Integrating Knowledge-Driven and Data-Driven Approaches to Modeling

Ljupčo Todorovski and Sašo Džeroski

Abstract
In this paper, we present a modeling framework that integrates the knowledge-based theoretical approach to modeling with the data-driven empirical modeling of dynamic systems. 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, their models, and includes guidelines for combining models of individual processes into a model of the entire observed system. We present a method for automated translation of the knowledge into the operational form of grammars that constrain the space of candidate models considered during the induction process. The developed framework is applied to two tasks of modeling dynamic systems from noisy measurement data in the domains of population and hydrodynamics.

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 involves observations and measurements of the system behavior under various conditions, selecting a set of system variables that are important for modeling, and formulating the model itself. In this paper, we deal with dynamics systems, i.e. systems that change their state over time, and their models expressed in the form of ordinary differential equations [2].

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 the system identification [6] focus mainly on solving the parameter estimation problem and make one of the following two assumptions. The first is that the structure of the model is provided by a human expert and the second assumption is 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 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 structure of the model equations from some general class of structures (such as linear or polynomial) that he/she believes to be adequate, fits its constant 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 the 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 can integrate domain-specific knowledge in the process of data-driven model induction, where knowledge is used to tailor the space of candidate model structures. Each candidate model is matched against given data and the model that fits the data best is chosen. Before we can use the domain-specific knowledge for guiding the induction process, we have to encode it - in the first part of the paper (Section 2), we present a formalism for encoding domain-specific modeling knowledge and illustrate its use in the domain of population dynamics. In Section 3, we show how to transform the knowledge into operational form of grammars that guide the process of induction. Section 4 reports on applications of the framework on two tasks of inducing real-world models from measured data. Finally, Section 5 concludes the paper with a summary, brief discussion of related work 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 influence the change of one or more system variables, while the model of a basic process specifies the equations used to model its influence. So our domain-specific knowledge encoding formalism will target 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 formalism also encodes knowledge about how to combine 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 domain of population dynamics [7]. Population dynamics area 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 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. Figure 1 presents an example taxonomy of process classes in the population dynamics domain. On the highest level of the taxonomy, we distinguish between basic processes that involve single species or inorganic nutrient and processes that represent interactions between two or more species and 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 the domain experts.

Table 1 presents the formalization of several process classes from the population dynamics taxonomy. 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 limited carrying capacity for species. In such cases, alternative Logistic_growth model is more appropriate.

Figure 1: A taxonomy of classes of basic processes in the population dynamics domain.

The definition of each process class consist of three parts.
First, we specify which types of variables are involved in the processes from the class. For example, each process in the Growth process class involves a single population \( p \). The processes in the Feeds_on class involve one population variable \( p \) and one variable \( c \) 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: Exponential_growth class inherit 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 the processes. The condition \( p \in cs \) in the Feeds_on process class specifies that a population cannot feed (predate) on itself.

```plaintext
process class Growth(Population p)
process class Exponential_growth is Population_growth
    expression const(growth_rate,0,1,Inf) * p
process class Logistic_growth is Population_growth
    expression const(growth_rate,0,1,Inf) * p * (1 - p / 
    const(capacity,0,1,Inf))

process class Feeds_on(Population p, set of Concentration cs)
    condition p \in cs
    expression p * \prod_{c \in cs} c
```

Table 1
Formalization of some of the population dynamics process classes from the taxonomy presented in Figure 1

The third part of the process class definition specifies the model (i.e., equation) structure used to model the influences of the processes in the 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. Symbol \( \text{const(name, lower_bound, initial, upper_bound)} \) is used to specify a generic constant parameter: its name, lower bound, default, and upper bound values. For example, the Exponential_growth model involves a single nonnegative (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 parameter fitting method.
combining scheme Population_dynamics(Inorganic i)
\[ \frac{di}{dt} = -\sum_{food \in i} \text{const}(0,1,\text{Inf}) * \text{Feeds_on}(p, \text{food}) \]

combining scheme Population_dynamics(Population p)
\[ \frac{dp}{dt} = \text{Growth}(p) - \text{Decay}(p) + \sum_{food \in p} \text{const}(0,1,\text{Inf}) * \text{Feeds_on}(p, \text{food}) - \sum_{pred \in p} \text{const}(0,1,\text{Inf}) * \text{Feeds_on}(\text{pred}, \text{food}) \]

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

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 time change of an inorganic nutrient \( i \), which is represented as the time derivative \( i \) of \( i \). The change of \( i \) is negatively influenced by all the \text{Feeds_on} interactions in which \( i \) is being consumed by an arbitrary population \( p \). The \text{Feeds_on}(p, i) denotes the model of the \text{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 time change of a population \( p \). The first line specifies that the time derivative of \( p \) increases with the population growth \( \text{Growth}(p) \) and decreases with its decay \( \text{Decay}(p) \). \text{Feeds_on} processes that involve \( p \) as a predator (or consumer) positively influence the change of \( p \), while the \text{Feeds_on} processes where \( p \) is involved as a prey negatively influence its change. Again, influences of these processes are summed up, as shown in Table 2.

Figure 2: An automated modeling framework based on the integration of domain-specific modeling knowledge in the process of equation discovery.
3. The modeling framework

The knowledge we encoded so far is independent of the particular modeling task and allows automated modeling of an arbitrary system in the population dynamics domain. Before applied to a particular task of automated modeling, we require user to specify the modeling task at hand. The task specification defines the types of the measured system variables along with the process classes that are expected to influence the system behavior. Given the task specification and the encoded knowledge about basic processes in the domain, we can build an operational representation of the knowledge that can be used to guide the process of induction. It takes a form of a grammar, as the ones used in linguistics. In our case, the grammar is used to specify the space of candidate models considered in the induction process. Figure 2 schematizes the modeling framework. Note that for induction, we used the equation discovery method LAGRAMGE [9] that searches through the space of candidate model structures and finds the one that fits measured data best.

variable Inorganic nut
variables Population phyto, zoo

processes Growth(phyto), Decay(zoo)
processes Feeds_on(phyto, {nut}), Feeds_on(zoo, {phyto})

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.

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 species of phytoplankton and zooplankton. The first two (growth and decay) processes specify that the populations of phytoplankton/zooplankton tend to increase/decrease in absence of any interactions with the environment and other species. The following two Feeds_on processes specify that phytoplankton consumes inorganic nutrient and that zooplankton feeds on phytoplankton.
4. Modeling environmental dynamic systems

In this section, we present the applications of our framework to two modeling tasks that involve the induction of process-based models from real-world measurements. The first task is from the population dynamics domain and the second is from the hydrodynamics domain.

4.1 Modeling algal growth in the Lagoon of Venice

The Lagoon of Venice is heavily influenced by anthropogenic inflow of (nitrogen and phosphorus) nutrients that cause excessive growth of algae [3]. The data were sampled weekly for slightly more than one year at four different locations in the Lagoon. The sampled quantities are nitrogen in ammonia $NH_3$, nitrogen in nitrate $NO_3$, phosphorus in orthophosphate $PO_4$ (all in $[\mu g/l]$), dissolved oxygen DO (in percentage of saturation), temperature $T$ ([degrees C]), and algal biomass $B$ (dry weight in $[g/m^2]$).

In the previous experiments with building models of population dynamics in the Lagoon of Venice, we used GOLDHORN method [3]. GOLDHORN could find an acceptable model only if we pre-processed the data and pre-compute two additional variables that reflect growth and mortality rates. The obtained model does not fit the data perfectly, but it still predicts most of the peaks and crashes of the biomass concentration correctly. 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.

variable Inorganic temp, DO, NH3, NO3, PO4
variable Population biomass
process Growth(biomass) biomass_growth
process Decay(biomass) biomass_decay
process Feeds_on(biomass, *) biomass_grazing

Table 4:
A task specification used for modeling biomass growth in the Lagoon of Venice.

The task of modeling algal growth in the Lagoon of Venice from Table 4 specifies the types of the observed system variables and the processes that are important for the biomass (algae) growth in the lagoon. Note that the specification of the bio-
mass grasing process leaves the nutrient parameter of the Feeds_on process class unspecified (denoted using the symbol *). Since we do not know the limiting factors for the biomass growth, we let LAGRAMGE to search for the model that would reveal them.

![Graph showing simulated and measured biomass growth](image)

**Figure 3:** Simulations of the biomass growth model in the Lagoon of Venice, induced by LAGRAMGE, compared to the measured biomass concentration.

The model induced by LAGRAMGE tells us that the limiting factors for the biomass growth are temperature $temp$, dissolved oxygen $DO$, and nitrogen in ammonia $NH_3$. Figure 3 compares the measured and simulated values of the biomass. We ran long-term simulation of the model from the initial value of the biomass without restarting the simulation process at each measurement point. For values of all other system variables needed during the simulation, we used the measurement at the nearest time point in the past. The model induced with LAGRAMGE did not fit the measured data perfectly. However, it correctly predicts most of the peaks and crashes of the biomass concentration. These events are more important to ecologists than the degree of fit. Note an important advantage of these models over the one induced with GOLDHORN. While the GOLDHORN model is black-box, the model induced with LAGRAMGE identifies the most important limiting factors for the biomass growth in the Lagoon of Venice.
4.2 Modeling the water level variation in Ringkøbing fjord

Ringkøbing fjord is a shallow estuary located at the Danish west coast, where it experiences mainly easterly and westerly winds. 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:

\[
\frac{dh}{dt} = \frac{f(a)}{A}(h_{sea} - h + h_0) + \frac{Q_f}{A} + g(W_{vel}, W_{dir}).
\]

The water level response to the wind forcing, dependent on both wind speed (variable \( W_{vel} \) measured in \([\text{m/s}]\)) 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 \([\text{m}^3/\text{s}]\)). 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 number of gate parts being open (\( a \)). Namely, the gate consists of 14 parts and allows for opening some parts and closing others. The data about the observed variables is collected by hourly measurements within the period from 1st of January to 10th of December 1999.

---

<table>
<thead>
<tr>
<th>Process Class</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salt_water_drive</td>
<td>( (h_{sea} - h + h_0) )</td>
</tr>
<tr>
<td>Fresh_water_flow</td>
<td>( Q_f/A )</td>
</tr>
</tbody>
</table>

Combining scheme Water_level_change:

\[
\frac{dh}{dt} = f(a) \cdot Salt_water_drive(a, h_{sea}, h, A) + Fresh_water_flow(Q_f, A) + G(W_{vel}, W_{dir})
\]

---

Table 5: Formalization of the partially specified model of the water level variation in the Ringkøbing fjord.
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 5 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.

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 $F_{\cdot0}$ and $G_{\cdot0}$ for $f$ and $g$ are the simplest possible models, i.e., constants. Next two alternatives ($F_{\cdot1}$ and $G_{\cdot1}$) are polynomials of the appropriate system variables with maximal degree of five. Finally, we used one additional modeling alternative for the $g$ function ($G_{\cdot2}$) that replaces the wind direction value (that represents angle) with the sine and cosine transformation thereof in the polynomial.

<table>
<thead>
<tr>
<th>task specification</th>
<th>training RMSE</th>
<th>10-CV RMSE</th>
<th>#CMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{\cdot0} + G_{\cdot0}$</td>
<td>0.0848</td>
<td>0.106</td>
<td>1</td>
</tr>
<tr>
<td>$F_{\cdot1} + G_{\cdot1}$</td>
<td>0.0655</td>
<td>0.0931</td>
<td>378</td>
</tr>
<tr>
<td>$F_{\cdot2} + G_{\cdot2}$</td>
<td>0.0585</td>
<td>0.0903</td>
<td>2184</td>
</tr>
<tr>
<td>polynomial</td>
<td>0.0556</td>
<td>2.39</td>
<td>2801</td>
</tr>
</tbody>
</table>

Table 6:
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 with (three first rows) and without (last row) using the partial model specification provided by domain experts. Last column gives number of candidate model structures (\#CMS) considered by LAGRAMGE.

Table 6 summarizes the results of the experiments with LAGRAMGE. The best cross-validated performance is gained using the partial model specification provided by the experts in combination with $F_{\cdot1}$ and $G_{\cdot2}$ modeling alternatives for the unspecified parts of the structure. The polynomial model of the water level variation that ignores the partial structure specification 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 (2.39) is much larger than the cross-validated RMSE of the models that follow the partial structure specification.
5. 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 approach presented here builds up on previous work on inducing process-based models presented in [5]. The formalism for building process-based model presented there has been extended to incorporate the features presented in this paper. Note that this extended approach has been already presented and evaluated on synthetic data [8]. However, this is the first study, where the presented modeling framework is applied on realistic tasks of inducing models from real-world measurements. The presented modeling framework is in spirit of compositional modeling paradigm [1] used for building qualitative models of real-world systems [4].

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.
Bibliography


