Mach Learn
DOI 10.1007/s10994-007-5042-6

Inductive process modeling

Will Bridewell · Pat Langley · Ljupčo Todorovski · Sašo Džeroski

Received: 29 August 2005 / Revised: 13 November 2007 / Accepted: 28 November 2007

Abstract In this paper, we pose a novel research problem for machine learning that involves constructing a *process model* from continuous data. We claim that casting learned knowledge in terms of processes with associated equations is desirable for scientific and engineering domains, where such notations are commonly used. We also argue that existing induction methods are not well suited to this task, although some techniques hold partial solutions. In response, we describe an approach to learning process models from time-series data and illustrate its behavior in three domains. In closing, we describe open issues in process model induction and encourage other researchers to tackle this important problem.

Keywords Scientific discovery · Process models · Compositional modeling · System identification · Ecosystem modeling

1 Introduction and motivation

Many scientific and engineering domains involve continuous variables that change over time. The increasing availability of data from such systems presents both an opportunity and a challenge for machine learning. Successful applications of induction methods hold

Editor: David Page.

W. Bridewell (✉) · P. Langley
Computational Learning Laboratory, Center for the Study of Language and Information,
Stanford University, Stanford, CA 94305, USA
e-mail: willb@csli.stanford.edu

P. Langley
e-mail: langley@csli.stanford.edu

L. Todorovski · S. Džeroski
Department of Knowledge Technologies, Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

L. Todorovski
e-mail: ljupco.todorovski@ijs.si

S. Džeroski
e-mail: saso.dzeroski@ijs.si

obvious benefits, and there exist large literatures on computational methods for regression and time-series prediction. But independent of accuracy, the predictive models that these techniques learn from data make little contact with the formalisms and concepts used by scientists and engineers. As Schwabacher and Langley (2001) have argued, domain experts will benefit more from knowledge that is cast in *communicable* forms, utilizing notations already familiar to them. And as Pazzani et al. (2001) have shown, experts in some domains will reject a learning system's output, even when very accurate, unless it makes contact with their prior knowledge.

Research on discovering numeric laws (e.g., Langley 1981; Żytkow et al. 1990; Džeroski and Todorovski 1993; Washio et al. 2000) addresses this concern, in that many scientists find equations familiar. However, the resulting knowledge, which generalizes beyond the training data, is typically *descriptive* in that it directly relates observed variables. In contrast, models in science and engineering often provide *explanations* which include variables, objects, or mechanisms that are unobserved, but that help predict the behavior of the observed variables. Moreover, explanations posit causal structures that link these elements to observed variables and often use general concepts or relations that occur in different models. Compare Kepler's third law, which relates a planet's period to its distance from the sun, with Newton's theory of gravitation, which introduces a theoretical variable for gravitational force. Kepler's model answers a *what* question about planetary motion, whereas the latter answers a *why* question, thereby providing an explanation.

We claim that scientists and engineers often state their explanations in terms of mechanisms (Machamer et al. 2000),[1] which illustrate how *processes* affect the properties of entities. These processes describe one or more causal relations among the entities.[2] We develop here a particular class of processes that are represented in terms of differential equations (for modeling change over time) and algebraic equations (for modeling instantaneous effects). Each such process may also include conditions, stated as threshold tests on its input variables, that describe when it is active. This representation ties the conceptual aspects of the model—how the entities are related—with the mathematical formulations used in systems biology, ecology, and other disciplines. A *process model* consists of a set of processes that link observable variables with each other causally, possibly through unobserved theoretical terms.[3]

Table 1 shows a simple process model for a predator–prey relationship between two protists (microorganisms). In this ecosystem, *Didinium nasutum* preys upon *Paramecium aurelia*. The model includes three processes that explain the changes in the concentrations of the two species. The first of these states that the *aurelia* concentration increases logistically, limited by the environment's carrying capacity, while the second process states that *nasutum* decays exponentially. The third process explains the interaction between the two species using a Holling type 1 response (Holling 1959). Here, the prey population decreases in relation to the size of both populations and the rate of predation. Likewise the predator population increases correspondingly, with an additional term reflecting the efficiency of predation (intuitively, the number of predators produced by consuming one prey). Given initial conditions (e.g., $nasutum = 64.67$, $aurelia = 276.60$), this model can simulate changes in the

[1] Much debate still exists in the philosophy of science over the nature of mechanisms. For instance, see work by Bechtel and Abrahamsen (2005), Glennan (2002), and Woodward (2002).

[2] A causal relation between entities X and Y means that the value of a property of X partially determines the value of at least one property in Y or vice versa.

[3] Our framework can be viewed readily as a quantitative version of Forbus' (1984) qualitative process theory, from which we have borrowed many ideas.

Table 1 A quantitative process model of a protist ecosystem with one predator (*D. nasutum*) and one prey (*P. aurelia*). Prey growth is logistic, predator decay is exponential, and predation is explained using a Holling type 1 response. The notation $d[X, t, 1]$ indicates the first derivative of X with respect to time t

model PredatorPrey;
variables $aurelia\{prey\}$, $nasutum\{predator\}$;
observable *aurelia*, *nasutum*;
process aurelia_growth;
equations $d[aurelia, t, 1] = 1.81 * aurelia * (1 - 0.0003 * aurelia)$;
process nasutum_decay;
equations $d[nasutum, t, 1] = -1 * 1.04 * nasutum$;
process predation_holling_1;
equations $d[aurelia, t, 1] = -1 * 0.03 * aurelia * nasutum$;
$d[nasutum, t, 1] = 0.30 * 0.03 * aurelia * nasutum$;

two species' concentrations over time. Note that, as expected, the model predicts oscillatory behavior, but it does not state this explicitly, as would descriptive laws. For comparison, the model is equivalent to the two differential equations

$$\dot{a} = 1.81 \cdot a \cdot (1 - 0.0003 \cdot a) - 0.03 \cdot a \cdot n,$$
$$\dot{n} = -1.04 \cdot n + 0.30 \cdot 0.03 \cdot a \cdot n$$

where $n = nasutum$, $a = aurelia$, and $\dot{x}$ denotes the first derivative of x with respect to time. Here we assume that the effects of multiple processes on the same variable are additive.

We assume that the specific processes in such a model are instances of some set of generic processes that, taken together, compose background knowledge about the domain. For example, each process in Table 1 instantiates a generic counterpart from Table 2. This direct mapping between the model's equations and the background knowledge provides two benefits. First, the explicit relationship between the two levels of knowledge removes the need to infer such a mapping, which can be difficult even for domain scientists. Second, the knowledge encoded in the generic processes provides an important source of constraints on the specific models one might consider as explanations for a data set.

We maintain that process models of the sort in Table 1 occur frequently in science and engineering, and that inducing them from data is a worthwhile task for machine learning researchers to address. We can state the task of *inductive process modeling* as:

- *Given:* Observations for a set of continuous variables as they change over time;
- *Given:* A set of observed, unobserved, and exogenous (i.e., unpredicted or forcing) variables that the model may include;
- *Given:* Generic processes that specify causal relations among variables using generalized functional forms;
- *Given:* Constraints, such as variable type information, that determine which processes may relate particular variables;
- *Find:* A specific *process model* that, when given initial values for the modeled variables and values for any exogenous variables, explains the observed data and predicts unseen data accurately.

Note that this formulation distinguishes between the generic processes given as input and the specific processes in the induced model, which mention particular variables and para-

Table 2 Five generic processes for population dynamics with constraints on their variables and parameters. The variable type constraints are denoted in braces following the variable's name, while parameter bounds are specified within brackets. The notation $d[S, t, 1]$ indicates the first derivative of S with respect to t

```
library pred_prey;
generic process logistic_growth;
   variables S{species};
   parameters gr[0, 3], ic[0, 0.1];
   equations d[S, t, 1] = gr * S * (1 − ic * S);
generic process exponential_growth;
   variables S{species};
   parameters gr[0, 3];
   equations d[S, t, 1] = gr * S;
generic process exponential_decay;
   variables S{species};
   parameters dr[0, 2];
   equations d[S, t, 1] = −1 * dr * S;
generic process holling_1;
   variables S1{prey}, S2{predator};
   parameters ar[0.01, 10], ef[0.001, 0.8];
   equations d[S1, t, 1] = −1 * ar * S1 * S2;
             d[S2, t, 1] = ef * ar * S1 * S2;
generic process holling_2;
   variables S1{prey}, S2{predator};
   parameters ar[0.01, 10], ef[0.001, 0.8], ht[1, 5];
   equations d[S1, t, 1] = −1 * ar * S1 * S2/(1 + ht * ar * S1);
             d[S2, t, 1] = ef * ar * S1 * S2/(1 + ht * ar * S1);
type species subtypeof number;
type predator subtypeof species;
type prey subtypeof species;
```

meter values. Later, we will see that this background knowledge provides important search constraints through the model space.

Earlier in the paper, we claimed that scientists often explain systems in terms of mechanisms, and the task definition incorporates this emphasis on explanation. In one sense, we consider process models explanatory because they associate pieces of equations with identifiable, causal relationships among entities. We illustrated this notion with the model in Table 1 and the set of equations that it encodes. In a deeper sense, we appeal to Hempel and Oppenheim (1948), who view an explanation as a collection of general laws and situation specific conditions that support the deduction of observations. Although process modeling does not directly map onto their framework, we can view the generic processes as general laws and the model, which incorporates some of the situational conditions, as the formal explanation of observed trajectories. This perspective serves as a primary distinction between inductive process modeling and analysis techniques such as autoregression and recurrent neural networks, which only describe how the measurements vary. The connection between processes and general domain knowledge supplies the explanatory power of the process modeling representation.

Another key point of the task definition is the insistence on simulation from minimal input. That is, the model should predict the behavior of a dynamic system solely from the starting values of all the variables and traces for the exogenous variables (i.e., those that drive the model and that we assume the modeled processes do not affect).

Although it is not a strict requirement on the inductive process modeling task, we make one other assumption that simplifies model construction: the dynamical systems that the models explain are generally viewed as deterministic. The observations themselves may well contain noise, which can complicate matters for this paradigm as it does for others. However, the framework posits that processes are always active whenever their conditions are met and that their equations have the specified effects. Scientists and engineers often treat the systems they study as deterministic, and we will operate under the same assumption.

This paper takes the form of an exploratory research report in the sense described by Dietterich (1990). Following his advice, we state clearly a promising new problem for machine learning and explore its various facets. In the section that follows, we consider some challenges posed by the problem of inducing quantitative process models. We then review a variety of established induction paradigms, concluding that none can be applied directly to this task, though some hold promising ideas on which we can build. After this, we describe a candidate approach to process model induction and illustrate it with some initial results. Finally, we close by suggesting an agenda for future research on this important topic. We do not report extensive experimental studies, but we do present results on three domains as evidence for the generality of both the task and our initial approach to it.

2 The task of inductive process modeling

We have claimed that the induction of process models differs from the problems typically studied in machine learning. Thus, before proceeding further, we review the distinguishing characteristics of this task, with a particular emphasis on the input and output that define the problem. We also discuss the inability of current methods to address quantitative process modeling and identify those approaches that relate most closely to the task.

2.1 Characteristics of inductive process modeling

Since process models characterize the behavior of dynamic systems over time,[4] the data that they explain are more appropriate to time-series regression than to most induction tasks. First, the variables, which represent quantitative measurements of the system under study, are primarily continuous. Second, the observed values are not independently and identically distributed, since those observed at later time steps depend on those measured earlier. Thus, the data violate an assumption made by the great majority of available learning algorithms. Finally, the training data are primarily unsupervised, in that they describe a set of variables that change over time, with no variable being singled out for special attention. Such unsupervised learning is generally viewed as less constrained than learning from supervised data.

In addition to data, process modeling requires as input a collection of generic components that can explain aspects of a system's behavior. Use of this knowledge offsets two difficulties inherent to the problem. First, process models can include theoretical variables, which are unobserved. The generic processes may indicate when such variables can appear and where they fit within the model's structure. Second, the processes that affect a variable are also unobservable. Without domain knowledge, the equations representing these processes could take an infinite number of forms, many of which would be implausible when viewed in the

[4] In some cases, the dynamic systems may be observed in a steady state.

modeled system's context. Hence, the generic processes (along with variable types) limit the model's structural form to ones that plausibly relate the variables.

The use of such domain knowledge leads to mechanistic models that explain the data rather than empirical (i.e., descriptive) ones that merely summarize the trajectories. By referring to processes operating in the environment, we can specify not only *which* variables interact but also *how* they do so. For example, the predator–prey model in Table 1 instantiates the holling_1 predation process from Table 2, which suggests that the amount of time taken for a predator to consume a prey is unimportant. In contrast, a model that includes holling_2 would support a direct influence of the handling time (ht) on the ecosystem's dynamics. Unobserved variables combine with the processes to enhance a model's explanatory power by enabling a rich set of relationships without the need to measure every suspected entity in the environment. *These characteristics of process models address how entities in the world behave as opposed to how measurements vary, and thus differentiate the output of inductive process modeling from the purely descriptive models generated by neural networks, autoregressive techniques, and even traditional methods of equation discovery.*

2.2 Related research

Due to the unique inputs and outputs of process model induction, previous research in machine learning provides little support for the task. For instance, research on *theory revision* appears relevant because it combines background knowledge with inductive learning to improve the predictive accuracy of a given model (e.g., Ourston and Mooney 1990). This framework often states domain theories as Horn clause programs, and the revision methods employ operators for adding and removing rules' conditions, as well as adding and removing entire rules. Although alternative formalisms exist, the paradigm assumes an explanatory model at the outset, rather than constructing one from available components. More importantly, theory-revision methods emphasize learning classifiers from supervised data rather than dealing with unsupervised regression. Thus, this approach seems like a poor match, although we will return to it when we discuss open issues.

Inductive logic programming (e.g., Lavrač and Džeroski 1994) has a somewhat closer connection to the task of learning process models. This framework takes advantage of background knowledge, stated as Horn clauses and ground literals, to learn from training cases. The resulting knowledge is itself cast as a set of Horn clauses, possibly with nonterminal (i.e., theoretical) symbols, and thus can have an explanatory character. But as usually practiced, inductive logic programming employs supervised learning for classification tasks and often generates descriptive rules. Moreover, the research emphasis has been on generating logical structures rather than numerical ones. However, Garrett et al. (2007) have used this approach to infer metabolic pathways that involve biochemical processes and unobserved entities, making them similar to our own models. Importantly, their system learns qualitative models only and requires one to transform quantitative trajectories into qualitative states. Even so, inductive logic programming holds promise as an approach to process model construction, although both the performance and learning methods must be extended to support numeric equations and continuous time.

Linear causal modeling constitutes another relevant paradigm, as it deals with unsupervised quantitative data, produces a causal model, and can even incorporate latent (theoretical) variables.[5] Researchers have developed techniques to both construct (Glymour et al.

[5] Some readers may view linear causal models as Bayesian networks with Gaussian units and additive combining functions, but the literature on them goes back much farther (e.g., Simon 1954).

1987) and revise (Bay et al. 2002) linear causal models from scientific data. Nevertheless, this approach has several drawbacks that limit its usefulness for inductive process modeling. In particular, such models are limited to linear relations, they do not deal with continuous time, and they do not organize causal links into processes. Thus we may be able to incorporate some ideas from linear causal modeling, but we cannot directly apply the basic approach.

Since the task of inductive process modeling involves data appropriate for time-series regression, we should also consider the applicability of ARIMA methods (Box et al. 1994). These produce predictors stated as difference equations cast as a weighted, linear combination of one or more prior observations and random shocks. Given initial conditions, ARIMA equations can produce forecasts of arbitrary length for a fixed-size time step, but, regardless of their accuracy, such models cannot be interpreted as explanations. An ARIMA model provides a concise description of the observed behavior, but it does not offer any hypothesis about the underlying mechanisms. As a result, this approach fails to satisfy a primary requirement of inductive process modeling.

Hidden Markov models (e.g., Poritz 1988) can also describe systems that change over time, which makes them potentially relevant to our learning problem. Such models can play an explanatory role, as evidenced by research in neuroscience that investigates the mechanisms behind ion-channel gating (Zheng et al. 2001), but the emphasis lies on explicit transitions between inferred states, as opposed to the processes from which the state changes emerge. For those fields primarily concerned with causality and relationships among entities as opposed to general system state, hidden Markov models provide a poor conceptual match. In addition, the probabilistic assumptions of Markov models are unnecessary for scientific and engineering domains that treat nature as deterministic.[6] Therefore this approach fails to fulfill the representational needs established by inductive process modeling.

Superficially, inductive process modeling appears closely related to work on equation discovery (e.g., Langley 1981; Langley et al. 1987; Washio et al. 2000), in that both produce numeric equations cast in the same forms that scientists use. However, such methods focus on inferring descriptive models that summarize and predict the data without explaining them, much as occurred in the early days of physics and chemistry. Research on the discovery of differential equations (e.g., Džeroski and Todorovski 1993; Todorovski and Džeroski 1997; Todorovski 2003) deals with change over time, but it does not produce models that include theoretical variables or, more importantly, that group terms into processes familiar to domain scientists. In general, work in this tradition generates understandable models but does not formulate explanatory accounts from candidate structures, making it quite distinct from inductive process modeling.

Research on system identification (e.g., Åström and Eykhoff 1971) also has little relevance to our concerns, for similar reasons. Here the standard task is to infer parameters for a system of differential equations to reproduce and predict observed time series, which could be useful for a component parameter estimation task but not the overall problem of process model induction. Methods for structural system identification (e.g., Bradley et al. 2001) also induce the forms of differential equations, but this approach lacks a means for encoding process information and requires only that the models be consistent with observed behavior. Work on hybrid systems (e.g., Ghosh and Tomlin 2001) includes methods that construct finite-state models with embedded differential equations, which lets them provide

[6] A similar argument holds for dynamic Bayesian networks (e.g., Ghahramani 1998), which have a similar flavor.

somewhat deeper accounts of observations, but, like hidden Markov models, they focus on system states rather than mechanistic processes.

Perhaps the most relevant work comes not from machine learning but from qualitative physics. We have already mentioned how our representation of models draws on Forbus' (1984) qualitative process theory by extending his qualitative formalism into a quantitative one. Moreover, Forbus and Falkenhainer (1990) report an approach to compositional modeling, which selects and combines known model fragments to produce desired qualitative behavior. The resulting structures explain the behavior in terms of component processes, much as required for the inductive process modeling as we have defined it. However, we must extend this idea to compose and parameterize quantitative processes that account for detailed trajectories, which involves additional challenges.

In summary, no existing approach addresses all the issues that arise in process model induction, which suggests that we need novel learning methods to solve this new problem.

3 Inducing process models

Our primary aim in this paper is to both characterize and advocate the problem of inducing process models from background knowledge and data. To this end, we have built a baseline system that provides evidence for the task's feasibility and that can serve as a nontrivial comparator for future work. Following common practice, we describe our system in terms of the formalism that encodes the resulting models, the inputs that drive learning, the performance element that uses them, and the learning method that constructs the models. We close the section with a methodology for evaluating inductive process modelers.

3.1 Representing background knowledge and models

We have already seen one example (Table 1) that involves a three-process model for a protist ecosystem. To reiterate, a process model specifies a set of processes that characterize quantitative relations among a set of observed, and possibly unobserved, variables. Each process includes zero or more conditions under which it is active, along with at least one causal equation that characterizes the influence that one or more variables exert on another. Although the processes in a given model are unordered and operate in parallel, we can organize them into a causal graph that equates the outputs of some processes with the inputs of others (Iwasaki and Simon 1994).

Table 2 presents a set of processes for population dynamics, which concerns changes in species' population levels over time (Murray 2004). The table contains *generic* processes that serve as background knowledge for learning; unlike *specific* processes, these do not commit to particular variables or parameter values, but they can indicate constraints on them. For example, the process for exponential growth states that its variable S must have type *species* and that its equation's coefficient gr must fall between zero and two. The background knowledge also contains a hierarchy of variable types that stem from the base type *number*. Note that processes in this domain include no conditions, so they are continuously active.

In addition to the generic processes, an inductive process modeler needs a set of typed variables and training data for those declared observable. For example, to construct the model in Table 1, the program would require *nasutum* to have type *predator* and *aurelia* to have type *prey*. Combined with the set of generic processes, this information defines the space of model structures that the induction system will search. Since both variables are

observable, two trajectories must be supplied. These data provide a means both to direct the search for parameters and to evaluate the instantiated process model.

As one can see, both the input and the output of inductive process modeling differ from those of other induction tasks. The richness of the background knowledge and the use of temporal data support the development of detailed causal models that carry more information than simpler influence diagrams. As a result, a quantitative process model can explain observed trajectories for its variables, which lets scientists analyze the effects of system perturbation and infer the behavior of theoretical entities.

3.2 Making predictions with process models

To produce the trajectories useful to scientists, one needs a performance element that can use the learned knowledge. In this case, we require an interpreter that can generate a predicted trajectory for each observable variable by carrying out forward simulation of a quantitative process model. To this end, we have implemented a module that operates in two phases. The first of these combines the components from each of the processes into a system of differential and algebraic equations.[7] During this conversion, the system ensures both that the algebraic equations will be solved according to their causal ordering and that the conditions are correctly associated with the equations.

The module's second phase evaluates the model using CVODE, an established method for solving first-order differential equations (Cohen and Hindmarsh 1996), which we couple with basic arithmetic operations for handling the algebraic equations. For this phase, we must provide:

- initial values for observed and unobserved variables,
- time series for the exogenous variables,
- specification of the simulation time points.

From this input, the simulator uses the initial values to solve the system of equations for the second time point as determined by the temporal resolution. The output of this step and the values of the exogenous variables at this time serve as the input for the third point, and so on until the program reaches the specified end time. This iterative procedure produces a trajectory for each variable, such as those Fig. 3 shows in Sect. 4.2.[8]

3.3 Constructing a model from components

Once provided with background knowledge, which includes generic processes and a type hierarchy for variables, a collection of variables to be modeled, and time-series data about the particular quantitative variables one wants to explain, an induction system can carry out constrained search through the space of process models. We have implemented such a system, called IPM, wherein the search mechanism operates in three distinct stages.[9]

[7] As we mentioned in Sect. 1, the system assumes that, when multiple processes influence the same variable, the effects are additive.

[8] Numerical solvers such as CVODE provide approximate solutions to systems of nonlinear differential equations. Although these tools take great care in the assessment and control of error, one should be aware of their limitations, which generally follow from limited-precision calculations. For the reported work, we use of CVODE's error control features, but leave an detailed study of their effects for future research.

[9] IPM serves as one component of PROMETHEUS, an integrated environment for process modeling that we have described elsewhere (Bridewell et al. 2006).

Table 3 Permissible bindings of generic processes from the population dynamics domain with the variables *nasutum* of type *predator* and *aurelia* of type *prey*

logistic_growth:	$S \rightarrow aurelia$	exponential_decay:	$S \rightarrow nasutum$
logistic_growth:	$S \rightarrow nasutum$	holling_1:	$S1 \rightarrow aurelia$
exponential_growth:	$S \rightarrow aurelia$		$S2 \rightarrow nasutum$
exponential_growth:	$S \rightarrow nasutum$	holling_2:	$S1 \rightarrow aurelia$
exponential_decay:	$S \rightarrow aurelia$		$S2 \rightarrow nasutum$

Table 4 A generic model from the predator–prey domain

process logistic_growth;
  parameters $gr[0, 3], ic[0, 0.1]$;
  equations $d[aurelia, t, 1] = gr * aurelia * (1 - ic * aurelia)$;
process exponential_decay;
  parameters $dr[0, 2]$;
  equations $d[nasutum, t, 1] = -1 * dr * nasutum$;
process holling_1;
  parameters $ar[0.01, 10], ef[0.001, 0.8]$;
  equations $d[aurelia, t, 1] = -1 * ar * aurelia * nasutum$;
  $d[nasutum, t, 1] = ef * ar * aurelia * nasutum$;

In its first stage, IPM finds all the permissible instantiations of the generic processes with the specified variables. For instance, the exponential_growth process from Table 2 requires a variable with type *species*. Consider the two variables *nasutum* with type *predator* and *aurelia* with type *prey*. Since the types of these variables are subtypes of *species*, IPM will create two instantiations of the exponential_growth process—one for each variable binding. Table 3 shows all eight permissible bindings between the two variables and the generic process library given in Table 2. After creating these bindings, IPM must determine which processes a model should include and what values the associated parameters should take.

The program's second stage uses subsets of this collection of partially instantiated processes to form *generic models*, each of which specifies an explanatory structure. In IPM, each permissible binding is an element in a set P, and each member of the power set of P is a generic model. The system carries out an exhaustive search of model structures by enumerating this power set, retaining as candidate models only those members that satisfy user-provided constraints, which currently includes the maximum number of processes in the model and the list of generic processes that must be instantiated. Table 4 shows one valid generic model built from the bound processes of Table 3. Here all the variables are bound, but the parameters lack specific values.

The third stage infers the parameter values for each generic model using two estimation strategies. At the core of this stage, IPM employs a nonlinear least-squares algorithm (Bunch et al. 1993) that carries out a second-order gradient descent search through the parameter space. For the first strategy, the program guides this estimation routine by simulating the full set of trajectories from initial values and providing information about the residuals for each observable variable. Search terminates based on a set of convergence criteria described by Dennis et al. (1981) and Gay (1983). To further avoid entrapment in local minima, IPM restarts the estimation routine from multiple random points in the parameter space.

When all the variables are observable, IPM augments the above search with a second strategy that uses *teacher forcing* (Williams and Zipser 1989), an alternative method that

does not evaluate full simulations but rather aims to minimize the error in predicting values at time $\mathcal{T}_{i+1}$ from those at $\mathcal{T}_i$. Given n samples, the program performs $n - 1$ one-step simulations, forwarding the information about the residuals to the same nonlinear least-squares algorithm mentioned above. Since local minima can still cause problems, IPM restarts the teacher forcing approach at multiple randomly selected points. Experience suggests that this technique finds a good region of the parameter space from which to start the more costly search based on full simulations, so the parameters selected by this technique seed one of the restarts for the full-simulation strategy.

From among the results for each estimation run, IPM selects the best set of parameters according to a fitness measure. The system currently uses the sum of squared error (SSE) for this step, which is defined as

$$\sum_{i=1}^{n} SSE(x_i, x_i^{obs}) = \sum_{i=1}^{n} \sum_{k=1}^{m} (x_{i,k} - x_{i,k}^{obs})^2$$

for variables $x_1, \ldots, x_n$ with m observed values each. To support modeling variables of different scale, one can use the relative mean squared error, which is defined as

$$\frac{\sum_{i=1}^{n} \frac{SSE(x_i, x_i^{obs})}{s^2(x_i^{obs})}}{nm},$$

where $s^2(x_i^{obs})$ gives the unbiased sample variance of the observations for x_i. Like SSE, lower scores on this measure indicate better fit, but due to the rescaling, we can compare values across data sets. In addition, a relative mean squared error of 1.0 indicates that the model performs as poorly (or as well) as predicting the mean value of the trajectory. Measures that account for the parametric complexity of a model (e.g., the Bayesian information criterion), are also reasonable but they would require a different parameter estimation routine due to the input required by Bunch et al.'s (1993) method.

Notably, IPM also lets one treat the initial values of the simulation as parameters. Often scientists and engineers can give only plausible ranges or approximate values for theoretical variables. In these cases, allowing variability in the initial values can help the program achieve a better fit to the observed trajectories. Moreover, noise can exist in the measurements of observable variables, so the system can also treat their initial values as parameters. Upon completion of its search, IPM returns a list of models, which are associated with their error scores and initial values.

Since IPM carries out an exhaustive search of model structures, the system's power rests in its knowledge of the modeled domain (i.e., the list of generic processes) and in the capabilities of its parameter estimation routine. As the number of model structures increases, exhaustive exploration becomes prohibitive, and one must either define other constraints to reduce this number or develop new search strategies appropriate to the space of nonlinear, dynamic models. Section 4.3 illustrates the role of additional structural constraints, but we leave the development of sophisticated search algorithms to future research.

3.4 Evaluation of inductive process modeling

To contribute to understanding a dynamic system, an induced process model should be comprehensible, plausible, and accurate. The first two of these characteristics reflect human judgment, and are both subjective and domain dependent. In this paper, we touch on these aspects briefly and somewhat anecdotally, choosing to look more closely at measures

Table 5 An alternative set of generic processes that capture the interaction between predators and prey

```
generic process predation;
   variables S1{prey}, S2{predator}, R{rate};
   parameters ef[0.001, 0.8];
   equations d[S1, t, 1] = −1 ∗ R ∗ S2;
             d[S2, t, 1] = ef ∗ R ∗ S2;
generic process holling_1;
   variables S1{prey}, R{rate};
   parameters ar[0.01, 10];
   equations R = ar ∗ S1;
generic process holling_2;
   variables S1{prey}, R{rate};
   parameters ar[0.01, 10], ht[1, 5];
   equations R = ar ∗ S1/(1 + ht ∗ ar ∗ S1);
```

of quantitative accuracy. The objectiveness of fitness scores suggests their use in ranking process models and evaluating process-modeling systems. After describing comprehensibility and plausibility in more detail, we outline an evaluation methodology based on model accuracy.

3.4.1 Comprehensibility and plausibility

We separate comprehensibility into structural and syntactic components, whereas we stress that plausibility is a function of structure alone. Support for the *syntactic* comprehensibility of process models follows from our claim that a modeling notation must be familiar to scientists. To this end, we based the process modeling language on the systems of differential and algebraic equations that scientists often use to explain observed phenomena. Moreover, the language is similar enough that one could build a model by placing an entire system of equations into a single process with the minimal syntactic overhead of providing variable declarations and a process name. We claim that the added ability to associate parts of equations with the processes that they represent increases comprehensibility by clarifying the mechanism of the modeled system.

One's choices about which processes to represent and how they behave affect a model's *structural* comprehensibility. Scientists may differ in their beliefs along these dimensions, but a capable representation will help them communicate the differences in their views. To illustrate, in Table 2 we represented predation in the generic processes holling_1 and holling_2. Alternatively, we could have introduced a separate predation process and isolated those equation elements that set the rate of consumption, as shown in Table 5. The two decompositions are mathematically equivalent but structurally different. Nevertheless, the names of the processes and the participating variable types help elucidate the meaning and organization of these two libraries. Our experience with ecologists and other biologists suggests that the quantitative process modeling formalism is comprehensible in terms of both syntax and structure (Bridewell et al. 2006). In particular, researchers familiar with differential equations and their use in biological modeling needed only a brief orientation to the language.

As with the comprehensibility, plausibility refers to an instantiated model's structure, in that implausible models violate constraints on the processes that may appear. For example, the explicit requirement that an exponential loss process occur in a model renders structures without that component invalid and hence implausible. In addition to the constraints

supported by IPM, scientists place other limitations on model structure that our current representation cannot capture. To illustrate, an ecologist may assert that a population dynamics model must contain either an exponential or a logistic growth process but not both. However, IPM lacks a means for specifying this type of knowledge and will consider structures that violate this constraint. In Sect. 4 we include a brief discussion of model plausibility in three problem domains and report on extensions to IPM's constraints.

3.4.2 Quantitative accuracy

Whereas comprehensibility and plausibility are subjective measures, quantitative accuracy provides an objective means for model evaluation. For inductive process modeling, we posit two approaches to assess accuracy: one based on interpolation and the other based on extrapolation. Briefly, interpolation involves the removal of randomly selected measurement vectors from the given time series. One then trains on the remaining points and predicts (or postdicts) the points that were held out for testing. In comparison, extrapolation uses a fixed boundary such that one uses the measurements taken before the selected time for training and holds out those following it for testing. We use both approaches as the basis of an experimental methodology for inductive process modeling.

For the interpolation task, we modify k-fold cross validation to make it applicable to time-series data. Given a data set composed of multivariate temporal trajectories, we remove the values for the initial time point $\mathcal{T}_0$ and generate the k subsets by filling them with the remaining data selected uniformly at random and without replacement, as in standard cross validation. We then create the k folds, prepending observed values for $\mathcal{T}_0$ to each training set. After IPM produces a model, it simulates the trajectories using its own estimated values for $\mathcal{T}_0$ and overlays each trajectory onto the appropriate test set. This procedure temporally aligns the interpolated values with the observed ones. Thus we measure the model's ability to predict values in the past based on surrounding observations. Research in archeology and paleontology provides a useful metaphor, where scientists reconstruct the communal or biological occurrences that fall between data from multiple eras. By adjusting the value of k, we control the amount of training data.

To evaluate according to extrapolation, we identify boundaries between the training and test data by percentage. For instance, given some time series, we can train on the first half of the observations and test on the second half. Shifting the boundary forward in time will help determine the effect of adding more training data on forecast accuracy. With this method, we gauge performance on the test set by simulating forward from $\mathcal{T}_0$ of the training data through the end of the trajectories. Models that incorporate exogenous variables present a curious problem in that the future values of those variables are generally unknown. However, for purposes of evaluation, we assume the availability of these values during testing and issue a caveat that the use of exogenous trajectories simulated by a separate model may introduce further error.[10]

These approaches assess a model and a modeling approach according to two of the many aspects on which one might judge it. Interpolation reveals how well IPM can learn from sparse data and to what extent more frequent sampling will improve its ability to learn an

[10]The proposed approach to extrapolation conflates three experimental factors. That is, increasing the amount of training data both decreases the amount of testing data and alters the boundary between the two sets. Using fewer test data reduces a model's burden to be accurate. In addition, some regions of a trajectory may be generally more difficult to predict than others, so a model need not be as accurate if the hold-out boundary's position is in a "simple" section of the data. We do not explore these issues in detail for this paper.

accurate model. Extrapolation provides a strong test of a model's suitability for forecasting purposes. Although we emphasize these tasks in this paper, we note that they form a single step in the argument for a model's general utility, which includes appeals to acceptability, plausibility, and generalization to qualitative phenomena not used to guide model development.[11] In the next section, we evaluate IPM's quantitative performance on these two criteria in three scientific domains.

4 Applications and results

In previous sections, we characterized the task of inductive process modeling and described one approach to it. In this section, we report results for our current implementation on three scientific domains. In the first we apply an extension of the generic processes given in Table 2 to predator–prey data. The second domain centers on the population dynamics for the aquatic ecosystem of the Ross Sea. The final modeling task involves the dynamics of Ringkøbing Fjord as its water level responds to various environmental conditions. In each case, we discuss both the inputs to IPM and the models it produces. We judge success based on predictive accuracy and model comprehensibility.

4.1 Predator–Prey interactions

The relationships between predators and their prey serve as the core aspect of many ecological models. Researchers believe that such interactions drive the evolution of the species involved, leading to the development of defense mechanisms in the prey and corresponding adaptations in the predators. In addition, by understanding predation one gains knowledge about sustainability, which can determine the effects of species introduction or removal upon a particular ecosystem. Research in this area effectively began in the last century through the work of Lotka and Volterra, as Berryman (1992) has recounted.

The Lotka–Volterra model of predation effectively assumes three fundamental processes: prey growth, predator decay, and predation. We can state the original formulation as

$$\dot{F} = \gamma F - \alpha F C,$$
$$\dot{C} = \epsilon \alpha F C - \delta C,$$

where F represents the prey population and C the predator population. Here the growth rate, γ, controls the exponential increase of prey over time, whereas the decay rate, δ, similarly specifies the natural decrease of the predator population. Predation correspondingly decreases the number of prey while increasing the number of predators, as influenced by the density of both populations and a fixed attack rate, α. An efficiency factor, ϵ, corresponds to the ratio of predators produced with respect to the number of prey consumed, so that smaller values slow predator growth. Since the introduction of this model, ecologists have suggested a variety of alternative forms with different mathematical expressions for the growth, loss, and predation processes.

For our experiments, we combined the partial library in Table 5 with the generic processes in Table 6 and others to define two types of growth for the prey, the predation process, 13 ways to determine the predation rate, and one type of loss for the predator. Both

[11] See Arrigo et al. (2003) for a paradigm of such an argument.

Table 6 Additional generic processes for the predator–prey domain. Variable type constraints are denoted in braces following the local name, while parameter bounds are specified within brackets

```
generic process ratio_dependent_2;
    variables S1{prey}, S2{predator}, R{rate};
    parameters ar[0.01, 10], ht[1, 5];
    equations R = ar * S1 * S2/(S2 + ht * ar * S1);
generic process ivlev;
    variables S1{prey}, S2{predator}, R{rate};
    parameters delta[0.001, 1], sat[0.0001, 1];
    parameters ar[0.01, 10], ht[1, 5];
    equations R = S2 * sat * (1 - e^(-delta*S1));
generic process hassell_varley_1;
    variables S1{prey}, S2{predator}, R{rate};
    parameters sat[0.0001, 1], mu[1, 100];
    equations R = sat * S1 * S2^-mu * S2;
generic process deangelis_beddington;
    variables S1{prey}, S2{predator}, R{rate};
    parameters ar[0.01, 10], ht[1, 5], mi[0, 1];
    equations R = ar * S1 * S2/(1 + ht * ar * S1 + mi * S2);
generic process crowley_martin;
    variables S1{prey}, S2{predator}, R{rate};
    parameters ar[0.01, 10], ht[1, 5], mi[0, 1];
    equations R = ar * S1 * S2/((1 + ht * ar * S1) * (1 + mi * S2));
```

exponential growth and loss come from the Lotka–Volterra model, whereas logistic growth restates the Malthus–Verhulst equation. We took most of the functional forms for the predation rate from Jost and Ellner (2000), who place them into two general categories. The Holling and Ivlev processes treat predation solely as a function of prey density, while the rest also incorporate the predator density. No single form of these processes sufficiently accounts for every predatory interaction, so the explanation of an observed system requires a search through the space of model structures in addition to parameter estimation.

We provided IPM with the expanded set of generic processes along with data for the protist ecosystem introduced in Sect. 1. These data, originally collected by Vellieux (1979), consist of twice daily recordings of the densities of both *Didinium nasutum* (the predator) and *Paramecium aurelia* (the prey).[12] We selected two data sets that consist of 71 and 52 recordings each. Visual inspection of the data indicated irregularities during the first few days of both experiments, which suggested that different regimes were operating at those times. As a result, we truncated the data sets to 54 and 28 samples, respectively, concentrating our attention on the more regular, though still noisy, sections of the trajectories. Figure 1 shows the full trajectories, along with lines that indicate points of truncation. Throughout this section, we refer to the truncated data in Fig. 1(a) as pd-a and that in Fig. 1(b) as pd-b.

To evaluate IPM, we used the technique described in Sect. 3.4. Specifically, we carried out two-, three-, four-, and five-fold cross validation with the system on each of the Vellieux data sets to determine the effect of training set size on interpolative performance. We also

[12] Jost and Ellner (2000) extracted the data from the original paper and made them publicly available at http://www.pubs.royalsoc.ac.uk as an appendix.

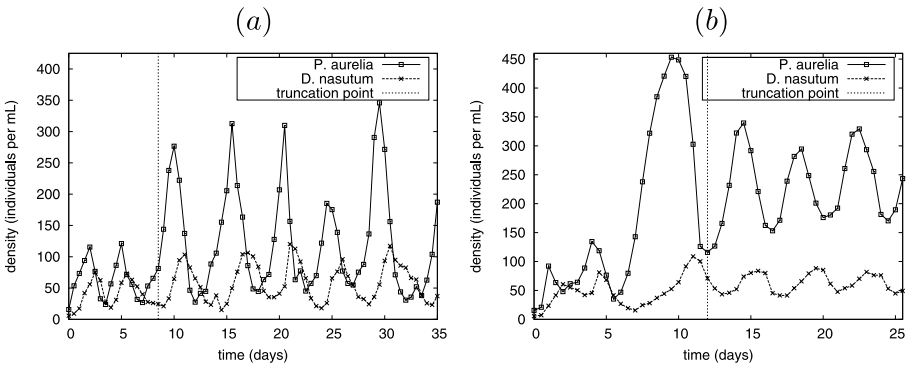

Fig. 1 Trajectories for two predator–prey data sets. Samples to the *left of the vertical lines* were not used because they appeared to involve different regimes

carried out forecasting experiments, in which we set the hold-out boundary at 10% intervals from 50% and 90%. Finally, we had IPM build a model from each complete data set. During training, the program searched for the model with the lowest relative mean squared error. In each case, we told it to search for models with four or fewer processes. The size limitation stems from domain knowledge about the nature of predation; that is, an explanatory model would typically include one process for each of prey growth, predator loss, and predatory interaction. By decoupling the predatory interaction process from the calculation of the predation rate, we expect plausible models to contain no more than four processes. The resulting search space consisted of 3,214 structures.

Training IPM on all the data in pd-a led to a model with a relative mean squared error (reMSE) of 0.35 and a coefficient of determination,[13] or r^2, of 0.72. In comparison, the program's best fit to pd-b had an reMSE of 0.097 and an r^2 of 0.98. Figure 2 shows the values of these measures as they vary with the amount of training data in the interpolation and extrapolation experiments.

The graph on the left of Fig. 2 shows the results of the interpolation task. There are two primary findings from these experiments. First, notice that the number of cross-validation folds has no substantial effect when fitting the data from pd-b. Moreover, the reMSE for the two-fold case, in which we reserved 50% of the data for testing in each fold, was 0.13 and the r^2 was 0.97. Both values are fairly close to the fitness of a model trained on all of pd-b, and they are roughly identical to the five-fold case. This observation suggests that IPM is very robust to data sparsity. The second finding is that, in the face of more challenging data, no discernible trend surfaces that would relate fitness to sparsity. The scores associated with pd-a reach their best values in the three-fold case, where IPM induced models from 2/3 of the data, but they decline thereafter. An experiment using ten-fold cross validation, not shown in the figure, produced an reMSE of 0.40 and an r^2 of 0.66. These values are slightly better than those for five-fold cross validation, but remain worse than those from the three-fold run. This finding further suggests the robustness of the system to sparse data.

[13] For IPM, the coefficient of determination is the square of the correlation coefficient. As such, values range between 0.0 and 1.0, where a larger value indicates that the model accounts for more of the variance in the data. Higher values of r^2 tend to reflect that a simulation better fits the shape of an observed trajectory. In comparison, reMSE is unbounded, but predicting the mean value results in a score of 1.0, and lower scores indicate better fit.

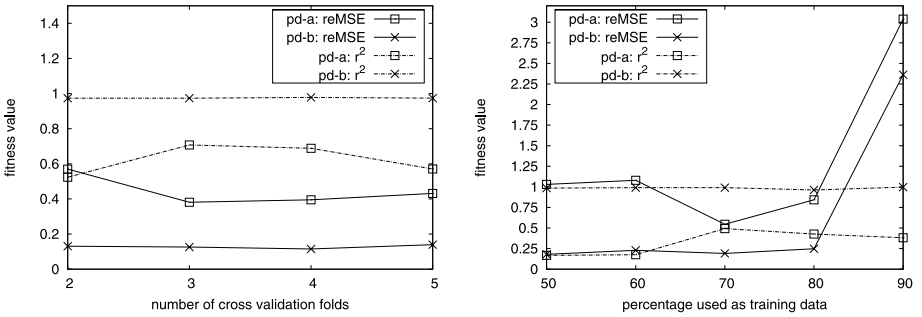

Fig. 2 The effect of the amount of training data on model fitness in the population dynamics domain. Results from cross validation appear *on the left* and those from forecasting appear *on the right*

The graph on the right of Fig. 2 shows the results of the forecasting task. In the associated experiments, we judged how the amount of training data affects the forecasting capabilities of the induced models. Before we discuss the results, we note that the high reMSE scores associated with training on 90% of the data result from a poor estimate of variance in both pd-a, where the test set consisted of five points, and pd-b, which contained only two observations for each variable. To be more specific, the variance of the *D. nasutum* observations was an order of magnitude smaller than the value calculated from the complete trajectory. If we use the variance from the full trajectory, the fitness scores fall in line with the others in the graph. In addition, we recognize that the length of the forecast may interfere with the actual predictive accuracy of a model (i.e., we expect forecasts that reach further into the future to be less reliable than those that predict a more local change), but the results in this study give a reasonable indication of IPM's capabilities.

The main finding from the extrapolation experiments is that, as one might expect, forecasting power tends to decrease with the amount of training data. The effect surfaces in the experiments on pp-a when IPM uses only 70% of the training data. Interestingly, the experiments for pd-b indicate exceptional performance in extrapolation even in conditions of minimal data. Given this result, we ran further tests to see at what point IPM was unable to produce a model of reasonable quality. With 70% held out, the best model gives an reMSE of 0.64 and an r^2 of 0.89, both of which are still reasonable. To give an idea of how much data were used, the hold-out boundary in this condition fell just before the first trough in the prey measurements (see *P. aurelia* in Fig. 1(b)). Finally, we held out 80% of the data, which puts the boundary in the middle of the first prey peak. The best model from this experiment produced an reMSE of 4.90 and an r^2 of 0.39 on the test data, which indicates that the model performed worse than predicting the average values.

For explanatory models, predictive accuracy provides only a partial means for gauging a model's utility. One must also evaluate its plausibility in terms of its adherence to theoretical constraints. Consider the quantitative process model shown in Table 7. The structure consists of a growth process for the prey, a loss process for the predator, a predation process, and a process that sets the rate of predation. Apart from the use of logistic growth instead of the exponential form, this model corresponds directly to the Lotka–Volterra system of equations. The corresponding model for data set pd-b was similar, although it used an alternative form for the predation rate.

That IPM learns a plausible model in this case is not surprising. Recall that one may both restrict generic processes to act on variables of certain types and limit the number of instantiated process in a model. For these experiments, we limited the growth processes to

Table 7 The model with the best fit for pp-a

model pd_a;
variables $aurelia\{prey\}$, $nasutum\{predator\}$, $predation\_rate\{rate\}$;
observable *aurelia*, *nasutum*;
process logistic_growth;
  equations $d[aurelia, t, 1] = 1.62259 * aurelia * (1 - 0.00023 * aurelia)$;
process exponential_loss;
  equations $d[nasutum, t, 1] = -1 * 1.10775 * nasutum$;
process predation;
  equations $d[aurelia, t, 1] = -1 * predation\_rate * nasutum$;
    $d[nasutum, t, 1] = 0.34053 * predation\_rate * nasutum$;
process holling_1_rate;
  equations $predation\_rate = 0.02628 * aurelia$;
initial conditions: $aurelia = 71.25362$, $nasutum = 27.40818$

variables with type *prey*, the loss process to those with type *predator*, and the model to at most four processes. Due to the properties of the equations, a model lacking alternatives for any of the four that occur in Table 7 could not produce oscillations and would provide a poor fit to either data set. The important aspect of this analysis is that IPM's means for constraining the model space suffices to ensure a plausible structure in this scenario.

4.2 The Ross Sea ecosystem

Although studying isolated pairs of species can provide useful information about population dynamics, most ecosystems comprise complex interrelationships among multiple species and energy sources. In these environments, processes such as growth and loss exhibit more complicated forms. In particular, remineralization, which replenishes crucial nutrients, results from organism decay, while the availability of various energy sources may limit species growth. To aid our task, we worked directly with experts in oceanography and ecology to determine the required knowledge for modeling an aquatic ecosystem.

Table 8 presents a few of the resulting generic processes that relate nutrients, animals, plants, and environmental factors. At an abstract level, an aquatic ecosystem consists of plants (*p_species*) that absorb available nutrients (*fe_nutrient* and *n_nutrient*) at a rate mediated by environmental conditions. In addition, animals (*z_species*) graze on the plants, and a portion of any dead organic matter becomes part of the detrital pool. Finally, the processes of remineralization, which converts detritus into its constituent nutrients, and mixing, which introduces nutrients from lower depths of the ocean, ensure a continued source of food for the plants.

We used a superset of the processes in Table 8 to develop a model of phytoplankton growth in the Ross Sea—an environment that holds particular interest for ecologists (e.g., Arrigo et al. 2003) both for its relative simplicity and for the size of its phytoplankton blooms. The data included measurements of nitrate and phytoplankton concentrations from cruises in the Southern Ocean, which the domain experts interpolated to produce values for each of 188 days. We also used daily satellite measurements of the sea surface temperature, the available light, and the percentage of ice coverage as forcing data for exogenous terms. In addition to the measured variables, we introduced theoretical ones that could help explain the observed behaviors. For example, we told IPM to consider models that include zooplankton concentration even though we lacked data for the grazers and were unsure of the

Table 8 A partial set of generic processes for an aquatic ecosystem. All variable types are direct subtypes of *number*

generic process producer_growth;
  variables $P\{p\_species\}, G\{p\_grate\}, L\{p\_glimit\}, T\{temperature\}, C\{ice\}$;
  parameters $max\_growth\_rate[0.1, 0.8]$;
  equations $G = max\_growth\_rate * exp(0.06933 * T * (1 - C) * L$;
    $d[P, t, 1] = G * P$;
 # 12.0107 g/mol is the atomic mass of carbon.
generic process nitrate_uptake;
  variables $P\{p\_species\}, N\{n\_nutrient\}, ntoc\{n\_const\}, G\{p\_grate\}$;
  equations $d[N, t, 1] = -1 * 1/(ntoc * 12.0107) * G * P$;
generic process iron_uptake;
  variables $P\{p\_species\}, F\{fe\_nutrient\}, fetoc\{fe\_const\}, G\{p\_grate\}$;
  equations $d[F, t, 1] = -1 * 1/(fetoc * 12.0107) * G * P$;
generic process grazing;
  variables $Z\{z\_species\}, P\{p\_species\}, D\{detritus\}, ZR\{z\_rate\}, B\{ratio\}$;
  parameters $efficiency[0.05, 0.4]$;
  equations $ZR = 0$;
    $d[Z, t, 1] = efficiency * ZR * Z$;
    $d[P, t, 1] = -1 * ZR * Z$;
    $d[D, t, 1] = (1 - B) * (1 - efficiency) * ZR * Z$;
generic process monod;
  variables $ZR\{z\_rate\}, P\{p\_species\}$;
  parameters $delta[0, 10], max\_grazing\_rate[0.3, 0.5]$;
  equations $ZR = max\_grazing\_rate * P/(delta + P)$;
generic process zoo_loss;
  variables $Z\{z\_species\}, D\{detritus\}, B\{ratio\}$;
  parameters $loss\_rate[0, 0.5]$;
  equations $d[Z, t, 1] = -1 * loss\_rate * Z$;
    $d[D, t, 1] = (1 - B) * loss\_rate * Z$;
generic process nitrate_mixing;
  variables $N\{n\_nutrient\}, T\{temperature\}$;
  parameters $max\_mixing\_rate[0.000001, 4], avg\_deep\_conc[31, 32]$;
  equations $d[N, t, 1] = (avg\_deep\_conc - N) * max\_mixing\_rate *$
    $(datamax(T) - T)/(datamax(T) - datamin(T))$;
generic process nitrate_limitation;
  variables $N\{n\_nutrient\}, L\{p\_glimit\}$;
  parameters $lambda[0.0000001, 1]$;
  equations $L = N/(N + lambda)$;

extent of their effect on the phytoplankton concentration. We also included various terms, such as the growth rate of phytoplankton, that correspond to concepts in ecological theory.

We evaluated IPM on the Ross Sea data using the same interpolation and extrapolation experiments described in Sect. 4.1. During training, IPM searched for the model with the lowest relative mean squared error. The library contained a total of 15 generic processes, each of which had a single valid binding to the given variables. To reduce the structural

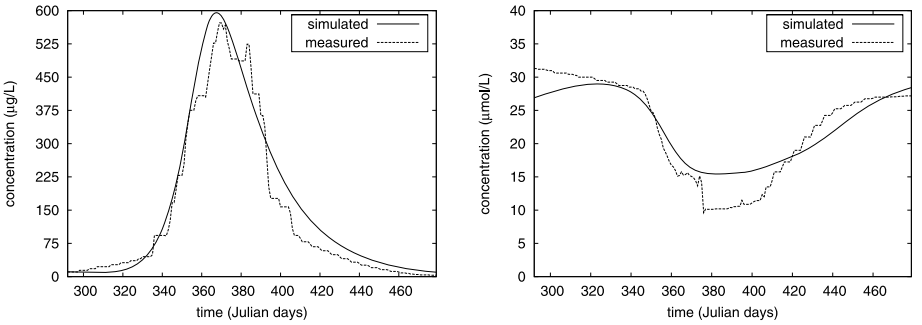

Fig. 3 Simulated and observed trajectories of phytoplankton (*left*) and nitrate (*right*) concentrations for the 1996–1997 Ross Sea bloom

search space, we identified a subset of generic processes that must occur in the model. As an illustration, the phytoplankton population must increase to fit the data. This requirement implies that the model contains an instantiation of producer_growth. In all, we forced the use of five generic processes and left the remaining ten optional, which resulted in a search space containing 1,024 structures. Finally, we altered the default combining scheme for the growth limitation variable to take the minimum of the influences. To clarify, each limiting factor (nitrate, iron, light) affects phytoplankton independently and, at any specific time, one of these will provide the strongest limit on growth. This relationship is not captured by summation, and although one can introduce auxiliary variables for each of the limiting factors and explicitly take the minimum, that solution would introduce other complexities.[14] The model produced from all the data had an reMSE score of 0.098, an r^2 value of 0.966, and led to the trajectories shown in Fig. 3.

IPM's scores on the interpolation task matched those from the full data. Specifically, reMSE ranged between 0.093 and 0.099, whereas r^2 fell between 0.959 and 0.965. These results, coupled with those from the population dynamics domain, suggest that the program is robust to data sets where values are missing at random.

The results on extrapolation tell a more complicated story. To begin, the reMSE scores for the Ross Sea are generally uninformative. In each experimental condition, IPM underestimated the variance of phytoplankton in the test data by one or more orders of magnitude. This result comes about because much of the test data resides in the tail of the bloom where the values vary by relatively minor amounts. The effect is strong enough that it eclipses any negative effect caused by the use of fewer training data and longer forecasts. To adjust for this problem, we present the root mean squared error (RMSE), which can show trends within a particular modeling task even though single values are less interpretable. Figure 4 shows the RMSE and the r^2 scores of the models returned by IPM.

Both the r^2 and RMSE scores show a decline in accuracy when IPM uses less training data (and more testing data). Before discussing the anomalously low r^2 score at 40%, we look more closely into the steep increase in RMSE. The data are noticeably less dynamic toward the end of the trajectories, so we expect that the bulk of the error surfaces in the

[14] Since the number of limiting factors would vary depending on the presence of limitation processes in the model, the program would require either a means to support variable numbers of participants in a process or several processes, each of which accounts for one possible combination of limiting factors. Adding a new combining scheme provides a more elegant solution.

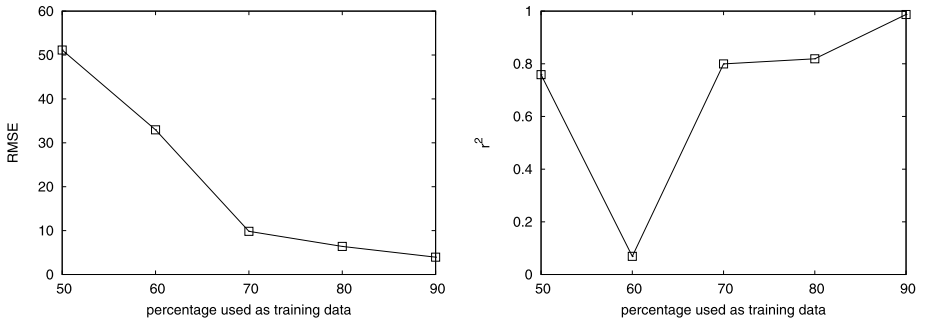

Fig. 4 Forecast accuracy in RMSE (*left*) and r^2 (*right*) for models of the Ross Sea with varying levels of training data

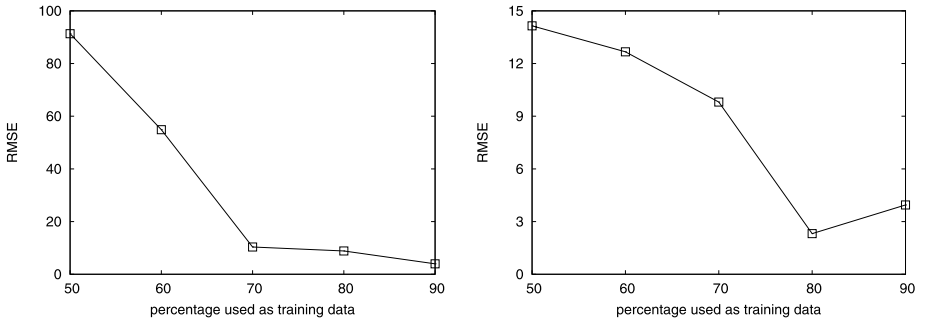

Fig. 5 Forecast accuracy in RMSE of the 10% of the data immediately following the hold-out boundary (*left*) and at the end of the Ross Sea trajectories (*right*)

downward slope of the phytoplankton bloom. To investigate, we examined the RMSE in the 10% of the total data directly after the hold-out boundary, as well as the final 10% of the trajectories. Figure 5 displays the results of these calculations. The graphs support our conjecture. Although accuracy increases with the amount of training data, the change is more pronounced when the test data fall directly after the hold-out boundary. Interestingly, the location of the boundary also contributes to the anomalous score in Fig. 4.

Inspection of the simulated trajectories in Fig. 6 shows why the r^2 score is low at the 40% boundary and why IPM failed to produce an accurate model. Notice that, on the test set, the selected model predicts what appear to be constant values for both the phytoplankton and nitrate concentrations. Examining the numerical values shows that minor variation exists in the forecast for phytoplankton and leads to a nonzero, but low, r^2. IPM's poor performance on this case stems from the position of the test boundary. Notice that the training data contained little evidence that the nitrate concentration would increase in the future. Consequently, IPM lacked a reason to prefer models that replenish nitrate over those that do not. More generally, *the system cannot anticipate what it has no reason to expect.*

This analysis begs the question of why the r^2 for the 50% boundary was much higher. Referring to the phytoplankton data, the training set in this case ends at the 386th Julian day, which occurs just after the peak of the bloom. When simulated, the induced model produces trajectories quite similar to those in Fig. 6. The main distinctions are that this model accounts for a larger region of phytoplankton decline and includes a minimal increase in the nitrate

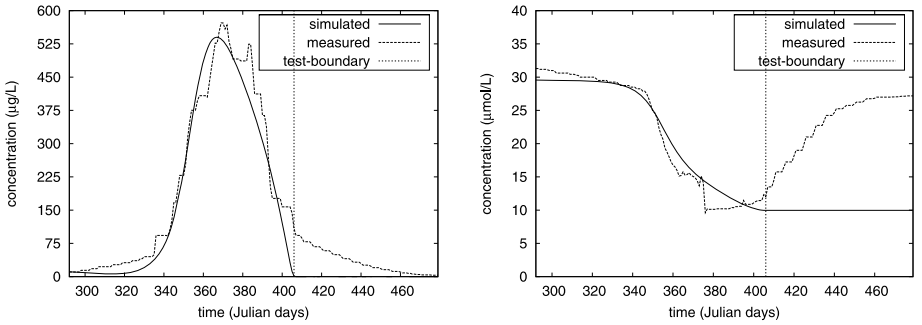

Fig. 6 Simulations of phytoplankton (*left*) and nitrate (*right*) from the model learned from the first 60% of the available Ross Sea data. The boundary separating the training and testing data appears as a *vertical line*

concentration. These characteristics lead to a better r^2 that is slightly offset by a worsening of the RMSE.

As in the population dynamics domain, we also evaluate the plausibility of IPM's models. To this end, Table 9 presents the process model that the program induced when given the complete Ross Sea data set. With one exception, the presented structure provides a plausible explanation of Ross Sea dynamics. In addition to the processes that we required, this model includes mechanisms that account for the death of phytoplankton, the limitation of its growth by resources, and the replenishment of iron. Notice that the model includes a grazing process, but no mechanism for setting the grazing rate. Therefore, grazing has no effect on the system and appears in Table 9 only because we forced its inclusion to reduce the search space. In contrast, the limitation processes for nitrate, iron, and light were all optional, but the removal of those for both light and nitrate has no effect on simulation. This finding suggests that iron dominates the other limiting factors throughout the simulation and that their inclusion is superfluous.[15] Since IPM performs exhaustive search, we examined the execution trace and found that the model lacking these processes has an RMSE of 0.0994 and an r^2 of 0.9665. The differences are small enough to reflect variance in the parameter estimation routine and suggest the need for a simplicity bias.

Although the above experiments did not generate any new insights about the underlying ecosystem, research by Asgharbeygi et al. (2006), who used a previous version of IPM, discovered plausible, new ecological knowledge. While developing a model that generalized to unseen data, the authors found evidence that the nitrate-to-carbon ratio of phytoplankton varies with the light availability. Standard ecological theory assumes that this ratio is constant, but recent research by Needoba and Harrison (2004) supports the hypothesis that the light regime directly affects the nitrate uptake of some phytoplankton species.

4.3 The Ringkøbing Fjord

For the third modeling task, we focused on water-level variation in Ringkøbing Fjord, a shallow estuary on the Danish west coast. Here the water level depends on three distinct aspects

[15] Substantial empirical evidence suggests that iron is the primary limiter of phytoplankton growth in the Ross Sea (Olson et al. 2000; Martin et al. 1991). Unfortunately, we lack time series for that nutrient's concentration, but IPM allows its inclusion as a theoretical variable. Iron's prominent role in the most accurate model lends plausibility to the discovered structure and highlights the importance of supporting theoretical terms.

Table 9 The model with the best fit for the Ross Sea data

model ross_sea;
variables $phyto\{p\_species\}$, $zoo\{z\_species\}$, $nitrate\{n\_nutrient\}$, $iron\{fe\_nutrient\}$,
$detritus\{detritus\}$, $light\{light\}$, $temperature\{temperature\}$, $ice\{ice\}$,
$fetoc\{fe\_const\}$, $beta\{ratio\}$, $phtyogr\{p\_grate\}$, $phytogl\{p\_glimit\}$,
$zoor\{z\_rate\}$, $ntoc\{n\_const\}$;
observable $phyto$, $nitrate$;
exogenous $light$, $temperature$, ice;
process phytoplankton_growth;
equations $phytogr = 0.299244 * exp(0.06933 * temperature * (1 - ice) * phytogl$;
$d[phyto, t, 1] = phytogr * phyto$;
process phytoplankton_loss;
equations $d[phyto, t, 1] = -0.04 * phyto$;
$d[detritus, t, 1] = (1 - beta) * 0.04 * phyto$;
process nitrate_uptake;
equations $d[nitrate, t, 1] = -1 * 1/(ntoc * 12.0107) * phytogr * phyto$;
process iron_uptake;
equations $d[iron, t, 1] = -1 * 1/(fetoc * 12.0107) * phytogr * phyto$;
process nitrate_limitation;
equations $phytogl = nitrate/(nitrate + 0.232262)$;
process iron_limitation;
equations $phytogl = iron/(iron + 0.000703)$;
process light_limitation;
equations $phytogl = (1 - exp(-1 * light/(16.0866/(1 + 7.44413 *$
$exp(light * exp(1.089 - 2.12 * log10(16.0866)))))))$;
process grazing;
equations $zoor = 0$;
$d[zoo, t, 1] = 0.320041 * zoor * zoo$;
$d[phyto, t, 1] = -1 * zoor * zoo$;
$d[detritus, t, 1] = (1 - beta) * (1 - 0.320041) * zoor * zoo$;
process nitrate_mixing;
equations $d[nitrate, t, 1] = (31 - nitrate) * 0.0343555 * (datamax(temperature) -$
$temperature)/(datamax(temperature) - datamin(temperature))$;
process remineralization;
equations $d[detritus, t, 1] = -0.0391038 * detritus$;
$d[iron, t, 1] = 0.0391038 * (1/(fetoc * 12.0107)) * detritus$;
process set_constants;
equations $beta = 0.999981$;
$ntoc = 6.6$;
$fetoc = 146411$;
initial conditions: $phyto = 10.9999$, $nitrate = 26.8066$, $zoo = 1.5$,
$detritus = 0.01745$, $iron = 0.0006951$

of the environment: the fresh water supply, the water exchange with the North Sea, and the local wind currents. The first two factors dominate the variation in the level, with the second being controlled through a 14 part gate. However, westerly wind currents also cause a rise in

Table 10 A set of generic processes for modeling the Ringkøbing Fjord. Variable types not explicitly defined are direct subtypes of *number*

generic process gate_influence_0;
  variables $GI\{gate\_influence\}$;
  parameters $r[-100000, 100000]$;
  equations $GI = r$;
generic process gate_influence_1;
  variables $GI\{gate\_influence\}$, $GO\{gate\_open\}$;
  parameters $r[-100000, 100000]$;
  equations $GI = r * GO$;
generic process wind_forcing_0;
  variables $WI\{wind\_influence\}$;
  parameters $r[-100000, 100000]$;
  equations $WI = r$;
generic process wind_forcing_1d;
  variables $WI\{winf\}$, $WD\{wdirection\}$;
  parameters $r[-100000, 100000]$;
  equations $WI = r * WD$;
generic process wind_forcing_1d_sin;
  variables $WI\{wind\_influence\}$, $WD\{wind\_direction\}$;
  parameters $r[-100000, 100000]$;
  equations $WI = r * sin(WD * 3.14159/180)$;
generic process wind_forcing_2dv_sin;
  variables $WI\{wind\_influence\}$, $WD\{wind\_direction\}$, $WV\{wind\_velocity\}$;
  parameters $r[-100000, 100000]$;
  equations $WI = r * WV * cos(WD * 3.14159/180)$;

the level measured at the gate. Accurate modeling of the influences on estuary height would be useful in a control system for the fjord's gate.

This domain differs from the two ecological cases in that we have a partial explanation of variation in fjord height:

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

The equation expresses that the water level in the estuary, h, changes at a rate determined by some function of the number of open gate parts, $f(a)$, divided by the surface area of the fjord, A. When the gates are opened, the difference between the water level in the open sea, h_{sea}, h, and a constant measurement error, h_0, influence h. Additionally, fresh water accumulates in the estuary, which causes an increase of Q_f/A, while the velocity, W_{vel}, and direction, W_{dir}, of the wind alter the water level according to some unknown function $g(\cdot)$. All variables except h and h_0 are measured and considered exogenous to the model. Thus IPM's induction task consists of identifying which functions for $f(\cdot)$ and $g(\cdot)$ most accurately model the change in the observed water level.

Table 10 lists 5 of the 16 candidate processes provided to IPM as background knowledge, wherein all variable types are direct subtypes of *number*. The gate-influence processes fill the role of $f(\cdot)$, specifying the magnitude of the effect produced by opening one or more sections of the gate. The various forms of wind-forcing serve as components of $g(\cdot)$ and

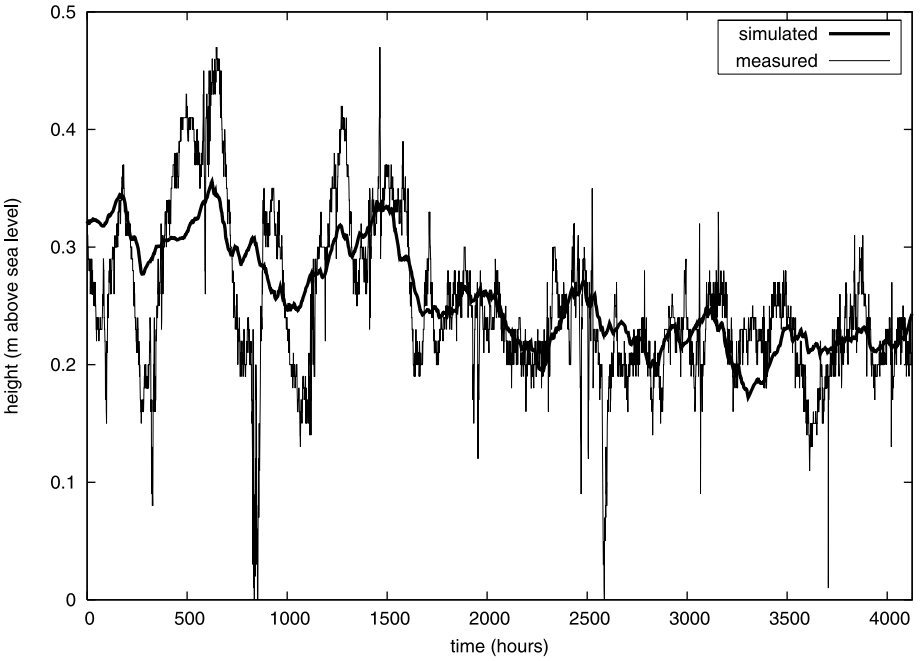

Fig. 7 The observed water level trajectory for the Ringkøbing Fjord between January 1 and December 10, 1999 and the simulated trajectory for the same period. The simulation comes from the induced model with the best quantitative fit

vary based on their use of the direction and velocity of the wind. For instance, some forms calculate the cosine or sine of the direction as alternatives to the angle, which appears in the measurements.

The described library creates a search space containing over 65,000 model structures, which is prohibitively large based on average computation time. To reduce this number, we hand coded two types of constraints on model composition. First, along with the base equation, we required both $f(\cdot)$ and $g(\cdot)$ to have constant terms, which could be set to zero (see gate_influence_0 in Table 10). Second, we created groups of mutually exclusive generic processes to reduce model complexity. For instance, an valid structure could have at most one process that introduced a first-order effect of wind direction. Thus, only one instantiation of wind_forcing_1d, wind_forcing_1d_sin, or the corresponding process for cosine could occur within a model. These constraints reduced the search space to 1,280 candidate structures.

The data from the Ringkøbing Fjord are the same as those used by Todorovski (2003) and consist of hourly measurements taken between January 1 and December 10 of 1999. There are 8,254 records for each observed variable. In the following experiments, we used the first 4,125 records for our initial evaluation and reserved the last section of the time series for examining accuracy on a forecasting task. The model that IPM produced from the initial examples had an reMSE score of 0.581, an r^2 score of 0.421, and an RMSE of 0.052. Figure 7 shows the model's simulated trajectory. These values compare favorably with those reported for the LAGRANGE system (Todorovski 2003), which, when trained on all 8,254 observations, produced a model with an RMSE of 0.059 and r^2 of 0.434.

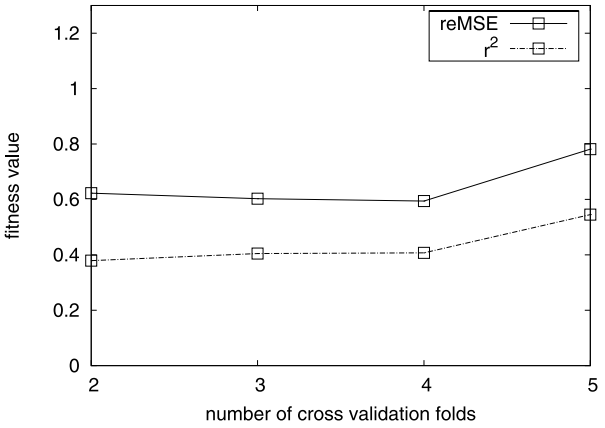

Fig. 8 Results from interpolation experiments in the Ringkøbing Fjord domain

Table 11 Results from simulating the full test set for the Ringkøbing Fjord domain. Values are reported for training sets covering roughly one quarter of a year (50%) to roughly one half of a year (90%)

Training data	reMSE	RMSE	r^2
90%	1.694	0.110	0.004
80%	2.564	0.136	0.128
70%	2.357	0.130	0.293
60%	18.669	0.367	0.010
50%	37.545	0.520	0.009

As with the previous domains, we examined IPM's performance on the interpolation task, where reMSE provided the guiding measurement. Figure 8 shows the results, in which the reMSE scores ranged between 0.594 and 0.781, and in which the r^2 values fell between 0.379 and 0.545. Although there is no discernible trend in the reMSE scores, performance in terms of r^2 appears to improve when we allocate more data to the training set for each fold. Due to an abundance of exogenous variables and the lack of unobserved variables in this domain, one might expect less variation in the interpolation scores. However, assuming an indirect relationship between the amount of knowledge one has and the amount of data that one requires, these results suggest that our knowledge of this domain (e.g., the relevant variables, the meaningful relationships) is far from complete.

For the extrapolation task, we trained IPM on 50%–90% of the first section of the data and predicted the dynamics of the second section. Table 11 shows the results of these experiments. In general, these numbers indicate that the models produced by IPM were unsuitable for long-term forecasting. Moreover, the reMSE and RMSE scores degrade considerably as we reduce the number of training data.

To some degree, the difficulty of the task tempers these results, but it also raises more questions. Initially we extrapolated the fjord dynamics over half a year by giving each model an initial water height and supporting it with exogenous variables throughout the time period. Although this seems like a sufficient amount of input, any error in the water height will remain uncorrected and can influence later predictions. With this possibility in mind, we conjectured that routine corrections of the water height would produce more accurate predictions.

To investigate this hypothesis, we altered the performance element of the extrapolation task. Instead of forecasting the full time course from a single initial condition, we simulated

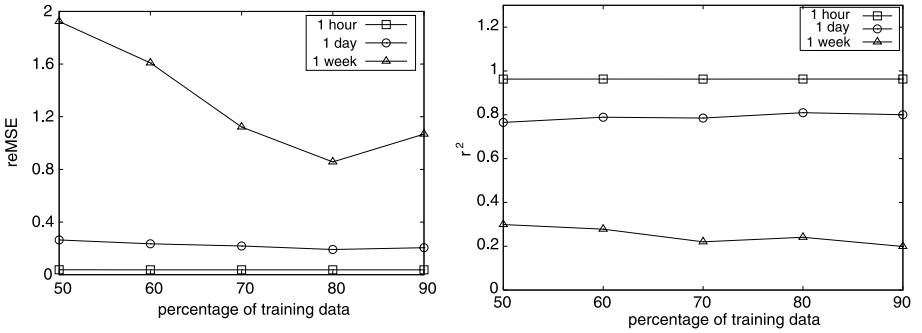

Fig. 9 Extrapolation results for one hour, one day, and one week forward predictions. These graphs show how accuracy changes with the amount of training data. The individual lines correspond to particular prediction intervals. The reMSE (*left*) and r^2 (*right*) scores show that the length of these intervals has a greater effect on predictive accuracy than does the amount of training data

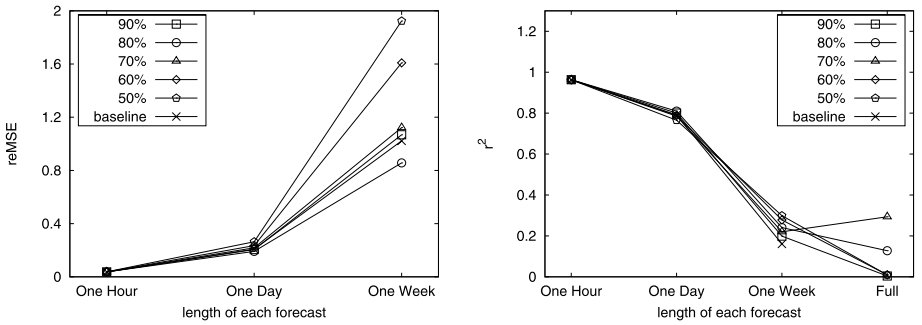

Fig. 10 Extrapolation results for one hour, one day, and one week forward predictions. These graphs show how accuracy changes with the size of the prediction interval. All but one of the lines correspond to the performance of a model induced from a fixed amount of training data (50%–90% of the first section of the year). The remaining line shows the results of the baseline algorithm

each model so that, given a starting time and the necessary exogenous inputs, it would produce predictions for one hour, one day, and one week in advance. We then compiled the one-hour predictions into a single trajectory, the one-day predictions into another, and the one-week values into a third. These resulting trajectories let us explore the effect of forecast distance on predictive accuracy in this domain. To place the results in context, we compared them to a naive baseline approach that predicts "no change" from the currently observed value. To create these trajectories, we shifted the observed time series forward by each lookahead interval.

Figures 9 and 10 show the results of these modified experiments. The plots in Fig. 9 suggest that the amount of training data has less influence on forecasting than the lookahead distance. Although reducing the number of training examples tends to degrade performance, we see nowhere near the trends shown Table 11. Instead, r^2 values stay fairly constant across a fixed prediction interval and, with the exception of the one-week case, the same is true for reMSE.

Figure 10 presents a different view of the same results and includes the baseline case for reference. In these graphs, one can better see how the length of the prediction interval

affects accuracy. For one hour or one day predictions, any model will do. Interestingly, even the baseline is sufficient for near-term estimates of the system dynamics. As the model looks farther into the future, the picture gets hazier. In terms of reMSE scores, the baseline performs poorly, but it generally scores better than the quantitative process models. The r^2 scores indicate that, as the prediction interval increases, the baseline ceases to account for the variability in water height as well as IPM's models.

The results in the fjord domain reveal how strongly IPM's utility depends on the knowledge captured in the generic process library. Although we specified a rich search space, we have little reason to believe that the defined library adequately represents the relationships between water height and the effects of both the gating mechanism and the wind. IPM's poor forecasting performance in this domain reinforces our doubts and strongly suggests that we lack sufficient knowledge of the relevant processes. Previous work by Todorovski (2003) indicated that the library suffices for interpolation, but our extrapolation experiments indicate a weakness. To accurately determine the root of these findings, we need to carry out a detailed study that disentangles the effects of IPM's various components.

In general, the experiments described in this section reveal interesting characteristics of IPM. First, in some situations, the program can generate accurate models from severely limited data. Second, as the experiments in the Ross Sea domain show, the system benefits from a "representative sample", which is a concept in need of better definition with respect to time series. Third, modeling and domain knowledge is crucial for the accurate and plausible explanation of a system's dynamics. Our future research will explore these and other findings in finer detail. In the next section, we outline a broader research agenda for inductive process modeling as a new subfield of machine learning.

5 A proposed research agenda

Although our initial results with IPM suggest the viability of inducing process models from observational data, they leave many questions unanswered. Before closing, we discuss some issues that future work in the area should address and consider some promising approaches that should be explored within this research agenda.

5.1 Representation

Our initial foray into process modeling uses a language of somewhat limited scope that suggests numerous extensions. For instance, scientists often organize their models in terms of systems and subsystems to increase manageability. Likewise, processes may be decomposed into subprocesses, thereby creating a behavioral hierarchy that complements a scientist's structural one. In the same vein, scientists often organize related properties into groups that belong to particular objects within the system. Since the task of inductive process modeling includes the development of comprehensible models as a primary goal, researchers should investigate methods of incorporating these common forms of knowledge organization into the modeling framework.

Along with structural extensions to the representation, future work should explore methods for specifying and learning models with varying degrees of certainty. In particular, a model might consist of both quantitative and *qualitative* processes (Forbus 1984), which use proportionalities to describe relations between variables. In this framework, exponential and logistic growth would map onto a single qualitative process stating that a population's growth rate is proportional to its size. Such models are appropriate for domains like microbiology, where scientists often state their knowledge in qualitative form. Moreover, qualitative

processes generally have fewer effective parameters than quantitative ones, which makes them useful for situations with few observations. Many issues that arise with quantitative models also occur with their qualitative analogs, so we also need work on this front.

5.2 Induction

In addition to dealing with representational issues, we need research that addresses the traditional problems of model robustness and search efficiency. Overfitting the training data can arise in nearly every learning task, and we need ways to guard against this tendency, especially as we develop algorithms that generate more complex process models. One avenue would examine analogs to methods that have proven successful in other induction paradigms. These include techniques for early halting in decision-tree construction using minimum description length and methods for postpruning using cross validation. Other techniques include ensemble methods like bagging, although this would require an adaptation of bootstrap sampling to handle time-series data (Efron and Tibshirani 1993; Härdle et al. 2003) and a means for generating a single, comprehensible model from the ensemble (e.g., Domingos 1997). In addition, researchers should explore other defenses against overfitting that are specific to process models.

While experimenting with IPM, we noticed that over 99% of the system's runtime was spent on parameter estimation, which suggests another avenue of research. Our current routines use variants of uninformed, gradient descent search. In contrast, search through the space of model structures relies heavily on domain knowledge, so it seems reasonable that one could use similar information to guide parameter estimation. For instance, our conversations with scientists indicate that they concentrate more on the high-level features of the trajectories (e.g., peak placement and height) than the residuals. Thus, developing methods for characterizing these features could lead to a more rational and more efficient means of tuning a model's parameters.

Research on inductive process modeling should also lead to extensions of the general search mechanism. For example, the current process-level constraints based on variable types could be augmented by model-level constraints based on processes or process types, so that a scientist can express both required and mutually exclusive processes. Knowledge about the dimensional units of variables would also constrain model induction, as would the introduction of knowledge that certain variables are conserved over time. Research should also continue on Bradley et al.'s (2001) use of qualitative patterns to characterize certain classes of equations. Other work should develop methods for learning new process forms.

An alternative approach to aiding model induction borrows an idea from work on theory revision. Rather than constructing a process model from scratch, one can instead start with a specific model and revise details to improve its fit to observations. Research on this topic should explore ways to revise a specific model's parameters, change the conditions on its component processes, replace these processes with others that relate the same variables, and even alter the basic structure of the initial model. Model revision will require the ability to remove components as well as add them, but otherwise the same issues arise as in the basic problem of process model induction.

Since dynamic systems may operate in several qualitatively different regimes, an inductive process modeler should appropriately handle shifts among them. One approach would involve identifying different regimes from time series and explaining the behavior of each one separately. A second method relies on the ability to include conditions within processes. One could extend the process modeler to learn these conditions so that processes are activated or inactivated as the system passes through each regime.

5.3 Evaluation

Finally, future work should identify appropriate methods for evaluating process models. Since the data composing the trajectories fail to meet the independent and identically distributed assumption made by many classification algorithms, standard evaluation techniques such as cross validation must be adapted to better measure model performance. Additionally, dynamic domains can pass through several operating regimes, which poses additional difficulties for any evaluation method that samples from the original trajectory. Successful modeling would require the observation of each of these regimes during training. The approach to evaluation we used in our experiments addresses these issues, but researchers should continue to explore other avenues.

Although these challenges should be met, the evaluation of process models must move beyond an emphasis on predictive accuracy for the new paradigm to be useful to scientists. Research must also take into account considerations of model robustness and explanatory power. For example, biologists sometimes evaluate a model based on both its dependence on specific parameters and its ability to match the general *shape* of observed trajectories. Such evaluation will require moving beyond one-to-one comparisons of observations and predictions to incorporate sensitivity analysis and to measure agreement with meaningful trends in the data.

In pursuing this research agenda, we should follow the accepted standards for established induction paradigms. Thus, papers should make explicit claims about a method's abilities and support them with experimental or theoretical evidence. Ideally, experimental studies should include a mixture of natural domains to ensure relevance and synthetic domains that let one vary dimensions of interest. However, the focus on familiarity and background knowledge recommends studies that involve collaborations with domain scientists or engineers, which remain uncommon in the machine learning literature. Finally, despite the distinctive nature of process model induction, researchers should incorporate ideas from other learning tasks and utilize existing methods as subroutines whenever sensible.

6 Concluding remarks

In this paper, we proposed a new problem for machine learning researchers that addresses the induction of process models from observations. We defined this task as the construction of models that combine known component processes to explain the observed behavior of continuous dynamical systems. We considered the challenges posed by process model induction and the potential of established techniques to address them, concluding that it demands research on new methods specialized to process modeling. We also presented an initial algorithm of this sort and demonstrated its functionality in multiple domains, after which we outlined a research agenda for future work on the topic.

Process models constitute a novel representation of knowledge that differs from the formalisms traditionally used in machine learning. These models are cast in the same terms as many scientific and engineering models, which should make them more communicable to practitioners in those fields. However, they have the same modularity as other formalisms that support learning, and they provide a clear facility for incorporating domain knowledge into learning mechanisms. We maintain that research on process model induction will broaden the scope of machine learning in significant ways, and we encourage others to join us in exploring computational methods that address this important new problem.

Acknowledgements The research reported in this paper was supported by NSF Grant Number IIS-0326059. We thank Javier Sánchez, Kazumi Saito, Dileep George, Stephen Bay, and Nima Asgharbeygi for their work on early versions of the IPM system, along with Kevin Arrigo and Gert van Dijken for providing data and expertise on the Ross Sea. We are especially grateful to Stuart Borrett for facilitating communication between Arrigo's lab and our own, and we thank the reviewers for their helpful comments. A shorter version of this paper appeared as "Inducing Process Models from Continuous Data" in the *Proceedings of the Nineteenth International Conference on Machine Learning.*

References

Arrigo, K. R., Worthen, D. L., & Robinson, D. H. (2003). A coupled ocean-ecosystem model of the Ross Sea: 2. Iron regulation of phytoplankton taxonomic variability and primary production. *Journal of Geophysical Research*, *108*, 3231.

Asgharbeygi, N., Bay, S., Langley, P., & Arrigo, K. R. (2006). Inductive revision of quantitative process models. *Ecological Modelling*, *194*, 70–79.

Åström, K. J., & Eykhoff, P. (1971). System identification—a survey. *Automatica*, *7*, 123–167.

Bay, S. D., Shrager, J., Pohorille, A., & Langley, P. (2002). Revising regulatory networks: from expression data to linear causal models. *Journal of Biomedical Informatics*, *35*, 289–297.

Bechtel, W., & Abrahamsen, A. (2005). Explanation: a mechanistic alternative. *Studies in History and Philosophy of the Biological and Biomedical Sciences*, *36*, 421–441.

Berryman, A. A. (1992). The origins and evolution of predator–prey theory. *Ecology*, *73*, 1530–1535.

Box, G., Jenkins, G. M., & Reinsel, G. (1994). *Time series analysis: forecasting & control* (3rd ed.). Englewood Cliffs: Prentice Hall.

Bradley, E., Easley, M., & Stolle, R. (2001). Reasoning about nonlinear system identification. *Artificial Intelligence*, *133*, 139–188.

Bridewell, W., Sánchez, J. N., Langley, P., & Billman, D. (2006). An interactive environment for the modeling and discovery of scientific knowledge. *International Journal of Human–Computer Studies*, *64*, 1099–1114.

Bunch, D., Gay, D., & Welsch, R. (1993). Algorithm 717: subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models. *ACM Transactions on Mathematical Software*, *19*, 109–130.

Cohen, S., & Hindmarsh, A. (1996). CVODE, a stiff/nonstiff ODE solver in C. *Computers in Physics*, *10*, 138–143.

Dennis, J. E. Jr., Gay, D. M., & Welsch, R. E. (1981). An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software*, *7*, 348–368.

Dietterich, T. G. (1990). Exploratory research in machine learning. *Machine Learning*, *5*, 5–9.

Domingos, P. (1997). Knowledge acquisition from examples via multiple models. In *Proceedings of the fourteenth international conference on machine learning* (pp. 98–106). Nashville: Kaufmann.

Džeroski, S., & Todorovski, L. (1993). Discovering dynamics. In *Proceedings of the tenth international conference on machine learning* (pp. 97–103). Amherst: Kaufmann.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York City: Chapman & Hall.

Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence*, *24*, 85–168.

Forbus, K. D., & Falkenhainer, B. (1990). Self-explanatory simulations: an integration of qualitative and quantitative knowledge. In *Proceedings of the eighth national conference on artificial intelligence* (pp. 380–387). Boston: AAAI Press.

Garrett, S., Coghill, G. M., Srinivasan, A., & King, R. D. (2007). Learning qualitative models of physical and biological systems. In S. D. Džeroski & L. Todorovski (Eds.), *Computational discovery of scientific knowledge*. Berlin: Springer.

Gay, D. M. (1983). Algorithm 611: Subroutines for unconstrained minimization using a model/trust-region approach. *ACM Transactions on Mathematical Software*, *9*, 503–524.

Ghahramani, Z. (1998). Learning dynamic Bayesian networks. In C. L. Giles & M. Gori (Eds.), *Adaptive processing of sequences and data structures*. Berlin: Springer.

Ghosh, R., & Tomlin, C. J. (2001). Lateral inhibition through delta-notch signaling: a piecewise affine hybrid model. In *Proceedings of the fourth international workshop on hybrid systems: computation and control* (pp. 232–246). Springer: Rome.

Glennan, S. (2002). Rethinking mechanistic explanation. *Philosophy of Science*, *69*, S342–S353.

Glymour, C., Scheines, R., Spirtes, P., & Kelly, K. (1987). *Discovering causal structure: artificial intelligence, philosophy of science, and statistical modeling*. San Diego: Academic Press.

Härdle, W., Horowitz, J., & Kreiss, J. (2003). Bootstrap methods for time series. *International Statistical Review*, *70*, 435–459.

Hempel, C. G., & Oppenheim, P. (1948). Studies in the logic of explanation. *Philosophy of Science*, *15*, 135–175.

Holling, C. S. (1959). The components of predation as revealed by a study of small-mammal predation of the European pine sawfly. *Canadian Entomologist*, *91*, 293–320.

Iwasaki, Y., & Simon, H. A. (1994). Causality and model abstraction. *Artificial Intelligence*, *67*, 143–194.

Jost, C., & Ellner, S. (2000). Testing for predator dependence in predator–prey dynamics: a non-parametric approach. *Proceedings of the Royal Society of London B*, *267*, 1611–1620.

Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science*, *5*, 31–54.

Langley, P., Simon, H. A., Bradshaw, G. L., & Żytkow, J. M. (1987). *Scientific discovery: computational explorations of the creative processes*. Cambridge: MIT Press.

Lavrač, N. L., & Džeroski, S. D. (1994). *Inductive logic programming: techniques and applications*. New York City: Ellis Horwood.

Machamer, P., Darden, L., & Craver, C. F. (2000). Thinking about mechanisms. *Philosophy of Science*, *67*, 1–25.

Martin, J. H., Gordon, R. M., & Fitzwater, S. E. (1991). The case for iron. *Limnology and Oceanography*, *36*, 1793–1802.

Murray, J. D. (2004). *Mathematical biology, I: an introduction* (3rd ed.). Berlin: Springer.

Needoba, J. A., & Harrison, P. J. (2004). Influence of low light and a light: dark cycle on NO_3^- uptake, intracellular NO_3^- and nitrogen isotope fractionation by marine phytoplankton. *Journal of Phycology*, *40*, 505–516.

Olson, R. J., Sosik, H. M., Chekalyuk, A. M., & Shalapyonok, A. (2000). Effects of iron enrichment on phytoplankton in the Southern Ocean during late summer: active fluorescence and flow cytometric analyses. *Deep-Sea Research Part II-Topical Studies in Oceanography*, *47*, 3181–3200.

Ourston, D., & Mooney, R. J. (1990). Changing the rules: a comprehensive approach to theory refinement. In *Proceedings of the eighth national conference on artificial intelligence* (pp. 815–820). Boston: AAAI Press.

Pazzani, M. J., Mani, S., & Shankle, W. R. (2001). Acceptance by medical experts of rules generated by machine learning. *Methods of Information in Medicine*, *40*, 380–385.

Poritz, A. (1988). Hidden Markov models: a guided tour. In *Proceedings of the international conference on acoustic, speech and signal processing* (pp. 7–13). New York City: IEEE Press.

Schwabacher, M., & Langley, P. (2001). Discovering communicable scientific knowledge from spatio-temporal data. In *Proceedings of the eighteenth international conference on machine learning* (pp. 489–496). Williamstown: Kaufmann.

Simon, H. A. (1954). Spurious correlation: a causal interpretation. *Journal of the American Statistical Association*, *49*, 467–479.

Todorovski, L. (2003). *Using domain knowledge for automated modeling of dynamic systems with equation discovery*. Doctoral dissertation, Faculty of Computer and Information Science, University of Ljubljana. Ljubljana, Slovenia.

Todorovski, L., & Džeroski, S. (1997). Declarative bias in equation discovery. In *Proceedings of the fourteenth international conference on machine learning* (pp. 376–384). Nashville: Kaufmann.

Veilleux, B. G. (1979). An analysis of predatory interaction between paramecium and didinium. *Journal of Animal Ecology*, *48*, 787–803.

Washio, T., Motoda, H., & Niwa, Y. (2000). Enhancing the plausibility of law equation discovery. In *Proceedings of the seventeenth international conference on machine learning* (pp. 1127–1134). Stanford: Kaufmann.

Williams, R., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*, 270–280.

Woodward, J. (2002). What is a mechanism? A counterfactual account. *Philosophy of Science*, *69*, S366–S377.

Zheng, J., Vankataramanan, L., & Sigworth, F. J. (2001). Hidden Markov model analysis of intermediate gating steps associated with the pore gate of *Shaker* potassium channels. *Journal of General Physiology*, *118*, 547–562.

Żytkow, J. M., Zhu, J., & Hussam, A. (1990). Automated discovery in a chemistry laboratory. In *Proceedings of the eighth national conference on artificial intelligence* (pp. 889–894). Boston: AAAI Press.