Incorporating Model Identifiability into Equation Discovery of ODE Systems

Dirk J.W. De Pauw
KERMIT: Research Unit
"Knowledge-based Systems"
Department of Applied Mathematics,
Biometrics and Process Control
Ghent University
Coupure links 653
Gent, Belgium
dirk.depauw@ugent.be

Bernard De Baets
KERMIT: Research Unit
"Knowledge-based Systems"
Department of Applied Mathematics,
Biometrics and Process Control
Ghent University
Coupure links 653
Gent, Belgium
bernard.debaets@ugent.be

ABSTRACT

Equation discovery is a machine learning technique that tries to automate the discovery of equations from measured data. In this contribution an equation discovery system based on genetic programming was developed in order to generate mechanistic models for systems described by ordinary differential equations. A problem often encountered with automatic model generation is that overly complex models are generated that "overfit" the measured data. This issue was addressed by incorporating a model identifiability measure (expressing which fraction of the model parameters can be given a unique value given the available data) into the fitness function of the individuals. Using noisy artificially generated data for a river water quality example case, it was shown that the developed system was able to generate model equations that fitted the data well and were also fully identifiable. Correct model equations were generated when starting from a model with minimum prior knowledge but also when starting from an overly complex model. As such, it was demonstrated that the developed equation discovery system is able to generate models with optimal complexity with regard to the available data.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming—*program synthesis, program modification*

General Terms

Algorithms

Keywords

genetic programming, equation discovery, identifiability, model complexity, overfitting

1. INTRODUCTION

Building mathematical models has long been of interest to scientists and practitioners. Indeed, models not only allow to describe the current and future behaviour of a system but are also considered valuable learning and instructing tools. In many fields of science and engineering, intense research is focused on the study of dynamic systems which will also be the subject of this research. These systems, that change over time, are often modelled using ordinary differential equations (ODEs).

For each modelling exercise two steps can be distinguished. In a first step, a valid model structure needs to be found, based on background knowledge about the system, which describes the relationships between the measured variables and how they change over time. In a subsequent step, acceptable values for the model parameters need to be determined. Modern-day modelling focuses mainly on the second step, very often selecting a model structure from literature and assuming it to be valid. The reason why the first step is often not properly addressed is that it can be a very time-consuming exercise to generate and test different model structures.

In order to automate the model building process, several machine learning techniques were developed during the past decades. Equation discovery is such a technique that tries to automate the discovery of equations from measured data [13, 21]. This technique is closely related to inductive process modelling [12, 4] in which models are automatically constructed drawing heavily on system knowledge encoded in, for example, a library of candidate model structures or substructures. Much progress has been made in this field of research, especially on how to incorporate domain knowledge into the algorithms, e.g. by using context-free grammars and high level model description languages [20, 1, 19]. This contribution, however, focuses on another frequently encountered problem in equation discovery and inductive process modelling, namely model complexity and overfitting issues.

Several researchers have remarked that automatic model building algorithms tend to produce overly complex models [11, 3]. These models tend to fit the training data very well but perform poorly when confronted with new data. This problem is often referred to as overfitting. The most common method used to deal with this issue is to include a measure of complexity in the objective function that is used to determine how well a model describes the training data set. Several complexity measures have been proposed,

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

most of them being a function of the length of the generated equation and/or the number of parameters or variables [20, 11, 7, 9]. As such, complex models are penalized when they are compared to less complex models that describe the data equally well. The idea behind this approach is closely related to the concept of parsimony and Occam's razor principle which states that the explanation of any phenomenon should make as few assumptions as possible, eliminating those that make no difference in the observable predictions of the explanatory hypothesis or theory.

Another approach for dealing with model complexity is to analyze the model's identifiability. Such analysis can be used to determine how many model parameters can be given a unique value given the quality and quantity of the available data. A model for which not all parameters can be given unique values is said to be unidentifiable and overparameterized. This is an undesirable model property since overparameterization is the root cause of overfitting. The reason why overparameterized models tend to perform poorly when confronted with new data is that the unidentifiable parameters need to be given values that cannot be determined from the data. Typically literature values are assumed which might not correspond to the true system values.

Model identifiability can be regarded as an aid that can assist in the construction of models that fit the data and have an optimal complexity. A study in which identifiability analysis was applied manually can be found in [8] where it was used to reduce the complexity of an existing model. In this paper, model identifiability will be used in a genetic programming based equation discovery system in order to deal with complexity and overfitting issues and, as such, improve the practical usefulness of the automatically generated models.

2. A GENETIC PROGRAMMING BASED EQUATION DISCOVERY SYSTEM

Genetic programming is an evolutionary algorithm that can be used to build "programs" that perform a user-defined task. Within the context of equation discovery, these "programs" are mathematical models that are used to describe a measurement data set as good as possible. Several authors have already demonstrated that genetic programming can be used successfully in the search for new or improved mathematical models [10, 2]. The following subsections will describe the equation discovery system developed and used for this research in more detail.

2.1 Individual representation

The task of the equation discovery system described here is to find models that can be written as sets of first order ordinary differential equations. The models are also required to be "mechanistic", meaning that the model parameters are required to have a physical meaning.

Probably the most commonly used structure to represent individuals in genetic programming is a tree structure. This structure has also been used here to represent the equations of the variables selected for discovery. Since the system supports the discovery of several equations simultaneously, each genetic programming individual consists of set of trees in which each tree represents one equation. For this application binary trees are used, composed of interconnected operator and terminal nodes. The operator nodes can be any of the four classical arithmetic operators: add, subtract, divide and multiply, while the terminal nodes can represent different model quantities like: parameters, inputs or other model variables.

Rather than starting from scratch to build a model, a model revision approach is followed here in which the modeller supplies the system with an initial model that is believed to be a good representation of the system under study. As such, the modeller is able to supply existing domain knowledge to the system and outline the global structure of the model, i.e. define the variables and processes involved in the system. A similar approach was used in [17] where its usefulness was clearly demonstrated. Once the initial model is specified, the modeller has to define for which variables the equations need to be discovered (or modified) and for which variables the equations of the initial model are considered "correct" and need not be modified.

Beside the definition of an initial model, the user can also supply domain knowledge in another way. For each equation to be discovered, it can be specified which model variables this equation can be function of and which of the four operators are allowed.

2.2 Mutation and crossover operators

Genetic programming, like many other evolutionary algorithms, tries to mimic the processes of natural evolution in order to evolve the individuals of a population into new individuals. This is accomplished through the use of genetic operators like mutation and crossover.

When developing mutation and crossover operators within the context of equation discovery an important aspect of mechanistic modelling should not be neglected, namely the unit consistency of the equations. What is meant by this is that the units of the right-hand side of an equation should match the units of the left-hand side. Unit consistency is especially important when models are considered in which parameters are allowed to appear several times in the equations, something that occurs in most models. The requirement of unit consistency greatly affects the design of the mutation and crossover operators.

A mutation operator typically acts on a single parent individual to form an offspring individual. In our implementation, a mutation is performed by selecting one node of the tree at random and replacing it (and its subtree) with a new randomly generated subtree. For unit consistency, the root node of the newly generated subtree should have identical units as the node that it replaces. A new subtree is constructed in an iterative way by randomly choosing either an operator node or a terminal node. If an operator node is generated, random subtrees for each of the operator's child nodes are generated. The probability of choosing an operator is taken to be 0.4. This causes slightly more terminals to be generated resulting in relatively compact subtrees. During the construction of subtrees, the mutation operator also allows new parameters to be added to the model. This allows to increase the complexity of the model equations but is often also required for reasons of unit consistency.

As is mostly the case, the crossover operator used in this system acts on two parent individuals to form two new offspring individuals by interchanging subtrees between the parents. From the first parent a node (and its attached subtree) is selected at random and exchanged with a node (and its subtree) of the second parent. For unit consistency,

only nodes with identical units as the node selected from the first parent are allowed to be selected from the second parent. As such, the newly created individuals are always unit consistent.

As already discussed above, several equations are allowed to be discovered simultaneously. It was chosen to give each of the equations of an individual only a 50% chance of being affected by mutation or crossover. This approach was required in order to give good equations a fair chance of being transferred to offspring individuals without being affected by the, mostly, destructive mutation and crossover operations.

2.3 Fitness determination

Each model generated by the equation discovery system is assigned a fitness value based on two components: model fit and identifiability. Both measures are calculated after the model has been calibrated, i.e. a value is determined for each of its parameters by minimizing the squared difference between model prediction and measured data. This minimization was performed using the Simplex optimization algorithm [16].

In order to judge the fit of the model to the data, the model efficiency criterion (also called Nash-Sutcliffe criterion) was used [15]:

$$ME = 1 - \frac{\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2}, \qquad (1)$$

where Y_i is an observation, $\hat{Y}_i$ a predicted value and $\bar{Y}$ the average of all n observations. For a perfect fit, this statistic results in a value equal to one. The (theoretical) lower bound of ME is negative infinity and for ME values lower than zero, the model-predicted values are worse than the observed mean. In our application, negative ME values are also assigned a value of zero.

Model complexity is quantified using the collinearity index identifiability measure [5, 8]. This measure quantifies the linear dependency between the sensitivity functions of the model variables to the model parameters. A model parameter set is said to be unidentifiable when its collinearity index exceeds 15 [5]. In such case, strong parameter correlations exist and unique parameter values cannot be assigned. In our approach, the fraction of identifiable parameters is determined for each generated model and serves as a measure for model complexity. A model that is fully identifiable will have an identifiability measure of one while a model that does not contain any identifiable parameters will have a value of zero.

Finally, the fitness measure assigned to each model is the sum of the model efficiency and the fraction of identifiable parameters. Since both quantities have a value between zero and one, the highest fitness value obtainable is two. This corresponds to a perfect fitting model that is fully identifiable.

2.4 Main genetic programming algorithm

Several steps can be distinguished in a genetic programming algorithm. As a first step, an initial population has to be created. In our system this is achieved by creating offspring through mutation of the initial model that is supplied to the system. It was chosen to work with a population size of 100 individuals. Once the population is created, the fitness of each of the models is determined and roulette wheel selection is used to select parent individuals. Using the selected parents, offspring individuals are created through crossover (40% probability) or mutation (60% probability). This process of selection and offspring generation continues until an offspring population of 100 individuals is formed. In a next step, both populations are combined and the 100 most fit individuals retained in order to form the next parent population. These step are repeated until 100 generations are created at which point the algorithm terminates.

Table 1: True and estimated parameter values and initial conditions for the example case model

Name	Unit	True	Estimated
$BOD_{(t=0)}$	mg.l^{-1}	7.33	7.4244
$DO_{(t=0)}$	mg.l^{-1}	8.5	8.5470
k_1	min^{-1}	0.3	0.3037
k_2	min^{-1}	0.4	0.4023

3. EXAMPLE CASE

Surface water contamination can be regarded as a serious problem throughout the whole world, affecting both developed and underdeveloped countries. One of the most prominent manifestations of surface water contamination with organic waste is a reduction of the dissolved oxygen concentration. Indeed, as the organic waste is decomposed by bacteria, oxygen in the water is consumed, leaving the water "oxygen depleted". Such oxygen-depleted water may not be able to support aquatic life that depends on the oxygen for survival.

In 1925, a relatively simple model describing the decreased oxygen concentration downstream of a polluting discharge was proposed by Streeter and Phelps [18]. Although their model is based on several assumptions, it is still often used as the basis for more complicated river water quality models. The model consists of two differential equations:

$$\frac{\partial BOD}{\partial t} = BOD_{in} - k_1.BOD \qquad (2)$$

$$\frac{\partial DO}{\partial t} = k_2.(DO_{sat} - DO) - k_1.BOD \qquad (3)$$

where BOD (biochemical oxygen demand) is the amount of oxygen required to decompose a certain amount of organic waste (mg.l^{-1}), BOD_{in} the waste inflow rate ($\text{mg.l}^{-1}.\text{min}^{-1}$), k_1 a degradation rate (min^{-1}), DO the dissolved oxygen concentration (mg.l^{-1}), k_2 the reaeration rate (min^{-1}) and DO_{sat} the oxygen saturation constant (mg.l^{-1}). Eq. (2) describes the change in BOD concentration downstream of the discharge which is determined by the waste inflow rate and the speed at which bacteria degrade the waste. The change of DO concentration is described by Eq. (3) and is driven by the reaeration of the water through its contact with the air and the oxygen consumption due to the bacterial waste decomposition.

Fig. 1 shows simulation results of BOD and DO using the "true" parameter values and initial conditions given in Table 1 for a river with a flow rate of $20\ \text{m.min}^{-1}$. Note that Table 1 does not list values for BOD_{in} and DO_{sat} which are assumed to be model constants and are assigned a value of $1\ \text{mg.l}^{-1}.\text{min}^{-1}$ and $11\ \text{mg.l}^{-1}$, respectively.

Synthetic measurement data (also shown in Fig. 1) was constructed by adding normally distributed noise to the sim-

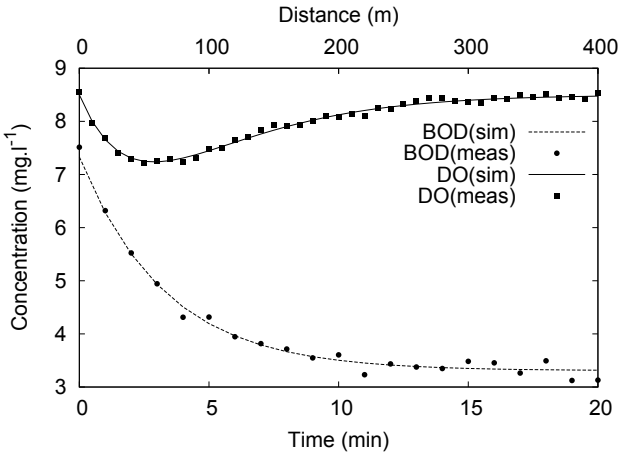

Figure 1: Synthetic data and simulated BOD and DO concentrations for a river with a flow rate of $20\,\mathrm{m.min^{-1}}$

Table 2: Model efficiency and identifiability criteria values for the true, initial and generated models

Model	Model efficiency	Identifiability
True	0.9892	1
Complex initial	0.9892	0.7142
Complex generated	0.9895	1
Simple initial	0.6516	1
Simple generated	0.9896	1

ulation results. In practice, BOD measurements are performed offline in the lab, based on water samples taken from the river. Therefore, a measurement interval of 1 min was chosen and a measurement noise standard deviation of $0.1\,\mathrm{mg.l^{-1}}$ assumed. For DO, measurements can be performed using an online sensor that allows for more frequent and more accurate measurements. Therefore, a measurement interval of 0.5 min and a measurement noise standard deviation of $0.05\,\mathrm{mg.l^{-1}}$ were chosen to generate the synthetic dissolved oxygen measurements.

Based on the synthetic data, the model was calibrated. The estimated parameter and initial condition values are listed in Table 1. The model efficiency and identifiability criteria values are listed in Table 2 ("True"). From these values it is clear that the "true" model is identifiable and results in a perfect fit. The fact that the model efficiency criterion does not reach one is normal and is caused by the presence of noise on the measurements.

4. RESULTS

The example case presented in the previous section was used to test the ability of the equation discovery system to construct a model that fits the data well and, at the same time, is fully identifiable. For this, two tests were conducted, each test starting from another initial model with different model efficiency and identifiability properties.

As a first test, an initial model that fitted the data well but is not fully identifiable ("Complex initial" in Table 2), was supplied to the system. For the organic pollution degradation, a Monod type kinetic was chosen rather than the first order degradation of the "true" model. The reaeration rate

Table 3: Estimated parameter values and initial conditions of the generated model starting from the complex model

Name	Unit	Value
$BOD_{(t=0)}$	$\mathrm{mg.l^{-1}}$	7.4749
$DO_{(t=0)}$	$\mathrm{mg.l^{-1}}$	8.5282
p_{c1}	$\mathrm{mg.l^{-1}.min^{-1}}$	1.0380
k_{1max}	$\mathrm{min^{-1}}$	0.3139
p_{c2}	$\mathrm{min^{-1}}$	0.3913
p_{c3}	$\mathrm{min^{-1}}$	-0.4097

was also assumed to be more complex and to be described by two terms, one term due to the river's hydraulics and one term due to the wind speed and river depth. The model equations are as follows:

$$\frac{\partial BOD}{\partial t} = BOD_{in} - k_{1max}.\frac{BOD}{BOD_{half}+BOD}.BOD \quad (4)$$

$$\frac{\partial DO}{\partial t} = \left(k_{2h}+\frac{k_{2w}}{h}\right)(DO_{sat}-DO) - k_{1max}.\frac{BOD}{BOD_{half}+BOD}.BOD \quad (5)$$

where k_{1max} is the maximum degradation rate ($\mathrm{min^{-1}}$), BOD_{half} the BOD concentration at which half of the degradation rate is reached ($\mathrm{mg.l^{-1}}$), k_{2h} the reaeration contribution due to the river's hydraulics, k_{2w} the reaeration contribution due to the wind ($\mathrm{m.min^{-1}}$) and h the river depth (m). The identifiability analysis showed that only 70% of the parameters were identifiable and that strong correlations exist in two parameter groups: [k_{1max}; BOD_{half}] and [k_{2h}; k_{2w}; h].

Using the initial model defined by Eqs. (4) and (5), the equation discovery system converged to following model:

$$\frac{\partial BOD}{\partial t} = p_{c1} - k_{1max}.BOD \quad (6)$$

$$\frac{\partial DO}{\partial t} = p_{c2}.(DO_{sat}-DO) - (p_{c2}+p_{c3}+k_{1max}).BOD \quad (7)$$

As can be seen from Table 2 ("Complex generated"), this model is fully identifiable and fits the data as good as the "true" model. Table 3 lists the estimated parameter values and initial conditions.

Looking at the BOD equation (Eq. (6)), one can see that it is identical to Eq. (2) of the true model. One observes that the Monod kinetic has been removed from the initial model and replaced by a first order degradation rate (k_{1max}) with value $0.3139\,\mathrm{min^{-1}}$ which corresponds to the degradation rate (k_1) of the true model. Further, the BOD_{in} model constant was removed from the model and replaced by a parameter, p_{c1}, with identical units and an estimated value of $1.0380\,\mathrm{mg.l^{-1}.min^{-1}}$. This values is close to the true BOD_{in} value of $1\,\mathrm{mg.l^{-1}.min^{-1}}$.

An analysis of the DO equation (Eq. (7)) shows that it is also very similar to the DO equation of the true model (Eq. (3)). The too complex reaeration rate term of the initial model has been removed from the model (along with its parameters) and replaced with a rate (p_{c2}) equal in value

Table 4: Estimated parameter values and initial conditions of the generated model starting from the simple model

Name	Unit	Value
$BOD_{(t=0)}$	mg.l^{-1}	7.4749
$DO_{(t=0)}$	mg.l^{-1}	8.5430
p_{s1}	$\text{mg}^2\text{.l}^{-2}\text{.min}^{-1}$	11.3917
p_{s2}	$\text{mg.l}^{-1}\text{.min}^{-1}$	3.4457
p_{s3}	min^{-1}	0.2799
p_{s4}	$-$	-1.1588
p_{s5}	l.mg^{-1}	-0.1035

to that of k_2 of the true model. For the second part of the equation, related to the BOD degradation, the Monod kinetic has been removed successfully. At first sight, this term still appears to be more complex. However, closer inspection of the parameters and their value reveals that the sum of parameters p_{c3} and p_{c2} almost equals zero and could thus be removed from the model, making the equation identical to the one from the true model.

A second test was performed using an initial model with minimum prior system knowledge. A constant change of BOD and DO over time was assumed:

$$\frac{\partial BOD}{\partial t} = a \tag{8}$$

$$\frac{\partial DO}{\partial t} = b \tag{9}$$

It is clear from Table 2 ("Simple initial") that this model is unable to adequately describe the generated data and better equations need to be discovered. Some additional system knowledge was provided to the equation discovery system: both BOD and DO equations can be function of BOD and DO and the use of model constants DO_{sat} and BOD_{in} is allowed in each of these equations.

The model that was generated by the system is given by Eqs. (10) and (11), its estimated parameters values and initial conditions are listed in Table 4.

$$\frac{\partial BOD}{\partial t} = \frac{p_{s1} - BOD.p_{s2}}{DO_{sat}} \tag{10}$$

$$\frac{\partial DO}{\partial t} = p_{s3}.(DO_{sat} + p_{s4}.DO) + p_{s3}.p_{s5}.BOD^2 \tag{11}$$

The model efficiency and identifiability criteria for this model ("Simple generated" in Table 2) show that the model fits the data very well and that it is fully identifiable given the available data.

Analyzing the BOD equation (Eq. (10)) shows that the true model equation was discovered. This becomes clear when considering the estimated parameter values of Table 4. Indeed, p_{s1}/DO_{sat} equals $1.0356\ \text{mg.l}^{-1}\text{.min}^{-1}$ which corresponds to BOD_{in} and p_{s2}/DO_{sat} equals $0.3132\ \text{min}^{-1}$ which corresponds to the k_1 degradation rate of the true model.

In contrast to the BOD equation, the DO equation of the true model was not discovered by the system. However, it has to be remarked that both models describe the data equally well and are both fully identifiable. The main difference between Eqs. (3) and (11) is that the DO equation of the generated model is function of BOD^2 instead of BOD. Further, an additional scaling factor (p_{s4}) is introduced to the DO variable in the equation. Beside these differences, the general structure of both equations is very similar.

5. DISCUSSION

The results presented above show that the equation discovery system is able to generate useful models that not only fit the synthetic data well but are also fully identifiable. By introducing the identifiability criterion as a component of the fitness measure, models without an excessive amount of parameters are formed. However, in some instances additional simplifications could have been made to the equations. For example, in Eq. (7) the sum of parameters p_{c3} and p_{c2} almost equals zero and could have been removed from the model. Also, in Eq. (10) the ratios p_{s1}/DO_{sat} and p_{s2}/DO_{sat} could have been replaced by single parameters. Such simplifications cannot be performed automatically by the system yet, but it is clear that such functionality would improve the quality of the generated models.

An increasingly important topic in genetic programming is code bloat, which is the excessive growth of individuals due to pieces of code that do not contribute to the fitness of the individuals. The introduction of an identifiability measure clearly prevents the models to grow too excessively, although some code bloat (with respect to the model variables) can still be observed. For example, an equation term DO/DO equals one and would have no meaning if multiplied with another equation term. Much research has been performed with respect to code bloat issues and several solutions have been proposed [14]. With respect to equation discovery, a symbolic analysis and simplification of the equations might be a way to solve these problems.

As was shown with Eq. (11), the equation discovery system does not always generate model equations that are identical to the ones that were used to generate the data (although most of the time it does). However, these equations describe the data equally well and are also fully identifiable. This issue is certainly not problematic since it is logical to assume that alternative model structures can exist that describe a certain data set equally well. One option to deal with such alternative model structures would be to design a discriminating experiment and gather new data which would allow to decide on the most likely model [6].

The research performed in this study has been on a case of artificially generated data based on a known model. Although this step is required to illustrate and prove that the system can discover correct equations, a more challenging task would be to work with real-world data and an unknown system model. However, before such cases are considered, the system will need to be extended with support for power, exponent, trigonometric and other more complicated functions. This will improve the chance of the system to find adequate models for real-world cases since many processes exist that can be described by these functions.

6. CONCLUSIONS

In this contribution, an equation discovery system based on genetic programming was developed in order to generate mechanistic models for systems described by ordinary differential equations. Using a model identifiability measure as one of the components of the fitness function, the issues of overfitting and model complexity were addressed.

Another important aspect of the developed system is that only unit consistent equations are generated, i.e. the units of the left-hand side of the equation match those of the right-hand side. This requirement had also implications for the design of the crossover and mutation operators.

Background knowledge about the system under study can be supplied to the equation discovery system in the form of an initial model. This enables the modeller to supply a model that is believed to be a good representation of the system. Using this model as a template, the initial population of the genetic programming algorithm is generated.

Using noisy artificially generated data for a river water quality example case, it was shown that the developed system was able to generate model equations that fitted the data well and were also fully identifiable. Correct model equations were generated starting from an initial model that was too simple but also from one that was too complex, illustrating the ability of the equation discovery system to generate models with optimal complexity with regard to the available data.

7. ACKNOWLEDGMENTS

This research was made possible due to the Postdoctoral Fellow funding granted to the first author by the Special Research Fund (BOF) of Ghent University.

8. REFERENCES

[1] N. Atanasova, L. Todorovski, S. Dzeroski, and B. Kompare. Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling*, 194(1-3):14–36, Mar. 2006.

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

[3] W. Bridewell, N. Bani Asadi, P. Langley, and L. Todorovski. Reducing overfitting in process model induction. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 81 – 88, 2005.

[4] W. Bridewell, P. Langley, L. Todorovski, and S. Dzeroski. Inductive process modeling. *Machine Learning*, 71(1):1–32, Apr. 2008.

[5] R. Brun, M. Kuhni, H. Siegrist, W. Gujer, and P. Reichert. Practical identifiability of asm2d parameters - systematic selection and tuning of parameter subsets. *Water Research*, 36(16):4113–4127, Sept. 2002.

[6] G. Buzzi-Ferraris and P. Forzatti. A new sequential experimental design procedure for discriminating among rival models. *Chemical Engineering Science*, 38(2):225–232, Feb. 1983.

[7] G. M. Cox, J. M. Gibbons, A. T. A. Wood, J. Craigon, S. J. Ramsden, and N. M. J. Crout. Towards the systematic simplification of mechanistic models. *Ecological Modelling*, 198(1-2):240–246, Sept. 2006.

[8] D. J. W. De Pauw, K. Steppe, and B. De Baets. Identifiability analysis and improvement of a tree water flow and storage model. *Mathematical Biosciences*, 211(2):314–332, Feb. 2008.

[9] O. Giustolisi and D. A. Savic. A symbolic data-driven technique based on evolutionary polynomial regression. *Journal of Hydroinformatics*, 8(3):207–222, Mar. 2006.

[10] G. J. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman, and T. Weinbrenner. Nonlinear model structure identification using genetic programming. *Control Engineering Practice*, 6(11):1341–1352, Nov. 1998.

[11] P. Langley, D. George, S. Bay, and K. Saito. Robust induction of process models from time-series data. In *Proceedings of the 20th International Conference on Machine Learning*, pages 432–439, 2003.

[12] P. Langley, O. Shiran, J. Shrager, L. Todorovski, and A. Pohorille. Constructing explanatory process models from biological data and knowledge. *Artificial Intelligence in Medicine*, 37(3):191–201, July 2006.

[13] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow. *Scientific Discovery: Computational Explorations of the Creative Processes*. MIT Press, 1987.

[14] S. Luke and L. Partait. A comparison of bloat control methods for genetic programming. *Evolutionary Computation*, 14(3):309–344, Sept. 2006.

[15] J. E. Nash and J. V. Sutcliffe. River flow forecasting through conceptual models part i - a discussion of principles. *Journal of Hydrology*, 10(3):282–290, Mar. 1970.

[16] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[17] K. Saito, P. Langley, T. Grenager, C. Potter, A. Torregrosa, and S. A. Klooster. Computational revision of quantitative scientific models. In *Proceedings of the 4th International Conference on Discovery Science*, pages 336 – 349, 2001.

[18] H. W. Streeter and E. B. Phelps. *A study of the pollution and natural purification of the Ohio river*. U.S. Public Health Service Bulletin No. 146. Washington D.C., 1925.

[19] L. Todorovski and S. Dzeroski. Integrating knowledge-driven and data-driven approaches to modeling. *Ecological Modelling*, 194(1-3):3–13, Mar. 2006.

[20] L. Todorovski, S. Dzeroski, and B. Kompare. Modelling and prediction of phytoplankton growth with equation discovery. *Ecological Modelling*, 113(1-3):71–81, Nov. 1998.

[21] L. Todorovski, S. Dzeroski, P. Langley, and C. Potter. Using equation discovery to revise an earth ecosystem model of the carbon net production. *Ecological Modelling*, 170(2-3):141–154, Dec. 2003.