Integrating Knowledge-Based and Data-Driven Modeling of Population Dynamics

Sašo Džeroski and Ljupčo Todorovski[1]

Abstract

The paper is concerned with integrating knowledge-based modeling or modeling from first principles, with data-driven or automated modeling of dynamic systems. We propose an approach to representing knowledge about processes in population dynamics domains and a method to transform such knowledge into an operational form that could be used by systems for discovery of differential equations. In this way, we improve the ability of computer systems to exploit both knowledge and data in the process of automated modeling of dynamic systems.

1. Population dynamics modeling

Population ecology studies the structure and dynamics of populations, where each population is a group of individuals of the same species inhabiting the same area. Structure of the population is characterized by the population density and its age, sex and spatial distribution. The focus of this paper is on modeling the dynamics of populations, especially the dynamics of change of their density. We consider here models of predator-prey population dynamics, where the interaction between predator and pray is antagonistic in the sense that it causes increase of the predator population and decrease of the prey population. The models take form of systems of differential equations (Murray 1993).

For example, consider a simple model based on populations of fox and rabbits. The latter are grazing on grass and foxes are carnivores that hunt rabbits. If the rabbit population density is high, the fox population rapidly grows. However, this causes many rabbits to be eaten, thus diminishing the rabbits population to the point where there is not sufficient food for foxes. Consequently, the fox population decreases, which causes faster growth of the rabbit population. This oscillatory behavior of these two population densities can be modeled using the well known Volterra-Lotka population dynamics model (Murray 1993). It can be generalized with the following schema:

$$\begin{aligned} \dot{N} &= \mathit{growth\_rate}(N) - \mathit{feeds\_on}(P, N) \\ \dot{P} &= \mathit{feeds\_on}(P, N) - \mathit{decay\_rate}(P). \end{aligned}$$

[1]Jožef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia,
email: Saso.Dzeroski@ijs.si, Ljupco.Todorovski@ijs.si

where N is the prey/rabbits population density and P the predator/fox population density.

Using this model schema, we can build models of predator-prey population dynamics with different complexity. The term *growth_rate*(N) defines the model of the prey population growth in absence of predation. Two models of single population growth are usually used (Murray 1993):

$$(a)\ growth\_rate(N) = aN; \quad (b)\ growth\_rate(N) = aN(1 - \frac{N}{K}).$$

The first model (a) assumes that the population growth is exponential and unlimited. However, there are real-world environments that have some carrying capacity for the population, which limits the density of the population. For example, this can be the limited supply of grass, which rabbits graze on. In such cases, an alternative logistic growth model (b) can be used, where K is a constant, determining the carrying capacity of the environment.

The second assumption made in simple population models is that the predation rate is proportional to the densities of predator and prey populations ($feeds\_on(N, P) = bPN$). Just like for growth, this means that the predation growth is exponential and unlimited. Again, in some cases the predators have limited predation capacity. When the prey population density is small the predation rate is proportional to it, but when the prey population becomes abundant, the predation capacity saturates to some limit value ($feeds\_on(N, P) = bPs(N)$). Several different terms can be used to model the predator saturation response to the increase of the prey density (Murray 1993):

$$(a)\ s(N) = A\frac{N}{N+B}; \quad (b)\ s(N) = A\frac{N^2}{N^2+B}; \quad (c)\ s(N) = A(1 - e^{-BN}),$$

where A is the limit value of the predation capacity saturation, and B is the constant, which determines the saturation rate.

2. Converting modeling and domain knowledge into context-free grammars

In the formalism for encoding the population dynamics knowledge, two types of knowledge can be provided. First, specific knowledge about the population dynamics domain (specifying the populations in the domain and pairs of populations where predator-prey interaction appears) is specified. The second type of knowledge domain independent modeling knowledge about population dynamic processes. Combining these two types of knowledge, a grammar for equation discovery is automatically generated.

2.1 Domain-specific knowledge

First, we provide a specification of the food-chain in the domain. An example for the simple example domain consisting of two populations of rabbits and foxes is given in Table 1. We use three first-order predicates: the

`domain(domain_name)` predicate is used to specify the name of the domain at hand, `foxes_and_rabbits` in the example; each population in the domain is specified using the predicate `population(domain_name, population_name)`; finally, we use the predicate `feeds_on(domain_name, predator_population, prey_population)` to specify each interaction between two populations. For now, only predator-prey interactions can be specified using the `feeds_on` predicate. However, the formalism can be easily generalized to allow specifying other types of interactions between populations, such as parasytism, competitive exclusion and symbiosis (Murray 1993).

Table 1: Description of a simple population dynamics domain consisting of two populations.

```
% Description of the Volterra-Lotka population dynamics domain ...
domain(foxes_and_rabbits).

% with two populations of foxes and rabbits ...
population(foxes_and_rabbits, fox).
population(foxes_and_rabbits, rabbit).

% where foxes are predators that feed on rabbits.
feeds_on(foxes_and_rabbits, fox, rabbit).
```

Please note that by using predicates `population` and `feeds_on`, the user is allowed to specify an arbitrary number of populations and interactions between them. For illustrative purposes, we include here a simple example domain with two populations and one interaction between them only.

2.2 Modeling knowledge

The second part of the modeling knowledge, which is domain independent, is given in Table 2. We use the predicate `template` to specify a set of alternative models for population dynamics processes like population growth and saturation, described in Section 1. Note that the symbol `const` is used to specify a constant parameter, whose value has to be fitted against measured data in the process of equation discovery.

2.3 Grammar construction

Using the definitions of the background knowledge from Tables 1 and 2 we can write a grammar for equation discovery. We automated this process, using the predator-prey model schema presented in Section 1. We have written a program that transforms a given food chain into a grammar that derives expressions for the differential equations.

The top level productions follow the template of the generalized Volterra-Lotka model. For each prey only population (no outgoing links) a growth term is added. For each other population Y, a decay term of the form − const * Y

Table 2: Templates with alternative sub-expressions used for modeling the processes of saturation and population growth.

```
% no saturation
template(saturation, X, (const * X) ).
% three different saturation models
template(saturation, X, (const * X / (X + const)) ).
template(saturation, X, (const * X * X / (X * X + const)) ).
template(saturation, X, (const * (1 - exp(const * X))) ).

% exponential ...
template(growth, X, (const * X) ).
% and logistic growth model
template(growth, X, (const * X * (1 - X / const)) ).
```

is added. In the equation for a predator population, a positive term is added for each outgoing feeds-on link. In the equation of a prey, a negative term is added for each incoming feeds-on link.

Productions are then added for each of the basic processes (growth, decay, predation) taking place in the domain. For each feeds-on(X,Y) link, a production specifies the form the associated term can take: the predator density X is multiplied by the term nutrient(Y). Productions for nutrient(Y) allow for different saturation terms, depending on the templates specified.

Table 3: A grammar for equation discovery constructed from the background knowledge in Tables 1 and 2.

```
foxes_and_rabbits ->
        time_derivative(rabbit) = growth(rabbit) - feeds_on(fox,rabbit);
        time_derivative(fox) = feeds_on(fox,rabbit) - const * fox

growth(rabbit) -> const * rabbit
growth(rabbit) -> const * rabbit * (1 - rabbit / const)

feeds_on(fox,rabbit) -> fox * nutrient(rabbit)

nutrient(rabbit) -> const * rabbit
nutrient(rabbit) -> const * rabbit / (rabbit + const)
nutrient(rabbit) -> const * rabbit * rabbit / (rabbit * rabbit + const)
nutrient(rabbit) -> const * (1 - exp(const * rabbit))
```

The grammar generated by the program for the example population dynamics domai , given Tables 1 and 2 as input, is given in Table 3. The starting non-terminal symbol in the grammar `foxes_and_rabbits` is used to generate both equations of the population dynamics model, using the schema from Section 1. The growth of the population of rabbits in absence of predation is modeled using the non-terminal symbol `growth(rabbit)` with two alternative productions, reflecting the two `template` predicates for growth from Table 2.

The third non-terminal symbol `feeds_on(fox,rabbit)` models the predation of foxes on rabbits. The predation rate is always proportional to the density of the fox population.

The non-terminal `nutrient(rabbit)` is used to introduce the model of predator response to the increase of rabbits population density, which can be proportional (first production `nutrient(rabbit) -> rabbit`) or can saturate for high rabbits population density (last three productions). The terminal symbols `fox` and `rabbit` are used to introduce the measured system variables of the population densities. Finally, the terminal symbol `const` denotes a constant parameter in the equations that has to be fitted to measured data.

3. Using the generated grammars for equation discovery

The overall setting in which the knowledge mentioned above is to be used is as follows. On one hand, we are given measured data about a population dynamics system. On the other hand, we are given domain specific knowledge about this system (food chains) and general knowledge on population dynamics modeling. The task is to find a model consistent with the given knowledge, which fits the given measured data.

In knowledge-based modeling, a domain expert uses his knowledge of the processes in the system to write down a model in the form of a set of differential equations. The structure of these equations is the crucial component of the model and reflects the processes in the system: constants in the equations can be calibrated using measured data. Mainstream system identification methods (Ljung 1993), which work under the assumption that the form of the equations is known, are typically used for the latter task.

In data-driven modeling, measured data about the dynamic system is used to derive both the structure and the constants of the differential equations in an automated fashion. Equation discovery is the area of machine learning that develops methods for automated discovery of quantitative laws, expressed in the form of equations, in collections of measured data (Džeroski et al. 1999). Equation discovery systems search a space of possible equation structures and are appropriate for data-driven modeling.

The equation discovery system LAGRAMGE (Todorovski/Džeroski 1997) can discover differential equations from measured data. It can also use background knowledge in the form of context-free grammars in addition to measured data. We thus propose to use the grammars derived from knowledge about population dynamics together with measured data as inputs to LAGRAMGE. Preliminary experiments indicate that this yields better performance than learning from measured data only. In particular, the noise robustness of the equation discovery process is greatly improved and the resulting models are much more understandable. Together with the ease of specification of food chains, this makes the approach much more useful as compared to earlier general-purpose equation discovery systems.

4. Discussion

The proposed approach allows for easy encoding of two types of knowledge about modeling population dynamics: basic knowledge about predator-prey interactions between populations and modeling knowledge about different population dynamic processes, such as growth of a population and saturation of predation. This high-level knowledge is then transformed automatically to the operational form of grammars, which can be used to guide the search for equations / models in the process of equation discovery. This can be done for arbitrarily complex predator-prey models consisting of any number of populations and interactions between them.

The grammars generated derive whole models, i.e., sets of equations, which take into account the interactions between populations. This significantly reduces the set of possible models to consider in equation discovery as compared to earlier approaches to equation discovery that do not consider such interactions. We expect this to lead to improved efficiency of equation discovery, as well as improved understandability of the models discovered.

We have a preliminary integration of the proposed approach within a system for discovery of differential equations, which has yielded improved performance in initial experiments. At present, the formalism proposed only considers predator-prey interactions between populations. Further work will also consider other types of interactions, such as parasytism, competitive exclusion and symbiosis, which can be easily introduced.

Bibliography

Džeroski, S., Todorovski, L., Bratko, I., Kompare, B., Križman, V. (1999): Equation discovery with ecological applications. In A. H. Fielding, editor, *Machine Learning Methods for Ecological Applications*, pages 185–207. Kluwer Academic Publishers, Boston, MA.

Ljung, L. (1993): Modelling of industrial systems. In *Proceedings of Seventh International Symposium on Methodologies for Intelligent Systems*, pages 338–349, Trondheim, Norway, Springer.

Murray, J. D. (1993): *Mathematical biology.* Springer, Berlin. Second, corrected edition.

Todorovski, L., Džeroski, S. (1997): Declarative bias in equation discovery. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 376–384, Nashville, MA. Morgan Kaufmann.