Application of Symbolic Regression on Blast Furnace and Temper Mill Datasets

Michael Kommenda[1], Gabriel Kronberger[1], Christoph Feilmayr[2], Leonhard Schickmair[2], Michael Affenzeller[1], Stephan Winkler[1], and Stefan Wagner[1]

[1] Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media
Upper Austria University of Applied Sciences, Campus Hagenberg
Softwarepark 11, 4232 Hagenberg, Austria
{michael.kommenda,gabriel.kronberger,michael.affenzeller,
stephan.winkler,stefan.wagner}@fh-hagenberg.at

[2] voestalpine Stahl GmbH, voestalpine-Strae 3, 4020 Linz, Austria,
{christoph.feilmayr,leonhard.schickmair}voestalpine.com

Abstract. This work concentrates on three different modifications of a genetic programming system for symbolic regression analysis. The coefficient of correlation R^2 is used as fitness function instead of the mean squared error and offspring selection is used to ensure a steady improvement of the achieved solutions. Additionally, as the fitness evaluation consumes most of the execution time, the generated solutions are only evaluated on parts of the training data to speed up the whole algorithm. These three algorithmic adaptations are incorporated in the symbolic regression algorithm and their impact is tested on two real world datasets describing a blast furnace and a temper mill process. The effect on the achieved solution quality as well as on the produced models are compared to results generated by a symbolic regression algorithm without the mentioned modifications and the benefits are highlighted.

Keywords: Symbolic Regression, Genetic Programming, Offspring Selection

1 Introduction

This paper describes the application of an adapted symbolic regression system on two different steel production datasets. The first dataset contains measurements from a blast furnace process which is the most common method for the production of hot metal (liquid iron). Although the chemical and physical reactions are well understood, subtle relations between single process parameters (e.g., heat loss in certain areas of the furnace) are not completely understood. The knowledge about such relationships can be used to optimize the blast furnace process (for example the quality of the products or the stability of the process) and therefore modeling the blast furnace process based on collected real world data is of special interest.

The second investigated process is temper rolling, a finishing step in the production of steel sheets. Temper rolling flattens, slightly lengthens, and roughens the surface of the steel sheet and determines the mechanical properties of the end product. During this process two major influence factors, the strip tension and the rolling force, must be adapted and initially preset to achieve the desired product quality [11]. A good prediction model for these two parameters is of major importance because it reduces the effort of tuning these parameters during production. Hence the amount of scrap material that does not fulfill customer requirements can be reduced.

A number of different methods have been used to model these two processes, such as neural networks [10, 3], support vector regression, or mathematical models [2]. In [8] as well as in [6] first modeling results with symbolic regression have been presented concerning identified models of the blast furnace and the temper mill. Unlike [8, 6] this paper focuses on three adaptations to a symbolic regression system and their effect when used in combination. Section 2 describes the used symbolic regression system and its adaptations, whereas Section 3 states the concrete parameter settings, implementation details and the content of the two datasets. The results are presented and interpreted in Section 4 and Section 5 concludes the paper and outlines future work and open issues.

1.1 Symbolic Regression

Regression analysis is a sub field of data mining attempting to reveal knowledge contained in a given dataset. More precisely, a model to describe a dependent variable is built using a set of independent variables (input variables) and weights. The identified model is learnt on a part of the dataset called training partition and its generalization capabilities are estimated on the test partition that must not be used for learning the model. The task of symbolic regression [7] is also to model the dependent variable, but contrary to other regression methods the model structure is not predefined.

In this paper symbolic regression is performed using a tree-based genetic programming system to evolve mathematical formulas. Genetic programming (GP) [7] is an evolutionary algorithm that produces programs to solve a given problem. GP follows the 'survival of the fittest' paradigm and is based on genetic algorithms, which work with a set of candidate solutions called population. The population is first initialized with random individuals (solutions). At every generation parts of the population are replaced by new individuals, created by combining the information of two parent individuals and optionally mutating the newly created child. Often the best n individuals (elitists) are directly passed to the next generation without manipulating them, to ensure the the best individual during the evolutionary process is not lost and hence the quality of the best individual per generation steadily increases.

2 Methods

In our modeling approach three algorithmic aspects have been incorporated and compared to a standard symbolic regression approach. Precisely we tested the effects of offspring selection, using the coefficient of correlation R^2 as fitness function and additionally sample the evaluated samples or fitness cases for each individual.

2.1 Offspring selection

Offspring selection (OS) [1] is an additional selection step in genetic algorithms and genetic programming that is applied after recombination (crossover) and mutation. OS only adds newly generated children to the next generation if this individual surpasses a given criterion. Mostly a fitness related criterion is used for OS, e.g. comparing the quality of the child to the quality of the parents. The question remains which quality should be used as comparison value. This is managed by the comparison factor c that states how the parent qualities are combined to act as comparison value for the newly generated child , e.g. $c = 0$ means the the child must outperform the less fit parent, $c = 0.5$ the child must outperform the average of both parents' fitness and $c = 1$ the child must outperform the parent with the better fitness. Additionally the success ratio sr determines the relative amount of the new population that must pass the offspring selection. If offspring selection is applied with the parameters $c = 1$ and $sr = 1$, it is commonly referred to as strict OS. Strict OS has the property that children with worse quality compared to its better parent are automatically discarded and therefore the overall quality of the population steadily increases.

2.2 Fitness function

The fitness of an individual in symbolic regression analysis is commonly calculated as the mean squared error (MSE) between the predicted values of the model and the observed values of the target variable. A drawback of the MSE as fitness function is that models which fit the characteristics of the target variable quite well but are different in location or scale have a larger MSE than models that do not fit the characteristics of the target variable but are located in the same range. Thus the GP process first prefers models that are located in the same range as the target variable and learns the characteristics of the target variable afterwards. To overcome this limitation the coefficient of determination R^2 (Equation 1) is used as fitness function for the GP process [4].

$$R^2(x, y) = \frac{\mathrm{Cov}(x, y)^2}{\mathrm{Var}(x) * \mathrm{Var}(y)} \tag{1}$$

A comparison of the predicted values with the target values is not directly possible because the predicted values could have a different scale or range than the target values. Therefore the model outcome must be linearly transformed

to allow an interpretation of the predicted values. The samples of the test partition must not be used for the transformation to allow an estimation of the generalization error without falsifying the results.

2.3 Sampling

In general most of the execution time of an evolutionary algorithm is consumed during the evaluation step of the generated individuals. Therefore reducing the amount of time spent for the evaluation can significantly reduce the execution time of the whole algorithm. This sampling technique has been used in [12] under the name goal softening.

In [5] two simpler sampling techniques have been proposed. More precisely a generational sampling technique, where all individuals in one generation are evaluated on the same subset of samples, was developed. The other possibility is to randomly select the subset of samples before each evaluation, which has also been used in this contribution. As the runtime of the algorithm does not rely solely on the evaluation step the speedup does not scale linearly with the reduction of the training samples. Nevertheless significant speedups can be achieved if sampling techniques are used.

3 Experiments

The algorithm adaptations mentioned in Section 2 have been tested on two real steel production datasets. A tree-based genetic programming approach with the parameter setting listed in Table 1 was used. The only difference among the configurations is that, with offspring selection enabled, gender specific selection [14] instead of tournament selection was performed to achieve a similar selection pressure in the algorithm run.

All experiments described in this section were performed with HeuristicLab [13]. HeuristicLab is an open source framework for modeling, executing and comparing different heuristic optimization techniques. All the described algorithmic adaptations are available in HeuristicLab 3.3.4, which can be obtained from `http://dev.heuristiclab.com`.

3.1 Datasets

The basis of the analysis of the algorithmic adaptations are two datasets originating from a blast furnace and a temper mill. The blast furnace dataset contains hourly process measurements collected between 2007 and 2010. The measurements describe the hot blast, the tuyere injection, the charging and tapping of the blast furnace, the top gas, as well as different general process parameters (e.g. stand stills, the melting rate or cooling losses). The hourly data was filtered to exclude rows with missing or incorrect values, for example when the blast furnace is in a faulty or maintenance state. Finally it consists of 126 columns and 16,000 rows of which the first 10,000 rows were used for training and 6,000 rows for

Parameter	Value
Population size	1000
Max. evaluated solutions	500,000
Sampling	10 %
	100 %
Parent selection	Tournament (group size = 7)
	Gender specific selection [14]
Offspring selection	No offspring selection
	Strict offspring selection
Replacement	1-Elitism
Initialization	PTC2 [9]
Crossover	Sub-tree-swapping
Mutation rate	15%
Mutation operators	One-point and Sub-tree replacement
Tree constraints	Max. expression size = 100
	Max. expression depth = 10
Stopping criterion	Max. evaluated solutions reached
Fitness function	MSE (minimization))
	R^2 (maximization
Function set	+,-,*,/,log,exp
Terminal set	constants, variable

Table 1. Symbolic regression parameters.

testing the produced models. From these 126 measured parameters, 23 were allowed to model the melting rate (Problem 1) and 63 were allowed for the carbon content of the hot metal (Problem 2).

The second dataset contains measured process parameters from a temper mill between 2002 and 2008. The dataset was joined with mechanical and chemical analysis of the rolled steel sheet. Afterwards, rows containing measurements when no temper rolling was performed or with missing values were removed, resulting in 32 columns and approximately 78,000 rows. The collected data contains exactly one row per produced steel sheet. Hence, the mean values of the rolling force (Problem 3) and the strip tension (Problem 4) during the whole temper rolling process of one specific steel sheet were predicted.

4 Results

The first analysis shows the effect of sampling on the execution time. The median execution time per configuration for Problem 1 is shown in Table 2. The standard deviation is about five minutes if sampling is used (10%) and 33 minutes without sampling (100%). The differences between the fitness functions is explained by the costlier calculation of the R^2. The genetic algorithm with offspring selection (OSGA) is faster than the genetic algorithm without offspring selection (GA) due to the fact that smaller models are produced. The most interesting result is that reducing the number of evaluated samples gives an algorithm

speedup of approximately five regardless of the concrete configuration used on the investigated dataset, without a significant worsening of the obtained quality.

	Samples	GA MSE	GA R^2	OSGA MSE	OSGA R^2
Execution time	10 %	00:55:12	00:56:40	00:38:37	00:44:58
Execution time	100 %	04:34:08	04:54:52	03:03:09	03:34:47
Median rel. error	10 %	5.90 %	2.73 %	5.46 %	3.08 %
STDEV of rel. error	10 %	3.45 %	0.50 %	1.33 %	0.50 %
Median rel. error	100 %	5.17 %	2.71 %	6.01 %	2.40 %
STDEV of rel. error	100 %	1.98 %	0.25 %	3.44 %	0.3 %

Table 2. Median execution times (hh:mm:ss) and median relative test error with standard deviation of different algorithm configurations for Problem 1.

The next analysis shows the qualities that were achieved with different algorithm configurations on all four problems. Table 3 lists the performance of the best training models on the training and test partition over 25 independent repetitions. The values are stated as the median and the standard deviation of the average relative error per sample, to make the results comparable across the different problems. It can been seen that the use of the coefficient of correlation R^2 as fitness function outperforms the mean squared error on all problems.

Problem	Algorithm	Fitness function	Training error	Test error
1	GA	MSE	3.75 % (1.45 %)	5.90 % (3.45 %)
1	OSGA	MSE	4.32 % (0.90 %)	5.46 % (1.33 %)
1	GA	R^2	1.58 % (0.07 %)	2.73 % (0.50 %)
1	OSGA	R^2	1.67 % (0.06 %)	3.08 % (0.50 %)
2	GA	MSE	1.81 % (0.44 %)	2.84 % (0.51 %)
2	OSGA	MSE	1.88 % (0.28 %)	2.78 % (0.51 %)
2	GA	R^2	1.38 % (0.03 %)	2.03 % (0.21 %)
2	OSGA	R^2	1.43 % (0.03 %)	2.00 % (0.59 %)
3	GA	MSE	17.57 % (2.60 %)	21.31 % (4.48 %)
3	OSGA	MSE	17.60 % (1.67 %)	21.46 % (2.86 %)
3	GA	R^2	15.22 % (1.88 %)	17.66 % (3.12 %)
3	OSGA	R^2	15.93 % (1.70 %)	18.35 % (2.87 %)
4	GA	MSE	28.50 % (7.73 %)	28.27 % (7.11 %)
4	OSGA	MSE	30.67 % (3.30 %)	32.47 % (3.27 %)
4	GA	R^2	25.01 % (5.14 %)	25.68 % (4.68 %)
4	OSGA	R^2	24.31 % (6.01 %)	24.22 % (5.89 %)

Table 3. Median and standard deviation of the relative error over 25 independent repetitions.

Furthermore, there is no indication that the use of offspring selection has any benefit or drawback regarding the achieved solution quality, although offspring

selection has a not to be underestimated influence of the algorithm dynamics. Therefore the tree sizes of the resulting model and the calculated number of generations per algorithm run have been examined. These results are shown in Table 4. It can been seen that the genetic algorithm without offspring selection (GA) builds models that max out the tree size constraint of 100. In contrast, models produced with offspring selection (OSGA) do not show this behavior. A possible explanation for this behavior, is that OSGA calculates less generations which reduces the chance of bloat, an increase of tree size without according fitness improvement, during the algorithm execution.

	Problem	GA MSE	GA R^2	OSGA MSE	OSGA R^2
Tree size	1	96.0 (19.0)	94 (9.2)	42.5 (16.3)	66.5 (17.4)
Generations	1	500.0	500.0	22.5 (1.9)	20.0 (0.9)
Tree size	2	87.5 (18.4)	85.5 (13.0)	45.0 (16.2)	47.5 (16.1)
Generations	2	500.0	500.0	21.0 (2.1)	15.0 (0.9)
Tree size	3	98.0 (19.6)	97.5 (12.7)	57.0 (13.8)	59.0 (19.5)
Generations	3	500.0	500.0	19.0 (1.2)	14.0 (1.0)
Tree size	4	94.0 (40.6)	95.0 (12.3)	43.0 (23.0)	74.0 (15.0)
Generations	4	500.0	500.0	18.0 (2.1)	17.0 (1.5)

Table 4. Median and standard deviation of the tree sizes and the calculated generations per configuration.

5 Conclusion

In this contribution three algorithmic adaptations, offspring selection, coefficient of determination R^2 as fitness function and sampling, to a rather standard symbolic regression system have been investigated. The effects of combining these adaptations have been demonstrated on real world datasets from two steel production processes. First of all the use of sampling reduces the execution time of the algorithm runs to 1/5 without affecting the resulting model quality (see Table 2) on the investigated problems.

The best improvements in terms of quality were achieved due to use of the R^2 as fitness function, which is coherent with the findings in [4]. Although the use of offspring selection in symbolic regression did not result in more accurate models, the models were generally smaller. The assumption is that models produced by offspring selection are less affected by bloat due to the fewer number of generations calculated, but this justification must be supported by further research. Additionally the use of sampling and offspring selection could add an bias to search for easily predicted samples instead of finding accurate models. This was not indicated by the obtained results, but could occur on different datasets.

Acknowledgments The work described in this paper was done within the Josef Ressel Centre for Heuristic Optimization *Heureka!* sponsored by the Austrian Research Promotion Agency (FFG).

References

1. Affenzeller, M., Winkler, S., Wagner, S., Beham, A.: Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications. Numerical Insights, CRC Press (2009)
2. Andahazy, D., Slaby, S., Löffler, G., Winter, F., Feilmayr, C., Bürgler, T.: Governing processes of gas and oil injection into the blast furnace. ISIJ International 46(4), 496–502 (2006)
3. Cho, S., Cho, Y., Yoon, S.: Reliable roll force prediction in cold mill using multiple neural networks 8(4), 874–882 (July 1997)
4. Keijzer, M.: Improving symbolic regression with interval arithmetic and linear scaling. In: EuroGP'03: Proceedings of the 6th European conference on Genetic programming. pp. 70–82. Springer-Verlag, Berlin, Heidelberg (2003)
5. Kommenda, M., Kronberger, G., Affenzeller, M., Winkler, S., Feilmayr, C., Wagner, S.: Symbolic regression with sampling. In: 22nd European Modeling and Simulation Symposium EMSS 2010. pp. 13–18. Fes, Morocco (October 2010)
6. Kommenda, M., Kronberger, G., Winkler, S., Affenzeller, M., Wagner, S., Schickmair, L., Lindner, B.: Application of genetic programming on temper mill datasets. In: Proceedings of the IEEE 2nd International Symposium on Logistics and Industrial Informatics (Lindi 2009). pp. 58–62. Linz, Austria (September 2009)
7. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press (1992)
8. Kronberger, G., Feilmayr, C., Kommenda, M., Winkler, S., Affenzeller, M., Thomas, B.: System identification of blast furnace processes with genetic programming. In: Proceedings of the IEEE 2nd International Symposium on Logistics and Industrial Informatics (Lindi 2009). pp. 63–68. Linz, Austria (September 2009)
9. Luke, S.: Two fast tree-creation algorithms for genetic programming. IEEE Transactions on Evolutionary Computation 4(3), 274–283 (Sep 2000)
10. Radhakrishnan, V.R., Mohamed, A.R.: Neural networks for the identification and control of blast furnace hot metal quality. Journal of Process Control 10(6), 509 – 524 (2000)
11. Stelzer, R., Ptz, P.D., Diegelmann, V., Gorgels, F., Piesack, D.: Optimum temper rolling degree: Pre-set and influencing effects of bending deformations. Steel research international 76(2-3), 105–110 (2005)
12. Vladislavleva, E.: Model-based problem solving through symbolic regression via pareto genetic programming. Open Access publications from Tilburg University urn:nbn:nl:ui:12-3125460, Tilburg University (2008), `http://ideas.repec.org/p/ner/tilbur/urnnbnnlui12-3125460.html`
13. Wagner, S.: Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment. Ph.D. thesis, Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria (2009)
14. Wagner, S., Affenzeller, M.: SexualGA: Gender-specific selection for genetic algorithms. In: Callaos, N., Lesso, W., Hansen, E. (eds.) Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI) 2005. vol. 4, pp. 76–81. International Institute of Informatics and Systemics (2005)