Knowledge Discovery through Symbolic Regression with HeuristicLab

Gabriel Kronberger, Stefan Wagner, Michael Kommenda, Andreas Beham, Andreas Scheibenpflug, and Michael Affenzeller

University of Applied Sciences Upper Austria, Campus Hagenberg
School for Informatics, Communication and Media
{gkronber,swagner,mkommend,abeham,ascheibe,maffenze}@heuristiclab.com
http://heal.heuristiclab.com/

Abstract. This contribution describes how symbolic regression can be used for knowledge discovery with the open-source software HeuristicLab. HeuristicLab includes a large set of algorithms and problems for combinatorial optimization and for regression and classification, including symbolic regression with genetic programming. It provides a rich GUI to analyze and compare algorithms and identified models. This contribution mainly focuses on specific aspects of symbolic regression that are unique to HeuristicLab, in particular, the identification of relevant variables and model simplification.

1 Introduction

HeuristicLab[1] is an open-source software system for heuristic optimization that features several metaheuristic optimization algorithms as well as several optimization problems. It is implemented in C# and based on the Microsoft .NET Framework and provides a rich graphical user interface for solving optimization problems and for experiment analysis. HeuristicLab is used in a number of fundamental and practical research projects [1], as well as in lectures on data mining and machine learning. In 2009 the project achieved the second place of the Microsoft Innovation Award in Austria and is released under the GNU General Public License. Some of the general features of HeuristicLab 3.3 are:

- comfortable and feature rich graphical user interface
- experiment designer to create and execute a large number of test runs
- graphical analysis and comparison of parameters and results
- graphical algorithm designer to create or modify algorithms
- plug-in based architecture which enables an easy integration of new algorithms and problems

Based on these general features HeuristicLab provides several well-known algorithms for classification and regression (e.g. linear regression, random forest,

[1] http://dev.heuristiclab.com
http://dev.heuristiclab.com/AdditionalMaterial/ECML-PKDD

P. Flach et al. (Eds.): ECML PKDD 2012, Part II, LNCS 7524, pp. 824–827, 2012.
© Springer-Verlag Berlin Heidelberg 2012

SVM, ...), which can be easily configured and applied to a given data set directly in the GUI. The experiment designer simplifies algorithm configuration and makes it very easy to find the best settings for a given problem.

Additionally to standard methods for regression HeuristicLab also provides an extensive implementation of symbolic regression based on genetic programming [2],[3]. In this contribution we show the core principle of symbolic regression and discuss how this method can be used for practical data mining applications.

2 User Groups and Related Work

HeuristicLab users can be categorized into three relevant groups: practitioners, experts, and students. Practitioners are trying to solve real-world problems with classical and advanced algorithms. Experts include researchers and graduate students who are developing new advanced algorithms. Students can learn about standard algorithms and can try different algorithms and parameter settings to various benchmark algorithms. The design and architecture of HeuristicLab is specifically tuned to these three user groups and we put a strong emphasis on the ease of use of the software.

Several software systems for symbolic regression or more generally for metaheuristic optimization have been developed in the recent years. *Formulize*[2] (or Eureqa-II) is notable as it also provides a user friendly GUI, uses powerful state-of-the-art algorithms and also provides a simple way to execute runs in the Cloud. However, it is much more specialized than HeuristicLab and is mainly designed for practitioners. *ECJ*[3] is an evolutionary computation system written in Java. It is well established as an implementation and experimentation platform in the EC community and also includes a framework for symbolic regression. A drawback ECJ is its limited GUI which makes it difficult to for example analyze the prediction accuracy of the discovered models.

3 Case Study: Tower Data

In this contribution we demonstrate the unique features of HeuristicLab and how it can be used for knowledge discovery in a real world application, in particular, for finding relevant driving factors in a chemical process and for the identification of white-box regression models for the process. The analysis is based on the *tower data set* which is kindly provided by Dr. Arthur Kordon from Dow Chemical [4]. In the demonstration we show the exact process to achieve the results, that can only be briefly described in the following two sections because of page constraints.

3.1 Identification of Non-linear Models

The following equation shows a non-linear model for the tower data set as identified by symbolic regression in HeuristicLab. The algorithm discovered the model

[2] `http://www.nutonian.com/eureqa-ii/`
[3] `http://cs.gmu.edu/~eclab/projects/ecj/`

structure and parameters automatically through an evolutionary process by assembling the basic building blocks: random constants, input variables, arithmetic operators, and the logarithm and exponential functions.

$$\frac{c_0 \cdot \mathrm{x23} \cdot \left(\left((c_1 \cdot \mathrm{x5} - c_2 \cdot \mathrm{x6}) - c_3 \cdot \mathrm{x14} \right) + \left(\frac{(c_4 \cdot \mathrm{x6} - c_5 \cdot \mathrm{x14})}{(c_6 \cdot \mathrm{x1} - c_7 \cdot \mathrm{x4})} - (c_8 - c_9 \cdot \mathrm{x4}) \right) \right)}{c_{16} \cdot \mathrm{x1}} \cdot c_{17}$$
$$+ \frac{(c_{10} \cdot \mathrm{x5} - c_{11} \cdot \mathrm{x6}) - (c_{12} \cdot \mathrm{x4} - (\log(c_{13} \cdot \mathrm{x4}) - (c_{14} \cdot \mathrm{x5} - c_{15} \cdot \mathrm{x24})))}{c_{16} \cdot \mathrm{x1}} \cdot c_{17} + c_{18}$$

3.2 Identification of Relevant Variables

Frequently it is not necessary to learn a full model of the functional relationship but instead only find a set of relevant variables for the process. This can be achieved easily with HeuristicLab through analysis of relative variable frequencies in the population of models. Figure 1 shows a variable frequency chart that clearly shows the six most relevant variables. Notably, the relevance of variables is determined based on non-linear models. So, non-linear influence factors and pair-wise interacting factors can be identified as well.

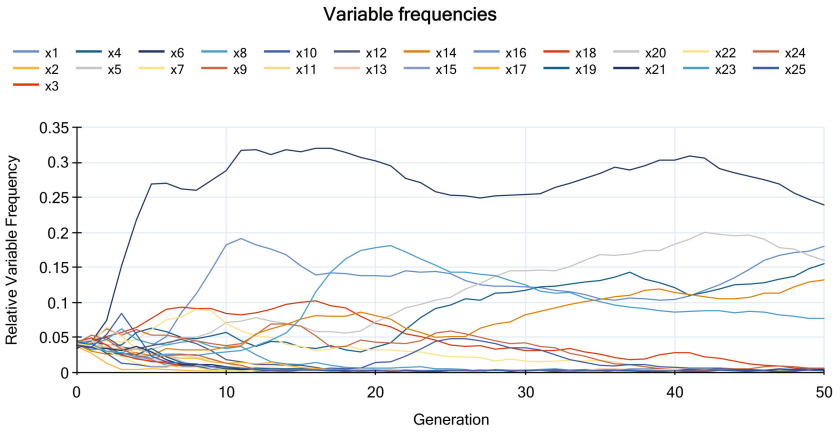

Fig. 1. Evolution of referenced input variables over a symbolic regression run

3.3 Simplification of Models with Visual Hints

A unique feature of HeuristicLab are visual hints for model simplification. By means of visual hints it is very easy to manually prune a complex model as shown above to find a good balance between complexity and accuracy. Figure 2 shows the GUI for model simplification. Green model fragments have a strong impact on the model output while white fragments can be pruned with minimal

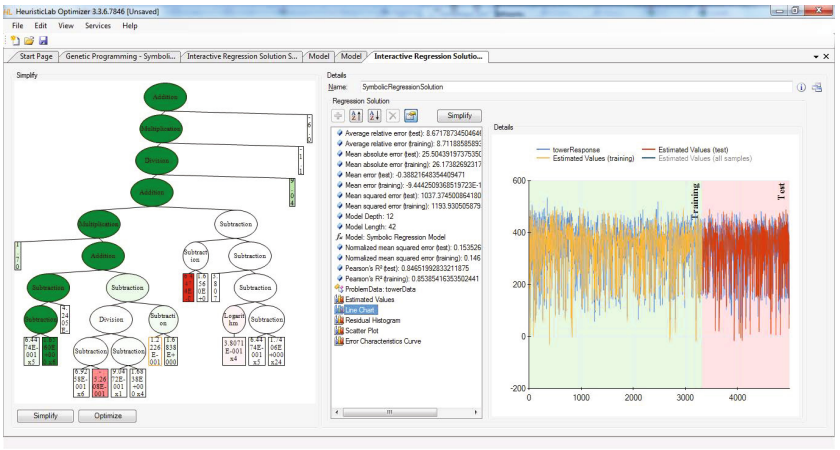

Fig. 2. Visual hints guide the user in manual simplification of non-linear models. Charts and accuracy metrics are updated in real-time to support the user.

losses in accuracy, in contrast red fragments increase the error of the model and should be removed. The GUI for model simplification immediately recalculates all error metrics and updates charts dynamically whenever a part of the model is changed by the user.

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. Kronberger, G.: Symbolic Regression for Knowledge Discovery - Bloat, Overfitting, and Variable Interaction Networks. Trauner Verlag, Linz (2011)
3. Veeramachaneni, K., Vladislavleva, E., O'Reilly, U.-M.: Knowledge mining sensory evaluation data: genetic programming, statistical techniques, and swarm optimization. Genetic Programming and Evolvable Machines 13(1), 103–133 (2012)
4. Vladislavleva, E.J., Smits, G.F., den Hertog, D.: Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. IEEE Transactions on Evolutionary Computation 13(2), 333–349 (2009)