Theory Revision in Equation Discovery

Ljupčo Todorovski, Sašo Džeroski

Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, 0.50 Ljubljana, Slovenia
Ljupco.Todorovski@ijs.si, Saso.Dzeroski@ijs.si

Abstract. State of the art equation discovery systems start the discovery process from scratch, rather than from an initial hypothesis in the space of equations. On the other hand, theory revision systems start from a given theory as an initial hypothesis and use new examples to improve its quality. Two quality criteria are usually used in theory revision systems. The first is the accuracy of the theory on new examples and the second is the minimality of change of the original theory. In this paper, we formulate the problem of theory revision in the context of equation discovery. Moreover, we propose a theory revision method suitable for use with the equation discovery system LAGRAMGE. The accuracy of the revised theory and the minimality of theory change are considered. The use of the method is illustrated on the problem of improving an existing equation based model of the net production of carbon in the Earth ecosystem. Experiments show that small changes in the model parameters and structure considerably improve the accuracy of the model.

1 Introduction

Most of the existing equation discovery systems make use of a very limited portion of the theoretical knowledge available in the domain of interest. Usually, the domain knowledge is used to constrain the search space of possible equations to the equations that make sense from the point of view of the domain experts. One of the aspects of the domain knowledge that is usually neglected by the equation discovery systems are the existing models in the domain. Rather than starting the search with an existing equation based model, equation discovery systems always start their search from scratch. In contrast with them, theory revision systems [9, 3] start with an existing model and use heuristic search to revise the model in order to improve its fit to observational data.

Most of the work on theory revision systems is on the revision of theories in propositional and first-order logic [9]. In this paper, we propose a flexible grammar based framework for theory revision in equation discovery. The existing initial model is transformed to a grammar, and alternative productions are used to define a space of possible revised equation models. The grammar based equation discovery system LAGRAMGE [6] is then used to search through the space of revised models and find the one that fits observational data best. The use of the proposed framework is illustrated on revising an equation based earth-science model of the net production of carbon in the Earth ecosystem.

The paper is organized as follows. The following section give a brief introduction to grammar based equation discovery. Typical approaches to revision of theories in propositional and first-order logic are briefly reviewed in Section 3. The grammar based framework for theory revision in equation discovery is presented in Section 4. Section 5 presents the experiments with revising the earth-science equation model. The last section summarizes the paper, discusses related work and gives direction for further work.

2 Equation Discovery

Equation discovery is the area of machine learning that develops methods for automated discovery of quantitative laws, expressed in the form of equations, in collections of measured data [1]. Equation discovery systems heuristically search through a subset of the space of all possible equations and try to find the equation which fits the measured data best.

Different equation discovery systems explore different spaces of possible equations. Early equation discovery systems used pre-defined (built-in) spaces that were small enough to allow effective heuristic (or exhaustive) search. However, this approach does not allow the user of the equation discovery system to tailor the space of possible equation to the domain of interest. On the other hand, recent equation discovery systems use different approaches to allow the user to restrict the space of the possible equations. In equation discovery systems that are based on genetic programming, the user is allowed to specify a set of algebraic operators that can be used. A similar approach has been used in the EF [10] equation discovery system. The equation discovery system SDS [7] effectively uses user provided scale-type information about the dimensions of the system variables and is capable of discovering complex equations from noisy data.

Finally, the equation discovery system LAGRAMGE [6] allows the user to specify the space of possible equations using a context free grammar. Note that grammars are a more general and powerful mechanism for tailoring the space of the equations to the domain of use than the ones used in SDS [7] and EF [10]. In the rest of this section we will describe this grammar based approach to equation discovery used in LAGRAMGE.

2.1 Grammar-Based Equation Discovery

The problem of grammar based equation discovery can be formalized as follows.
Given:
- a set of variables $V = v_1, v_2, \ldots, v_n$ of the observed system, including a target dependent variable $v_d \in V$;
- a grammar G; and
- a table M of observations (measured values) of the system variables.

Find a model E in the form of one or more algebraic or differential equations defining the target variable v_d that:

1. is derived by the grammar G; and

2. minimizes the discrepancy between the observed values of the target variable v_d and the values of v_d obtained with simulating the model.

An example of a grammar for equation discovery is given in Table 1. The grammar contains a set of two nonterminal symbols {`P_Vdiff, Vdiff`}, with a set of productions attached to each of them, and a set of three terminal symbols {`v1, v2, const[0:1]`}. The semantics of the terminal and nonterminal symbols in the grammar are explained below.

There are two types of terminal symbols used in the grammars for equation discovery. The first group is used to denote the variables of the observed system (`v1` and `v2` in the example grammar from Table 1). Another group of terminal symbols of the form `const[l:h]` is used to denote the constant parameter in the equation model whose value has to be fitted against the observational data from M. A constraint `[l:h]` specifies that the value of the constant parameter should be within the interval $l \leq v \leq h$.

Table 1. An example of a grammar for equation discovery defining the space of polynomials of a single variable $vdiff = v_1 - v_2$.

```
P_Vdiff -> const[0:1]
P_Vdiff -> const[0:1] + (P_Vdiff) * (Vdiff)
Vdiff -> v1 - v2
```

The nonterminal symbol `Vdiff` defines an intermediate variable which is the difference between two system variables `v1` and `v2`. This is done with the single production for the nonterminal symbol `Vdiff`. The other nonterminal symbol `P_Vdiff` is used to build polynomials of an arbitrary degree.

2.2 Lagramge

The equation discovery system Lagramge applies heuristic (or exhaustive) search through the space of models generated using user provided grammar G. The values constant parameters (terminal symbols `const`) in the generated models are fitted against input data M using standard non-linear constrained optimization method. After fitting the values of the constant parameters tho model is evaluated according to the sum of squared errors (SSE heuristic function [6]), i.e., the differences between observed values of the target variable v_d and the values of v_d calculated by the model. Alternative MDL heuristic function that takes into account the complexity of the model can be also used [6].

3 Theory Revision

The problem of theory revision can be defined as follows: **Given** an imperfect domain theory in the form of classification rules and a set of classified examples,

find an approximately minimal syntactic revision of the domain theory that correctly classifies all of the examples.

A representative system that addresses this problem is EITHER [3]. EITHER refines propositional Horn-clause theories using a suite of abductive, deductive and inductive techniques. Deduction is used to identify the problems with the domain theory, while abduction and induction are used to correct them. The problem of theory revision has received a lot of attention in the field of inductive logic programming [2], where a number of approaches have been developed for revising theories in the form of first-order Horn clause theories. For an overview, we refer the reader to [9].

Two kinds of problems are encountered within imperfect domain theories: over-generality occurs when an example is classified into a class other than the correct one, while over-specificity occurs when an example cannot be proven to belong to the correct class. Note that a single example can be misclassified both ways at the same time. Overly general rules are either specialized by adding new conditions to their antecedents or are deleted from the knowledge base. Problems of over-specificity are solved by generalizing the antecedents of existing rules, e.g., by removing conditions from them, or by the induction of new rules.

4 Grammar-Based Theory Revision of Equation Models

4.1 Problem Definition

The problem of grammar based theory revision can be formalized as follows. **Given**:

- a set of variables $V = v_1, v_2, \ldots, v_n$ of the observed system, including a target dependent variable $v_d \in V$;
- an existing model E, represented as an equation(s) defining the target variable v_d. Note that this can actually be a set of (algebraic or differential) equations defining the value of the target variable v_d;
- a grammar G that derives the model E; and
- a table M of observations (measured values) of the system variables.

Find a revised model E' (equation/set of equations as above) that:

1. is derived by the grammar G;
2. minimizes the discrepancy between the observed values of the target variable v_d and the values of v_d obtained with simulating the model; and
3. differs from the existing model E as little as possible.

Items 2. and 3. above would typically appear in a formulation of a general theory revision problem, regardless of the language in which the theories are expressed. In contrast to our formulation, however, the possible changes to the initial theory would be specified in terms of revision operators that can be applied to the initial and intermediate theories. As theories are typically logical theories in theory revision settings, operators typically include addition/deletion of entire rules (propositional or first-order Horn clauses) and addition/deletion of conditions in individual rules.

4.2 From an initial model to a grammar

In a typical setting of revising an existing scientific model, we would only have observational data and a model, i.e., an equation developed by scientists to explain a particular phenomenon. A grammar that would explain how this model was actually derived and provide options for alternative models is typically not available. The above is especially true for simpler models.

However, when the model (equation) is complex, it is only rarely written as a single equation defining the target variable, but rather as a set of equations defining the target variable, which typically contains equations defining intermediate variables. The latter typically define meaningful concepts in the domain of discourse. Often, alternative equations defining an intermediate variable would be possible and the modeling scientist would choose one of these: the alternatives would rarely (if ever) be documented in the model itself, but might be mentioned in a scientific article describing the derived model and the modeling process.

Table 2. Equations defining the NPPc variable in the CASA earth-science model.

$$
\begin{array}{l}
NPPc = \max(0, E \cdot IPAR) \\
\quad E = 0.389 \cdot T1 \cdot T2 \cdot W \\
\quad\quad T1 = 0.8 + 0.02 \cdot topt - 0.0005 \cdot topt^2 \\
\quad\quad T2 = 1.1814/((1 + e^{0.2 \cdot (TDIFF - 10)}) \cdot (1 + e^{0.3 \cdot (-TDIFF - 10)})) \\
\quad\quad\quad TDIFF = topt - tempc \\
\quad\quad W = 0.5 + 0.5 \cdot eet/PET \\
\quad\quad\quad PET = 1.6 \cdot (10 \cdot \max(tempc, 0)/ahi)^A \cdot pet\_tw\_m \\
\quad\quad\quad\quad A = 0.000000675 \cdot ahi^3 - 0.0000771 \cdot ahi^2 + 0.01792 \cdot ahi + 0.49239 \\
\quad IPAR = FPAR\_FAS \cdot monthly\_solar \cdot SOL\_CONV \cdot 0.5 \\
\quad\quad FPAR\_FAS = \min((SR\_FAS - 1.08)/srdiff, 0.95) \\
\quad\quad\quad SR\_FAS = (1 + fas\_ndvi/1000)/(1 - fas\_ndvi/1000) \\
\quad\quad SOL\_CONV = 0.0864 \cdot days\_per\_month
\end{array}
$$

A set of equations defining a target variable through some intermediate variables can easily be turned into a grammar, as demonstrated in Tables 2 and 3, which give an earth-science model and a grammar that derives this model only. Having the grammar in Table 3, however, enables us to specify alternative models through providing additional productions for the nonterminal symbols in the grammar. Additional productions for intermediate variables would specify alternative choices, only one of which will eventually be chosen for the final model. Observational data would be then used to select among combinations of such choices, if we apply a grammar based equation discovery system (such as Lagramge) with the grammar that includes additional productions to observational data as input.

While the presented approach from the previous paragraph does take into account the initial model, it may allow for a completely different model to be

Table 3. Grammar derived from the equations for NPPc variable in the CASA earth-science model in Table 2. The grammar generates the original equations only.

NPPc ->	max(const[0:0], E * IPAR)
E ->	const[0.389:0.389] * T1 * T2 * W
T1 ->	const[0.8:0.8] + const[0.02:0.02] * topt
	- const[0.0005:0.0005] * topt * topt
T2 ->	const[1.1814:1.1814] / ((const[1:1] + exp(const[0.2:0.2]
	* (TDIFF - const[10:10]))) * (const[1:1]
	+ exp(const[0.3:0.3] * (-TDIFF - const[10:10]))))
TDIFF ->	topt - tempc
W ->	const[0.5:0.5] + const[0.5:0.5] * eet / max(PET, const[0:0])
PET ->	const[1.6:1.6]
	* pow(const[10:10] * max(tempc, const[0:0]) / ahi, A)
	* pet_tw_m
A ->	const[0.000000675:0.000000675] * ahi * ahi * ahi
	- const[0.0000771:0.0000771] * ahi * ahi
	+ const[0.01792:0.01792] * ahi + const[0.49239:0.49239]
IPAR ->	FPAR_FAS * solar * SOL_CONV * const[0.5:0.5]
FPAR_FAS ->	min((SR_FAS - const[1.08:1.08]) / srdiff, const[0.95:0.95])
SR_FAS ->	(const[1:1] + fas_ndvi / const[1000:1000])
	/ (const[1:1] - fas_ndvi / const[1000:1000])
SOL_CONV ->	const[0.0864:0.0864] * days_per_month

derived, depending on whether productions for alternative definitions are provided for each of the intermediate variables. It is here that the minimal revision/change principle comes into play: among theories of similar quality (fit to the data), theories that are closer to the original theory are to be preferred. Since we are dealing with theories that are not necessarily expressed in logic (e.g., equations), only syntactic criteria of minimality of change are applicable in a straightforward fashion.

4.3 Typical Alternative Productions

Note that when an alternative production is specified for an intermediate variable, there are no restrictions (at least in principle) on these productions. For example, they can introduce new intermediate variables and productions defining them. They can also specify arbitrary functional forms (in the case of equations). However, they do have to eventually derive (in the context of the entire grammar) valid sub-expressions involving the set of terminal symbols (system variables) associated to the initial model.

A very common alternative production would replace the particular constants on the right-hand-side with generic constants, allowing the equation discovery system to re-fit them to the given observational data. In the grammar from Table 3 that change can be achieved by replacing a terminal symbol of the form `const[v:v]`, denoting a constant parameter with fixed value `v`, with a generic symbol `const` that allows for an arbitrary value of the particular constant parameter. In our experiments with the earth-science CASA model we allow for a 100% change of the original values of the constant parameters in the initial model. This can be specified by replacing the terminal symbol `const[v:v]` with `const[0:2*v]`, where interval $[0 : 2 \cdot v]$ is equal to $[(v - 100\% \cdot v) : (v + 100\% \cdot v)]$ (a 100% relative change).

A slightly more complex alternative production would replace a particular polynomial on the right-hand-side of a production with an arbitrary polynomial of the same (intermediate) variables. For example, in the grammar from Table 3 can be replaced by a grammar, similar to the example grammar from Table 1, for generating an arbitrary polynomial of the variable *topt*.

4.4 Current Implementation

Our current implementation of the theory revision approach to equation discovery outlined above involves applying LAGRAMGE to the given observational data and a grammar specifying the possible alternative productions to be used in theory revision. The observational data are used to select a particular combination of the possible alternatives: note that these also include leaving parts of the model unchanged (as the original productions are also a part of the grammar) even if alternative productions for these exist.

We currently do not have an implementation of the minimal change preference integrated within LAGRAMGE. This however, can be achieved in a relatively straightforward manner. One of the heuristic functions used by LAGRAMGE to search the space of equations, called MDL, takes into account the degree-of-fit (sum of square errors) as well as the size of the equation model. A reasonable approach to implement a minimality of change principle would be to replace the second term in the MDL heuristic: replace the size of the equation with a distance between the current model and the initial model. The distance measure can be a distance on tree-structured terms, which would take into account the number and complexity of the alternative productions taken to derive the current equation.

5 Experiments in Revising an Earth-Science Model

We illustrate the use of the proposed framework for theory revision in equation discovery on the problem of revising one part of the earth-science CASA model [4]. The CASA model predicts annual global fluxes in trace gas production on the basis of a number of measured (observed) variables, such as surface temperatures, satellite observations of the land surface, soil properties, etc. Because the whole CASA model is a quite complex system of difference and algebraic equations, we focused on the revision of the NPPc part of CASA (CASA-NPPc), presented in Table 2, that is used to predict the monthly net production of carbon at a given location.

The values of the input variables (terminal symbols in the grammar from Table 2) were measured (and/or calculated) for 303 locations on the Earth providing a data set with 303 examples. In order to evaluate the accuracy of the model on unseen data we applied standard ten-fold leave-one-out cross validation method. The error of the original and revised models was calculated as root mean squared error defined as $\sqrt{\sum_{i=1}^{N}(NPPc_i - \hat{NPPc})_i^2/N}$, where N is number of the data points; $NPPc_i$ and $\hat{NPPc}_i$ are the observed value and the value calculated by the model, respectively.

5.1 Revisions Used in the Experiments

As described in Section 4 we first transformed the given NPPc model into a grammar (given in Table 3) that derives that model only. Furthermore, we added alternative productions to the grammar that define the space of possible revisions. We used six alternative possibilities for the revision of the NPPc model, described below.

E-c-100 : we allowed a 100% relative change of the constant parameter 0.389 in the equation defining the intermediate variable E. Therefore, we replaced the original production for nonterminal symbol `E` in the grammar with `E -> const[0:0.778] * T1 * T2 * W`, i.e., changed the constraint on the value of the constant parameter from the original `const[0.389:0.389]`, which fixes the value of the constant parameter, to `const[0:0.778]`, which allows a 100% relative change of the original value of the constant parameter ($[0 : 0.778]$ being equal to $[(0.389 - 100\% \cdot 0.389) : (0.389 + 100\% \cdot 0.389)]$).

T1-c-100, T2-c-100 : we allowed the same revisions as the one described above on the right hand sides of the productions for `T1` and `T2`.

SR_FAS-c-20 : we allowed 20% relative change of the constant parameters values in the equation defining the intermediate variable SR_FAS. The relative change of 20% was used to avoid values of the constant parameters lower than 800, which would cause singularity (division by zero) problems in the formula for calculating SR_FAS.

T1-s : we allowed the original second degree polynomial for calculation of $T1 = 0.8 + 0.02 \cdot topt - 0.0005 \cdot topt^2$ with an arbitrary polynomial of the same variable *topt*. The following alternative productions were added to the grammar from Table 3 for this purpose: `T1 -> const` and `T1 -> const + (T1) * topt`.

T2-s : the graph of the dependency between the *T2* and *TDIFF* variables shows a Gaussian-like slightly asymmetrical dependency curve. Following the fact that this kind of dependency can be approximated also with a higher degree polynomial we replaced the original `T1` production in the grammar from Table 3 with two productions (similar to the ones for **T1-s**, presented above) that define an arbitrary polynomial of the *TDIFF* variable.

In addition to these six possibilities for revising the CASA-NPPc model we also used different combinations of them.

5.2 Results of the Experiments

The results of the experiments with different alternative grammars for revision are presented in Table 4.

Table 4. Error reduction (in %) gained with revising the original CASA-NPPc model using different grammars for revision.

Grammar	Reduction of RMSE (in %)
SR_FAS-c-20	14.93
T2-c-100	13.25
T1-s	13.05
T2-s	12.90
E-c-100	12.59
T1-c-100	12.39
SR_FAS-c-20 + T2-s	15.56
SR_FAS-c-20 + T1-s	15.46
T2-c-100 + T1-s	13.92
T2-s + T1-s	13.30
SR_FAS-c-20 + T2-c-100	11.55
SR_FAS-c-20 + T2-s + T1-s + E-c-100	**16.19**
SR_FAS-c-20 + T2-s + T1-c-100 + E-c-100	15.44
SR_FAS-c-20 + T2-c-100 + T1-s + E-c-100	14.82
SR_FAS-c-20 + T2-c-100 + T1-c-100 + E-c-100	12.92

The first six rows of Table 4 shows that revising the value of the constant parameters in the equation for calculating *SR_FAS* gives the greatest improvement of the original model. The original value of the parameters (equal to 1000) defines an almost linear dependence of *SR_FAS* on observed variable *srdiff*. The revised values of the constant parameters were equal to 800 (lowest possible values), which increase the non-linearity of the dependence. Allowing lower values of the parameters in the equation gives further improvement, but singularity (division by zero) problems appear due to the range of the *srdiff* variable.

The analysis of the results of the structural revisions shows the following. **T1-s** revision cause the second-degree polynomial for calculating the *T1* variable to be replaced by a fourth degree polynomial. On the other hand, the structural revision **T2-s** reduced the complex formula for calculating *T2* with a constant value. This is a surprising result that would have to be discussed with the Earth science experts that built the CASA model.

Furthermore, we tested pairwise combinations of the six model refinements. The results are presented in the second part of the Table 4. Results show that improvements gained using individual refinement grammars do not combine additively. However, combinations do increase the improvements: maximal improvement gained with pairwise combinations is 15.56% compared with the highest improvement of 14.93% gained using individual revisions.

Finally, the results of the experiments with combining all the refinements are presented in the last four rows of Table 4. Note however, that revisions of the T1 and T2 structures (T1-s and T2-s) are mutually exclusive with the respective revisions of the T1 and T2 constants (T1-c-100 and T2-c-100). Therefore, four possible combinations are possible, the one combining the structural revisions of the *T1* and *T2* formulas and revisions of the values of the constant parameters in formulas for the *SR_FAS* and E gives the maximal improvement of the accuracy of 16.19%.

In sum, the presented results of the experiments show that small revisions of the CASA-NPPc model parameters and structure considerably improve the accuracy of the model, the maximal improvement being above 16%. However, Earth science experts should also evaluate the comprehensibility and acceptability of the revised models. Nevertheless, if some of the revisions generate models that do not make sense from their point of view, new alternative productions would have to be defined to reflect the experts comments, and allow only revisions that lead to acceptable models.

Note here that the most of the error reduction is gained using a fairly simple revision operator of changing the values of the constant parameters in the *SR_FAS* equation. Only minor additional reductions can be obtained by combining this revision with any of the other five revision operators described above. Therefore, this revision would probably be the optimal one from the point of view of the minimality of change criterion, discussed in Section 4.

6 Conclusions and Discussion

We have presented a general framework for the revision of theories in the form of (sets of) quantitative equations. The method is based on grammars, which can be derived from the original theory. Domain experts can focus the revision process on parts of the model and guide it by providing relevant alternative productions. In this way, the revision process can be interactive, as is quite often the case when revising theories expressed in logic.

We have applied our approach to the problem of revising an existing equation based model of the net production of carbon in the Earth ecosystem. Experimental results show that small revisions in both the values of the constant parameters and the structure of equations considerably reduce the error of the model by 16%.

Saito et al. [5] address the same task of revising scientific models in the form of equations. Their approach is based on transforming parts of the model into a neural network, training the neural network, then transforming the trained network back into an expression/equation. This indirect approach is limited to revising the parameters or form of one equation in the model at a time. It also requires some handcrafting to encode the equations as a neural network – the authors state that “the need to to translate the existing CASA model into a declarative form that our discovery system can manipulate” is a challenge to their approach.

Our approach allows for a straightforward representation of existing scientific models as grammars, which can then be directly manipulated and used to perform theory revision. The transition from the initial model to a grammar is so straightforward that we consider automating this process as one of the topics for immediate further work. Revisions to several equations of the original model may be considered simultaneously, as illustrated by the experiments performed.

Whigham and Recknagel [8] also consider the specific task of revising an existing model for predicting chlorophyll-a by using measured data. They use a genetic algorithm to calibrate the equation parameters. They also use a grammar based genetic programming approach to revise the structure of two sub-parts (one at a time) of the initial model. A most general grammar that can derive an arbitrary expression using the allowed arithmetic operators and functions was used for each of the two sub-parts.

Unlike this paper, Whigham and Recknagel [8] do not present a general framework for the revision of quantitative scientific models. Their approach is similar to ours in that they use grammars to specify possible revisions. However, the grammars they use are too general to provide much information about the domain at hand. Also, they do not consider the notion of minimality of revision and genetic programming typically produces very large expressions without a simplicity bias.

As already mentioned, an immediate topic for further work is to automate the grammar generation from the initial model. Another challenge is to provide the domain experts an interactive tool for testing out different alternatives for revision. Furthermore, integrating the minimality of change criterion in LAGRAMGE is also an open issue. Minimal description length (MDL) heuristics in LAGRAMGE can be adapted to take into account the distance between the current and the initial equation model. Finally, we plan to apply the proposed framework to the task of revision of other portions of the CASA model as well as revision of other equation based environmental models.

Acknowledgments

We thank Christopher Potter, Steven Klooster and Alicia Torregrosa from NASA-Ames Research Center for making available both the CASA model and the relevant data set.

References

1. P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Żythow. *Scientific Discovery.* MIT Press, Cambridge, MA, 1987.
2. N. Lavrac and Sašo Džeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, Chichester, 1994. Freely available at `http://www-ai.ijs.si/SasoDzeroski/ILPBook/`.
3. D. Ourston and R. J. Mooney. Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, 66:273–309, 1994.

4. C. S. Potter and S.A. Klooster. Interannual variability in soil trace gas (CO_2, N_2O, NO) fluxes and analysis of controllers on regional to global scales. *Global Biogeochemical Cycles*, 12:621–635, 1998.
5. K. Saito, P. Langley, and T. Grenager. The computational revision of quantitative scientific models. 2001. Submitted to Discovery Science conference.
6. L. Todorovski and S. Džeroski. Declarative bias in equation discovery. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 376–384, Nashville, MA, 1997. Morgan Kaufmann.
7. T. Washio and H. Motoda. Discovering admissible models of complex systems based on scale-types and identity constraints. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, volume 2, pages 810–817, Nogoya, Japan, 1997. Morgan Kaufmann.
8. P. A. Whigham and F. Recknagel. Predicting chlorophyll-a in freshwater lakes by hybridising process-based models and genetic algorithms. In *Book of Abstracts of the Second International Conference on Applications of Machine Learning to Ecological Modeling*. Adelaide University, 2000.
9. S. Wrobel. First order theory refinement. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 14–33. IOS Press, 1996.
10. R. Zembowicz and J. M. Żytkow. Discovery of equations: Experimental evaluation of convergence. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 70–75, San Jose, CA, 1992. Morgan Kaufmann.