Revision of equation based models

Ljupčo Todorovski LJUPCO.TODOROVSKI@IJS.SI
Sašo Džeroski SASO.DZEROSKI@IJS.SI
Department of Intelligent Systems, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Abstract

State of the art equation discovery systems start the discovery process from scratch, rather than from a given initial equation. On the other hand, theory revision systems start from a given initial theory 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 revision of equation based models. We propose a theory revision method that allows for trade-off between the goodness of fit of the revised model and its similarity to the initial one. 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 relatively simple revisions of the parameters and structure of the initial model considerably improve its accuracy.

1. Introduction

An important type of domain specific modeling knowledge that is neglected by existing equation discovery methods [1, 9, 11] are the existing models already established in the domain at hand. Rather than starting the search with an existing equation based model, current equation discovery methods always start their search from scratch. In contrast with them, theory revision methods [3, 13] start with an existing theory and use heuristic search to revise the theory in order to improve its fit to a set of newly acquired observational data. However, theory revision research is mainly concerned with the revision of theories expressed in propositional or first-order logic. Therefore, the developed methods are not directly applicable to the task of revision of models based on equations.

In this paper, we propose a flexible, grammar based, equation discovery method for revision of equation based models. First, the given existing model is transformed into an initial grammar that can be used to derive the initial given model only. The nonterminals in the grammar and their productions reflect the structure of the initial model. Next, the initial grammar is extended with alternative productions that specify the possible modeling alternatives. The modeling alternatives can be specified by a domain expert. The extended grammar built in this manner specifies the space of possible revisions of the initial model. Therefore, in the last step, the equation discovery method LAGRAMGE [9] is applied to search through the space of possible revisions and find the one that fits the newly measured data best.

Theory revision methods follow the minimal revision principle: among theories of similar goodness of fit to the data, the ones that are closer to the original theory are to be preferred. In order to incorporate this principle in our method, we modify the MDL heuristic function used in LAGRAMGE that introduces preference toward simpler equations in the process of equation discovery [9]. The MDL heuristics takes into account complexity of an equation along with its goodness of fit to the data. We replace the complexity of an equation based model in the MDL heuristic with the distance of the model from the initial one. For measuring this distance, we use a standard measure of distance between tree-structured terms [8].

The proposed method for revising equation based models was applied to the task of revising one part of the CASA model of the net production of carbon by terrestrial plants in the Earth ecosystem [4]. Experimental results show that the method finds revisions that considerably reduce the error of the initial CASA model on the newly observed data.

The paper is organized as follows. The following Section 2 gives a brief introduction to grammar based equation discovery. Section 3 defines the problem of revising equation based models. Section 4 describes

the transformation of the given initial model into a grammar, while the process of adding modeling alternatives to the initial grammar is described in Section 5. Introducing the minimality of change principle in the process of revising equation based models is the topic of Section 6. The experimental methodology used for evaluating of the approach, as well as the experimental results are presented in Section 6.1. Finally, Section 7 summarizes the paper and discusses related research.

2. Equation Discovery

Equation discovery is the area of machine learning that develop methods for automated discovery of quantitative laws, expressed in the form of equations, in collections of measured data [1]. Equation discovery methods 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 methods explore different spaces of possible equations. Early equation discovery methods used pre-defined (built-in) spaces that were small enough to allow efficient heuristic (or exhaustive) search. EF method [14] searches the pre-defined space of polynomials, but allow the user to limit the polynomial degree as well as a set of functions that can be used to introduce new variables that can appear in the polynomials. Furthermore, equation discovery method SDS [11] uses user provided scale-type information about the dimensions of the system variables and is capable of discovering fairly complex equations from noisy data.

Experts from a specific domain of interest can usually provide much more modeling knowledge about the domain at hand than merely enumerating the measurement units used for measuring the variables of the observed system. In order to incorporate other types of knowledge in the process of equation discovery, we should provide the user with a more sophisticated declarative bias mechanisms. Equation discovery method LAGRAMGE [9] allows the user to specify the space of possible equations using a context free grammar. Grammars are a more general and powerful formalism that allows for tailoring the space of the equations to the domain of use than the ones used in SDS [11] and EF [14]. 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 V of the observed system variables, a target variable $v_d \in V$, a table M of their measured values, and a grammar G, **find** an equation based model M_E the target variable v_d that can be derived by the grammar G and 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 a single nonterminal symbol `T1`, a set of two productions attached to it, and a set of two terminal symbols {`topt, const[-10:0:10]`}. The semantics of the terminal and nonterminal symbols in the grammar are explained below.

Table 1. An example of a grammar for equation discovery that defines the space of polynomials of a single variable *topt*.

```
T1 -> const[-10:0:10]
T1 -> const[-10:0:10] + (T1) * topt
```

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 (`topt` in the example grammar from Table 1). Another group of terminal symbols of the form `const[l:i: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:i:h]` specifies that the value v of the constant parameter should be within the interval $l \leq v \leq h$ and its initial value should be i.

The nonterminal symbol `T1` is used to generate polynomial of arbitrary degree. First production generates zero degree polynomial, i.e., constant parameter only. Using the second production first and then second production generates a first degree polynomial, i.e., `T1 -> const[-10:0:10] + (T1) * topt -> const[-10:0:10] + (const[-10:0:10]) * topt`. Finally, by using the second production more then once an arbitrary degree polynomial can be generated.

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 [5]. After fitting the values of the constant parameters tho model is evaluated according to the sum of squared errors (SSE heuristic function

[9]), 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 [9].

3. Problem Definition

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 method 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 [13].

In analogy with theory revision, the problem of revising equation based models can be defined as follows: **Given**

- an imperfect existing model M_I of the observed system, expressed in form of equations and
- a set of observations or measurements of the system variables of the observed system,

find a revised model M_R that

- minimizes the discrepancy between the observed values of the system variable and the values of system variables obtained with simulating the model and
- differs from the initial model M_I as little as possible.

Note that the definition of the problem of revising equation based models is very similar to the definition of the theory revision problem. However, the possible changes to the initial equation based model would very much differ than the possible changes of an initial logical theory. As theories are typically logical theories in theory revision settings, the changes typically include the addition and deletion of entire rules (propositional or first-order Horn clauses), as well as the addition and deletion of conditions in individual rules. In the continuation of the paper, we will propose a framework for specifying plausible changes for the problem of revising equation based models.

4. Transforming the Initial Model Into a Grammar

In a typical setting of revising an existing equation based model, we would only have observational data and the 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 equations are complex, the model is rarely written as a single equation defining the target variable. Much more often it is written as a set of equations defining the target variable, which also contains equations defining intermediate unobserved variables. The latter define meaningful concepts in the domain of interest. Often, alternative equations defining an intermediate variable would be possible and the modeling scientist would choose one of these. The alternative equations 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.

An example of such a complex equation based model (CASA-NPPc) is given in Table 2. The model is one portion of the CASA earth-science model of the global production and absorption of biogenic trace gases in the Earth atmosphere. Further details about the model can be found latter in Section 6.1. The model defines the value of the *NPPc* variable (that denotes the net primary production of carbon) in terms of a set of observed system variables, such as *topt* and *tempc*. Lower case variable names are used to denote observable variables (with the exception of the dependent variable *NPPc*). The remaining variables are unobservable and must be computed from others using their defining equations.

A set of equations defining a target variable through some intermediate variables can easily be turned into a grammar presented in Table 3. The starting symbol of this grammar represents the dependent variable *NPPc*, the nonterminal symbols represent the intermediate variables, while the terminal symbols are used to denote the observed system variables and the constant parameters of the model. Each nonterminal symbol in

Table 2. The CASA-NPPc model consists of a portion of the CASA model defining the NPPc intermediate variable.

$$
\begin{array}{l}
NPPc = \max(0, E \cdot IPAR) \\
\quad E = 0.56 \cdot T1 \cdot T2 \cdot W \\
\quad\quad T1 = 0.8 + 0.02 \cdot topt - 0.0005 \cdot topt^2 \\
\quad\quad T2 = 1.1814/((1 + \exp(0.2 \cdot (TDIFF - 10))) \cdot (1 + \exp(0.3 \cdot (-TDIFF - 10)))) \\
\quad\quad\quad TDIFF = topt - tempc \\
\quad\quad W = 0.5 + 0.5 \cdot eet/PET \\
\quad\quad\quad PET = 1.6 \cdot (10 \cdot \max(tempc, 0)/ahi)^A \cdot pet\_tw\_m \\
\quad\quad\quad\quad A = 0.000000675 \cdot ahi^3 - 0.0000771 \cdot ahi^2 + 0.01792 \cdot ahi + 0.49239 \\
\quad IPAR = FPAR\_FAS \cdot monthly\_solar \cdot SOL\_CONV \cdot 0.5 \\
\quad\quad FPAR\_FAS = \min((SR\_FAS - 1.08)/srdiff, 0.95) \\
\quad\quad\quad SR\_FAS = (1 + fas\_ndvi/1000)/(1 - fas\_ndvi/1000) \\
\quad\quad SOL\_CONV = 0.0864 \cdot days\_per\_month
\end{array}
$$

the grammar has a single production that generates the model equation used to calculate the respective intermediate variable. Therefore, the grammar in Table 3 generates a single equation based model that is equivalent to the one from Table 2.

Note, however, that grammar in Table 3 enables us to specify an arbitrary number of alternative models for each intermediate variable through providing additional productions for the nonterminal symbols in the grammar. These additional productions would specify alternative modeling choices, only one of which will eventually be chosen for the final (revised) model. Observational data would be then used to select among combinations of such choices, if we apply a grammar based equation discovery system LAGRAMGE with the grammar that includes additional productions and observational data as input.

5. Extending the initial grammar with alternative productions

Note that when alternative productions are 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. However, they do have to eventually derive (in the context of the entire grammar) valid sub-expressions involving the set of terminal symbols that represent observed system variables.

A very common alternative production would replace the particular constant parameter value on the right hand side of an existing production with a generic unspecified constant parameter, allowing the equation discovery system to re-fit them to the given observational data. The change can be achieved by replacing a terminal symbol that denotes a fixed value constant parameter with the generic symbol `const` that allows for an arbitrary value of the constant parameter. In our experiments with the CASA-NPPc model, we use alternative productions that allow for a 100% relative change of the initial value of a constant parameter. This can be specified by replacing the fixed value constant parameter `v` with a terminal symbol `const[0:v:2 · v]`. Thus, the lower bound for the newly introduced constant parameter is set to $v - 100\% \cdot v = 0$, while the upper bound is set to $v + 100\% \cdot v = 2 \cdot v$. The default value of the constant parameter is the same as its initial value, i.e. it is set to v.

Slightly more complex alternative productions would allow for replacing a particular polynomial on the right hand side of a production with an arbitrary polynomial of the same (intermediate) variable(s). An example of such alternative productions for the nonterminal symbol `T1` from the grammar in Table 3 is given in Table 1. These productions can be used to generate an arbitrary polynomial of the system variable *topt*.

This grammar based framework allows human experts to point out what are the parts of the model they are completely confident in. These parts should be left intact in the revision process, i.e., no alternative productions should be specified for the respective nonterminal symbols. For example, earth science experts that built the CASA model pointed out what are the "weak" parts of the NPPc portion of the CASA model. They pointed out four intermediate variables, for which they are not very confident in the equations used to calculate their values. Therefore, these are the variables for which alternative productions should be added to the initial grammar.

6. Minimality of change principle

While the approach presented above does take into account the initial model, it may allow for a completely different model to be derived, depending on whether and what kind of productions for alternative models are provided for each of the intermediate variables. It is here that the minimal revision/change principle

Table 3. A grammar derived from the CASA-NPPc model in Table 2. The grammar generates the original CASA-NPPc model only.

```
NPPc ->       max(0, E * IPAR)
E ->          0.56 * T1 * T2 * W
T1 ->         0.8 + 0.02 * topt - 0.0005 * topt * topt
T2 ->         1.1814 / ((1 + exp(0.2 * (TDIFF-10))) * (1 + exp(0.3 * (-TDIFF-10))))
TDIFF ->      topt - tempc
W ->          0.5 + 0.5 * eet / max(PET, 0)
PET ->        1.6 * pow(10 * max(tempc, 0) / ahi, A) * pet_tw_m
A ->          0.000000675*ahi*ahi*ahi - 0.0000771*ahi*ahi + 0.01792*ahi + 0.49239
IPAR ->       FPAR_FAS * solar * SOL_CONV * 0.5
FPAR_FAS ->   min((SR_FAS - 1.08) / srdiff, 0.95)
SR_FAS ->     (1 + fas_ndvi / 1000) / (1 - fas_ndvi / 1000)
SOL_CONV ->   0.0864 * days_per_month
```

comes into play: among theories of similar quality (fit to the data), theories that are closer to the original theory are to be preferred.

The crucial concept that is necessary in order to implement the minimality of change principle is the measure of change or distance between the (potential) revised model and the initial model. Since parse trees are used in LAGRAMGE to represent models, we use a measure of distance between tree structured terms as a measure of distance between models. Thus, the distance measure we use assesses syntactic structural distance, i.e., the amount of change in the structure of the equations of the model.

A common approach to computing distances between strings or tree structured terms is the *editing* approach, leading to *edit distance measure*. Following the editing approach, a set of basic edit operations is first defined. The edit operations available for editing trees are relabeling (changing the label), deleting and inserting a node in the tree. Costs are assigned to these operations, depending on the labels of the nodes involved. The problem of computing the distance between two tree structured terms $\mathcal{T}_1$ and $\mathcal{T}_2$ is then transformed to the problem of finding a minimal cost sequence of basic editing operations that transforms a tree $\mathcal{T}_1$ into a tree $\mathcal{T}_2$.

This problem is $\mathcal{NP}$-complete for the case of unordered tree structures, i.e., structures where the left-to-right order of the children of a node is unimportant. In our case, we are dealing with parse trees which are ordered, since the left-to-right order of the children is important and determined by the production applied to the nonterminal in the node. The distance between ordered tree structures can be efficiently computed. An overview of algorithms that can be used for computing an edit distance between ordered tree structures is given in [8]. The computation of distances between parse trees can be even more efficient, as illustrated in [6].

Therefore, for the purpose of calculating the edit distance between equation based models (or more precisely their parse trees), we use the algorithm proposed in [6]. The costs of the basic edit operations are defined as follows:

Deleting a node has a cost of 1.

Inserting a node has a cost of 1.

Relabeling a node has a cost of 1, if the label is actually changed, or 0 otherwise. Note that for nonterminal symbols that denote constant parameters, the actual value of the constant parameter is used a label.

Once we have defined a distance measure between models, we can incorporate it into LAGRAMGE by modifying the MDL heuristic function used in LAGRAMGE to introduce preference toward simpler equations. The MDL heuristics takes into account the complexity of an equation along with its goodness of fit to the data [9], i.e.,

$$\mathrm{MDL}(E) = \mathrm{SSE}(E) + \frac{l}{10 \cdot l_{\max}} \cdot \mathrm{SSE}(E_0),$$

where $\mathrm{MDL}(E)$ is the sum of squared errors of the current model on the training data, $\mathrm{SSE}(E_0)$ is the error of the simplest model, l is the length of the current model (in number of terminal symbols) and $l_{\max}$ the length of the most complex equation is the search space. Since the LAGRAMGE search space consists of parse trees with limited depth, the maximal length $l_{\max}$ can be easily computed in advance. Roughly speaking, the second part of the MDL heuristic function of LAGRAMGE adds a penalty for equation complexity to the sum of squared errors.

By analogy to the MDL heuristic, we can define MC (minimality of change) heuristic function as follows:

$$\mathrm{MC}(E) = \mathrm{SSE}(E) + \frac{\mathrm{distance}(E, E_0)}{C} \cdot \mathrm{SSE}(E_0),$$

where $\mathrm{distance}(E, E_0)$ is the distance between the current model E and the initial model E_0. Note that the maximal distance is not available as in the case of maximal length for MDL, so we introduce a user defined parameter C. This parameter can be used to trade-off between goodness of fit of the current model and minimality of change with respect to the initial model. Large values of C will diminish the "change penalty" term of the MC heuristic, leading to a preference toward accurate models, not necessarily similar to the initial one. On the other hand, small values of C will increase the "change penalty" term, leading to a preference toward models that are similar to the initial model E_0.

6.1 Revising the CASA Earth-Science Model

Data from the latest generation of satellites, combined with readings from ground sources, hold great promise for testing and improving on existing scientific models of the Earth's biosphere. One such model, CASA, developed by Potter and Klooster [4] at NASA Ames, accounts for the global production and absorption of biogenic trace gases in the Earth atmosphere, as well as predicting changes in the geographic patterns of major vegetation types (e.g., grasslands, forest, tundra, and desert) on the land.

CASA predicts, with reasonable accuracy, annual global fluxes in trace gas production as a function of surface temperature, moisture levels, and soil properties, together with global satellite observations of the land surface. The model incorporates difference equations that represent the terrestrial carbon cycle, as well as processes that mineralize nitrogen and control vegetation type. These equations describe relations among quantitative variables and lead to changes in the modeled outputs over time. CASA operates on gridded input at different levels of resolution, but typical usage involves grid cells that are eight kilometers square, which matches the resolution for satellite observations of the land surface.

Because the overall CASA model is quite complex, involving many variables and equations, we decided to focus on one portion that lies on the model's 'fringes' and that does not involve any difference equations. As Table 2 indicates, the model predicts this quantity as the product of two unobservable variables, the photosynthetic efficiency, E, at a site and the solar energy intercepted, $IPAR$, at that site.

Photosynthetic efficiency is in turn calculated as the product of the maximum efficiency (0.56) and three stress factors that reduce this efficiency. One stress term, *T2*, takes into account the difference between the optimum temperature, *topt*, and actual temperature, *tempc*, for a site. The second factor, *T1*, involves the nearness of *topt* to a global optimum for all sites. The third term, *W*, represents stress that results from lack of moisture as reflected by *eet*, the estimated water loss due to evaporation and transpiration, and *PET*, the water loss due to these processes given an unlimited water supply. In turn, *PET* is defined in terms of the annual heat index, *ahi*, for a site, and *pet_tw_m*, a modifier on *PET* to account for day length at differing locations and times of year.

The energy intercepted from the sun, *IPAR*, is computed as the product of *FPAR_FAS*, the fraction of energy absorbed photo-synthetically for a given vegetation type, *monthly_solar*, the average radiation for a given month, and *SOL_CONV*, the number of days in that month. *FPAR_FAS* is a function of *fas_ndvi*, which indicates overall greenness at a site as observed from space, and *srdiff*, an intrinsic property that takes on different numeric values for different vegetation types.

Of the variables we have mentioned, *NPPc*, *tempc*, *ahi*, *monthly_solar*, *SOL_CONV*, and *fas_ndvi*, are observable. Two additional terms, *eet* and *pet_tw_m*, are defined elsewhere in the model, but we assume their definitions are correct and thus we can treat them as observables. The remaining variables are unobservable and must be computed from the others using their definitions. This portion of the model also contains a number of numeric parameters, as shown in the equations in Table 2.

6.2 Experimental data and methodology

The training data set used in the experiments of CASA-NPPc model revision consists of 303 data points. Each data point contains measurements of the observed system variables for a distinct location on the Earth.

The quality of the revised models is assessed through the discrepancy between the predicted and observed values of the dependent variable. The smaller the discrepancy is, the better is the model. The discrepancy is measured using standard root mean squared error (RMSE) measure, calculated as: $\sqrt{\sum_{i=1}^{303}(NPPc_i - \hat{NPPc_i})^2/303}$, where $NPPc_i$ and

$\hat{NPPc_i}$ are the observed and the predicted value of *NPPc*, respectively. The RMSE of the initial model on the training data set is 465.214.

In order to estimate the error of the revised models on test data unseen during the process of revision, we applied standard 30-fold cross validation methodology. Following this methodology the data set consisting of 303 examples is randomly partitioned into 30 partitions, with approximately the same number of (10) examples in each of them. In each iteration of the cross validation procedure, twenty-nine out of thirty partitions are used as a training data set for revision of the initial model and the revised model is then used to predict the values of the dependent variable *NPPc* on the remaining partition, unseen during the revision phase. By repeating this iteration thirty times, once for each partition, we obtain 303 predictions of the *NPPc* value for all the data points in the training data set.

6.3 Grammar for the revision of the CASA-NPPc model

As described in Section 4, the given CASA-NPPc model was first transformed into the initial grammar presented in Table 3. In addition, alternative predictions were added to this initial grammar for the four intermediate variables that experts pointed out that they are not very confident in the equations used to calculate their values. Each of these alternative productions specifies one or more possible revisions of the initial CASA-NPPc model. The complete list of alternative productions added to the initial grammar is given in Table 4.

Alternative productions for E

`Ec-100` allows a revision of the constant parameter (with the initial value of 0.56) in the equation for the intermediate variable *E*. The alternative production allow for a 100% relative change of the initial value of the constant parameter.

`Es-exp` allows for a replacement of the three terms product from the initial *E* equation (i.e., the product $T1 \cdot T2 \cdot W$) with a product that allows for arbitrary exponents on these terms (i.e., product of the form $T1^{c_1} \cdot T2^{c_2} \cdot W^{c_3}$). The initial values of the exponents are set to 1, in that case the product is equivalent to the product in the initial *E* equation.

Alternative productions for T1

`T1c-100` allow for a 100% relative change of the initial values of the constant parameters in the *T1* equation.

`T1s-poly` allows for a replacement of the initial second degree polynomial that defines the value of *T1* with an arbitrary degree polynomial of the variable *topt*. In addition, the maximal depth of the parse trees considered by LAGRAMGE was set to allow the maximal polynomial degree of five.

Alternative productions for T2

`T2c-100` allow 100% relative change of the initial values of the constant parameters in the equation for `T2`.

`T2s-poly` allows for replacement of the initial equation that defines the value of *T2* with an arbitrary degree polynomial of the variable *TDIFF*. Again, the maximal degree of the polynomial was limited to five.

Alternative productions for SR_FAS

`SR_FASc-25` allows for a 25% relative change of the initial values of the constant parameters in the *SR_FAS* equation. The relative change of 25% was used to avoid values of the constant parameters lower than 750, which would cause singularity (division by zero) problems in the equation for *SR_FAS*.

Note, however, that arbitrary combination of these alternative productions can be added to the initial grammar. If all the alternative productions are added at the same time, then LAGRAMGE will find the most beneficial combination of revisions, i.e., the one that leads to the best revision of the initial model.

6.4 Experimental results

The results of the experiments with different modeling alternatives, presented above are summarized in Table 5.

When we allow only a single of the seven presented alternatives (the first seven rows of Table 5), revising the value of the structure of the *E* equation gives the largest reduction of the error of the initial CASA-NPPc model. This is also the most interesting structural revision proposing the following equation:

$$E = 0.610 \cdot T1^{2.83} \cdot T2^{0.638} \cdot W^{0}.$$

The proposed value of 0 for the exponent of the watter stress factor W suggests that the watter stress factor

Table 4. Alternative productions added to the initial grammar from Table 3. Each of them specifies one or more revisions of the initial CASA-NPPc model.

`Ec-100: E ->`	`const[0:0.56:1.12] * T1 * T2 * W`
`Es-exp: E ->`	`const[0:0.56:1.12] * pow(T1, const[0:1:]) * pow(T2, const[0:1:]) * pow(W, const[0:1:])`
`T1c-100: T1 ->`	`const[0:0.8:1.6] + const[0:0.02:0.04] * topt - const[0:0.0005:0.001] * topt * topt`
`T1s-poly: T1 ->`	`const | const + (T1) * topt`
`T2c-100: T2 ->`	`const[0:1.1814:2.3628] / ((1 + exp(const[0:0.2:0.4] * (TDIFF - const[0:10:20]))) * (1 + exp(const[0:0.3:0.6] * (-TDIFF - const[0:10:20])))`
`T2s-poly: T2 ->`	`const | const + (T2) * TDIFF`
`SR_FASc-25: SR_FAS ->`	`(1 + fas_ndvi / const[750:1000:1250]) / (1 - fas_ndvi / const[750:1000:1250])`

Table 5. The root squared mean error (RMSE) of the revised model, the percentage of relative error reduction (RER) of the RMSE of the revised model when compared to the RMSE of the initial CASA-NPPc model (with RMSE of 465.213) and distance (DIST) of the revised model from the initial one. The RMSE was estimated both on training data (training - the left-hand side of the table) and using 30-fold cross validation (CV - the right-hand side of the table).

alternative	training			CV		
production(s)	RMSE	RER (%)	DIST	RMSE	RER (%)	DIST
`Ec-100`	458.626	1.42	1	460.5	1.01	0.9
`Es-exp`	443.029	4.77	16	443.032	4.77	16.0
`T1c-100`	458.301	1.49	3	460.799	0.95	3.0
`T1s-poly`	450.265	3.21	46	457.37	1.69	45.8
`T2c-100`	457.018	1.76	3	459.633	1.20	3.0
`T2s-poly`	450.972	3.06	71	461.642	0.77	73.4
`SR_FASc-25`	453.157	2.59	2	455.281	2.13	2.0
All combined	414.739	**10.85**	104	423.684	**8.93**	67.4

is not important for predicting the photosynthetic efficiency E. Earth scientists proposed a possible explanation that the influence of the watter stress factor is already being captured by the satellite measurements of the relative greenness *fas_ndvi*.

The results of the experiments with searching for the optimal combination of all the alternative revisions are presented in the last row of Table 5. As expected, the optimal combination leads to the maximal relative reduction of the RMSE of more than 10% on training data and almost 9% when cross-validated. The combination of `Es-exp`, `T1c-100`, `T2s-poly`, and `SR_FASc-25` alternative productions leads to the best revised model. Note that the experimental results also show that the reductions obtained with allowing a single alternative production at a time, sum up, i.e., the error reduction obtained with the combination of alternative productions equals the sum of the error reductions obtained with individual ones.

Furthermore, in the next series of experiments, we explored the influence of the minimality of change principle on the revised models. For this purpose, we used the MC heuristic function (see Section 6) in LAGRAMGE with 7 different values of the C parameter: 32, 64, 128, 256, 512, 1024 and 2048. Recall from Section 6 that the C parameter is used to trade-off between goodness of fit of the model and minimality of change with respect to the initial model: smaller values lead to a higher preference toward models that are similar to the initial one.

The results of these experiments are summarized in Figure 1. As expected, the distance of the revised model from the initial model constantly increases as we increase the value of the C parameter. The distance is maximal when SSE heuristic function is used, i.e., the minimality of change principle is neglected. The trend of the relative reduction of the error of the initial model, estimated on training data, is the same: it constantly increases and reaches maximum when the SSE heuristic function is used. Thus, the more distant is revised model, the more accurate it is on training data.

The revised model that is most similar to the initial one (i.e., the one found using MC heuristic with $C = 32$) is obtained with revising the values of the constant parameters (`Ec-100` and `SR_FASc-25`) of the initial model, leading to an error reduction of 5.29%. This shows that the revisions of the initial equations

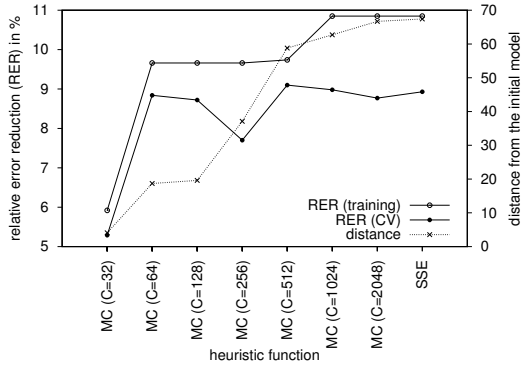

Figure 1. Relative error reduction and distance from the initial model for revised models obtained using SSE and MC (minimality of change) heuristic function with different values of the C parameter.

for E and especially *SR_FAS* are necessary and very important for the error reduction, even if we prefer a minimal change of the initial CASA-NPPc model. The analysis of the second revised model obtained with $C = 64$ gives further support for this claim. Namely, it leads to (almost maximal) error reduction of 8.84% with revising these two equations agian. However, in this case, more complex structural revision (`Es-exp`) of the initial E equation has been proposed.

The increasing trend of the performance of the revised models on training data can easily lead to overfitting, especially in cases when arbitrary revisions are allowed. Even in the experiments with a limited set of revision alternatives performed here, we can see that the cross-validated error reduction does not constantly increase. Therefore, models that are closer to the initial one can perform better on test data. In our experiments, the model obtained using the minimality of change heuristic with $C = 512$ (error reduction of 9.10%) slightly outperforms the model obtained using the SSE heuristic (error reduction of 8.93%), when cross-validated. The revised model, obtained using the MC heuristic with $C = 512$, leaves the *T2* equation unchanged and has a structure that is otherwise identical to the structure of the model obtained using the SSE heuristic (the values of the constant parameters in the equations are slightly different). This result shows that the revision of the *T2* equation is not really important for the reduction of the error of the initial CASA-NPPc model.

7. Discussion

In this paper, we propose a flexible grammar based equation discovery method for revision of equation based models. To support revision of existing equation based models with equation discovery, we use the transformation principle. First, the given existing model is transformed into an initial grammar that can be used to derive the initial given model only. Then, domain experts can than focus the revision process on parts of the model and guide it by providing relevant modeling alternatives that are added to the initial grammar as alternative productions. In this way, the revision process can be interactive, as is quite often the case when revising theories expressed in logic. The method also incorporate the minimality of change principle, in a way that allows the trade-off between the goodness of fit of the revised model and its similarity to the initial one.

Finally, note that the complexity of the model revision task is independent from the complexity of the initial model. Thus, the proposed approach scales up with the complexity of the initial model without problems. The complexity of the revision task depends only on the number of modeling alternatives provided by domain experts. In case of a very complex space of possible revisions, alternative non-exhaustive strategies (such as beam search strategy) for searching the space of possible revisions can be used in LAGRAMGE [9].

We have applied our approach to the real-world problem of revising a portion of an existing equation based model named CASA of the net production of carbon by terrestrial plants in the Earth ecosystem. Experimental results show that small revisions of both the values of the constant parameters and the structure of equations considerably reduce the error of the model by 9%. This improvement is regarded as a non-trivial by Earth scientists that developed the CASA model. The experiments also show the importance of the minimality of change principle in model revision from two aspects. First, the use of the minimality of change principle can slightly improve the accuracy of the revised model on test data, unused during the process of revision. Second, it can be used to identify what are the most important revisions that lead to the largest improvements of the accuracy of the initial model.

The research presented in the chapter is very related to several other lines of work, presented in brief below.

In [7], authors address the same task of revising models based on equations. Their approach is based on transforming a part of the model into a neural network, training the neural network, then transforming the trained network back into an expression/equation. They also obtained revised models with a considerably smaller error rate than the original one. Their method gained slightly lower improvement of the initial model

accuracy then our method. Also, their 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. Finally, the method presented in [7] do not incorporate the minimality of change principle in their approach.

The approach of transforming equation based models to neural networks and use these for refinement is similar in spirit to the KBANN approach proposed in [10]. There, an initial theory based on classification rules is first encoded as neural network. Then, the topology of the network is refined and the network is re-trained with the newly observed data. Finally, the network is transformed back into rules. However, the application of KBANN is limited to theories and models expressed as classification rules.

In [12], authors consider the 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 the work presented here, [12] does not present a general framework for the revision of equation based 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 incorporate the minimality of change principle in their approach. This can be considered as a very weak point of their approach, since genetic programming methods tend to produce very large expressions without any simplicity bias.

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. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, Chichester, 1994.

[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. Global model estimates of carbon and nitrogen storage in litter and soil pools: Response to change in vegetation quality and biomass allocation. *Tellus*, 49B:1–17, 1997.

[5] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterlin. *Numerical Recipes.* Cambridge University Press, Cambridge, MA, 1986.

[6] T. Richter. A new measure of the distance between ordered trees and its applications. Technical report, Department of Computer Science IV, University of Bonn, Bonn, Germany, 1997.

[7] K. Saito, P. Langley, T. Grenager, C. Potter, A. Torregrosa, and S. A. Klooster. The computational revision of quantitative scientific models. In *Proceedings of the Fourth International Conference on Discovery Science*, pages 336–349, Berlin, 2001. Springer.

[8] D. Shasha and K. Zhang. Approximate tree pattern matching. In *Pattern Matching Algorithms*, pages 341–371. Oxford University Press, 1997.

[9] L. Todorovski and S. Džeroski. Declarative bias in equation discovery. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 376–384, San Mateo, CA, 1997. Morgan Kaufmann.

[10] G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70:119–165, 1994.

[11] 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, San Mateo, CA, 1997. Morgan Kaufmann.

[12] 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.

[13] S. Wrobel. First order theory refinement. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 14–33. IOS Press, 1996.

[14] 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 Mateo, CA, 1992. Morgan Kaufmann.