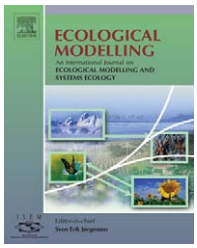
available at www.sciencedirect.com

SCIENCE DIRECT®

journal homepage: www.elsevier.com/locate/ecolmodel



Integrating knowledge-driven and data-driven approaches to modeling

*Ljupčo Todorovski, Sašo Džeroski**

Jožef Stefan Institute, Department of Knowledge Technologies, Jamova 39, SI-1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Available online 29 November 2005

Keywords:
Computational scientific discovery
Machine learning
Dynamic systems
Aquatic ecosystems
Hydrodynamics

ABSTRACT

In this paper, we present a framework for modeling dynamic systems that integrates the knowledge-based theoretical approach to modeling with the data-driven empirical modeling. The framework allows for integration of modeling knowledge specific to the domain of interest in the process of model induction from measured data. The knowledge is organized around the central notion of basic processes in the domain and it includes models thereof as well as guidelines for combining models of individual processes into a model of the entire observed system. The presented framework is applied to three tasks of modeling dynamic environmental systems from noisy measurement data in the domains of population and hydro dynamics. In all applications, the models induced with the framework can be used both to accurately predict and explain the behavior of the observed dynamic systems.

1. Introduction

Scientists and engineers build mathematical models in order to analyze and better understand the behavior of real world systems. Establishing an acceptable model of an observed system is a challenging task that occupies a major portion of the mathematical modeler's work. It involves observations and measurements of the system behavior under various conditions, selecting a set of variables that are important for modeling, and formulating the model itself. The first milestone in the process of modeling a real-world system is the choice of the modeling formalism. Ordinary differential equations (ODEs) are one of the most widely accepted formalisms for modeling of dynamic systems, i.e., systems that change their state over time (Gershenfeld, 1999). In this paper, we deal with inducing models of dynamics systems based on ODEs from observed behavior of the selected system variables.

There are two main aspects of formulating a model of a real-world system. First, an appropriate model (equations) structure has to be established (the structure identification problem). Second, acceptably accurate values for the parameters are to be determined (the parameter estimation problem). Methods for system identification (Ljung, 1993) focus mainly on solving the parameter estimation problem and make one of the following two assumptions. They either assume that the structure of the model is provided by a human expert, or assume that the structure is chosen from some general well-known class of model structures, such as linear equations, polynomials, or neural networks.

Formulating a model based on the assumption that the structure identification problem is solved by a human expert is known as the theoretical or knowledge-driven approach. Following this approach, the expert first identifies the processes that govern the behavior of the observed system. Then, using domain-specific knowledge about the identified processes, the expert writes down a proper structure of the model equations. In contrast to the theoretical approach, the empirical approach adopts a data-driven trial-and-error paradigm. The expert first chooses a general class of structures (such as linear or polynomial) that he/she believes to be adequate, fits its constant

* *Corresponding author*. Fax: +386 1 425 1038.
E-mail address: Saso.Dzeroski@ijs.si (S. Džeroski).
0304-3800/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.ecolmodel.2005.10.001

parameters, and checks how well the simulation of the model matches the observed data. If the match is not close enough, the procedure is repeated until an adequate model is found. A very limited portion (if any) of existing domain-specific knowledge is used in the modeling process.

In this paper, we aim at integrating the theoretical and empirical approaches to modeling. We present a modeling framework that integrates domain-specific knowledge in the process of data-driven induction of models based on ODEs. The knowledge is used to constrain the space of candidate models considered in the induction process. Before we can use the knowledge, we have to encode it—in the first part of the paper, we present a formalism for encoding knowledge in a form of taxonomy of basic processes and their models. The encoded knowledge includes also guidelines for composing models of the whole system from the models of the basic processes that govern its behavior. We show that this kind of knowledge, available in many textbooks on ecological modelling, can be efficiently transformed to grammars that specify the space of candidate models and can be used to guide the induction process. Our framework then uses the grammar-based equation discovery method Lagramge (Todorovski and Džeroski, 1997) to heuristically search through the space of candidate models, match them against data, and find the one that fits the data best. We illustrate the generality and usefulness of the framework by applying it to three tasks of modeling environmental dynamic systems from noisy measurement data.

The paper is organized as follows. We present the formalism for encoding domain-specific modeling knowledge in Section 2. Section 3 presents the automated modeling framework, which is then applied to three tasks of inducing models of environmental dynamic systems. We present the results of these applications in Section 4. Section 5 puts our work in the context of related research. Finally, Section 6 concludes the paper with a summary and directions for further research.

2. Encoding modeling knowledge

Models of dynamic systems are often stated in terms of *basic processes* that govern the behavior of the observed system. Each basic process influences the change of one or more variables of the observed system, while the model of a basic process specifies the equations used to model its influence. Our formalism for encoding domain-specific knowledge targets knowledge about what are the basic processes in the domain of interest as well as what models of their influence on the system behavior are typically used by domain experts. The encoded knowledge also includes guidelines for combining the models of the individual basic processes into a single model of the entire observed system.

We will illustrate the use of the formalism on the example domain of population dynamics (Murray, 1993). Population dynamics studies the structure and dynamics of populations, where each population is a group of individuals of the same species sharing a common environment. We consider models of the dynamic change of population concentrations (or densities) that take the form of ordinary differential equations.

2.1. *Taxonomy of basic process classes*

The knowledge about basic processes in the domain of interest is encoded in the taxonomy of process classes. Fig. 1 presents an example taxonomy of process classes in the population dynamics domain. At the highest level of the taxonomy, we distinguish between basic processes that involve a single species or an inorganic nutrient and processes that represent interactions between two or more species and/or nutrients. Down the taxonomy tree, the process classes become more specific. For example, there are two kinds of processes that involve a single species: growth of a population or its decay. The leaf nodes represent particular modeling alternatives typically used by domain experts to model individual processes, such as exponential or logistic growth.

Table 1 presents the formalization of several process classes from the population dynamics. Note that we present the complete formalization of the process classes from Fig. 1 in Appendix A. The first process class *Growth* represents processes of species growth in the absence of any interaction with other species and inorganic nutrients. It has two sub-classes, specifying two models of growth. The *Exponential_growth* class specifies unlimited exponential growth of the observed population. This kind of growth is inappropriate in many real-world cases, since the environment typically has a limited carrying

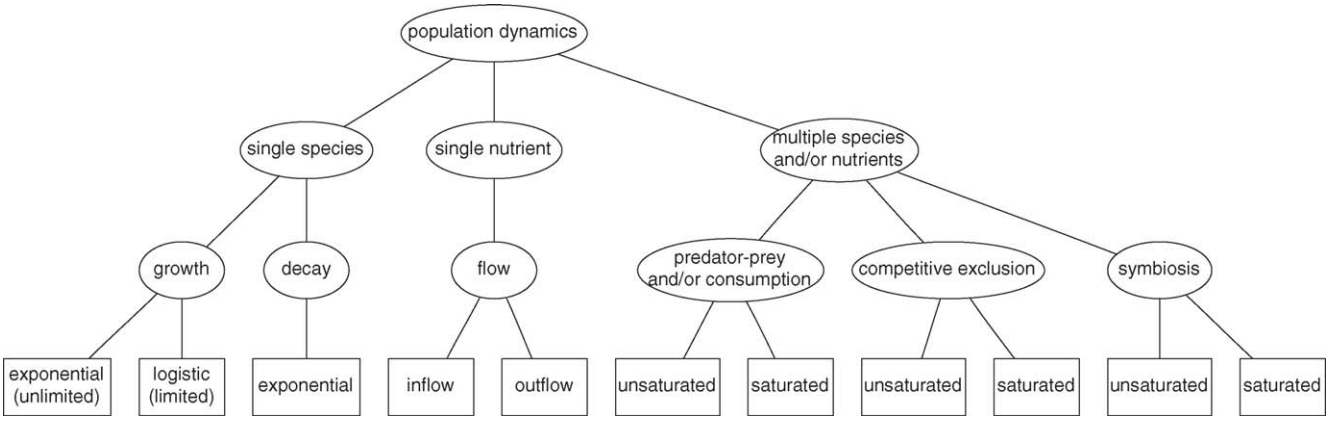


Fig. 1 – A taxonomy of classes of basic processes in the population dynamics domain.

Table 1 – Formalization of some of the population dynamics process classes from the taxonomy presented in Fig. 1

Process class growth(Population p)
Process class exponential_growth is Growth
  Expression const(growth_rate, 0, 1, Inf)$\times p$
Process class Logistic_growth is Growth
  Expression const(growth_rate, 0, 1, Inf)$\times p\times$
  $(1 - p/\text{const}(\text{capacity}, 0, 1, \text{Inf}))$

Process class decay(Population p)
Process class exponential_decay is Decay
  Expression const(decay_rate, 0, 1, Inf)$\times p$

Process class Feeds_on(Population p, set of concentration cs)
  Condition $p \notin$ cs
  Expression $p\times \prod_{c\in cs} c$

Table 2 – Schemes for combining the models of basic population dynamics processes into a model of the entire system

Combining scheme Population_dynamics(Inorganic i)
  $di/dt = -\sum_{food, f\in food} \text{const}(\_, 0, 1, \text{Inf})\times \text{Feeds\_on}(p, \text{food})$

Combining scheme Population_dynamics(Population p)
  $dp/dt = \text{Growth}(p) - \text{Decay}(p) + \sum_{food} \text{const}(\_, 0, 1, \text{Inf})\times$
  $\text{Feeds\_on}(p, \text{food}) - \sum_{pred, food, p\in food} \text{const}(\_, 0, 1, \text{Inf})\times$
  $\text{Feeds\_on}(\text{pred}, \text{food})$

capacity for the given species. In such cases, the alternative *Logistic_growth* model is more appropriate.

The definition of each process class consist of three parts, which concerns variables, conditions on the variables and the equation templates used to model processes of the class.

First, we specify which types of variables are involved in processes from the given class. For example, each process in the *Growth* and *Decay* process classes involves a single variable p of type *population*. The processes in the *Feeds_on* class involve one population variable p and a set of variables *cs* of type concentration, which can be either a population or an inorganic nutrient. The declarations of variable types are inherited through the taxonomy of process classes: the *Exponential_growth* class inherits from the parent class *Growth* the fact that growth processes involve a single variable of type *population*.

The second part of the process class definition specifies constraints on the variables involved in a process of the given class. The condition $p \notin$ cs in the *Feeds_on* process class specifies that a population cannot feed (predate) on itself.

The third part of the process class definition specifies the model (i.e., equation) structure used to model the influences of processes of the given class. The model structure is based on variables involved in the process and generic constant parameters. The values of the generic constant parameters are not known and are to be fitted against measurement data. The term *const*(*name, lower_bound, initial, upper_bound*) is used to specify a generic constant parameter: its name, as well as lower bound, default, and upper bound values. For example, the *Exponential_growth* model involves a single non-negative (note that a lower bound of 0 as well as infinite upper bound are specified) constant parameter that represents the growth rate with the default value of 1. The default value of the constant parameter is used as its initial value by the iterative non-linear method for parameter fitting, which matches the model structure against measured data and find the values of the model parameters that fit the data best.

2.2. *Combining scheme*

The knowledge representation formalism also encodes the scheme that is used to combine the models of individual basic processes into a model of the entire system. Table 2 presents two combining schemes for building population dynamics models.

The first combining scheme specifies how to build the equation for the rate of change of an inorganic nutrient concentration i, which is represented by the time derivative di/dt of i. The change of i is negatively influenced by all the *Feeds_on* interactions in which i is being consumed by an arbitrary population p. The term *Feeds_on*(p, i) denotes the model of the *Feeds_on* process influence. The $\sum$ aggregation function is used to sum up the influences of all such processes.

The second combining scheme specifies how to combine individual process models into a model of the rate of change of a population concentration p. The first line specifies that the time derivative of p increases with the population growth *Growth*(p) and decreases with its decay *Decay*(p). *Feeds_on* processes that involve p as a predator (or consumer) positively influence the rate of change of p, while the *Feeds_on* processes where p is involved as a prey negatively influence its rate of change. Again, influences of these processes are summed up, as shown in Table 2.

3. The modeling framework

The knowledge we encoded so far is independent of the particular modeling task at hand—it allows for modeling of an arbitrary system in the population dynamics domain. Before using the knowledge for modeling of a particular population dynamics system, we have to specify the modeling task. The specification includes system variables and their types along with the processes (and their process classes) that are expected to influence the system behavior. Given the modeling task specification, the encoded knowledge can be then used to generate a grammar that specifies the space of candidate model structures for the observed system. The grammar-based equation discovery method LAGRAMGE is then used to search through the space of candidate models and find the one that fits the measured data best. Fig. 2 gives a schematic representation of this modeling framework.

Table 3 gives an example of a modeling task specification for a simple ecosystem with three system variables *nut*, *phyto*, and *zoo* representing the concentrations of an inorganic nutrient and two populations, one of phytoplankton and one of zooplankton, respectively. The first two (growth and decay) processes specify that the populations of phytoplankton/zooplankton tend to increase/decrease in the absence of any interactions with the environment and other species. The

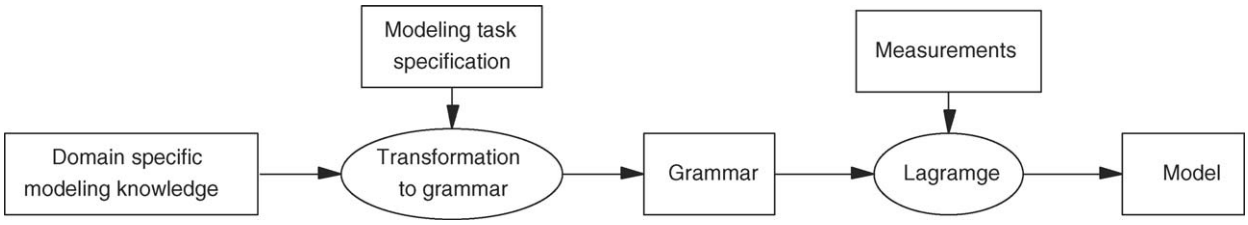


Fig. 2 – An automated modeling framework based on the integration of domain-specific modeling knowledge in the process of equation discovery.

following two *Feeds_on* processes specify that phytoplankton consumes the inorganic nutrient and that zooplankton feeds on phytoplankton.

The transformation of the modeling task specification to a grammar proceeds in a top-down manner. It starts with the starting symbol and assigns productions to it, then proceeds with other nonterminal symbols. Nonterminal symbols in the grammar denote process classes, while the alternative productions for each nonterminal symbol specify possible expressions for modeling the corresponding process class. The starting symbol of the grammar combines the expressions for individual processes into candidate models of the whole system, according to the combining scheme.

For example, consider the aquatic ecosystem example from Table 3. The start symbol uses the combining schemes from Table 2 to compose a model of the whole ecosystem (see the first production in the grammar from Table 4). The right hand side of the production for the starting symbol builds a system of three differential equations, one for each variable of the observed ecosystem. The second equation (for the time derivative of *phyto*) is built by summing up the growth process for the *phyto* population, i.e., *Growth*(*phyto*), the *phyto* decay process (0, since none of them are specified), the consumption process where *phyto* is involved as consumer, i.e., *Feeds_on*(*phyto*, {*nut*}), and the predator-prey process where *phyto* has the role of prey, i.e., *Feeds_on*(*zoo*, {*phyto*}). The productions for the other nonterminal symbols are built according to the specification of the taxonomy of process classes from Table 1.

The grammar from Table 4 generates a set of candidate model structures for modeling the simple aquatic ecosystem. Lagramge can heuristically search through this space and find the model that fits the data best. The Lagramge algorithm employs an iterative non-linear optimization method (Bunch et al., 1993) to fit the constant parameters of the candidate model structures. Heuristic function based on the discrepancy between simulated and observed values of the system variables, guides the search. Further details about Lagramge can be found in (Todorovski and Džeroski, 1997; Todorovski, 2003).

Table 3 – A task specification used for modeling a simple aquatic ecosystem consisting of two consumption interactions between three populations of inorganic nutrient, phytoplankton, and zooplankton

```
System variable Inorganic nut
System variable Population phyto, zoo

Processes Growth(phyto), Decay(zoo)
Processes Feeds_on(phyto, {nut}), Feeds_on(zoo, {phyto})
```

4. Modeling environmental dynamic systems

We applied our framework to three tasks of modeling dynamic systems from real-world data. Two tasks from the population dynamics domain have been already addressed by other equation discovery methods. The third task is from the domain of hydrodynamics. The evaluation criteria used in these experiments are predictive accuracy (as measured by root mean squared error, RMSE) and comprehensibility of the discovered models as evaluated by domain experts.

4.1. *Modeling algal growth in the Lagoon of Venice*

The Lagoon of Venice measures 550 km^2, but is very shallow, with an average depth of less than 1 m. It is heavily influ-

Table 4 – The grammar specifying the candidate models for modeling the simple aquatic ecosystem presented in Table 3

```
Start →
  Time_deriv(nut) = −const[_ : 0 : 1 :]×Feeds_on_phyto_nut;
  Time_deriv(phyto) = Growth_phyto − 0 + const[_ : 0 : 1 :]
    ×Feeds_on_phyto_nut − const[_ : 0 : 1 :]×Feeds_on_zoo_phyto;
  Time_deriv(zoo) = 0 − Decay_zoo + const[_ : 0 : 1 :]
    ×Feeds_on_zoo_phyto

Feeds_on_phyto_nut → Unsaturated_feeds_on_phyto_nut
Feeds_on_phyto_nut → Saturated_feeds_on_phyto_nut
Unsaturated_feeds_on_phyto_nut → phyto×nut
Saturated_phyto_nut → phyto×nut/
  (nut + const[saturation_rate : 0 : 1 :])

Growth_phyto → Exponential_growth_phyto
Exponential_growth_phyto → const[growth_rate : 0 : 1 :]×phyto
Logistic_growth_phyto → const[growth_rate : 0 : 1 :]
  ×phyto×(1 − const[capacity : 0 : 1 :])

Feeds_on_zoo_phyto → Unsaturated_feeds_on_zoo_phyto
Feeds_on_zoo_phyto → Saturated_feeds_on_zoo_phyto
Unsaturated_feeds_on_zoo_phyto → zoo×phyto
Saturated_zoo_phyto → zoo×phyto/
  (phyto + const[saturation_rate : 0 : 1 :])

Decay_zoo → Exponential_decay_zoo
Exponential_decay_zoo → const[decay_rate : 0 : 1 :]×zoo
```

enced by anthropogenic inflow of nutrients—7 mio kg/year of nitrogen and 1.4 mio kg/year of phosphorus (Bendoricchio et al., 1994). These (mainly nitrogen) loads are above the Lagoon's admissible trophic limit and generate its dystrophic behavior, which is characterized by excessive growth of algae, mainly *Ulva rigida*. Four sets of measured data were available (Coffaro et al., 1993) for modeling the growth of algae in the Lagoon. The data were sampled weekly for slightly more than one year at four different locations in the Lagoon. Location 0 was sampled in 1985/1986, locations 1, 2, and 3 in 1990/1991. The sampled quantities are nitrogen in ammonia NH_3, nitrogen in nitrate NO_3, phosphorus in orthophosphate PO_4 (all in [μg/l]), dissolved oxygen DO (in percentage of saturation), temperature T (°C), and algal biomass B (dry weight in [g/m^2]).

Previous experiments with automated modeling of algal growth in the Lagoon of Venice with equation discovery (Kompare and Džeroski, 1995) used the GoldHorn method (Križman, 1998). Since GoldHorn could not find an accurate model based on the set of measured variables, two additional variables were calculated and added to the set of variables. These are the growth and mortality rates, which are known quantities in ecological modeling and were calculated according to the simplified version of an existing model of algal growth in the lagoon proposed by (Coffaro et al., 1993). From the extended set of variables and data measured at Location 0, GoldHorn discovered a difference equation for predicting biomass that, due to the large measurement errors (estimated at the level of 20–50%), does not fit the data perfectly, but it still predicts most of the peaks and crashes of the biomass concentration correctly (Kompare and Džeroski, 1995). Although the equation model involves the mortality rate, as calculated by domain experts, the model itself is still a black-box model that does not reveal the limiting factors for the biomass growth in the lagoon.

The task specification of modeling algal growth in the Lagoon of Venice, given in Table 5, lists the types of the observed variables and the processes that are important for the growth of algae (and their biomass) in the lagoon. Note that we are only interested in a model of the biomass growth, so biomass is the only system variable. The specification of the `biomass_grazing` process leaves the nutrient parameter of the `Feeds_on` process class unspecified (denoted using the symbol *). Since ecologists did not know the limiting factors for biomass growth, they let Lagramge search for the model that would reveal them.

Table 5 – A task specification used for modeling the growth of algae biomass in the Lagoon of Venice

Variable Inorganic temp, DO, NH_3, NO_3, PO_4
System variable Population biomass
Process Growth(biomass) biomass_growth
Process Decay(biomass) biomass_decay
Process Feeds_on(biomass, *) biomass_grazing

The experiments with Lagramge were performed using a grammar automatically built from the task specification in Table 5 and the library of modeling knowledge outlined in Section 2 and completely specified in Appendix A. The grammar generates 6248 candidate model structures, among which the following model matches the measured data on the Location 0 best:

$$\begin{aligned}\mathrm{d}\mathtt{biomass}/\mathrm{dt} = {} & 6.17\times10^{-5}\,\mathtt{biomass}\left(1-\frac{\mathtt{biomass}}{1.80}\right)\\ & +3.01\times10^{-4}\,\mathtt{biomass}\,\mathrm{DO}\frac{\mathrm{NO_3}}{\mathrm{NO_3}+6.28}\\ & -0.0319\,\mathtt{biomass}.\end{aligned}$$

The model for Location 0 tells us that the limiting factors for the biomass growth of the *Ulva rigida* in the lagoon are dissolved oxygen (DO) and nitrogen in nitrate (NO_3).

Furthermore, Lagramge discovered another model from the Location 2 data:

$$\begin{aligned}\mathrm{d}\mathtt{biomass}/\mathrm{dt} = {} & 4.79\times10^{-5}\,\mathtt{biomass}\left(1-\frac{\mathtt{biomass}}{0.844}\right)\\ & +0.406\,\mathtt{biomass}(1-\mathrm{e}^{-0.216\mathtt{temp}})(1-\mathrm{e}^{-0.413\mathrm{DO}})\\ & \times\frac{\mathrm{NH_3}}{\mathrm{NH_3}+10}-0.0343\,\mathtt{biomass}.\end{aligned}$$

This model tells us that the limiting factors for the biomass growth are temperature (`temp`), dissolved oxygen (DO), and nitrogen in ammonia (NH_3). Although the two models are not identical, they both identify dissolved oxygen and nitrogen based nutrients to be limiting factors for biomass growth. The

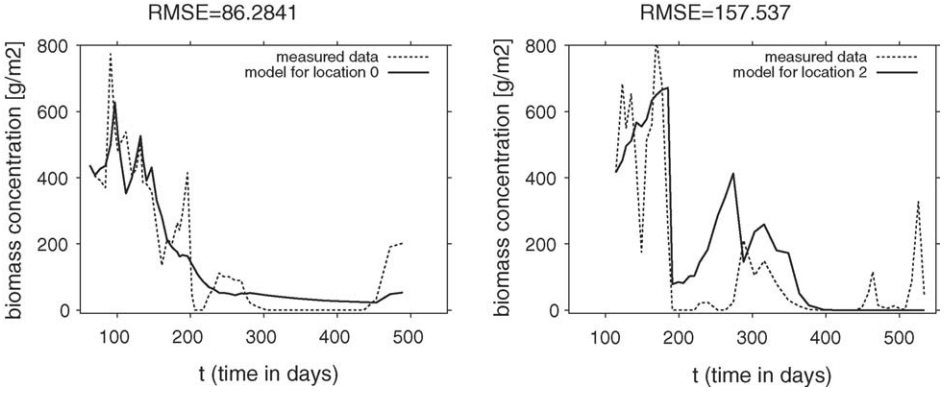


Fig. 3 – Simulations of the two models of the biomass growth in the Lagoon of Venice, discovered by Lagramge, compared to the measured biomass concentration (left-hand side: Location 0, right-hand side: Location 2).

differences between the two models may be due to the fact that the measurements were taken during two different time periods.

In the experiments with the data measured at the other two locations (1 and 3), LAGRAMGE did not find an accurate model of the biomass growth. Note that these results still compare favorably with the results obtained by GOLDHORN, which discovered an acceptable model for Location 0 only.

Fig. 3 compares the measured and simulated values of the biomass change over time for both models. We ran long-term simulations of the models from the initial value of the biomass without restarting the simulation process at each measurement point. For values of all other variables needed during the simulation, we used the measurement at the nearest time point in the past. As in the GOLDHORN experiments, due to the high measurement errors (of the order 20–50%), the models discovered by LAGRAMGE do not fit the measured data perfectly. However, they correctly predict most of the peaks and crashes of the biomass. These events are more important to ecologists than the degree of fit. Note an important advantage of these models over the one discovered by GOLDHORN. While the GOLDHORN model is black-box, the models discovered by LAGRAMGE identify the most important limiting factors for biomass growth in the Lagoon of Venice.

4.2. *Modeling phytoplankton growth in Lake Glumsø*

Lake Glumsø Jørgensen et al. (1986) is situated in a subglacial valley in Denmark. It is shallow with average depth of about 2 m and its surface area is 266,000 m^2. For several years, prior to the measurements discussed here, it was receiving mechanically-biologically treated waste water from a community with 3000 inhabitants and a surrounding area which was mainly agricultural with almost no industry. The high nitrogen and phosphorus concentration in the treated waste water had caused hypereutrophication. The lake contained no submerged vegetation, probably due to the low transparency of the water and the oxygen deficit at the bottom.

Domain experts considered the concentrations of phytoplankton (`phyto`), zooplankton (`zoo`), soluble nitrogen (`nitro`), soluble phosphorus (`phosp`), and the water temperature (`temp`) relevant for modeling the phytoplankton growth. These variables were measured at 14 distinct time points over a period of two months. This amount of measured data itself was insufficient for modeling, so additional processing was applied to obtain a suitable data set (Todorovski et al., 1998). First, dotted graphs of the measurements were plotted and given to three human experts to draw a curve that, in their own opinion, described the dynamic behavior of the observed variable between the measured points. A properly plotted expert curve can be regarded as an additional source of reliable data. Curves drawn by the human experts were then smoothed with Bezier splines. Finally, three data sets were obtained by sampling the splines derived from each of the three human experts' approximations at regular time intervals with time step $h = 0.03215$ day. The data set provided by the first expert was used for the experiments.

Table 6 – A specification of the modeling the phytoplankton growth in Lake Glumsø task

Variable Inorganic temp, nitro, phosp
Variable Population zoo
System variable Population phyto
Process Decay(phyto) phyto_decay
Process Feeds_on(phyto, *) phyto_grazing
Process Feeds_on(zoo, phyto) zoo_grazing

The task of modeling phytoplankton growth in Lake Glumsø, as specified in Table 6, lists the types of the observed variables and the processes that are important for the growth. Note that the specification of the `phyto_grazing` process leaves the nutrient parameter of the `Feeds_on` class unspecified (denoted using the symbol *). This is because the experts did not know the limiting factors for the biomass growth, and therefore we let LAGRAMGE search for the model that would identify them.

The experiments with LAGRAMGE were performed using a grammar automatically built form the task specification and the library of modeling knowledge outlined in Section 2. The grammar generates 496 candidate model structures. Among these, the model with the minimal value of the LAGRAMGE heuristic function on the measured data was:

$$\dot{\texttt{phyto}} = 0.553\,\texttt{temp}\frac{\texttt{phosp}}{0.0264 + \texttt{phosp}} - 4.35\times\texttt{phyto} - 8.67\texttt{phyto zoo}.$$

The structure of the discovered equation tells us that phosphorus is a limiting factor for phytoplankton growth in the lake and that the growth is temperature dependent.

Note that the same model was already discovered by LAGRAMGE (Todorovski et al., 1998) using a hand-crafted grammar based on the human expert knowledge about modeling population dynamics. However, there is an important difference between the experiment performed here and the one performed with the previous version of LAGRAMGE. In the previous study, we were not able to specify bounds on the values of the constant parameters, so the output of LAGRAMGE was manually post-processed in order to filter out the equations with invalid values of the constant parameters (e.g., negative growth or saturation rate). In the experiment presented here, there is no need for this additional step, since the knowledge about the valid values of the constant parameters was encoded within the library of domain-specific knowledge.

4.3. *Modeling the water level variation in Ringkøbing fjord*

In the last series of experiments, we illustrate that the proposed formalism allows for partial model specification. In such a case, a human expert specifies only some parts of the model structure and leaves others unspecified or partly specified. Our framework can be then used to determine both the structure and the parameters of the unspecified parts.

An example of such a task is modeling water level variation in Ringkøbing fjord, a shallow estuary located at the Danish

Table 7 – Formalization of the partially specified model of the water level variation in the Ringkøbing fjord and specification of the modeling task

```
Function class Salt_water_drive(Opening a, Level h_sea, Level h,
  Surface A
    Expression (h_sea − h + const[h_0 : −5 : 0.1 : 5])

Function class Fresh_water_flow(Flow Q_f, Surface A)
    Expression Q_f/A

Combining scheme Water_level_change(Level h)
    time_deriv(h) = (F(a)/A)×Salt_water_drive(a, h_sea, h, A) +
        Fresh_water_flow(Q_f, A) + G(W_vel, W_dir)
```

```
Variable Opening a
Variable Level h_sea
System variable Level h
Variable Surface A
Variable Flow Q_f
Variable Velocity W_vel
Variable Direction W_dir

Function F(a) flow_friction
Function G(W_vel, W_dir) wind_forcing
```

Table 8 – Formalization of the modeling alternatives for the unspecified parts of the model of the water level variation in the Ringkøbing fjord

```
Function class F(Opening a)
Function class F_0 is F
  Expression const[_ : −5000 : 0.1 : 5000]
Function class F_1 is F
  Expression polynomial({a}, const[_ : −5 : 0.1 : 5], 5)

Function class G(Velocity W_vel, Direction W_dir)
Function class G_0 is G
  Expression const[_ : −5000 : 0.1 : 5000]
Function class G_1 is G
  Expression Polynomial({W_vel, W_dir}, const[_ : −5 : 0.1 : 5], 5)
Function class G_2 is G
  Expression Polynomial({W_vel, sin(W_dir), cos(W_dir)},
    const[_ : −5 : 0.1 : 5], 5)
```

west coast, where it experiences mainly easterly and westerly winds.[1] Wind forcing causes large short term variation of the water level (h) measured at the gate between the estuary and the North Sea. Domain experts specified the following partial model for the temporal variation of the water level in the estuary:

$$\dot{h} = \frac{f(a)}{A}(h_{\mathrm{sea}} - h + h_0) + \frac{Q_{\mathrm{f}}}{A} + g(W_{\mathrm{vel}}, W_{\mathrm{dir}}).$$

The water level response to the wind forcing, dependent on both wind speed (variable W_{vel}, measured in [m/s]) and direction (W_{dir}, measured in degrees), is modeled by an unknown function g. Apart from wind forcing, the water level is dominated by the fresh water supply (Q_{f}, measured in [m^3/s]). When the gate is closed, fresh water is accumulated in the estuary causing a water level rise of Q_{f}/A, where A is the surface area of the estuary measured in squared meters. During periods when the gate is open, the stored fresh water is emptied in the North Sea. The gate is also opened in order to maintain sufficient water level in the estuary, in which case the water rise is driven by the difference between the water level in the open sea (variable h_{sea}, measured in meters), the water level in the estuary (h, measured in meters), and the constant parameter (h_0). The flow is restricted by the friction of the flow, modeled by an unknown function f of the number of gate parts being open (a). Namely, the gate consists of 14 parts and allows for opening some parts and closing others. The value of A is not directly observed, but a function that calculates A on the basis of h is provided, so A can be also treated as observed variable.

In order to apply our framework to the task of model completion, we first encode the partial specification within our formalism. The formalization of the partial model specification from Table 7 follows the partial model formula proposed by the domain experts. The formula is decomposed into two building blocks following the explanation of the partial model specification.[2]

In the second step, we formalize the modeling alternatives for each of the unspecified parts of the model, i.e., the f and g functions. In the experiments, we use simple constant and polynomial models due to the lack of additional domain knowledge. The modeling alternatives used in the experiments are presented in Table 8. The first modeling alternatives `F_0` and `G_0` for f and g are the simplest possible models, i.e., constants within the interval [−5000, 5000] with the initial values of 0.1. The next two alternatives (`F_1` and `G_1`) are polynomials of the appropriate system variables with constant parameters within the interval [−5, 5] (and initial values of 0.1). The maximal degree of the polynomials is five. Finally, we used one additional modeling alternative for the g function (`G_2`) that replaces the wind direction value (that represents angle) with the sine and cosine transformation thereof in the polynomial.

The data about the observed variables is collected by hourly measurements of all the observed variables within the period from 1st of January to 10th of December 1999. We used the task specification presented in Tables 7 and 8 to induce a model from the measurements with LAGRAMGE. We performed three experiments. In the first, we used the `F_0` and `G_0` modeling alternatives, in the second we used the `F_1` and `G_1` modeling alternatives, and in the third we used the `F_1` and `G_2` modeling alternatives. In order to evaluate the benefit of using partial model specification, we performed one additional experiment in which no knowledge about model structure was

[1] The task was used as an exercise within a post-graduate course on modeling dynamic systems organized in 2000. Since the Web page of the course is no longer available, we cannot provide a proper reference to the original task specification. Note also that we could not consult domain experts and therefore could not obtain expert comments on the induced models.

[2] In order to specify different alternatives for modeling flow friction F and wind forcing G, we introduce a taxonomy of function classes. The definitions of function classes are the same as definitions of process classes. We distinguish between functions and processes because the former do not represent processes from the domain; in this case, we use them to specify alternative models for the individual building blocks.

Table 9 – The root mean squared errors (RMSE, estimated on both training data and using 10-fold cross-validation) of the four water level variation models induced by LAGRAMGE with (three first rows) and without (last row) using the partial model specification provided by the domain experts

Task specification	Training RMSE	Cross-validated RMSE	#CMS
F_0 + G_0	0.0848	0.106	1
F_1 + G_1	0.0655	0.0931	378
F_1 + G_2	0.0585	0.0903	2184
Polynomial	0.0556	2.389	2801

The last column gives number of candidate model structures (#CMS) considered during the search.

used. In this last experiment, we used a polynomial model of the water level change in the fjord. We used 10-fold cross-validation for estimating the RMSE of the induced models.

Table 9 summarizes the results of the experiments. The best cross-validated performance is gained using the partial model specification provided by the experts in combination with the F_1 and G_2 modeling alternatives for the unspecified parts of the structure. LAGRAMGE proposed the following models for f and g:

$$f(a) = 5 + 5a + 5a^2 + 5a^3 - 1.01a^4$$

$$\begin{aligned} g(W_{vel}, W_{dir}) = &- 0.00137 - 0.0106 \cos W_{dir} \\ &+ 0.218 \cos W_{dir} \sin W_{dir} \\ &+ 0.0106 W_{vel} \cos W_{dir} \sin W_{dir} \\ &- 0.0128 W_{vel}^2 \cos W_{dir} \sin W_{dir} \\ &- 0.000428 W_{vel}^3 \cos W_{dir} \sin W_{dir}. \end{aligned}$$

The graph on the left-hand side of Fig. 4 shows the simulation of this model compared to the measured water level in the Ringkøbing fjord. We ran a long-term simulation of the model from the initial value of the water level without restarting the simulation process at any measurement point. For the values of all other variables needed during the simulation, we used the measurement at the nearest time point in the past.

Note that the long-term simulation of the model follows the general pattern of water level variation. However, despite the agreement with the general trend and relatively low error (RMSE of 0.0585), the model fails to precisely capture the short-term (hour) changes of the water level in the fjord. To test the short-term prediction power of the model, we performed two additional simulations, which we restarted with the true measured water level values at every hour and at every day (24 h). Table 10 presents the results of this analysis. They show that the model is also suitable for precise short-term prediction of the water level in the Ringkøbing fjord. Using the model, the water level in an hour can be predicted with an average error of 0.0168, which is below 2 cm and a correlation coefficient of 0.98.

Since the model induced by LAGRAMGE follows the partial structure specification provided by the human experts, further analysis of the model can be performed. For example, we can compare the influence of the gate opening (modeled by $f(a)(h_{sea} - h + h_0)/A$) with the effect of the wind (modeled by $g(W_{vel}, W_{dir})$). The graph on the right-hand side of Fig. 4 shows the ratio of the gate opening and the wind influences on the water level change in the Ringkøbing fjord. The low magnitude of the ratio shows that the influence of the wind prevails over the influence of the gate opening most of the time. The only exceptions occur in the period from 80 to 100 days from the beginning of the measurement, that is, the end of March and beginning of April 1999.[3]

Table 10 – The RMSE and correlation coefficient (r) for the short-term (one hour and one day) prediction of the water level in the Ringkøbing fjord compared to the RMSE and r of the simulation over the whole observation period

Prediction/simulation period	RMSE	r
1 h	0.0168	0.976
1 day	0.0425	0.845
Whole observation period	0.0585	0.659

The polynomial model of the water level variation that ignores the partial specification of the model performs best on the training data. However, the model's small RMSE is due to the overfitting of the training data, since the cross-validated RMSE of this model (2.389) is much larger than the cross-validated RMSE of the models that follow the partial structure specification.

In sum, the Ringkøbing fjord experiments show the capability of our framework to address modeling tasks in which human experts can partially specify the model structure and leave some of its parts unspecified.

5. Related work

The presented modeling framework follows the paradigm of compositional modeling (Falkenheiner and Forbus, 1991), an automated approach to building qualitative models from observations in presence of domain-specific modeling knowledge. In the compositional modeling, knowledge is organized as a library of model fragments. Given a modeling task specification (or scenario), compositional modeling methods compose a set of appropriate fragments into a model that is suitable for modeling the observed system. The obtained model is evaluated by qualitative simulation (Kuipers, 1994). The compositional modeling approach is mainly applied to the tasks of building qualitative models. For example, authors of (Coghill et al., 2002) apply this approach to the task of inducing qualitative models of chemical reaction pathways from noisy measurement data.

Although the concepts introduced within the QR area are also relevant for automated building of quantitative models of real-world systems, this idea has not been widely explored. A notable exception is the PRET reasoning system for automated modeling of dynamic systems (Bradley et al., 2001), which

[3] Note again that, unfortunately, we could not obtain expert comments on these results.

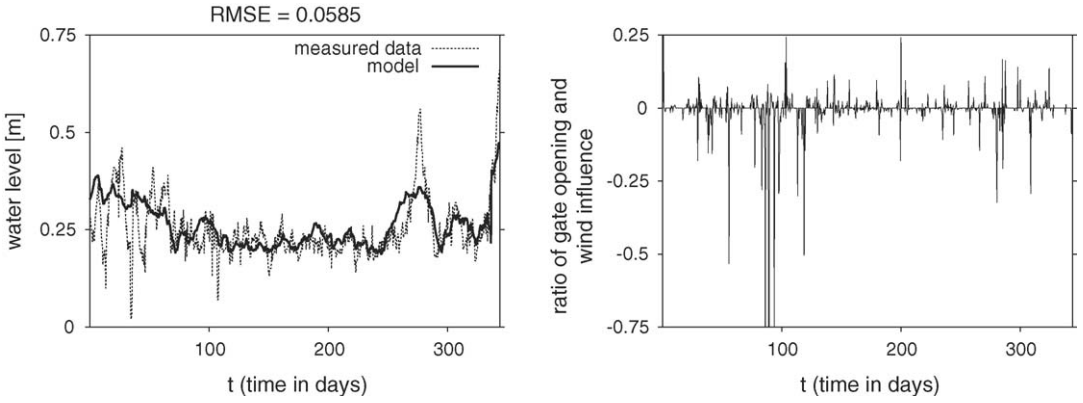


Fig. 4 – Simulation of the water level variation model induced by LAGRAMGE compared to the measured water level (left-hand side) and the ratio of the gate opening and the wind influences on the water level change in the Ringkøbing fjord as modeled by LAGRAMGE.

employs two kinds of knowledge. The first is domain-specific knowledge in the form of "conservation rules", such as Kirchoff's law in the domain of electrical circuits, which specifies that the sum of input and output currents at any observed point in the circuit is zero. Similarly, the force balance rule in the mechanics specifies that the sum of forces at any observed coordinate of the mechanical system is zero. These rules are more general than domain knowledge about model fragments used in compositional modeling approaches, and constrain the space of possible models much less. This kind of knowledge more appropriate for modeling in engineering domains such as electrical circuits and mechanics. Knowledge about basic processes in the domain of interest (that is more appropriate for building explanatory models in the biological or chemical domains) can not be fitted within PRET's framework.

The approach presented here builds up on previous work on inducing process-based models presented in (Langley et al., 2002). Note however, that the taxonomy of process classes introduced here allows for better organization of knowledge. Also, here we made the combining scheme explicit, as opposed to the implicit combining scheme used there—models of individual processes are always additively combined. Finally, note that methods for inducing equation-based models from data have been already used to induce models of environmental systems, (see, e.g., Kompare and Džeroski, 1995; Todorovski et al., 1998). They have been also successfully applied to tasks of revising existing models of environmental system, see, e.g., (Whigham and Recknagel, 2001; Saito et al., 2001; Todorovski et al., 2003).

6. Summary and further work

In this paper, we presented a framework for automated modeling of dynamic systems based on equation discovery. The framework integrates the theoretical knowledge-driven and the empirical data-driven approaches to modeling. The framework provides a formalism for encoding and integrating domain-specific knowledge in the process of model induction. The knowledge is organized in a taxonomy of process classes, each representing an important class of processes in the observed domain. This high-level knowledge representation can be automatically transformed to the operational form of grammars that specify the space of candidate models of the observed system. The equation discovery method LAGRAMGE can be then used to search through the space of candidate models and find the one that fits the measured data best.

The results of the application of the modeling framework to two real-world modeling tasks show that our framework is capable of inducing comprehensible dynamic systems' models from real-world measurement data. Our framework performs better than existing equation discovery methods on the tasks of modeling algae growth in Lagoon of Venice in terms of performance, flexibility, and comprehensibility of the discovered models. The experiment on modeling water level variation in Ringkøbing fjord illustrates the capability of our framework to address modeling tasks, in which a human expert partially specifies the model structure and leaves other parts unspecified—our framework can be then applied to induce the unspecified parts of the model from data.

Although the scope of this paper is modeling dynamic change of the observed system through time, the approach can be extended to incorporate knowledge about spatial processes also. In the population dynamics domain, these processes would reflect the spatial diffusion of populations through the environment. The extended approach would allow modeling of spatio-temporal dynamic systems with partial differential equations (Todorovski et al., 2000).

Appendix A

A.1. *Complete library of modeling knowledge for population dynamics*

```
type Concentration is real
type Concentrations is set(Concentration)
type Population is Concentration
type Populations is set(Population)
type Inorganic is Concentration
function class Saturation(Concentration c)
function class No_saturation() is Saturation
   expression c
function class Saturation_type_1() is Saturation
   expression c / (c + const(saturation_rate,0,1,Inf))

function class Saturation_type_2() is Saturation
   expression c * c / (c * c + const(saturation_rate,0,1,Inf))

function class Saturation_type_3() is Saturation
   expression 1 - exp(-const(saturation_rate,0,1,Inf) * c)
process class Growth(Population p)

process class Exponential_growth() is Growth
   expression const(growth_rate,0,1,Inf) * p

process class Logistic_growth() is Growth
   expression const(growth_rate,0,1,Inf) * p * (1 - p / const(capac,0,1,Inf))
process class Decay(Population p)

process class Exponential_decay() is Decay
   expression const(decay_rate,0,1,Inf) * p

process class Flow(Concentration c)

process class Constant_inflow() is Flow
   expression const(inflow_rate,0,1,Inf)

process class Constant_outflow() is Flow
   expression -const(outflow_rate,0,1,Inf)

process class Feeds_on(Population p, Concentrations cs)
   condition p not in cs
   expression p * product({c}, c in cs, Saturation(c))

process class Interaction(Populations ps)
   condition cardinality(ps) at least 2

process class Competitive_exclusion() is Interaction
   expression -product({p}, p in ps, Saturation(p))

process class Symbiosis() is Interaction
   expression product({p}, p in ps, Saturation(p))

combining scheme Population_dynamics(Inorganic i)
   time_deriv(i) = + sum({} , true, Flow(i))
      - sum({p, food}, i in food, const(_,0,1,Inf) * Feeds_on(p, food))
combining scheme Population_dynamics(Population p)
   time_deriv(p) = + sum({} , true, Growth(p))
      + sum({} , true, Flow(p))
      + sum({food}, true, const(_,0,1,Inf) * Feeds_on(p, food))
      - sum({} , true, Decay(p))
      - sum({p1, food}, p in food, const(_,0,1,Inf) * Feeds_on(p1, food))
      + sum({ps}, p in ps, const(_,0,1,Inf) * Interaction(ps))
```

REFERENCES

Bendoricchio, G., Coffaro, G., DeMarchi, C., 1994. A trophic model for Ulva Rigida in the Lagoon of Venice. Ecol. Model. 75–76, 485–496.

Bradley, E., Easley, M., Stolle, R., 2001. Reasoning about nonlinear system identification. Artificial Intell. 133, 139–188.

Bunch, D.S., Gay, D.M., Welsch, R.E., 1993. Algorithm 717: Subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models. ACM Trans. Mathematical Software 19, 109–130.

Coffaro, G., Carrer, G., Bendoricchio, G., 1993. Model for Ulva Rigida growth in the Lagoon of Venice. Technical report, University of Padova, Padova, Italy. UNESCO MURST Project: Venice Lagoon Ecosystem.

Coghill, G.M., Garrett, S.M., King, R.D., 2002. Learning qualitative models in the presence of noise. In: Proceedings of the 16th International Workshop on Qualitative Reasoning,.

Falkenheiner, B., Forbus, K.D., 1991. Compositional modeling: Finding the right model for the job. Artificial Intelligence 51, 95–143.

Gershenfeld, N., 1999. The Nature of Mathematical Modeling. Cambridge University Press, Cambridge, UK.

Jørgensen, S.E., Kamp-Nielsen, L., Chirstenen, T., Windolf-Nielsen, J., Westergaard, B., 1986. Validation of a prognosis based upon a eutrophication model. Ecol. Model. 32, 165–182.

Kompare, B., Džeroski, S., 1995. Getting more out of data: automated modelling of algal growth with machine learning. In: Proceedings of the International symposium on coastal ocean space utilisation, Yokohama, Japan. pp. 209–220.

Križman, V., 1998. Avtomatsko odkrivanje strukture modelov dinamičnih sistemov. Ph.D. thesis, Faculty of computer and information science, University of Ljubljana, Ljubljana, Slovenia. In Slovene.

Kuipers, B., 1994. Qualitative reasoning: modeling and simulation with incomplete knowledge. MIT Press, Cambridge, MA.

Langley, P., Sanchez, J., Todorovski, L., Džeroski, S., 2002. Inducing process models from continuous data. Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, pp. 347–354.

Ljung, L., 1993. Modelling of industrial systems. In: Proceedings of Seventh International Symposium on Methodologies for Intelligent Systems, Springer, Berlin, pp. 338–349.

Murray, J.D., 1993. Mathematical Biology, second corrected edition. Springer, Berlin.

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

Todorovski, L., 2003. Using domain knowledge for automated modeling of dynamic systems with equation discovery. PhD. thesis, Faculty of computer and information science, University of Ljubljana, Ljubljana, Slovenia. Available for download at http://www-ai.ijs.si/ ljupco/ed/.

Todorovski, L., Džeroski, S., 1997. Declarative bias in equation discovery. In: Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, pp.376–384.

Todorovski, L., Džeroski, S., Kompare, B., 1998. Modelling and prediction of phytoplankton growth with equation discovery. Ecol. Model. 113, 71–81.

Todorovski, L., Džeroski, S., Langley, P., Potter, C.S., 2003. Using equation discovery to revise an earth ecosystem model of the carbon netproduction. Ecol. Model. 170, 141–154.

Todorovski, L., Džeroski, S., Srinivasan, A., Whiteley, J., Gavaghan, D., 2000. Discovering the structure of partial differential equations from example behavior. In: Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, pp. 991–998.

Whigham, P.A., Recknagel, F., 2001. Predicting chlorophyll-a in freshwater lakes by hybridising process-based models and genetic algorithms. Ecol. Model. 146 (1–3), 243–252.