Flexible Model Induction through Heuristic Process Discovery

Pat Langley

Institute for the Study of
Learning and Expertise

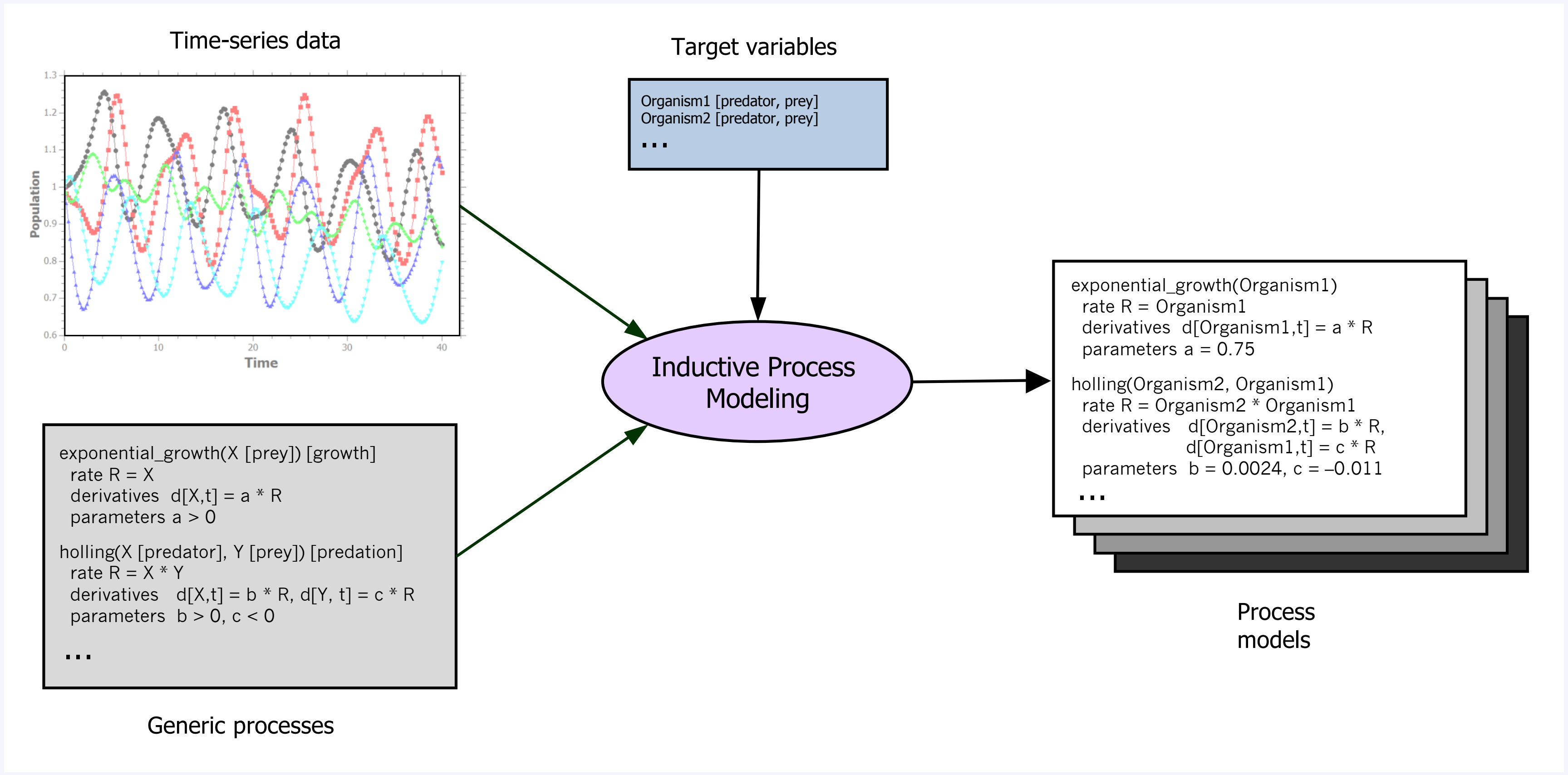
Adam Arvay

Department of Computer Science
University of Auckland

Thanks to W. Bridewell, S. Dzeroski, R. Morin, L. Todorovski, and others for their contributions to this research, which was funded by ONR Grant N00014-11-1-0107.

Inductive Process Modeling

Inductive process modeling construction of explanations for time series from background knowledge.



```
exponential_growth(X [prey]) [growth]
  rate R = X
  derivatives  d[X,t] = a * R
  parameters a > 0
holling(X [predator], Y [prey]) [predation]
  rate R = X * Y
  derivatives   d[X,t] = b * R, d[Y, t] = c * R
  parameters  b > 0, c < 0
 ...
```

```
exponential_growth(Organism1)
  rate R = Organism1
  derivatives  d[Organism1,t] = a * R
  parameters a = 0.75
holling(Organism2, Organism1)
  rate R = Organism2 * Organism1
  derivatives   d[Organism2,t] = b * R,
                d[Organism1,t] = c * R
  parameters  b = 0.0024, c = –0.011
 ...
```

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

A Formalism for Process Models

A quantitative process model comprises a set of processes *P*, each of which includes:

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

This formalism has important mathematical properties that aid in model induction.

The notation borrows directly from Forbus' (1984) notion of *qualitative processes*.

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[X] = A • r

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

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

These units serve as *building blocks* for constructing models.

RPM: Regression-Guided Process Modeling

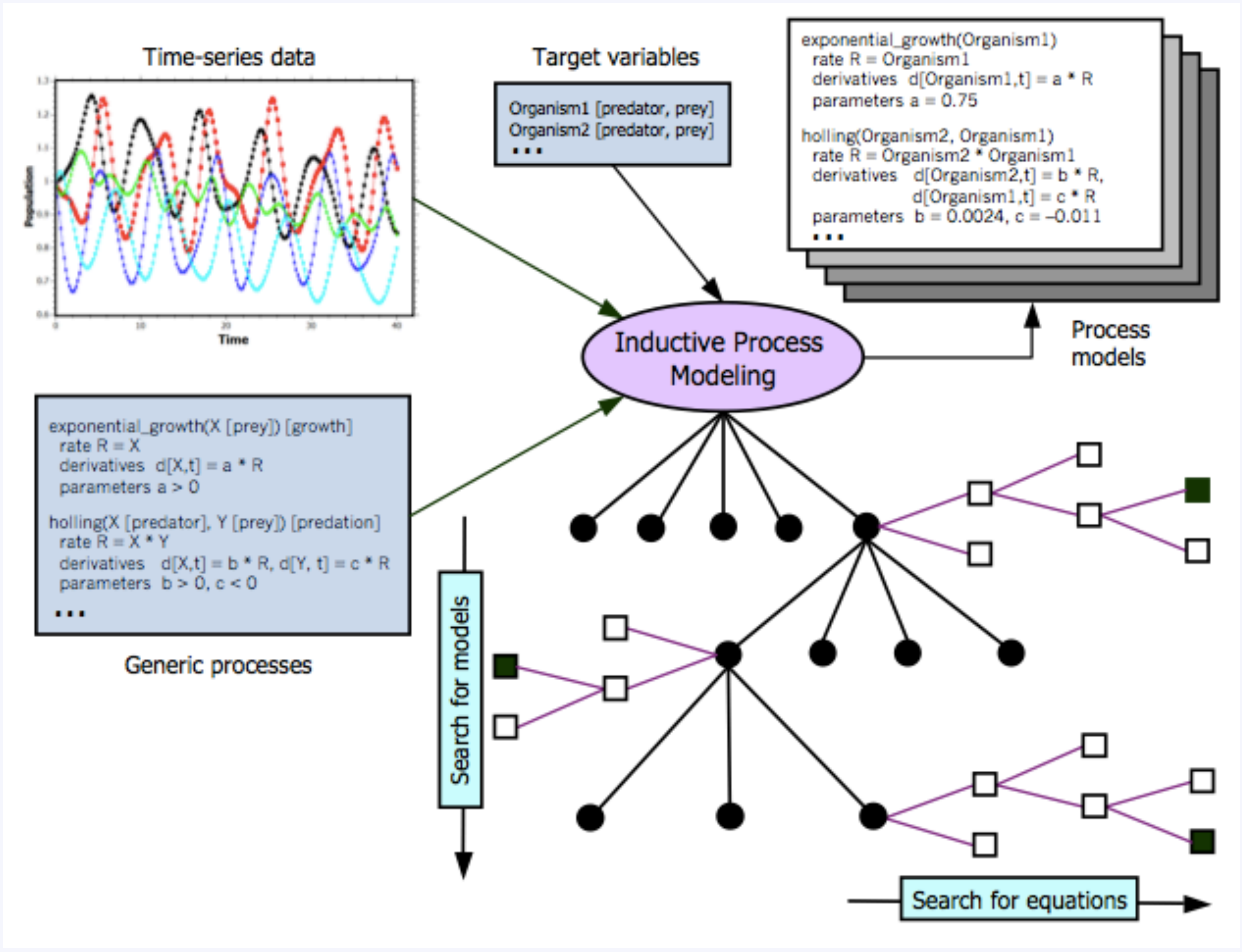
RPM (Langley & Arvay, 2015) is a system for process model induction that:

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

$\Big\}$ Assumes all variables observed
Rate expressions are parameter free

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

Two-Level Heuristic Search in RPM



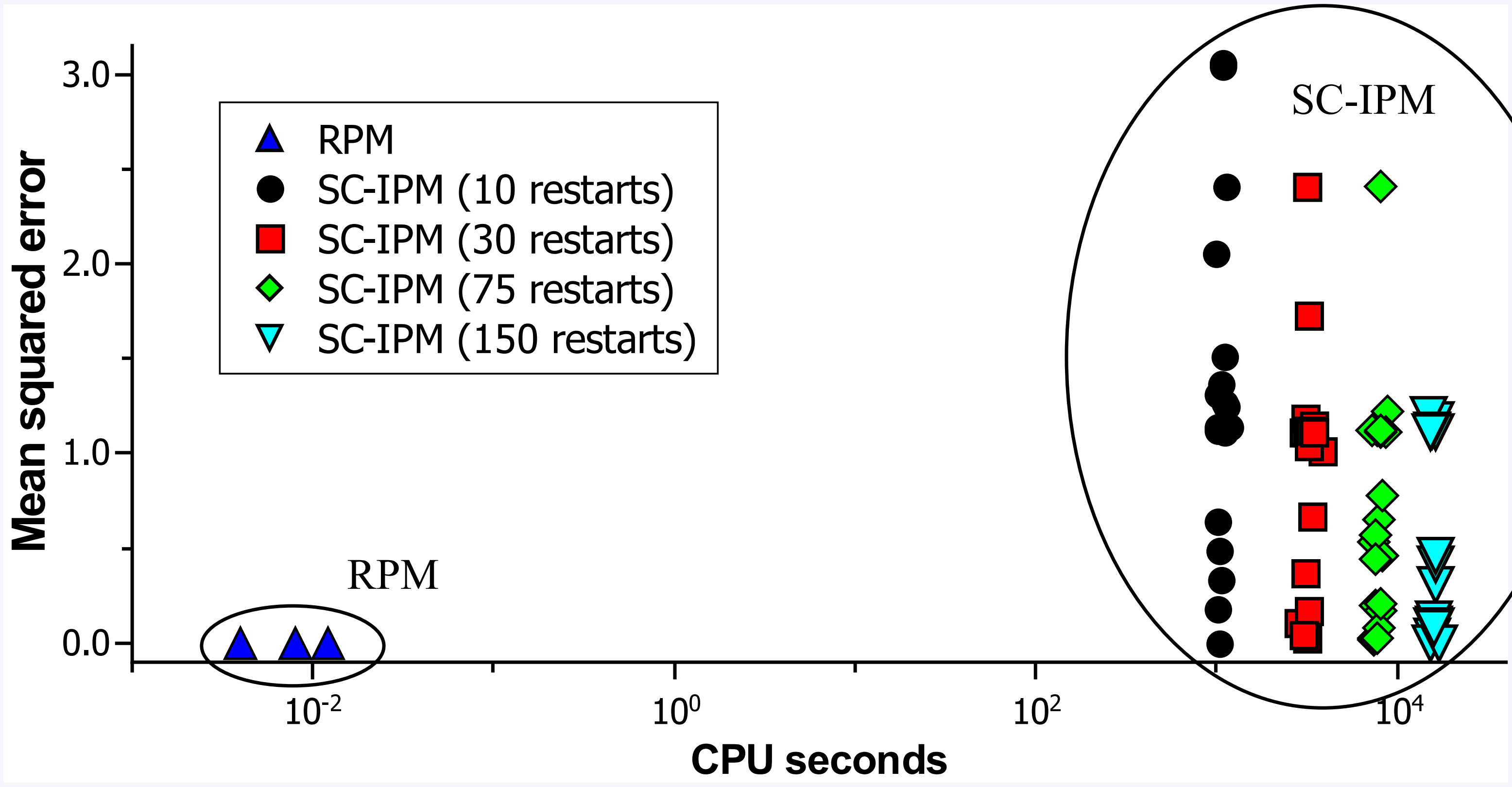
```
Organism1 [predator, prey]
Organism2 [predator, prey]
...
```

```
exponential_growth(Organism1)
  rate R = Organism1
  derivatives  d[Organism1,t] = a * R
  parameters a = 0.75

holling(Organism2, Organism1)
  rate R = Organism2 * Organism1
  derivatives   d[Organism2,t] = b * R,
                d[Organism1,t] = c * R
  parameters  b = 0.0024, c = –0.011
  ...
```

```
exponential_growth(X [prey]) [growth]
  rate R = X
  derivatives  d[X,t] = a * R
  parameters a > 0

holling(X [predator], Y [prey]) [predation]
  rate R = X * Y
  derivatives   d[X,t] = b * R, d[Y, t] = c * R
  parameters  b > 0, c < 0
  ...
```

RPM and SC-IPM

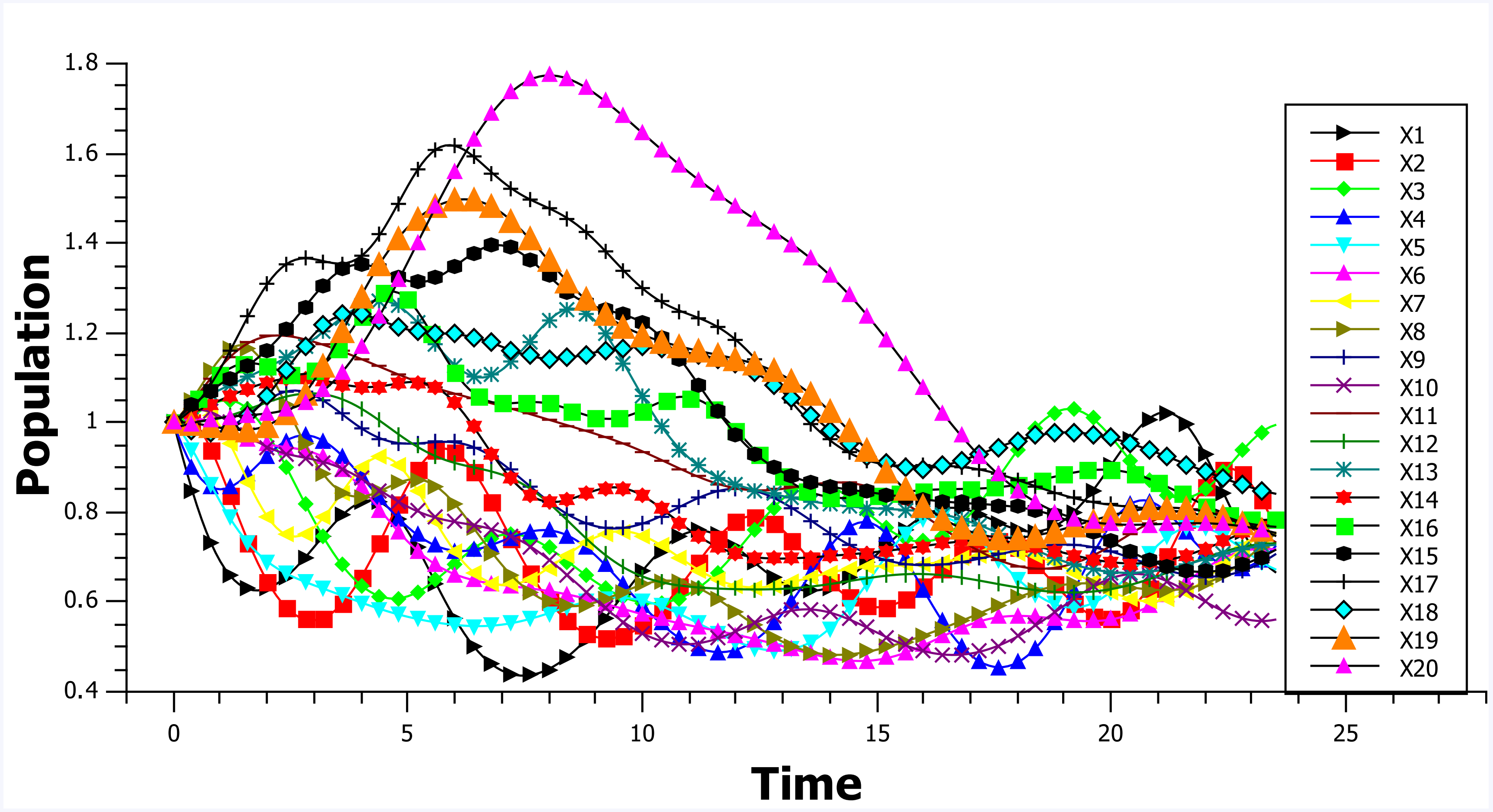
We compared RPM to SC-IPM, its predecessor, on synthetic data for a three-variable predator-prey ecosystem.



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

Behavior on Complex Synthetic Data

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



The system scales well to modeling tasks with many variables.

Selective Induction of Process Models

SPM (Arvay & Langley, 2016) incorporates three extensions that remedy three drawbacks of RPM:

- *Delaying binding* of some variables in generic processes until it finds evidence of a relationship;
- Combining *sampling* of processes with *backward elimination* to induce more complex equations;
- Finding *multiple* equations for each dependent variable and then searching for ways to *combine* them into consistent models.

Experiments showed these extensions give SPM much greater *coverage*, *scalability*, and *reliability* than its predecessor.

However, like RPM, it assumes that all generic processes are given in advance. We want to overcome this limitation.

A Revised Model Induction Task

Consider a new version of inductive process modeling that we will state as:

- *Given*: A set of typed variables and observed trajectories of their values over time;
- *Given:* A set of generic processes with type constraints and rate terms that might appear in explanations;
- *Given:* A set of *rate expressions* and *conceptual relations* that can serve as building blocks for new processes;
- *Find:* A model that uses these structures to explain the variables' observed trajectories;
- *Find:* New generic processes that are useful for the current and future modeling task.

We want computational methods that solve this revised problem.

Conceptual Processes and Rate Templates

```
(a) prey
     :variables ((a organism)(b organism))
     :inputs    (b)
     :outputs   (a)
    inflow
     :variables ((a organism))
     :inputs    ( )
     :outputs   (a)
(b) identity
     :expression p
    product
     :expression p · q
    ratio
     :expression p / q
```

A Naïve Approach to Process Discovery

Our first implemented system, FPM/N, takes a naive approach to flexible process modeling by:

- Calling on SPM to find acceptable process models using an initial set of generic processes;
- If this fails, using conceptual relations and rate templates to generate an expanded set of processes;
- Calling on SPM again to find acceptable models using this expanded set.

Given enough resources, this scheme should find reasonable models even when it must invent new processes.

However, the exhaustive approach of considering all possible processes seems unlikely to scale well.

A Heuristic Approach to Process Discovery

Our second system, FPM, instead takes a heuristic approach to flexible process modeling by:

- Calling on SPM to find acceptable process models using an initial set of generic processes;
- If this fails, identifying the variable V for which SPM could find no consistent equations;
- Using conceptual relations and rate templates to generate only those processes that incorporate V;
- Calling on SPM again to find acceptable models using this expanded set, repeating the procedure if necessary.

This heuristic approach focuses attention on relevant processes, which should let it scale much better.

Empirical Studies of Process Discovery

We have evaluated FPM/N and FPM using multiple methods on:

- Natural predator-prey data from Veilleux (1979)
- Synthetic data for a five-organism predator-prey system
- Synthetic data for an aquatic ecosystem with grazers and nutrients

In each case, they discovered the processes we had omitted and induce accurate models.

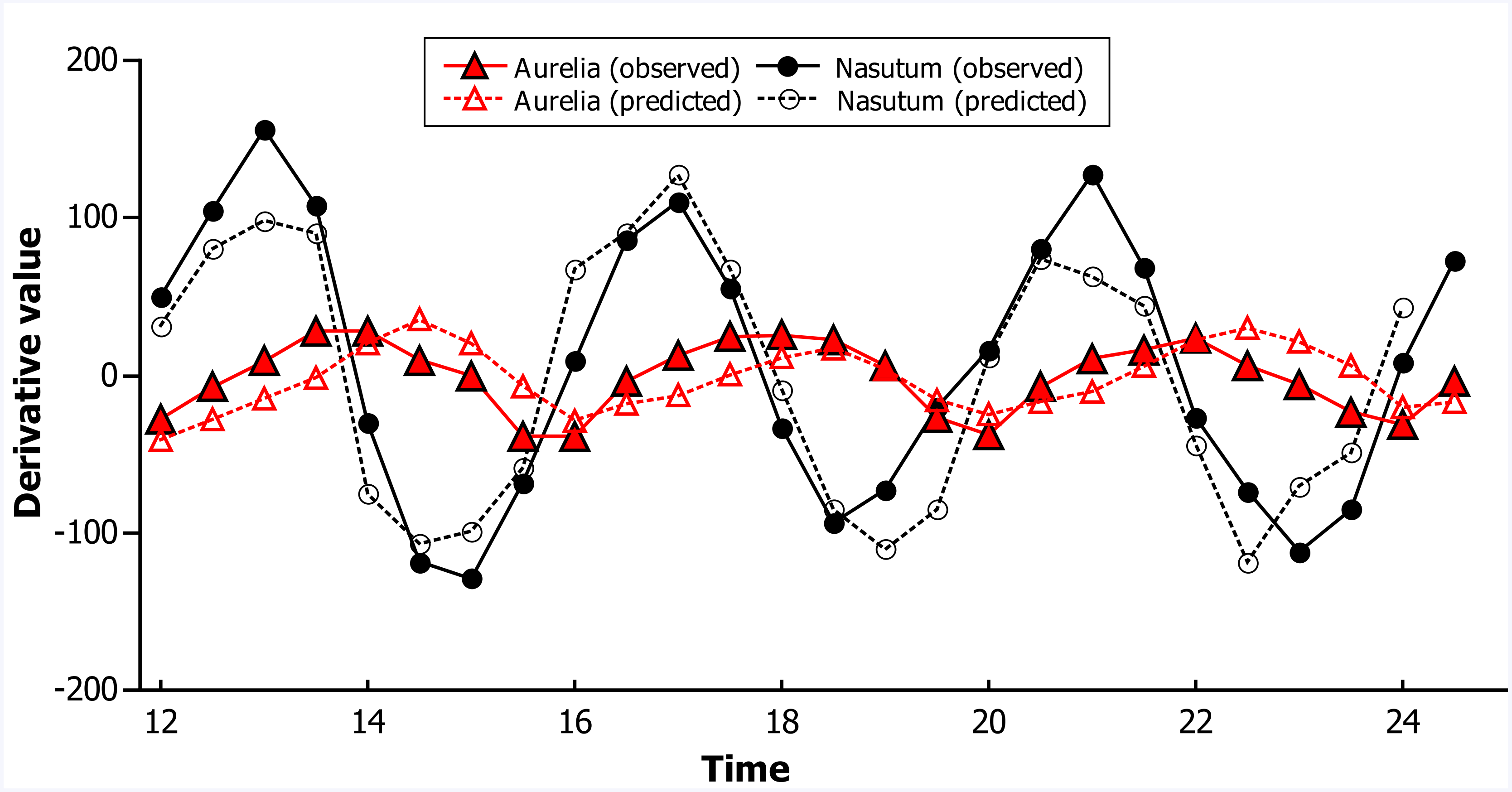
We also carried out additional experiments that demonstrated:

- Transfer and reuse of processes across modeling tasks
- Scaling to increased numbers of variables and rate templates

Together, these suggest that our approach is more flexible and robust than earlier systems.

Behavior on Natural Data

FPM invents a process for *predation* to fit Veilleux's (1979) data.

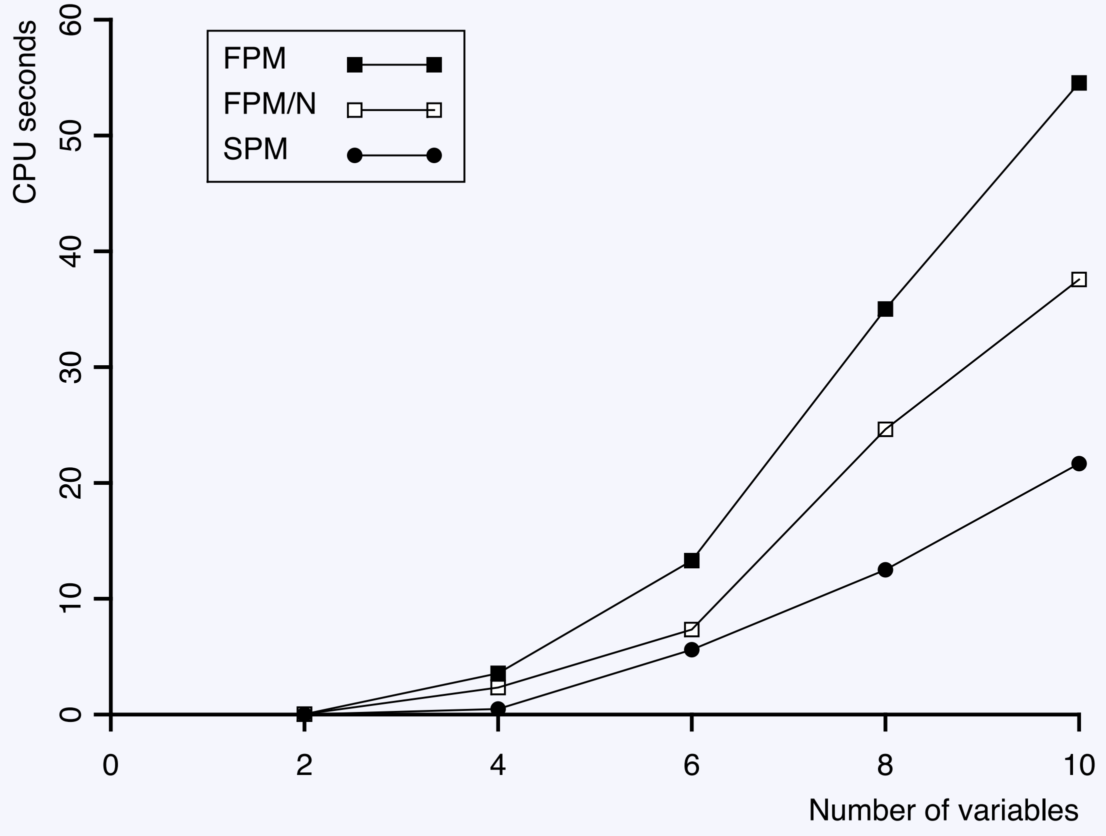


$$d[aurelia] = 0.75 \cdot aurelia - \boxed{0.11 \cdot nasutum \cdot aurelia}\ [r^2 = 0.84]$$

$$d[naustum] = \boxed{0.0024 \cdot nasutum \cdot aurelia} - 0.57 \cdot nasutum\ [r^2 = 0.71]$$

Scaling to Number of Variables

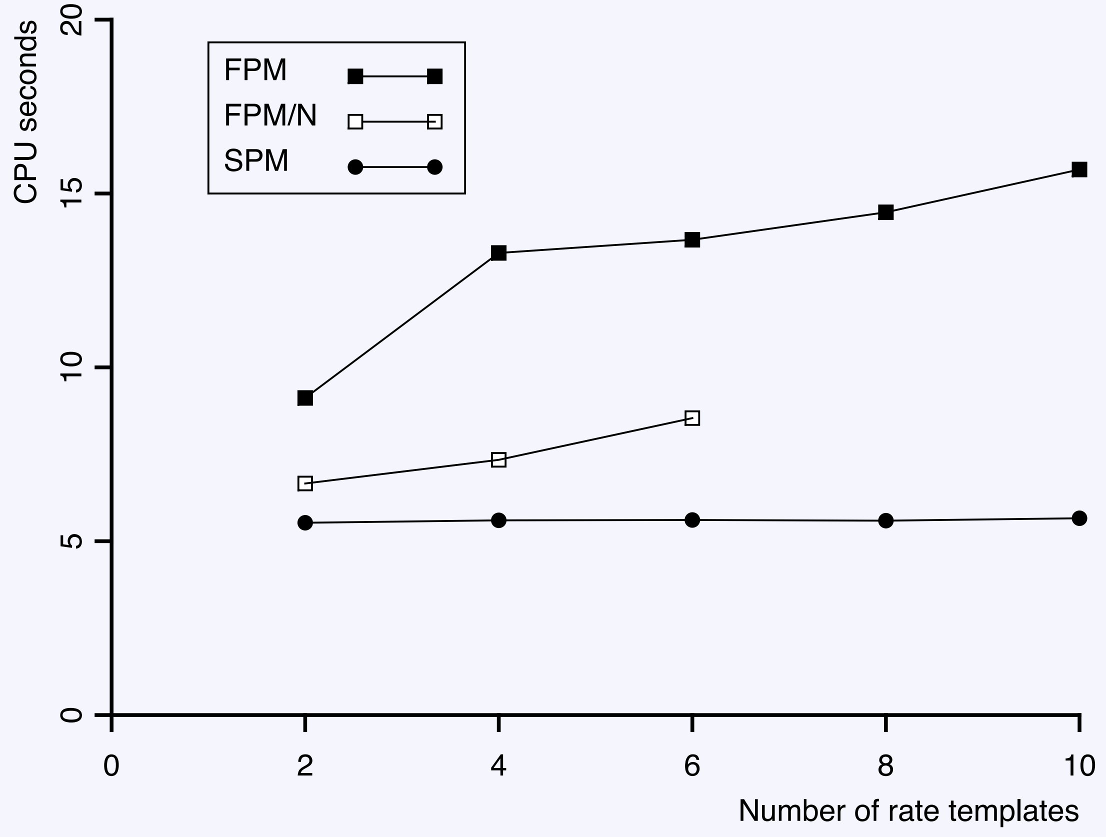
Surprisingly, with increasing number of terms in a target equation, induction time for FPM grows more rapidly than for FPM/N.



Here SPM was provided with all generic processes it needed.

Scaling to Number of Rate Templates

FPM’s induction time is not influenced strongly by the number of rate templates, but FPM/N fails when we omit more than six.



Again, SPM was provided with all generic processes it needed.

Related and Future Research

Our approach builds on ideas from earlier research, including:

- Qualitative representations of scientific models (Forbus, 1984)
- Inducing differential equations (Todorovski, 1995; Mai et al., 2016)
- Heuristic search and multiple linear regression
- Learning rules to fill in gaps (Sleeman, 1984; VanLehn et al., 1992)

Our plans for extending the SPM system include:

- Handling parametric rate expressions (gradient descent)
- Dealing with unobserved variables (iterative optimization)

Together, these should extend FPM's coverage and usefulness even further.

Summary Comments

We have reported an extension to inductive process modeling that extends earlier work by:

- Generating candidate processes from more basic elements
- Using impasses to focus generation on problematic variables
- Retaining useful processes for later use in model induction

We also described a new system, FPM, that incorporates these ideas and studied its behavior experimentally.

For more information, see *http://www.isle.org/process/* .

End of Presentation