HeuristicModeler: A Multi-Purpose Evolutionary Machine Learning Algorithm and its Applications in Medical Data Analysis

Stephan Winkler, Michael Affenzeller, and Stefan Wagner
Department of Software Engineering
Upper Austrian University of Applied Sciences
College of Information Technology at Hagenberg
Hauptstraße 117, 4232 Hagenberg, Austria
A-4232 Hagenberg, Austria
E-Mail: {stephan,michael,stefan}@heuristiclab.com

KEYWORDS

Medical Data Mining, Machine Learning, Regression, Classification, Time Series Analysis, Genetic Programming, Self-Adaption

ABSTRACT

The application of machine learning techniques for discovering patterns in data is becoming more and more important not only in computer science in general, but also especially in medical data mining. Regression modeling problems are to be solved in this context as well as classification problems, and also time series analysis methods are expected to become more and more important. In this paper we describe a multi-purpose machine learning approach based on various evolutionary computation concepts that is applicable for several medical data mining aspects in evidence based medicine. We show how regression, classification and time series problems can be attacked using this algorithm, and we also propose a hybrid approach combining time series analysis with regression and classification aspects.

1. INTRODUCTION

Data mining is understood as the practice of automatically searching large stores of data for patterns. Incredibly large (and quickly growing) amounts of data are collected not only in commercial, administrative, and scientific, but also in medical databases; this is the reason why intelligent computer systems that can extract useful information (such as general rules or interesting patterns) from large amounts of observations are needed. In short, "data mining is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad et al. 1996).
In medicine, in particular, more and more data collections storing measurements of patients' physical examinations are available including lots of information not discovered yet. This is why data based machine learning algorithms have to be applied in order to retrieve additional insights into human biological processes, how environment factors influence human health or how certain human parameters are related.
The following three classes of data analysis problems are relevant within medical data analysis: Regression, classification and time series analysis. In any of these cases, statistical algorithms are supposed to "learn" functions by analyzing a set of input-output examples ("training samples").

In statistics, *regression* analysis is understood as the act of modeling the relationship between variables, namely between one or more target ("dependent") variables and other variables (also called input or explanatory variables). I.e., the goal is to find a mathematical function f which can be used for calculating the target variable Y using the input variables $X_{1..p}$:

$$Y = f(X_1,...,X_p) \quad (1)$$

Various regression algorithms are frequently used, for example linear regression, correlation analysis, the Gauss-Newton algorithm, the general Levenberg-Marquardt algorithm (Gill et al. 1981) or, in more complex cases, artificial neural networks (ANNs) as for example explained in (Nelles 2001).

Classification is understood as the act of placing an object into a set of categories, based on the object's properties. Objects are classified according to an (in most cases hierarchical) classification scheme also called taxonomy. A statistical classification algorithm is supposed to take feature representations of objects and map them to a special, predefined classification label. Such a classification algorithm is designed to learn a function f which maps a vector of object features $X_1,...,X_p$ into one of several classes. A given sample x_i can so be classified using f and $X_1,...,X_p$:

$$Class(y_i) = f(X_{1(i)},...,X_{p(i)}) \quad (2)$$

There are several approaches which are nowadays used for solving classification problems; the most common ones are (as described in (Mitchell 2000), e.g.) decision tree learning, instance-based learning, inductive logic

programming (such as in Prolog, e.g.) and reinforcement learning.

There are two main goals of *time series analysis*: On the one hand one tries to identify the cause of a phenomenon represented by a sequence of observations and its relationships with other sequences of observations, and on the other hand the goal is to predicting future values of time series variables. Both of these goals require that the pattern of observed time series data is identified and more or less formally described. I.e., for the target variable *Y* one wants to identify a function *f* so that *Y* at time *t* can be calculated using values of other variables and (if available) also information about the history of *Y*:

$$Y_{(t)} = f(X_{1(t-\{0..z\})},...,X_{p_{(t-\{0..z\})}},Y_{(t-\{1..z\})}) \qquad (3)$$

where *z* is the maximum time offset for variables used in *f*. Detailed discussions of time series and methods applicable can for example be found in (Box and Jenkins 1976) or (Kendall and Ord 1990).

All these data mining problem classes are strongly relevant in the context of medical applications; medical data mining is thereby one of the necessary foundations of evidence based medicine. More and more data of patients are stored and can be used for scientific surveys making it possible to derive knowledge not discovered yet. This is why data based machine learning algorithms have to be applied in order to retrieve additional insights into human biological processes, how environment factors influence human health or how certain human parameters are related.

2. EVOLUTIONARY MACHINE LEARNING

2.1 Evolutionary Computation

Evolutionary computing is the collective name for heuristic problem-solving techniques based on the principles of biological evolution, which are natural selection and genetic inheritance. One of the greatest advantages of these techniques is that they can be applied to a variety of problems, ranging from leading-edge scientific research to practical applications in industry and commerce; the forms of evolutionary computation relevant for the work described in this paper are *Genetic Algorithms* (GAs) and *Genetic Programming*.

The fundamental principles of the *Genetic Algorithm* were first presented by Holland (Holland, 1975), overviews about GAs and their implementation in various fields were given for instance in (Goldberg, 1989), (Michalewicz 1996) and (Affenzeller 2003). A GA works with a set of candidate solutions (also known as individuals) called population. During the execution of the algorithm each individual has to be evaluated, which means that a value indicating the "fitness" or "goodness" is returned by a fitness function. New individuals are created on the one hand by combining the genetic make-up of two "parent" solution candidates (this procedure is called "crossover") producing a new "child", and on the other hand by mutating some individuals, i.e. changing randomly chosen parts of genetic information (normally a minor ratio of the algorithm's population is mutated in each generation).

Beside crossover and mutation, the third decisive aspect of genetic algorithms is selection. In analogy to biology this is a mechanism also called "survival of the fittest". Usually, the individual's probability to propagate its genetic information to the next generation is proportional to its fitness; the better a solution candidate's fitness value, the higher the probability, that its genetic information will be included in the next generation's population. This procedure of crossover, mutation and selection is repeated over many generations until some termination criterion is fulfilled.

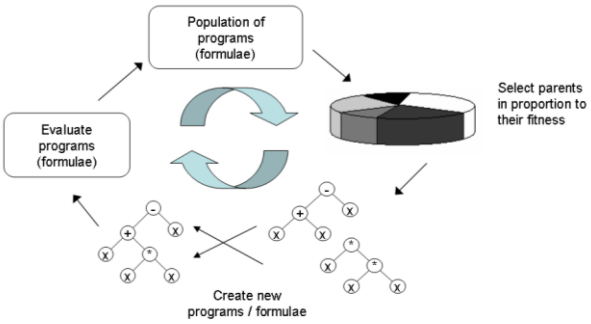

Figure 1: The Genetic Programming Cycle (Langdon and Poli 2002)

Genetic programming was first explored in depth in 1992 by John R. Koza who pointed out that virtually all problems in artificial intelligence, machine learning, adaptive systems, and automated learning can be recast as a search for a computer program, and that genetic programming provides a way to successfully conduct the search for a computer program in the space of computer programs (Koza 1992). Similar to GAs, GP works by imitating aspects of natural evolution: A Population of solution candidates evolves through many generations towards a solution using evolutionary operators (crossover and mutation) and a "survival-of-the-fittest" selection scheme. Whereas GAs are intended to find an array of characters or integers representing the solution of a given problem, the goal of a GP process is to produce a computer program (or, as in our case, a formula) solving the optimization problem at hand. As in every evolutionary process, new individuals (in GP's case, new programs) are created. They are tested, and the fitter ones in the population succeed in creating children of their own. Unfit ones die and are removed from the population (Langdon and Poli 2002). This procedure is graphically illustrated in Figure 1.

2.2 GP-Based Structure Identification

The concept of structure identification is not very common in the literature. Indeed, it is well known that every model consists of an equation set (the structure) and of values (parameters). System identification actually implies both, but usually the definition of the structure is considered either obvious or as the less critical issue,

while the consistent estimation of the parameters especially in presence of noise receives the largest part of the attention.
By its very general problem statement, GP allows to approach the problem of structure identification and the problem of parameter identification simultaneously. As a consequence, GP techniques are used for identifying various kinds of technical systems; some approaches use genetic programming to identify the structure in addition to standard parameter estimation techniques, many other ones use GP for determining both the structure and the parameters of the model of a nonlinear system as for example described in (Rodríguez-Vázquez and Fleming 2000) and (Beligiannis et al. 2005).

3. THE HEURISTICMODELER

On the basis of the GP-based structure identification methods and several other enhanced algorithmic and problem specific mechanisms we have compiled the *HeuristicModeler*, a multi-purpose machine learning algorithm that is able to evolve models for various different machine learning problem classes. The framework used for the implementation of the *HeuristicModeler* is the *HeuristicLab* (Wagner and Affenzeller 2005), a framework for prototyping and analyzing optimization techniques for which both generic concepts of evolutionary algorithms and many functions for analyzing them are available.

The algorithmic basis for the *HeuristicModeler* is an extended Genetic Algorithm implementation also called "SASEGASA" (Affenzeller and Wagner 2004). There are several new hybrid evolutionary concepts combined in this algorithmic basis, the most important ones being on the one hand the self-adaptive selection pressure steering and on the other hand the so-called Offspring Selection concept.

The selection pressure measures how hard it is to produce individuals out of the current population that improve the overall fitness. As soon as this internal selection pressure reaches a pre-defined maximum value, the algorithm is terminated and presents the best actual model as the result of the training process. Details can be found in (Affenzeller and Wagner 2004) and (Affenzeller 2005).

The basic idea of Offspring Selection is that individuals are first compared to their own parent solution candidates and accepted as members of the new generation's population if they meet certain criteria. In the context of structure identification and machine learning we have realized that the use of very rigid settings yields best results (Winkler et al. 2006b). I.e., new models are kept and thus inserted in the next generation's population only if they outperform their own parent individuals; this selection scheme is illustrated in Figure 2.

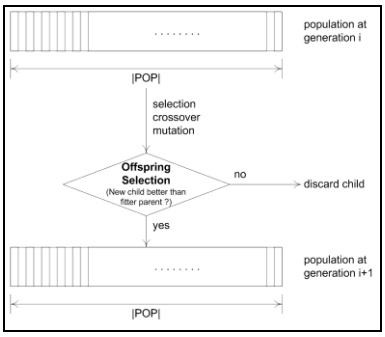

Figure 2: Embedding the Offspring Selection concept in GP-based machine learning.

The GP-based structure identification methods described in the previous section have been implemented as plug-ins for the *HeuristicLab* forming the problem specific basis of the *HeuristicModeler*. The following modeling specific extensions have been integrated into the general GP workflow:

- During the execution of a structure identification algorithm it can easily happen that a model showing a very suitable structure is assigned a very bad fitness value only due to inadequate parameter settings. Therefore we have implemented an additional local parameter optimization stage based on real-values encoded Evolution Strategies as for example explained in (Schwefel 1994) and integrated it into the execution of the Genetic Programming algorithm.
- As the GP-based model training algorithm tries to evolve better models, it can easily happen that models become more and more complex; the more complex models are, the better they can fit given training data, but they are also negative effects, namely increasing runtime consumption as well as the danger of overfitting. Therefore a heuristic tree pruning algorithm has also been integrated into the *HeuristicModeler*; in certain intervals, selected models included in the actual models pool are selected and pruned systematically, i.e. formula parts that do not seem to have a measurable influence on the model's evaluation are deleted in order to retrieve simpler models without significantly losing quality.

Due to its flexible and wide functional basis and the extended concepts described above, the GP-based modeling concept implemented in the *HeuristicModeler* is less exposed to the danger of overfitting than other machine learning algorithms; recent results and comparisons to other data-based modeling techniques in the context of medical data analysis are for example summarized in (Winkler et al. 2006c). Furthermore, as we will show in the following section, the results generated using the *HeuristicModeler* can easily be analyzed and interpreted using the *HeuristicModelAnalyzer*, a tool for analyzing solutions for data analysis problems that includes several enhanced evolutionary modeling aspects.

4. EXAMPLES AND APPLICATIONS IN THE CONTEXT OF MEDICAL DATA ANALYSIS

4.1 Regression

For demonstrating the use of our evolutionary machine learning approach for attacking regression problems we have generated a synthetic data set including 5 variables and 400 samples. This data was analyzed using the *HeuristicModeler* and a model was trained; this model is graphically shown in Figure 3.

There are several possibilities how to evaluate a regression model using the *HeuristicModelAnalyzer*: Apart from drawing the (original and estimated) values and a graphical representation of the formula as a structure tree, the average squared error can be calculated as well as an overview of the errors distribution (as exemplarily shown later in Figure 7).

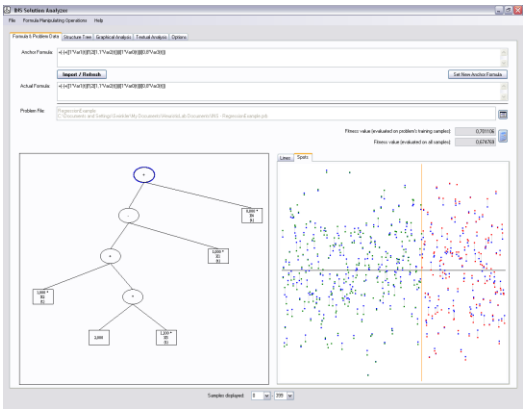

Figure 3: A solution to a regression problem, analyzed using the *HeuristicModelAnalyzer*

4.2 Classification

Several widely used benchmark classification datasets storing medical data (mainly survey records and diagnosis information) have already been analyzed using *HeuristicModeler* and *HeuristicModelAnalyzer*. In (Winkler et al. 2006a), (Winkler et al. 2006b) and (Winkler et al. 2006c) we have documented the results achieved for several medical classification benchmark problems, for example for the *Wisconsin* and the *Thyroid* datasets, which are parts of the UCI Machine Learning Repository (`http://www.ics.uci.edu/~mlearn/`).

Summarizing the results documented in the publications mentioned above, GP-based training of classifiers is able to outperform other training methods (kNN classification, linear modeling and ANNs) especially on test data. There are several possibilities how to evaluate a classification model using the *HeuristicModelAnalyzer*: Apart from drawing the (original and estimated) values and a graphical representation of the formula as a structure tree and calculating the average squared error, confusion matrices and (enhanced) receiver operating characteristics (ROC) curves can be generated. Furthermore, optimal thresholds are also identified automatically on the basis of a misclassification matrix storing information about how to weight misclassification dependent on the respective classes involved. This matrix is initially set so that all misclassifications are weighted equally; especially in medical application it can be necessary to manipulate this weighting as it is for example more critical misclassifying a diseased patient as not diseased than vice versa.

In Figure 4 we show a graphical representation of a solution for the *Wisconsin* classification problem that was generated using the *HeuristicModeler* and analyzed using the *HeuristicModelAnalyzer*. As confusion matrices are also frequently used for evaluating classifiers, these are also automatically displayed when analyzing a model using the *HeuristicModelAnalyzer*.

Figure 4: A solution for the *Wisconsin* classification problem, generated by the *HeuristicModeler* and analyzed using the *HeuristicModelAnalyzer*.

Last, but not least the *HeuristicModelAnalyzer* enables the evaluation of classifiers for multi-class classification problems on the basis of a multi-class extension of ROC curves. Basic ROC analysis provides a convenient graphical display of the trade-off between true and false positive classification rates for two class problems (Zweig and Campbell 1993). In the context of two class classification, ROC curves are calculated as follows: For each possible threshold value discriminating two given classes, the numbers of true and false classifications for one of the classes are calculated. For example, if the two classes “true” and “false” are to be discriminated using a given classifier, a fixed set of equidistant thresholds is tested and the true positives (TP) and the false positives (FP) are counted for each of them. Each pair of TP and FP values produces a point of the ROC curve.

The main idea of Multi-ROC charts as presented in (Winkler et al. 2006d) is that for each given class c_i the numbers of true and false classifications are calculated for each possible pair of thresholds between the classes c_{i-1} and c_i as well as between c_i and c_{i+1} (assuming that the n classes can be represented as real numbers and that $c_i<c_{i+1}$ holds for every $i \in [1,(n-1)]$). The resulting tuples of (FP,TP) values are stored in a matrix which can be plotted easily. This obviously yields a set of points which can be interpreted analog to the interpretation of “normal” ROC curves: the closer the point are located to the left upper corner, the higher is the quality of the

classifier at hand. For getting sets of ROC curves instead of ROC points, an arbitrary threshold t_a between the classes c_{i-1} and c_i is fixed and the FP and TP values for all possible thresholds t_b between c_i and c_{i+1} are calculated. This produces one single ROC curve; it is executed for all possible values of t_a.

An example showing 10 ROC curves is given in Figure 5; this MROC chart was generated for a classifier learned for a synthetical data set storing 2000 samples divided into 6 classes and is taken from (Winkler et al. 2006d).

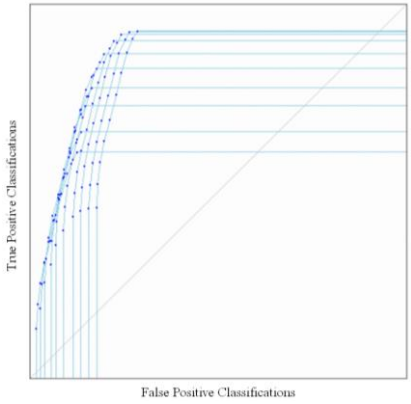

Figure 5: An exemplary Multi-ROC chart.

4.3 Timeseries Analysis

Time series analysis is in fact not a major topic in medical data analysis, there are not many research projects or results published in this area. There are also no prominent benchmark data sets for testing time series analysis algorithms on medical data. Still, there is a lot of experience using the *HeuristicModeler* for solving time series problems on data recorded in the context of mechatronical systems. For example, in (Del Re et al. 2005) we report on models trained for the NO_x emissions of Diesel engines using the GP-based identification method incorporated in the *HeuristicModeler*. Figures 6 and 7 show the evaluation of one of these models using the *HeuristicModelAnalyzer*: Apart from drawing the (original and estimated) values and a graphical representation of the formula as a structure tree, an overview of the errors distribution is given.

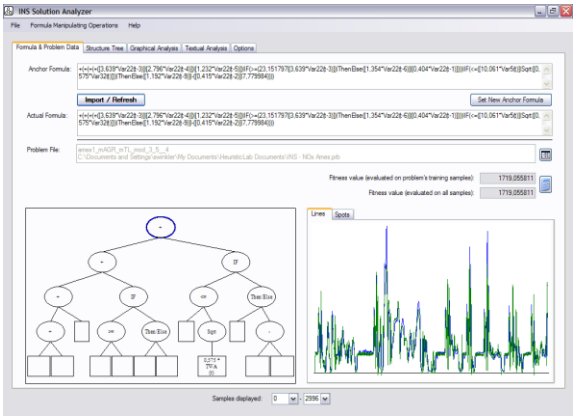

Figure 6: A model for the NOx emissions of a BMW Diesel engine, generated using the *HeuristicModeler*

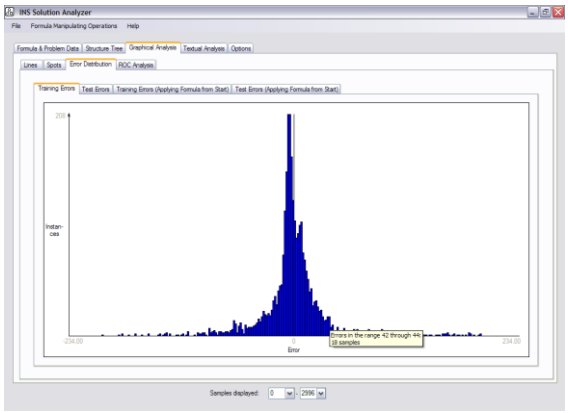

Figure 7: Evaluation of the model shown in Figure 11: Graphical display of the errors distribution.

5. CONCLUSIONS

In this paper we have described a multi-purpose machine learning approach, the *HeuristicModeler*, based on various evolutionary computation concepts that is applicable for several medical data mining aspects in evidence based medicine. We have exemplarily shown how regression, classification and time series problems can be attacked using this algorithm. Especially in the context of analyzing medical data we have already achieved very good results for classification problems; considering the enhanced concepts used and the quality of the result achieved for other time series data we are confident that the use of this machine learning algorithm will also yield satisfying results for medical time series data. Furthermore, we have also demonstrated how the *HeuristicModelAnalyzer*, a tool for analyzing the results for data mining problems as well as selected aspects of the underlying enhanced evolutionary algorithm.

ACKNOWLEDGEMENTS

The work described in this paper was done within the Translational Research Project L282 “GP-Based Techniques for the Design of Virtual Sensors” sponsored by the Austrian Science Fund (FWF). This project is executed as a joint venture by the Upper Austrian University of Applied Sciences Hagenberg, Austria and the Johannes Kepler University Linz, Austria.

REFERENCES

Affenzeller, M. 2003. *New Hybrid Variants of Genetic Algorithms - Theoretical and Practical Aspects*. Universitätsverlag Rudolf Trauner, Linz, Austria.

Affenzeller, M. and S. Wagner. 2004. “SASEGASA: A New Generic Parallel Evolutionary Algorithm for Achieving Highest Quality Results”. In *Journal of Heuristics - Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems*, vol. 10, 239-263. Kluwer Academic Publishers.

Affenzeller, M. 2005. *Population Genetics and Evolutionary Computation – Theoretical and Practical Aspects*. Universitätsverlag Rudolf Trauner, Linz, Austria.

Affenzeller, M. and S. Wagner. 2005. “Offspring Selection: A New Self-Adaptive Selection Scheme for Genetic

Algorithms". In *Adaptive and Natural Computing Algorithms*, 218-221. Springer Computer Science.

Beligiannis, G.N., L.V. Skarlas, S.D. Likothanassis and K. Perdikouri. 2005. "Nonlinear model structure identification of complex biomedical data using a genetic programming based technique". In *IEEE Transactions on Instrumentation and Measurement*, vol. 54:6, 2184- 2190.

Box, G.E.P and G.M. Jenkins. 1976. *Time Series Analysis - Forecasting and Control*. Holden Day, San Francisco.

Del Re, L., P. Langthaler, C. Furtmüller, S. Winkler and M. Affenzeller. 2005. "NOx Virtual Sensor Based on Structure Identification and Global Optimization". In *Proceedings of the SAE World Congress 2005*, paper number: 2005-01-0050.

Fayyad, U.M., G. Piatetsky-Shapiro and P. Smyth. 1996. "From Data Mining to Knowledge Discovery: An Overview". In *Advances in Knowledge Discovery and Data Mining*, 1-34. AAAI Press.

Gill, P.R., W. Murray and M.H. Wright. 1981. "The Levenberg-Marquardt method". *Practical Optimization*, 136–137.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman, Boston San Francisco New York.

Holland, J.H. 1975. *Adaption in Natural and Artificial Systems*. MIT Press, Cambridge, Mass.

Kendall, M. and J.K. Ord. 1990. *Time Series.* Edward Arnold, London.

Koza, J. 1992. *Genetic Programming: On the Programming of Computers by means of Natural Selection*. MIT Press, Cambridge, Mass.

Koza, J. 1995. "Genetic Programming for Econometric Modeling". In *Intelligent Systems for Finance and Business*, 1st edn. Wiley & Sons, 251-269.

Langdon, W. and R. Poli. 2002. *Foundations of Genetic Programming*. Springer, Berlin Heidelberg New York.

Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin Heidelberg New York.

Mitchell, T.M. 2000. *Machine Learning*. McGraw-Hill, New York.

Nelles, O. 2001. *Nonlinear System Identification*. Springer, Berlin Heidelberg New York.

Rechenberg, I. 1973. *Evolutionsstrategie*. Friedrich Frommann Verlag.

Rodríguez-Vázquez, K. and P.J. Fleming. 2000. "Use of genetic programming in the identification of rational model structures". In *Third European Conference on Genetic Programming EuroGP'2000, Lectures Notes in Computer Science 1802*, 181-192. Springer Computer Science.

Schwefel, H.P. 1994. *Numerische Optimierung von Computer-Modellen mittels Evolutionsstrategie*. Birkhäuser Verlag, Basel, Switzerland.

Wagner, S. and M. Affenzeller. 2005. "HeuristicLab: A Generic and Extensible Optimization Environment". In *Adaptive and Natural Computing Algorithms*, 538-541. Springer Computer Science.

Winkler, S., M. Affenzeller and S. Wagner. 2005. "New Methods for the Identification of Nonlinear Model Structures Based Upon Genetic Programming Techniques". In *Journal of Systems Science*, 31:1, 5-13. Oficyna Wydawnicza Politechniki Wroclawskiej.

Winkler, S., M. Affenzeller and S. Wagner. 2006. "Automatic Data Based Patient Classification Using Genetic Programming". In *Cybernetics and Systems 2006*, vol. 1, pp. 251-256. Austrian Society for Cybernetic Studies, ISBN 3-85206-172-5.

Winkler, S., M. Affenzeller and S. Wagner. 2006. "Advances in Applying Genetic Programming to Machine Learning, Focussing on Classification Problems". In *Proceedings of the 20th IEEE International Parallel & Distributed Processing Symposium IPDPS 2006, IEEE Catalog Number: 06TH8860*, paper number: NIDICS-012.

Winkler, S., M. Affenzeller and S. Wagner. 2006. "Using Enhanced Genetic Programming Techniques for Evolving Classifiers in the Context of Medical Diagnosis - An Empirical Study". In *Proceedings of the GECCO 2006 Workshop on Medical Applications of Genetic and Evolutionary Computation (MedGEC 2006).*

Winkler, S., M. Affenzeller and S. Wagner. 2006. "Sets of Receiver Operating Characteristic Curves and their Use in the Evaluation of Multi-Class Classification". In *Proceedings of the Genetic and Evolutionary Computation Conference 2006 (GECCO '06)*.

Zweig, M.H. and G. Campbell. 1993. "Receiver-Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine". *Clinical Chemistry*, 39:561-577.

AUTHOR BIOGRAPHIES

STEPHAN M. WINKLER received his MSc in Computer Science from Johannes Kepler University Linz, Austria in 2004. His research interests include Genetic Programming, Nonlinear Model Identification, Fault Detection and Machine Learning. Currently he is a research associate within the Translational Research Program L284 "GP-Based Techniques for the Design of Virtual Sensors", a research project funded by the Austrian Science Fund (FWF).

MICHAEL AFFENZELLER has published several papers and journal articles dealing with theoretical aspects of Genetic Algorithms and Evolutionary Computation in general. In 1997 he received his MSc in Industrial Mathematics and in 2001 his PhD in Computer Science, both from the Johannes Kepler University Linz, Austria. He is professor at the Upper Austrian University of Applied Sciences Hagenberg, Austria and associate professor at the Institute of Formal Models and Verification at Johannes Kepler University Linz, Austria since his habilitation in 2004.

STEFAN WAGNER also received his MSc in Computer Science from Johannes Kepler University Linz, Austria in 2004. He now holds the position of an associate professor at the Upper Austrian University of Applied Sciences Hagenberg, Austria. His research interests include Evolutionary Computation and Heuristic Optimization, Theory and Application of Genetic Algorithms, Machine Learning and Software Development.

The Web-pages of all three authors as well as further information about HeuristicLab and related scientific work can be found at `http://www.heuristiclab.com`.