Heuristic Induction of Rate-Based Process Models

Pat Langley

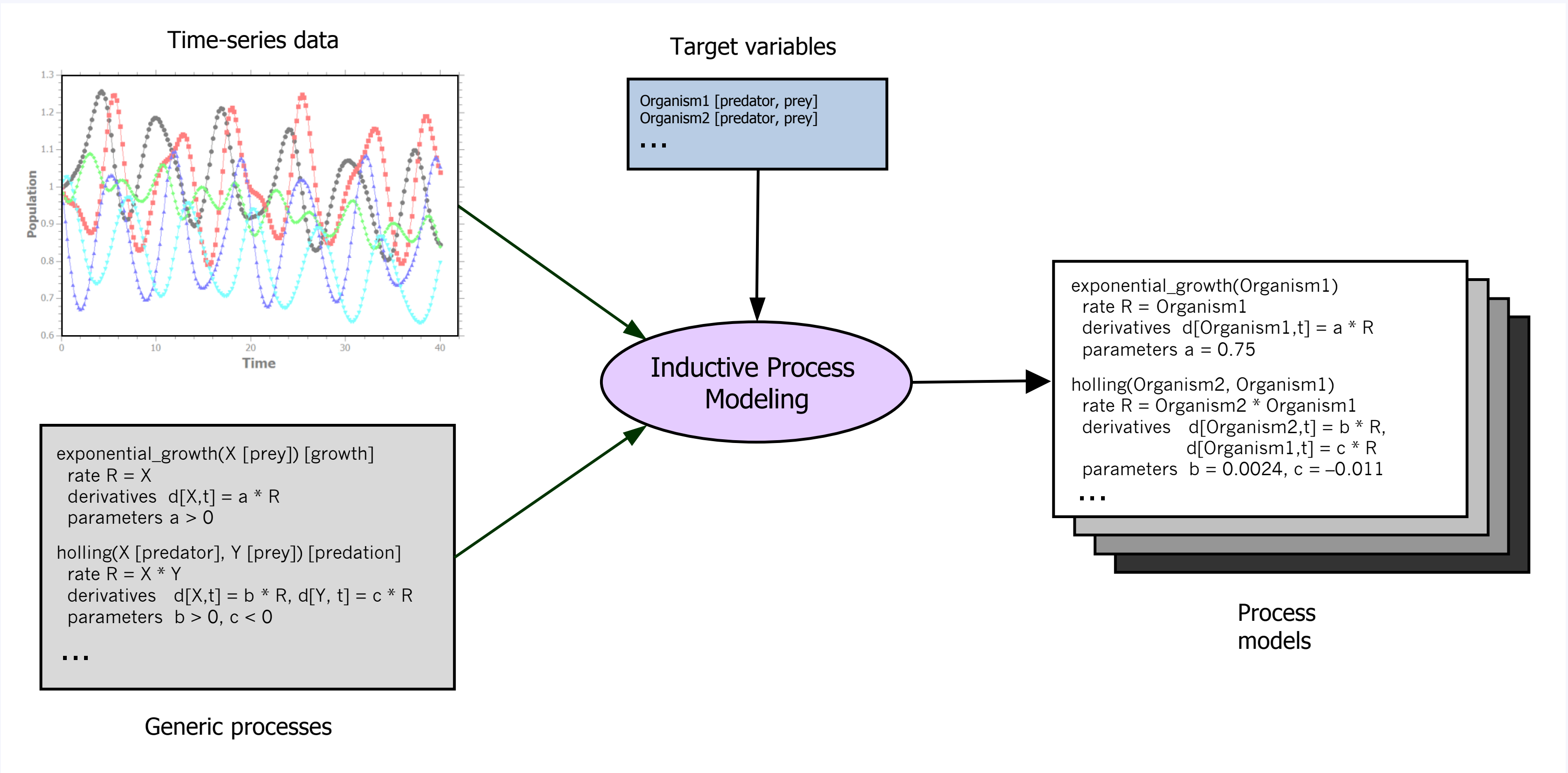
Adam Arvay

Department of Computer Science
University of Auckland
Auckland, NZ

Thanks to W. Bridewell, R. Morin, S. To, L. Todorovski, and others for contributions to this project, which is funded by ONR Grant No. N00014-11-1-0107.

Inductive Process Modeling

Inductive process modeling constructs explanations of time series from background knowledge (Langley et al., 2002) .



Models are stated as sets of *differential equations* organized into higher-level *processes*.

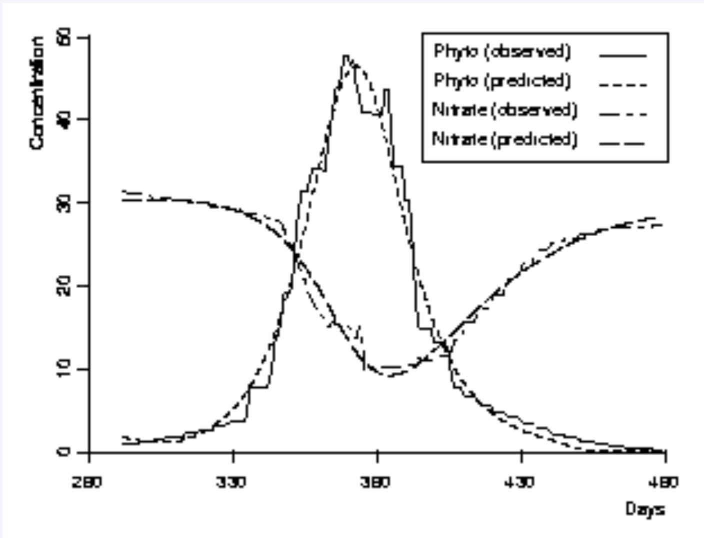
The SC-IPM System

Previously, we reported SC-IPM (Bridewell & Langley, 2010), a system for inductive process modeling that:

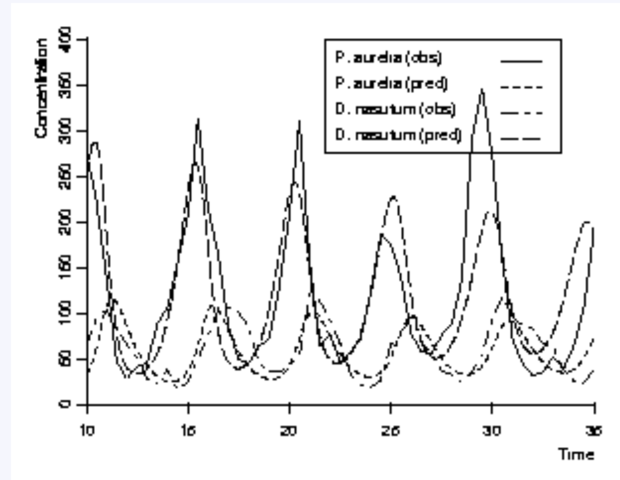
1. Uses background knowledge to generate *process instances*;
2. Combines them to produce possible *model structures*, rejecting ones that violate known constraints;
3. For each candidate model structure:
   a. Carries out gradient descent search through parameter space to find good coefficients;
   b. Invokes random restarts to decrease chances of local optima;
4. Returns the parameterized model with lowest squared error or a ranked list of models.

We have reported encouraging results with SC-IPM on a variety of scientific data sets.

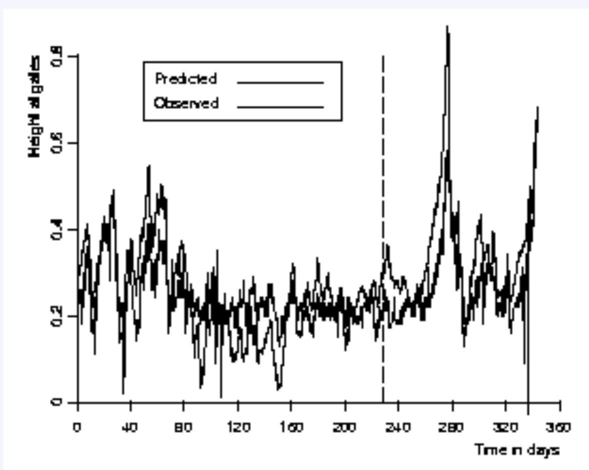
Some SC-IPM Successes



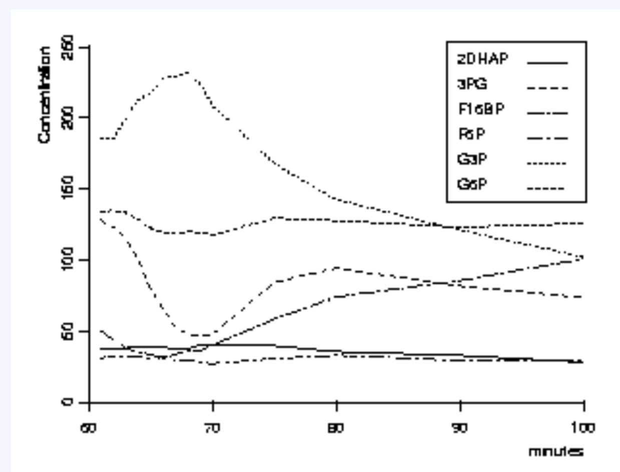
aquatic ecosystems



protist dynamics



hydrology



biochemical kinetics

Critiques of SC-IPM

Despite these successes, the SC-IPM system suffers from four key drawbacks, in that it:

- Evaluates *full model structures*, so disallows heuristic search;
- Requires *repeated simulation* to estimate model parameters;
- Invokes *random restarts* to reduce chances of local optima;
- Despite these steps, it can still find poorly-fitting models.

99.99 percent of CPU time

As a result, SC-IPM does not scale well to complex modeling tasks and it is not reliable.

In recent research, we have developed a new framework that avoids these problems.

A New Process Formalism

SC-IPM allowed processes with only algebraic equations, only differential equations, and mixtures of them.

In our new modeling formalism, each process P must include:

- A *rate* that denotes P's speed / activation on a given time step;
- An *algebraic equation* that describes P's rate as a *parameter-free* function of known variables;
- One or more *derivatives* that are proportional to P's rate.

This notation has important mathematical properties that assist model induction.

The revised formalism is also closer to Forbus' (1984) original Qualitative Process theory.

A Sample Process Model

Consider a process model for a simple predator-prey ecosystem:

```
exponential_growth[aurelia]
  rate        r = aurelia
  parameters  A = 0.75
  equations   d[aurelia] = A * r

exponential_loss[nasutum]
  rate        r = nasutum
  parameters  B = -0.57
  equations   d[nasutum] = B * r

holling_predation[nasutum, aurelia]
  rate        r = nasutum * aurelia
  parameters  C = 0.0024
              D = -0.011
  equations   d[nasutum] = C * r
              d[aurelia] = D * r
```

Each derivative is proportional to the algebraic rate expression.

A Sample Process Model

Consider a process model for a simple predator-prey ecosystem:

```
exponential_growth[aurelia]
  rate         r = aurelia
  parameters   A = 0.75
  equations    d[aurelia] = A * r

exponential_loss[nasutum]
  rate         r = nasutum
  parameters   B = -0.57
  equations    d[nasutum] = B * r

holling_predation[nasutum, aurelia]
  rate         r = nasutum * aurelia
  parameters   C = 0.0024
               D = -0.011
  equations    d[nasutum] = C * r
               d[aurelia] = D * r
```

This model compiles into a set of differential equations

```
d[aurelia] = 0.75 * aurelia – 0.011 * nasutum * aurelia
d[nasutum] = 0.0024 * nasutum * aurelia – 0.57 * nasutum
```

Some Generic Processes

Generic processes have a very similar but more abstract format:

```
exponential_growth(X [prey]) [growth]
  rate         r = X
  parameters   A = (> A 0.0)
  equations    d[prey] = A * r

exponential_loss(X [predator]) [loss]
  rate         r = predator
  parameters   B = (< B 0.0)
  equations    d[prey] = B * r

holling_predation(X [predator], Y [prey]) [predation]
  rate         r = X * Y
  parameters   C = (> C 0.0)
               D = (< D 0.0)
  equations    d[predator] = C * r
               d[prey] = D * r
```

These form the *building blocks* from which to compose models.

RPM: Regression-Guided Process Modeling

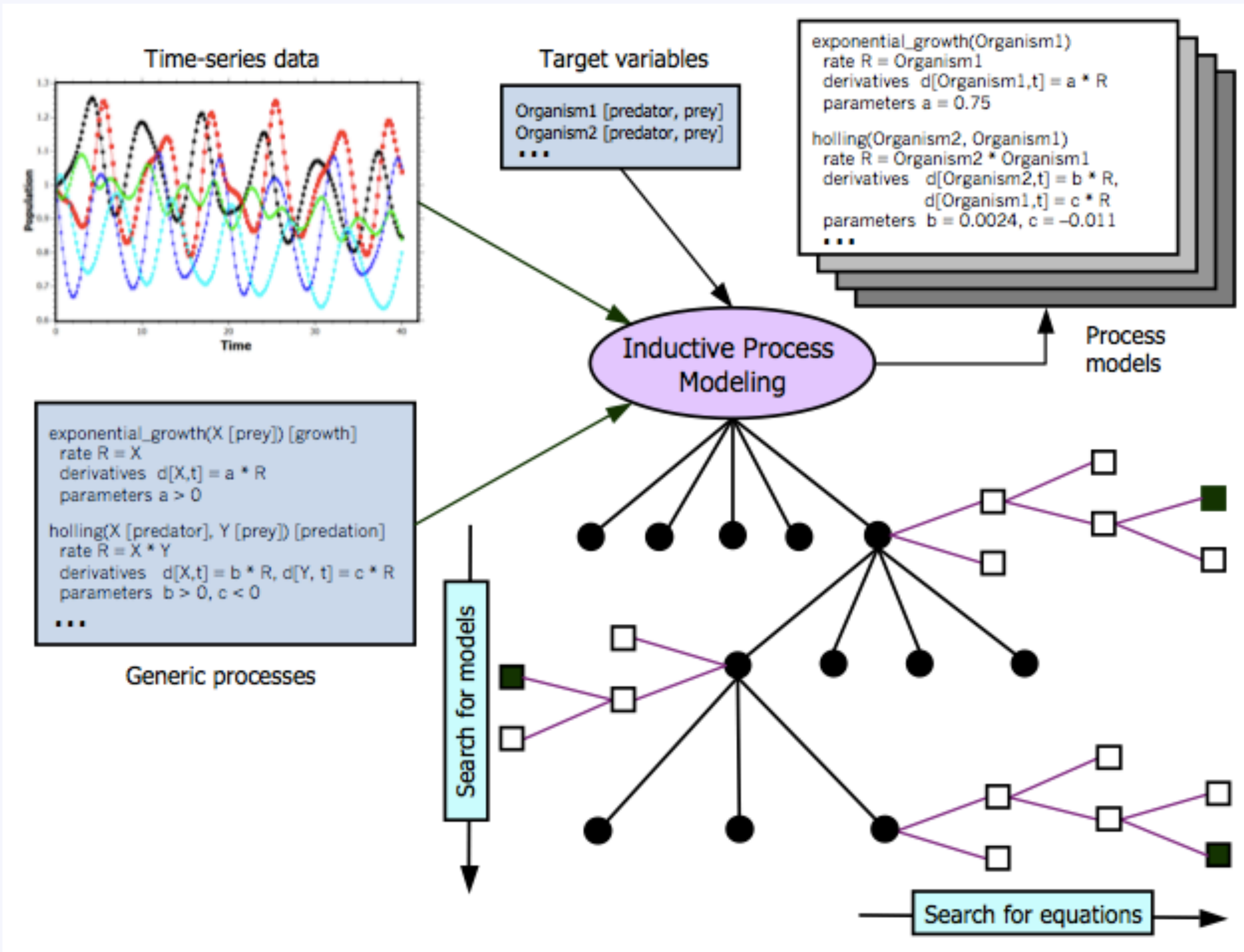
This suggests a new approach to inducing process models that our *RPM* system implements:

- Generate all process instances consistent with type constraints
- For each process P, calculate the *rate* for P on each time step
- For each dependent variable X,
  - Estimate *dX/dt* on each time step with center differencing,
  - For each subset of processes with up to *k* elements,
    - Find a regression equation for dX/dt in terms of process rates
    - If the equation's r^2 is high enough, retain for consideration
  - Add the equation with the highest r^2 to the process model

This approach factors the model construction task into a number of tractable components.

Assumes all variables observed
Rate expression is parameter free

Two-Level Heuristic Search in RPM



Heuristics for Model Induction

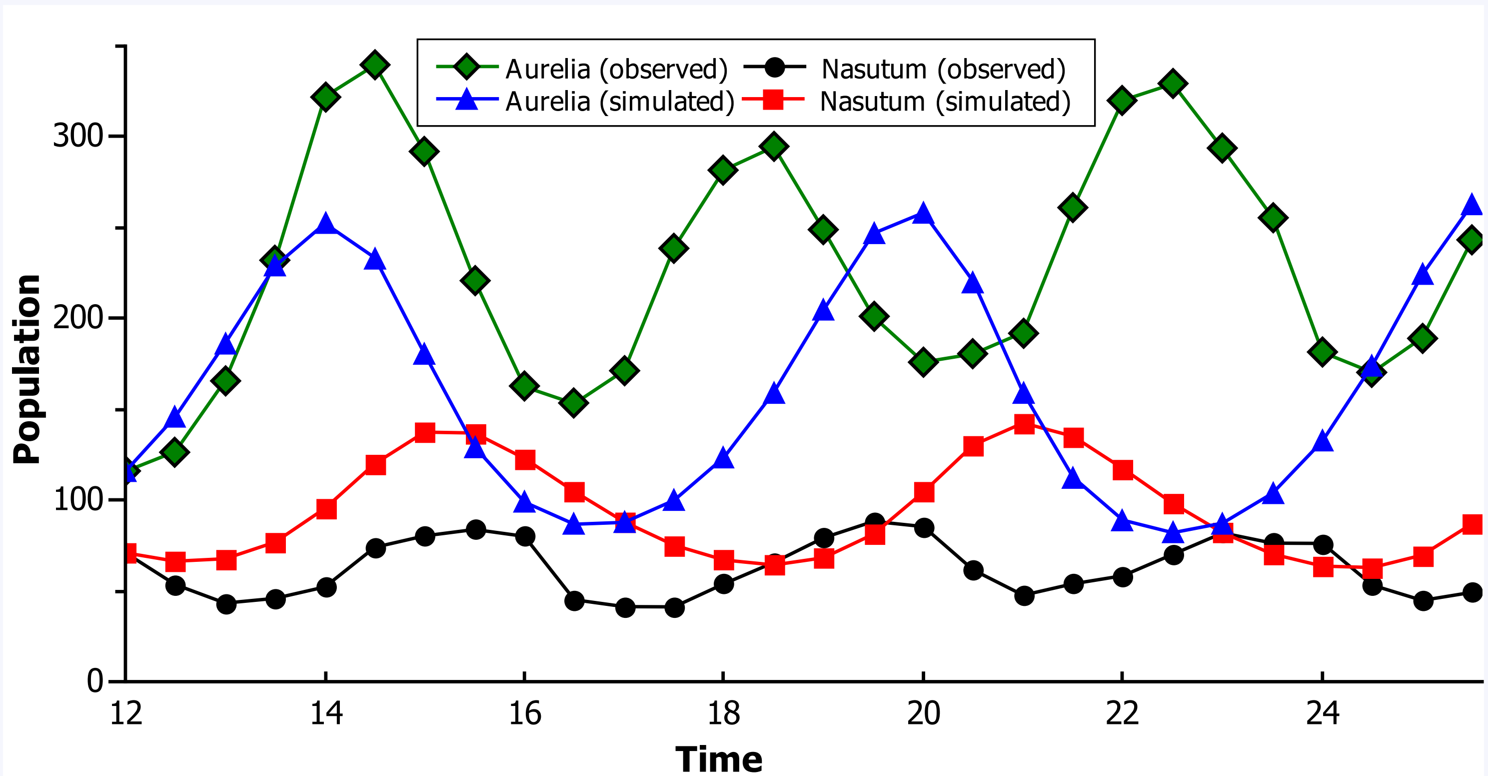
RPM uses four heuristics to guide its search through the space of process models:

- A model may include only one process instance of each type;
- Parameters must obey numeric constraints in generic processes;
- If an equation for one variable includes a process P, then P must appear in equations for other variables that P mentions;
- Incorporate variables that participate in more processes earlier than less constrained ones.

These heuristics reduce substantially the amount of search that RPM carries our during model induction.

Behavior on Natural Data

RPM matches the main trends for a simple predator-prey system.

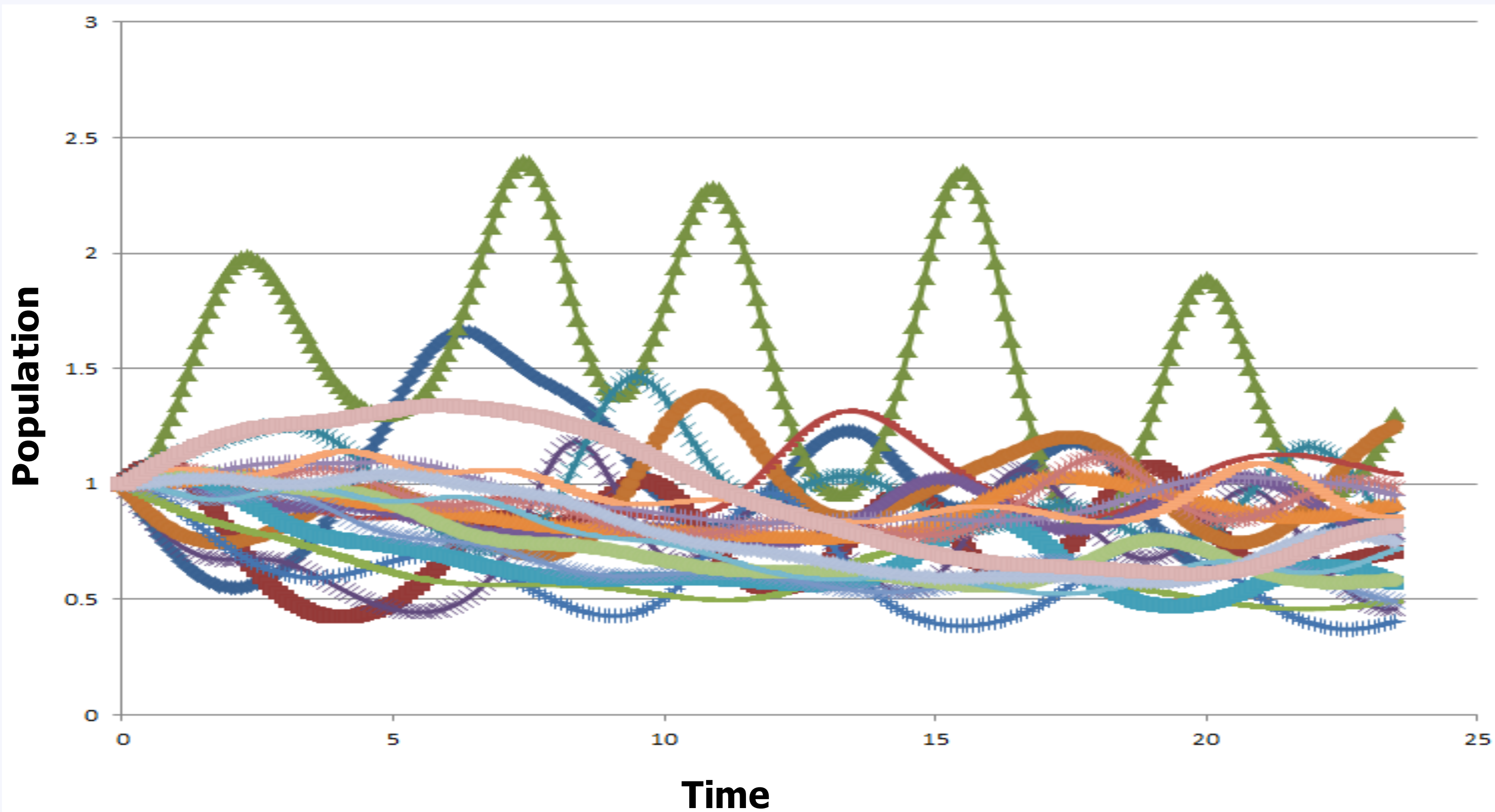


$$d[aurelia] = 0.75 * aurelia - 0.11 * nasutum * aurelia \ [r^2 = 0.84]$$

$$d[naustum] = 0.0024 * nasutum * aurelia - 0.57 * nasutum \ [r^2 = 0.71]$$

Behavior on Complex Synthetic Data

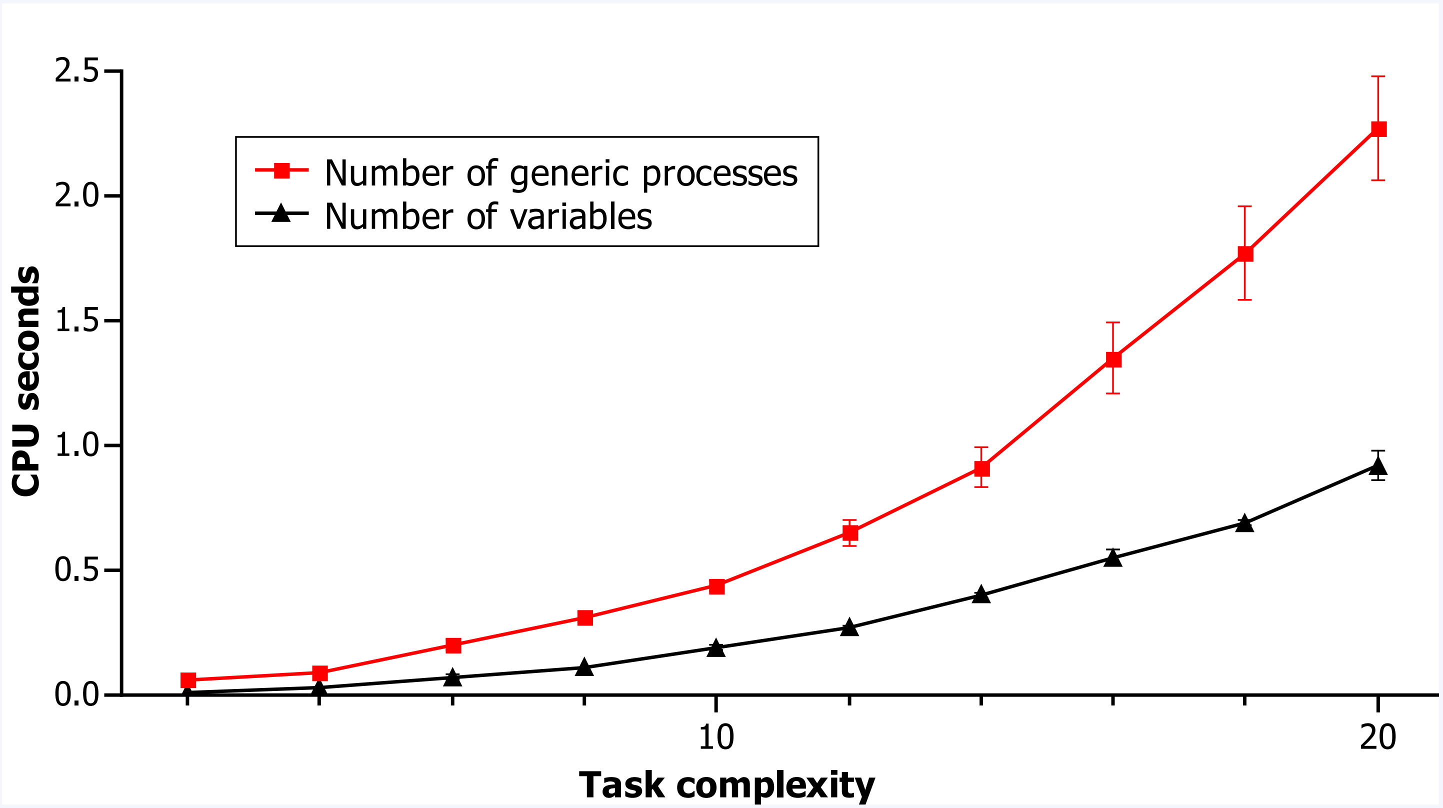
RPM also finds an accurate model for a 20-organism food chain.



This suggests the system scales well to difficult modeling tasks.

Handling Noise and Complexity

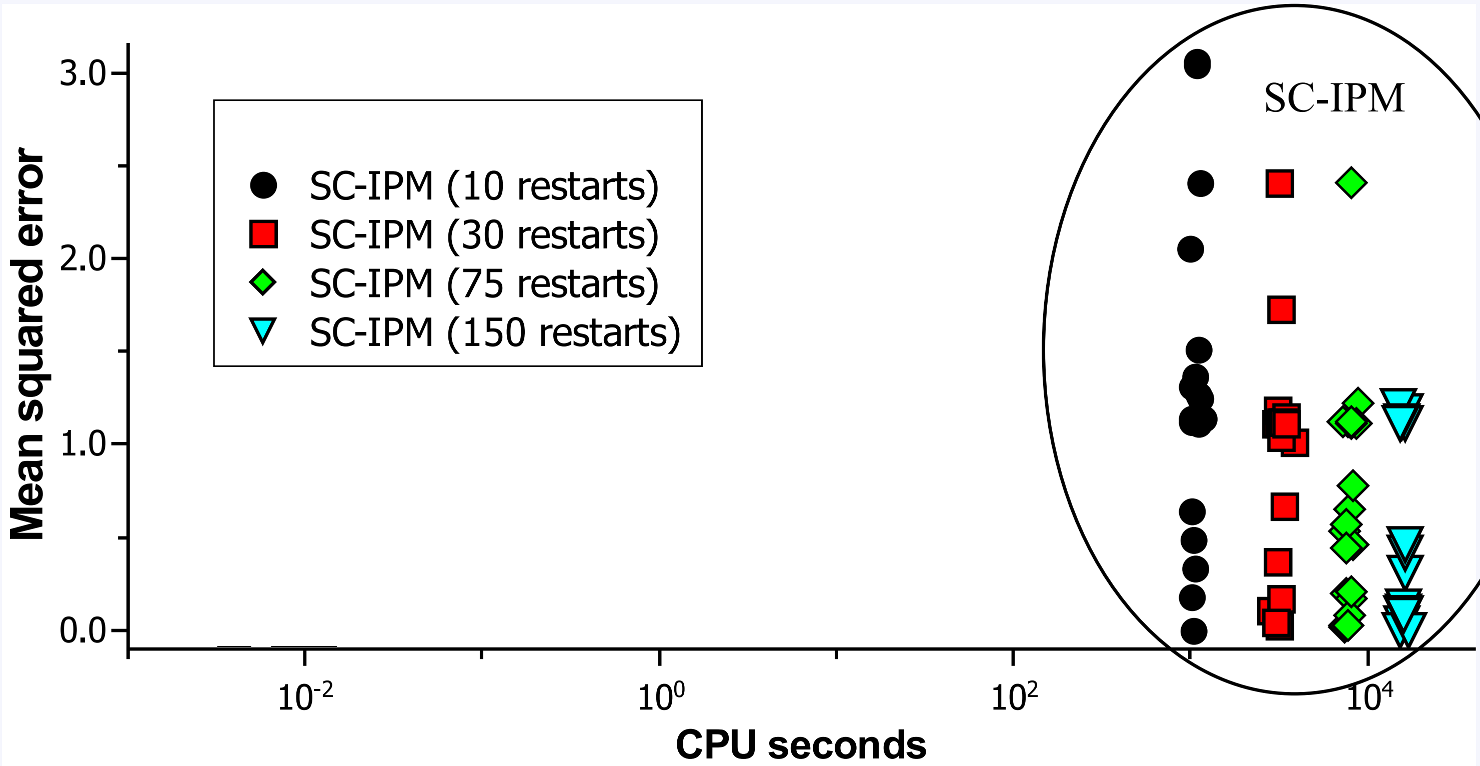
With smoothing, RPM can handle 10% noise on synthetic data.



The system also scales well to increasing numbers of generic processes and variables in the target model.

RPM and SC-IPM

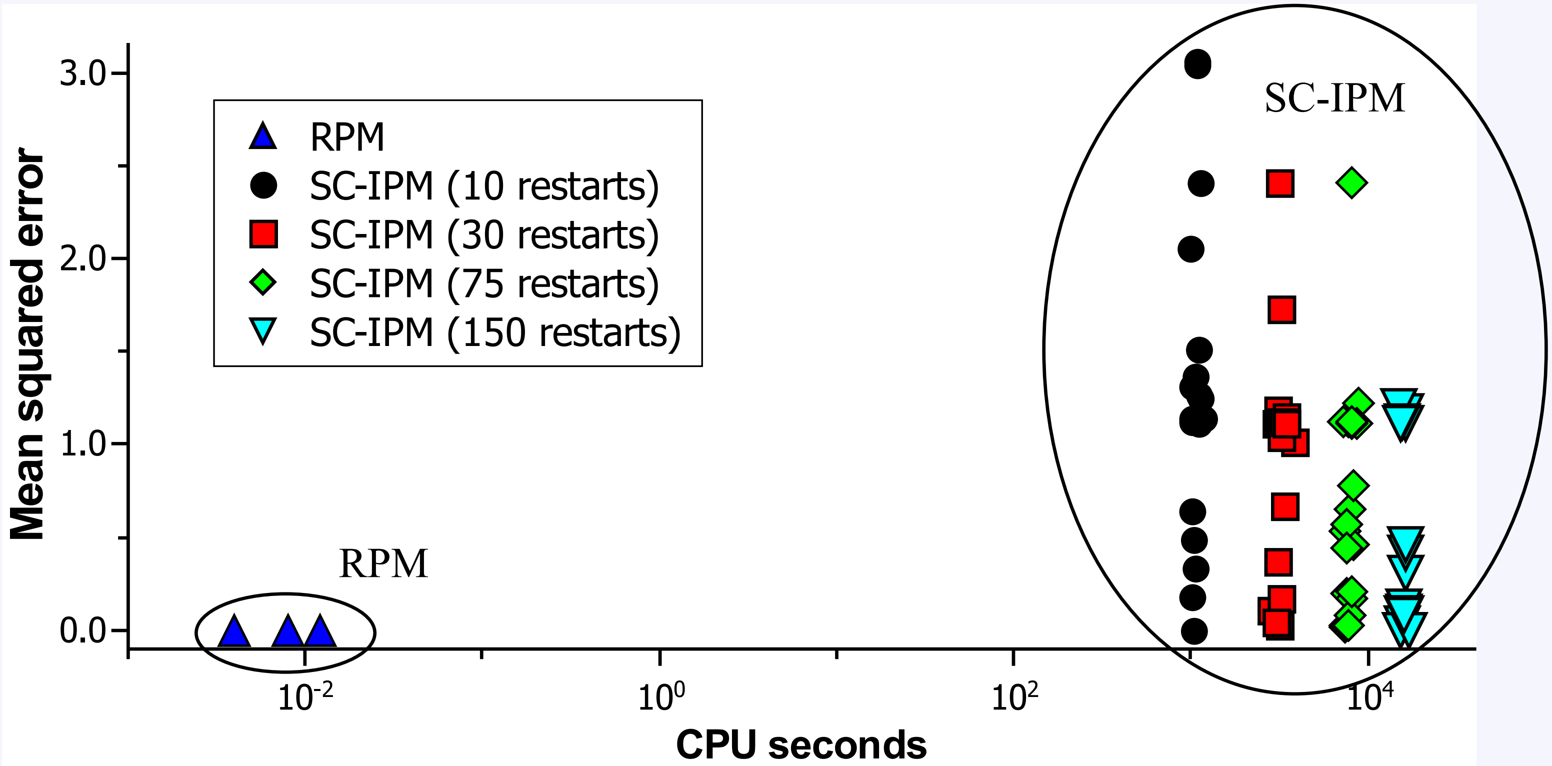
We compared RPM to SC-IPM, its predecessor, on synthetic data for a three-variable predator-prey ecosystem.



SC-IPM finds more accurate models with more restarts, but also takes longer to find them.

RPM and SC-IPM

We compared RPM to SC-IPM, its predecessor, on synthetic data for a three-variable predator-prey ecosystem.



RPM found accurate models far more reliably than SC-IPM and, at worst, ran *800,000 faster* than the earlier system.

Related and Future Research

Our approach builds on ideas from earlier research, including:

- Qualitative representations of scientific models (Forbus, 1984)
- Inducing differential equations (Todorovki, 1995; Bradley, 2001)
- Heuristic search and multiple linear regression

Our plans for extending the RPM system include:

- Replacing greedy search for models with beam search
- Adding heuristic search through the equation space
- Handling parametric rate expressions (e.g., using LMS)
- Dealing with unobserved variables (e.g., iterative optimization)

Together, these should extend RPM's coverage and usefulness.

Summary Remarks

In this talk, I presented a novel approach to inductive process modeling that:

- Incorporates a rate-based representation for processes
- Carries out heuristic search through the space of models
- Avoids the need for repeated simulation and random restarts
- Scales well to irrelevant variables and complex models
- Is more reliable and much more rapid than its predecessor

However, we can improve the framework's scalability further and reduce its reliance on simplifying assumptions.

Publications on Inductive Process Modeling

Todorovski, L., Bridewell, W., & Langley, P. (2012). Discovering constraints for inductive process modeling. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Toronto: AAAI Press.

Park, C., Bridewell, W., & Langley, P. (2010). Integrated systems for inducing spatio-temporal process models. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (pp. 1555-1560). Atlanta: AAAI Press.

Bridewell, W., & Todorovski, L. (2010). The induction and transfer of declarative bias. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (pp. 401-406). Atlanta: AAAI Press.

Bridewell, W., & Langley, P. (2010). Two kinds of knowledge in scientific discovery. *Topics in Cognitive Science*, *2*, 36-52.

Bridewell, W., Borrett, S. R., & Langley, P. (2009). Supporting innovative construction of explanatory scientific models. In A. B. Markman & K. L. Wood (Eds.), *Tools for Innovation*. Oxford: Oxford University Press.

Bridewell, W., Langley, P., Todorovski, L., & Dzeroski, S. (2008). Inductive process modeling. *Machine Learning*, *71*, 1-32.

Bridewell, W., Borrett, S., & Todorovski, L. (2007). Extracting constraints for process modeling. *Proceedings of the Fourth International Conference on Knowledge Capture* (pp. 87-94). Whistler, BC.

Bridewell, W., & Todorovski, L. (2007). Learning declarative bias. *Proceedings of the Seventeenth International Conference on Inductive Logic Programming*. Corvallis, OR.

Borrett, S. R., Bridewell, W., Langley, P., & Arrigo, K. R. (2007). A method for representing and developing process models. *Ecological Complexity*, *4*, 1-12.

Bridewell, W., Sanchez, 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.

Bridewell, W., Langley P., Racunas, S., & Borrett, S. R. (2006). Learning process models with missing data. *Proceedings of the Seventeenth European Conference on Machine Learning* (pp. 557-565). Berlin: Springer.

Langley, P., Shiran, O., Shrager, J., Todorovski, L., & Pohorille, A. (2006). Constructing explanatory process models from biological data and knowledge. *AI in Medicine*, *37*, 191-201.

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

Bridewell, W., Bani Asadi, N., Langley, P., & Todorovski, L. (2005). Reducing overfitting in process model induction. *Proceedings of the Twenty-Second International Conference on Machine Learning* (pp. 81-88). Bonn, Germany.

Todorovski, L., Bridewell, W., Shiran, O., & Langley, P. (2005). Inducing hierarchical process models in dynamic domains. *Proceedings of the Twentieth National Conference on Artificial Intelligence* (pp. 892-897). Pittsburgh, PA: AAAI Press.