortez and A. d. J. R. Morais. A data mining approach to predict forest fires using meteorological data. 2007.

[3] C. Gathercole and P. Ross. Dynamic training subset selection for supervised learning in genetic programming. In *Parallel Problem Solving from Nature — PPSN III*, volume 866 of *Lecture Notes in Computer Science*, pages 312–321. Springer Berlin Heidelberg, 1994.

[4] I. Gonçalves and S. Silva. Experiments on controlling overfitting in genetic programming. In *15th Portuguese Conference on Artificial Intelligence*, October 2011.

[5] I. Gonçalves and S. Silva. Balancing learning and overfitting in genetic programming with interleaved sampling of training data. In *Genetic Programming*, volume 7831 of *Lecture Notes in Computer Science*, pages 73–84. Springer Berlin Heidelberg, 2013.

[6] I. Gonçalves, S. Silva, J. Melo, and J. Carreiras. Random sampling technique for overfitting control in genetic programming. In *Genetic Programming*, volume 7244 of *Lecture Notes in Computer Science*, pages 218–229. Springer Berlin Heidelberg, 2012.

[7] J. Kools. 6 functions for generating artificial datasets. MATLAB Central File Exchange, Retrieved December 17, 2014. http://www.mathworks.com/matlabcentral/fileexchange/41459.

[8] Y. Liu and T. Khoshgoftaar. Reducing overfitting in genetic programming models for software quality classification. In *High Assurance Systems Engineering, 2004. Proceedings. Eighth IEEE International Symposium on*, pages 56–65, March 2004.

[9] R. Poli and N. McPhee. Parsimony pressure made easy: Solving the problem of bloat in gp. In *Theory and Principled Methods for the Design of Metaheuristics*, Natural Computing Series, pages 181–204. Springer Berlin Heidelberg, 2014.

[10] D. Robilliard and C. Fonlupt. Backwarding: An overfitting control for genetic programming in a remote sensing application. In *Artificial Evolution*, volume 2310 of *Lecture Notes in Computer Science*, pages 245–254. Springer Berlin Heidelberg, 2002.

[11] S. Silva and E. Costa. Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. *Genetic Programming and Evolvable Machines*, 10(2):141–179, 2009.

[12] L. Vanneschi and S. Silva. Using operator equalisation for prediction of drug toxicity with genetic programming. In *Progress in Artificial Intelligence*, volume 5816 of *Lecture Notes in Computer Science*, pages 65–76. Springer Berlin Heidelberg, 2009.

[13] J. Žegklitz and P. Pošík. Model selection and overfitting in genetic programming: Empirical study [extended version]. http://arxiv.org/abs/1504.08168, 2015.