knowledge-based framework for modeling dynamic environmental systems

sašo džeroski (joint work with **ljupčo todorovski**)
jožef stefan institute, ljubljana, slovenia

4th workshop on environmental applications of machine learning
4th european conference on ecological modelling

bled, september 2004

motivation: automated modeling

- computational support for building equation-based models
- existing modeling (system identification) methods:
  - assume that the model structure is known
  - or assume linear/NN structure (if the model is unknown)
  - task: determine the values of the model parameters
- in contrast, our (equation discovery based) method:
  - does not assume (a single) prescribed model structure
  - task: determine both structure and parameters of the model

motivation: the use of domain-specific knowledge

- two approaches to modeling of real-world systems:
  1. theoretical (knowledge-driven)
     domain expert derives a proper model structure
     based on domain-specific modeling knowledge
  2. empirical (data-driven)
     try different models to fit observed data (trial-and-error)

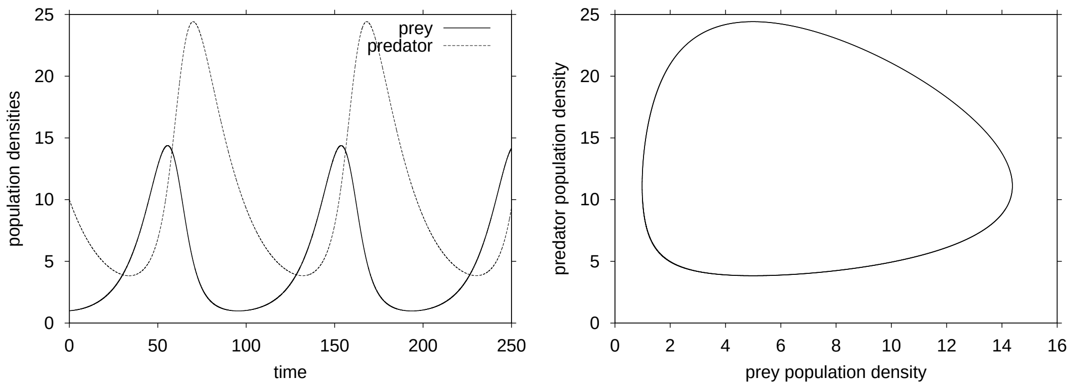
- in 1., a lot of domain knowledge and few data are needed;
  terms in equations typically correspond to processes
  (equations are meaningful to domain scientists)

- in 2., many data of good quality are needed;
  no domain knowledge can be used
  terms in equations are rarely meaningful to the domain scientist

- to integrate 1. and 2., that is the question!

discovering dynamics: problem definition and example

- GIVEN an example behavior of a dynamic system:



- FIND the system dynamics equations:

$$\dot{N} = a \cdot N - b \cdot NP$$
$$\dot{P} = c \cdot NP - d \cdot P$$

  – N is prey population density (hares)
  – P is predator population density (lynx)

discovering dynamics: a declarative bias approach

- space of possible equations (language bias):
  - user defined (declarative)
  - based on the domain-specific knowledge
- declarative bias formalism in lagramge: context-free grammar
  - prescribes the form of the expression on the right-hand side
  - generates legal expressions in the C programming language
  - result of the derived expressions: (double)
- discovered equations are of the form $\dot{v_d} = E$,
  where E is an expression derived using the given grammar

common language biases for equations

- universal grammar (arithmetical expressions):

$$
\begin{aligned}
E &\rightarrow E + F \mid E - F \mid F \\
F &\rightarrow F * T \mid F/T \mid T \\
T &\rightarrow const \mid v \mid (E)
\end{aligned}
$$

- multivariate polynomials:

$$
\begin{aligned}
E &\rightarrow const \mid const \cdot F \mid E + const \cdot F \\
F &\rightarrow v \mid v \cdot F
\end{aligned}
$$

monod example (subexpressions): context free grammar

- context free grammar used for environmental dynamic systems:

$$
\begin{aligned}
E &\rightarrow \; const \mid const \cdot F \mid E + const \cdot F \\
F &\rightarrow v \mid Y \mid v \cdot Y \\
Y &\rightarrow \texttt{monod}(const, v)
\end{aligned}
$$

$$Y : \frac{v}{v + const}$$

- user defined function `monod`:

```
double monod(double c, double v) {
   return(v / (v + c));
}
```

lagramge: search in the space of structures

- the grammar defines the space of possible equation structures: the structures that can be derived with at most N applications of rules from the grammar
- the grammar also
  - orders the search space
  - starting from the simplest equation structure ...
  - allows the use of different search strategies
- two search strategies implemented in lagramge:
  - beam search
  - exhaustive search

lagramge: constant parameters fitting

- nonlinear optimization method:
  - downhill simplex
  - minimize the difference between given and simulated data
  - use integration instead of differentiation:

$$\dot{V} = F(V) \longrightarrow V(t_i) = \int_{t_0}^{t_i} F(V)dt$$

- two different heuristic functions:
  - SSE = sum of squared errors
  - MDL = SSE + equation length penalty

lagramge: summary

- advantages of using declarative bias in discovering dynamics:
  - can use different search strategies (e.g., beam search)
  - can discover complex models
  - can use background knowledge from the study domain
  - and thus compensate for incomplete (and noisy) data
- Q: where do the grammars come from?

grammar source 1: domain-specific modeling knowledge

- knowledge organized around central notion of process
  - identify basic processes in the domain
  - what models are used for individual processes?
  - how are they combined into models of the entire system?
- similar to compositional modeling [Falkenheiner & Forbus, 1994]
  - model fragments + rules for combining them
  - model fragments = models of individual processes
  - building model = search for appropriate combination of fragments
  - support for building QUALITATIVE models only

example: population dynamics modeling knowledge

- collected from textbooks in modeling of biological systems
- population: group of individuals of the same species inhabiting the same area
- population dynamics: change of the population density
- basic population dynamics processes:
  - population growth, population decay
  - interaction between populations

example: simple population dynamics model

- Lotka-Volterra population dynamics model:

$$\dot{N} = a \cdot N - b \cdot NP$$
$$\dot{P} = c \cdot NP - d \cdot P$$

- general scheme:

$$\dot{N} = \mathrm{growth}(N) - b \cdot \mathrm{feeds\_on}(P, N)$$
$$\dot{P} = c \cdot \mathrm{feeds\_on}(P, N) - \mathrm{decay}(P)$$

- assumes:
  - unlimited prey growth – $\mathrm{growth}(N) = aN$
  - unlimited predator decay – $\mathrm{decay}(P) = dP$
  - unlimited predator capacity – $\mathrm{feeds\_on}(P, N) = PN$

relaxing the assumptions: models of population growth

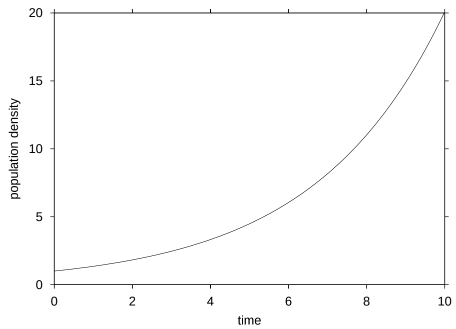
- exponential (unlimited) growth:

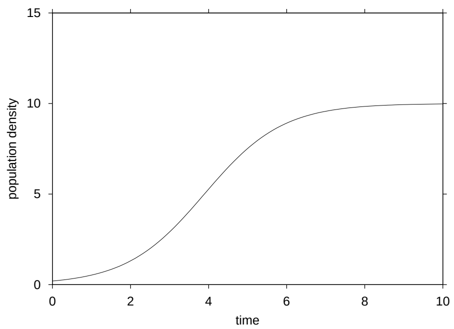
$$\text{growth}(N) = aN$$

- logistic (limited) growth:

$$\text{growth}(N) = aN(1 - N/K)$$

K is carrying capacity of the environment





models of predator-prey interaction

- unsaturated (unlimited) predation capacity:

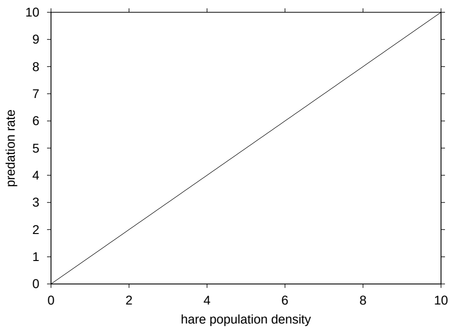
$$\text{feeds\_on}(P, N) = aPN$$

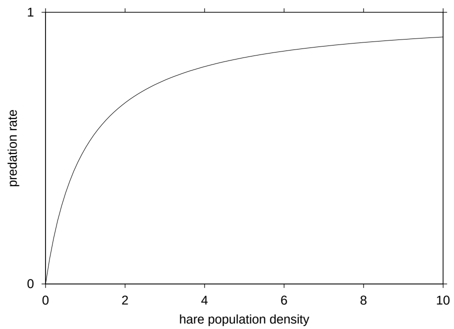
- saturated predation capacity:

$$\text{feeds\_on}(P, N) = P \cdot \text{saturation}(N)$$

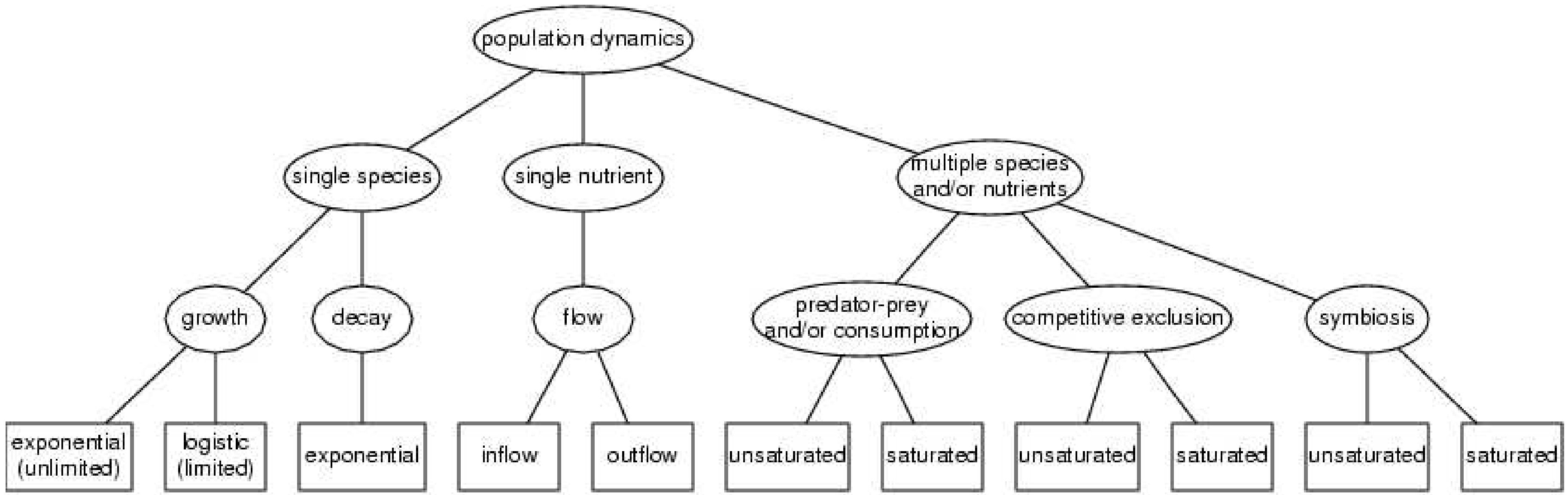
$$\text{saturation}(N) = AN/(N + B)$$

A is the limit of the predation saturation, B is the saturation rate





domain-specific knowledge (1): taxonomy of process classes



domain-specific knowledge (2): models of the individual processes

- alternative sub-model templates for modeling individual processes:

```
process class Growth(Population p)

process class Exponential_growth is Growth
   expression const(growth_rate,0,1,Inf) * p

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

```
process class Feeds_on(Population p, set of Concentration cs)
   condition p ∉ cs
   expression p * Π_{c∈cs} Saturation(c)
```

domain-specific knowledge (3): combining scheme

- combining models of individual processes into a model of the entire system:

```
combining scheme Population_dynamics(Inorganic i)
```

$$\frac{d}{dt}\mathtt{i} = + \mathtt{Flow(i)}$$
$$- \sum_{\mathtt{food,i} \in \mathtt{food}} \mathtt{const(\_,0,1,Inf)} * \mathtt{Feeds\_on(p,\ food)}$$

```
combining scheme Population_dynamics(Population p)
```

$$\frac{d}{dt}\mathtt{p} = + \mathtt{Growth(p)} - \mathtt{Decay(p)}$$
$$+ \sum_{\mathtt{food}} \mathtt{const(\_,0,1,Inf)} * \mathtt{Feeds\_on(p,\ food)}$$
$$- \sum_{\mathtt{pred,food,p} \in \mathtt{food}} \mathtt{const(\_,0,1,Inf)} * \mathtt{Feeds\_on(pred,\ food)}$$

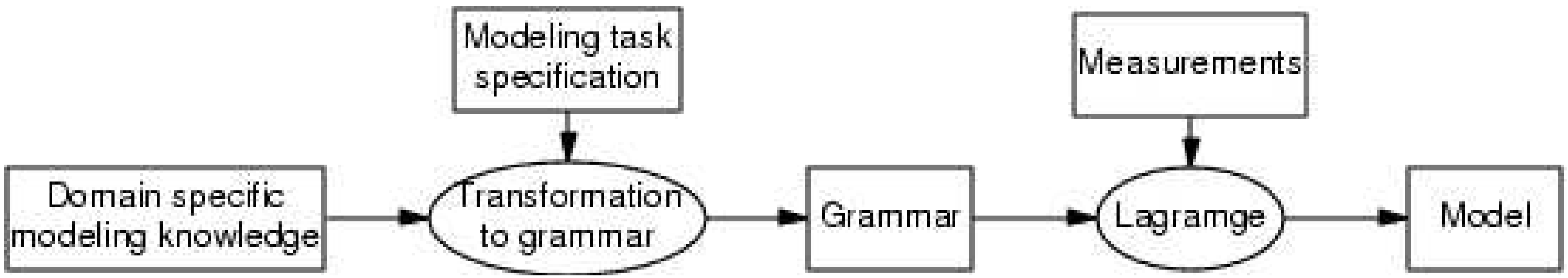
modeling task specification: Lotka-Volterra

```
variable Population hare
variable Population lynx

process Growth(hare)
process Decay(lynx)
process Feeds_on(hare, lynx)
```

- two system variables: hare (prey) and lynx (predator)
- three processes: growth, decay, and predator-prey interaction

integrating knowledge in the process of model induction



1. from task specification and knowledge to grammar
2. using the grammar for equation discovery with lagramge

domain-specific knowledge transformed into grammar

- context-dependent grammar for equation discovery:

```
general_lotka_volterra ->
    time_deriv(hare) = Growth(hare) - const[0:1:]  * Feeds_on(lynx,hare)
    time_deriv(lynx) = const[0:1:]  * Feeds_on(lynx,hare) - Decay(lynx)

Growth(hare) -> const[0:1:]  * hare
Growth(hare) -> const[0:1:]  * hare * (1 - hare / const[0:1:])

Decay(lynx) -> const[0:1:]  * lynx

Feeds_on(lynx,hare) -> lynx * Saturation(hare)

Saturation(hare) -> hare
Saturation(hare) -> hare / (hare + const[0:1:])
```

- note the:
  - context-dependent constraint
  - bounds on the constant parameters

Lagoon of Venice: task specification

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

- six observed variables (only `biomass` modeled)
- two fixed processes
- one process template (unknown limiting factors)

Lagoon of Venice: results

- relatively high model errors (rmse – 86.2841; 157.537)
  - due to high measurement errors (order 20% – 50%)
  - predict most of the biomass concentration peaks and crashes
- comprehensible white-box models induced:
  - ecology expert can easily understand them
  - they reveal the limiting factors for algae growth
  - dissolved oxygen, nitrogen-based nutrients, and temperature

Lagoon of Venice: results

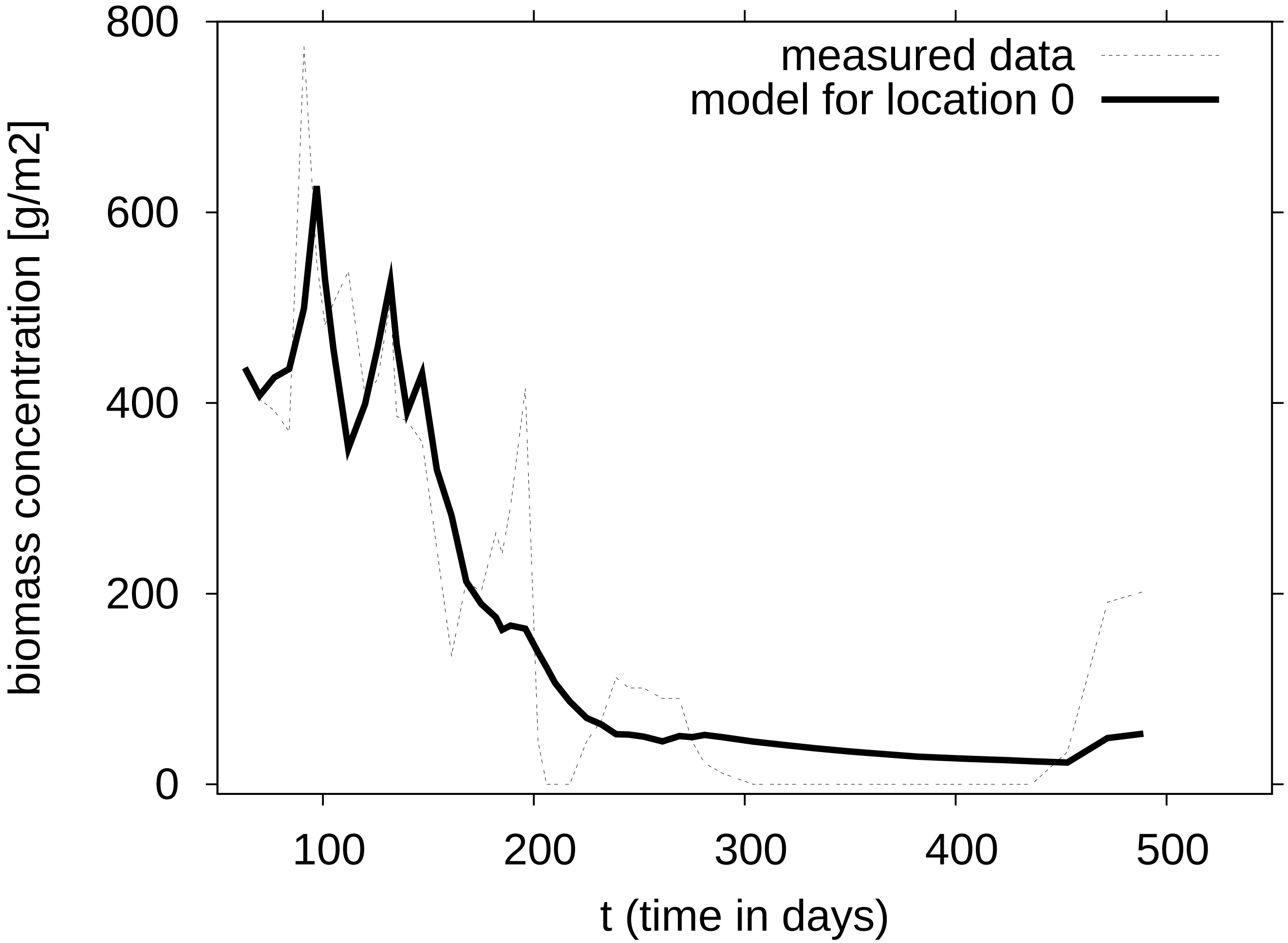
location 0:

$$\dot{\texttt{biomass}} = 6.17 \cdot 10^{-5} \cdot \texttt{biomass} \cdot (1 - \frac{\texttt{biomass}}{1.80}) + 3.01 \cdot 10^{-4} \cdot \texttt{biomass} \cdot \texttt{DO} \cdot \frac{\texttt{NO3}}{\texttt{NO3} + 6.28} - 0.0319 \cdot \texttt{biomass}.$$

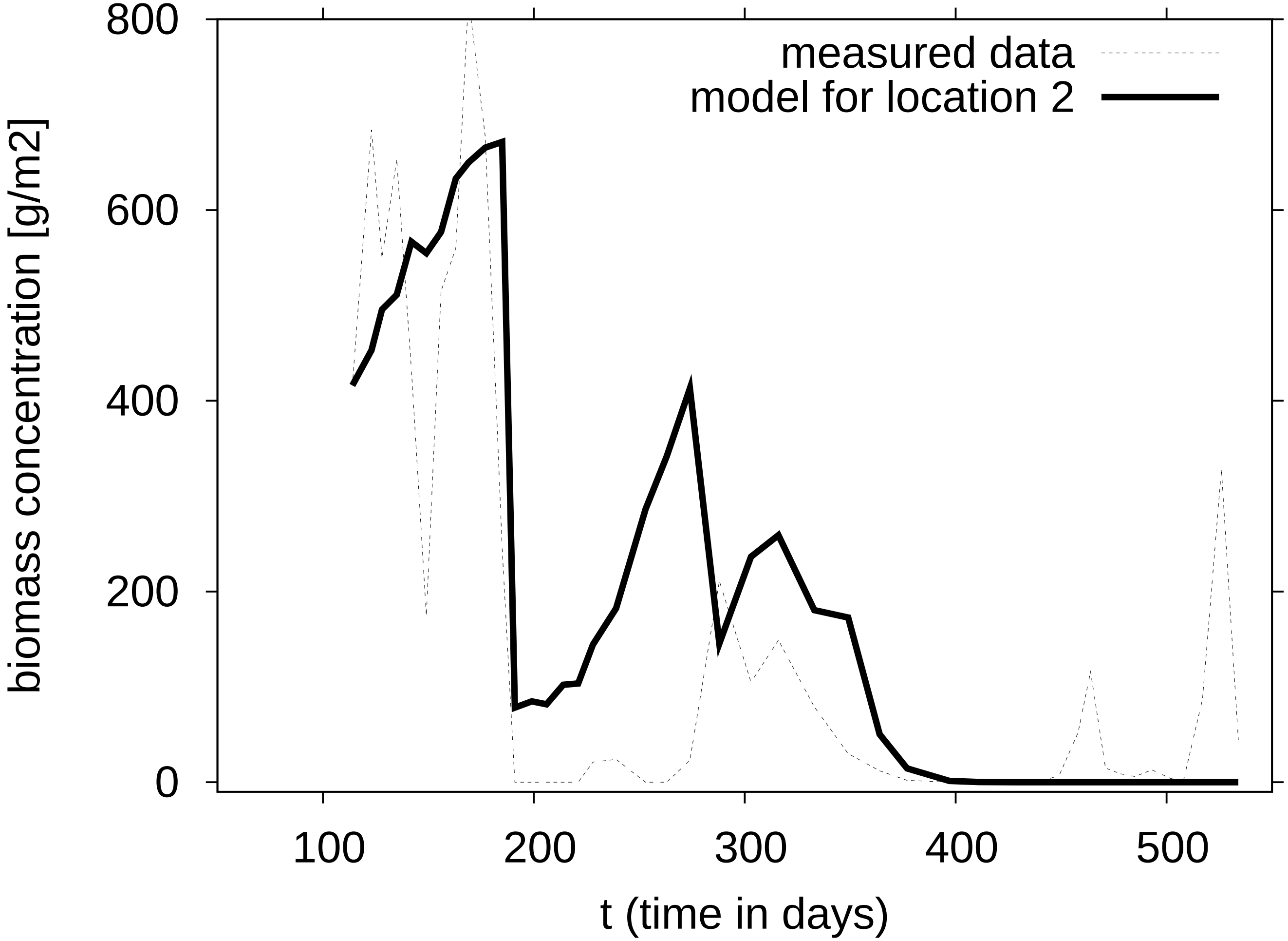
location 4:

$$\dot{\texttt{biomass}} = 4.79 \cdot 10^{-5} \cdot \texttt{biomass} \cdot (1 - \frac{\texttt{biomass}}{0.844}) + 0.406 \cdot \texttt{biomass} \cdot (1 - e^{-0.216 \cdot \texttt{temp}}) \cdot (1 - e^{-0.413 \cdot \texttt{DO}}) \cdot \frac{\texttt{NH3}}{\texttt{NH3} + 10} - 0.0343 \cdot \texttt{biomass}.$$

Lagoon of Venice: simulation of the induced model



Lagoon of Venice: simulation of the induced model



Lake Glumsoe

task specification

variable Inorganic temp, nitro, phosp
variable Population phyto, zoo

process Decay(phyto) phyto_decay
process Feeds_on(phyto, *) phyto_grazing
process Feeds_on(zoo, phyto) zoo_grazing

discovered model

$$\dot{\mathtt{phyto}} = 0.553 \cdot \mathtt{temp} \cdot \frac{\mathtt{phosp}}{0.0264 + \mathtt{phosp}} - 4.35 \cdot \mathtt{phyto} - 8.67 \cdot \mathtt{phyto} \cdot \mathtt{zoo}.$$

grammar source 2: Ringkøbing fjord

- expert specified only part of the model structure:

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

- two parts of the structure left unspecified:
  - gate opening f that depends on number of open gate parts
  - wind forcing g that depends on wind direction and speed

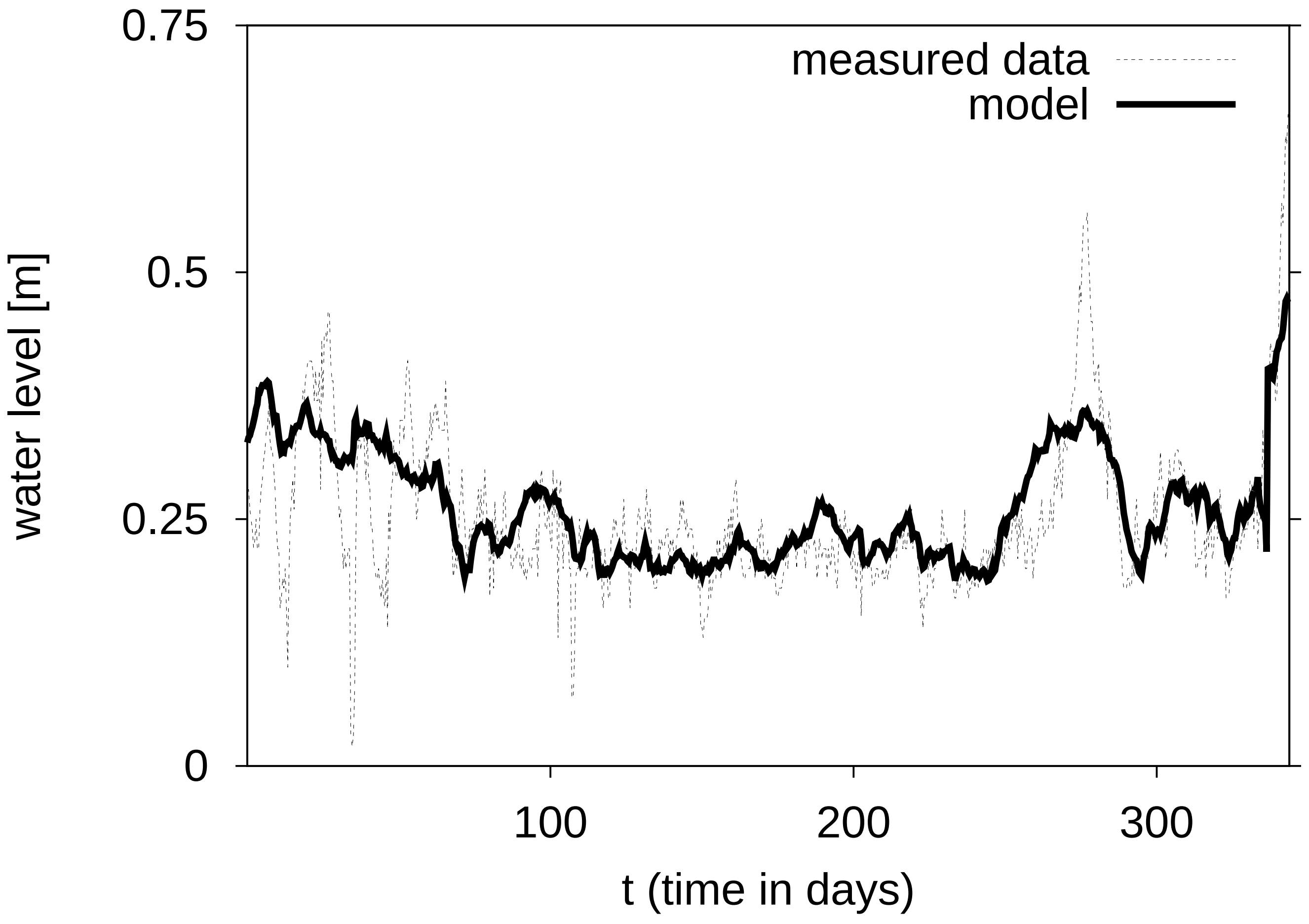
- experiments:
  - f – constant or polynomial
  - g – constant, polynomial, and trigonometric functions
  - maximal polynomial degree of 5

Ringkøbing fjord: results

task specification	training rmse	CV rmse	#CMS
constants	0.0848	0.106	1
polynomials	0.0655	0.0931	378
polynomials + sin/cos	0.0585	0.0903	2184
no partial structure (black-box)	0.0556	2.389	2801

- best model captures the long-term dynamics of the water level
- can also predict short-term changes (one hour or day)
- allows for comparison of wind and gate opening influence
- black-box polynomials overfit the training data

Ringkøbing fjord: simulation of the induced model



grammar source 3: existing models

- GIVEN:
  - an existing (imperfect) model M_0
  - a set of new observations/measurements
- FIND a revised model M_R that
  - minimizes the discrepancy between observed and measured values of the system variables
  - is as similar as possible to M_0

outline of the approach

- take an existing model
- construct a grammar that derives the model
- identify the unreliable parts of the model (expert)
- add alternative grammar rules for these parts (expert)
- use lagramge on the observations and grammar
- preference for models similar to the original one (minimality of change principle)

revising a part of a global vegetation model: CASA-NPPc initial

$$
\begin{aligned}
&NPPc = \max(0, E \cdot IPAR)\\
&\quad E = 0.389 \cdot T1 \cdot T2 \cdot W\\
&\quad\quad T1 = 0.8 + 0.02 \cdot topt - 0.0005 \cdot topt^2\\
&\quad\quad T2 = 1.1814/((1 + \exp(0.2 \cdot (TDIFF - 10))) \cdot (1 + \exp(0.3 \cdot (-TDIFF - 10))))\\
&\quad\quad\quad TDIFF = topt - tempc\\
&\quad\quad W = 0.5 + 0.5 \cdot eet/PET\\
&\quad\quad\quad PET = 1.6 \cdot (10 \cdot \max(tempc, 0)/ahi)^A \cdot pet\_tw\_m\\
&\quad\quad\quad\quad A = 0.000000675 \cdot ahi^3 - 0.0000771 \cdot ahi^2 + 0.01792 \cdot ahi + 0.49239\\
&\quad IPAR = FPAR\_FAS \cdot monthly\_solar \cdot SOL\_CONV \cdot 0.5\\
&\quad\quad FPAR\_FAS = \min((SR\_FAS - 1.08)/srdiff, 0.95)\\
&\quad\quad\quad SR\_FAS = (1 + fas\_ndvi/1000)/(1 - fas\_ndvi/1000)\\
&\quad\quad SOL\_CONV = 0.0864 \cdot days\_per\_month
\end{aligned}
$$

proposed alternatives

- experts identified four "weak" parts of the model:
  - equations for E, $T1$, $T2$, and SR_FAS
- two alternatives for $E = 0.0389 \cdot T1 \cdot T2 \cdot W$:
  1. $E = \texttt{const} \cdot T1 \cdot T2 \cdot W$
  2. $E = \texttt{const} \cdot T1^{\texttt{const}} \cdot T2^{\texttt{const}} \cdot W^{\texttt{const}}$
- two alternatives for $T1 = 0.8 + 0.02 \cdot topt - 0.0005 \cdot topt^2$:
  1. $T1 = \texttt{const} + \texttt{const} \cdot topt + \texttt{const} \cdot topt^2$
  2. $T1 \rightarrow \texttt{const} \mid \texttt{const} + (T1) * topt$

the revised CASA-NPPc model

$$NPPc = \max(0, E \cdot IPAR)$$
$$E = 0.402 \cdot T1^{0.624} \cdot T2^{0.215} \cdot W^{0}$$
$$T1 = 0.680 + 0.270 \cdot topt - 0 \cdot topt^{2}$$
$$T2 = 1.1814/((1 + \exp(0.2 \cdot (TDIFF - 10))) \cdot (1 + \exp(0.3 \cdot (-TDIFF - 10))))$$
$$TDIFF = topt - tempc$$
$$TDIFF = topt - tempc$$

$$W = 0.5 + 0.5 \cdot eet/PET$$
$$PET = 1.6 \cdot (10 \cdot \max(tempc, 0)/ahi)^{A} \cdot pet\_tw\_m$$
$$A = 0.000000675 \cdot ahi^{3} - 0.0000771 \cdot ahi^{2} + 0.01792 \cdot ahi + 0.49239$$

$$IPAR = FPAR\_FAS \cdot monthly\_solar \cdot SOL\_CONV \cdot 0.5$$
$$FPAR\_FAS = \min((SR\_FAS - 1.08)/srdiff, 0.95)$$
$$SR\_FAS = (1 + fas\_ndvi/750)/(1 - fas\_ndvi/750)$$
$$SOL\_CONV = 0.0864 \cdot days\_per\_month$$

- relative accuracy improvement 9%, revised model simpler

a brief summary of the talk

- integration of different aspects of knowledge:
  - taxonomy of basic processes
  - partial specification of the model structure
  - existing models
  - ...

  in equation discovery through grammars (lagramge)

- grammars can be
  - obtained by transforming the domain-specific knowledge
  - in terms of textbook modeling knowledge
  - or provided by human expert

further work

- establishing libraries of
  - modeling (process-based) knowledge in different domains
  - existing models
- application/evaluation on other real-world tasks
  - better parameters fitting procedure
  - modeling population dynamics in lake Bled
- inducing domain-specific knowledge (grammars):
  - from data
  - from existing models
  - from textbooks and articles on modeling