Discovering the Structure of Partial Differential Equations from Example Behavior

Ljupčo Todorovski LJUPCO.TODOROVSKI@IJS.SI
Sašo Džeroski SASO.DZEROSKI@IJS.SI
Department of Intelligent Systems, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Ashwin Srinivasan ASHWIN.SRINIVASAN@COMLAB.OX.AC.UK
Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, England

Jonathan Whiteley JONATHAN.WHITELEY@NDA.OX.AC.UK
David Gavaghan DAVID.GAVAGHAN@COMLAB.OX.AC.UK
Oxford University, Nuffield Department of Anaesthetics and Computing Laboratory (address above)

Abstract

One of the most powerful and widely accepted analytical formalisms for modeling biological and physical systems is that of the partial differential equation (PDE). Establishing an acceptable PDE model for a dynamic system occupies a major portion of the work of the mathematical modeler. There are two main aspects to this activity. First, an appropriate structure has to be determined for the equations involved (the model identification problem). Second, acceptably accurate values for parameters are to be determined (the parameter estimation problem). Of these, the first is more challenging, and is the focus of this paper. We propose a method for discovering the structure of PDE models from example behavior. For simple PDE models, we illustrate that a straightforward adaptation of existing equation discovery methods is sufficient. However, complex PDE models require a more sophisticated approach: a two-stage method is proposed in the paper. The efficacy of the approach is demonstrated initially by rediscovering the PDE models for several artificial problems. We also use it to obtain the structure of the classic FitzHugh-Nagumo model. This represents a very wide class of biological systems, making the model discovery method of interest to scientists concerned with the enterprise of obtaining a mathematical understanding of dynamic processes occurring in the life sciences.

1. Introduction

The celebrated work of Alan Hodgkin and Andrew Huxley (Nobelstiftelsen, 1972; Hodgkin & Huxley, 1952) examined the conductance of sodium and potassium ions across the cell membrane when the homeostasis of the cell is disturbed by some external stimulus. This causes a brief reversal in a nerve cell's electrical polarization; this phenomenon, known as an "action potential" results in the transmission of an impulse along a nerve axon. The set of simultaneous partial differential equations developed by Hodgkin and Huxley to model this behavior embodies principles applicable also to the impulses in muscles. The work remains a milestone on the road towards the understanding of the nature of excitability, and laid the foundations of modern computational neuroscience. It is also an early demonstration of what has since become one of the most powerful and widely accepted analytical formalisms for modeling biological systems: the partial differential equation (PDE). PDEs are now used routinely to model: physiological transport processes such as gas exchange mechanisms and fluid flow in arteries; predator-prey behavior; the movement and growth of carcinogenic cells; viral infection in humans; animal coat patterns; fluid-flow in arteries; nerve-transmission, etc. (Murray, 1993). It is these kinds of models that are envisaged in the Physiome Project, which seeks to utilize data obtained from sequencing the human genome to provide "...a quantitative description of the physiological dynamics or functions of the intact organism" (see `http://www.physiome.org/`).

Obtaining a PDE model for experimental observations is not easy. Often, the physical processes involved are known at the outset. What is less clear is which of these are the most important. For example, in examining oxygen transport to red blood cells, the physical processes are convection, diffusion and chemical reaction. In fact, convection makes a negligible contribution and reaction is only important for sick lungs. Once it is known that only the diffusion term is important, parameters such as the diffusion coefficient can then by found relatively easily. Currently, there is very little by way of automated assistance for identifying the key physical processes. Model-construction proceeds on a trial-and-error basis: the scientist selects those believed to be important, constructs a model and sees if solutions match the observed data. If not, the procedure is repeated until an adequate model is found. This paper describes an automated technique, named PADLES[1], aimed at assisting this process. Given data, PADLES extracts the structures of PDE models that may prove adequate, i.e., yield good models for the data when instantiated with appropriate coefficients. It adapts existing grammar-based equation-learning methods to achieve this.

The paper is organized as follows. Background concepts of partial differential equations and equation discovery are briefly presented in Section 2. In Section 3, a straightforward extension of the existing equation discovery methods for discovering PDE structure is presented. Section 4 first presents a more sophisticated two-stage method for discovering the structure of complex PDE models. It then demonstrates the use of the method for extracting the structure of two PDE models. The last section summarizes the paper and gives some directions for further work.

2. Background

2.1 Partial Differential Equations

Differential equations are used to describe the behavior of dynamic systems, i.e., systems whose state changes over time. In ordinary differential equations (ODEs), time is the only dimension along which change of state is considered. Partial differential equations (PDEs) consider change of state along several dimensions, e.g., time and space.

Consider first a function f of one-variable (time) $f(t)$. The (ordinary) derivative of f with respect to t is defined as

$$\frac{df}{dt} = \lim_{h \to 0} \frac{f(t+h) - f(t)}{h}.$$

Consider a two-dimensional function $u = u(x,t)$, i.e., a function of two independent variables x (one-dimensional space) and t (time). The partial derivative of u with respect to x is defined as

$$\frac{\partial u}{\partial x} = \lim_{h \to 0} \frac{u(x+h,t) - u(x,t)}{h}$$

The functions $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial t}$ are the first-order partial derivatives of u. The second-order ones are $\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}(\frac{\partial u}{\partial x})$, $\frac{\partial^2 u}{\partial t^2}$, and $\frac{\partial^2 u}{\partial x \partial t} = \frac{\partial}{\partial x}(\frac{\partial u}{\partial t}) = \frac{\partial}{\partial t}(\frac{\partial u}{\partial x})$.

A partial differential equation (PDE) is an equation involving one or more partial derivatives of an unknown function of several variables. The order of a PDE is the order of the highest-order derivative that appears in the equation. An example of a first-order PDE is the non-linear first-order wave equation $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$ and an example of a second-order PDE is the second-order wave equation $\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0$.

Given a model of a dynamic system in the form of an ODE, the behavior of the system can be simulated/derived by solving the ODE. To solve an ODE, an initial state has to be provided. A general numerical integration method (such as Runge-Kutta integration) can then be applied. For PDEs, the situation is more complicated. Boundary conditions, typically more complex than just specifying an initial state, need to be provided. A similar range of choices is available for performing the time integration as for ODEs, while the spatial derivatives might typically be handled using either finite difference or finite element methods. In either case, a suitable spatial mesh has to be generated, with, in general, a finer mesh giving a smaller numerical error of simulation but requiring a larger computational effort. Many PDE problems, including the FitzHugh-Nagumo model considered in this work, are also non-linear and may be very sensitive to slight changes in initial conditions, or display very different behavior for slight variations in equation parameters. Where experimental systems display such complex behavior it can be very difficult to determine the appropriate form of the equations and may require lengthy and painstaking observational work in the laboratory, as was the case for Hodgkin and Huxley.

2.2 Discovery of Differential Equations

Equation discovery systems help human experts to discover natural laws, expressed in the form of equations, in collections of observed data. Early approaches to

[1] The acronym PADLES stands for "PArtial DifferentiaL Equations Structure" discovery assistant.

equation discovery, such as Bacon (Langley et al., 1987), were concerned with rediscovering empirical laws from the history of different scientific disciplines, expressed in the form of algebraic equations. Other systems for discovery of algebraic equations, such as Fahrenheit (Żytkow & Zhu, 1991) and Abacus (Falkenheiner & Michalski, 1990) followed. In general, equation discovery systems search a space of possible equations to find equations that fit the observed data well. A recent addition to the family of equation discovery system, SDS (Washio & Motoda, 1997), reduces the space of algebraic equations searched by making use of user provided dimensional (scale-type) information about the system variables.

Lagrange (Džeroski & Todorovski, 1995) extended the scope of equation discovery to ordinary differential equations and thus modeling of dynamic systems. The basic idea was to introduce the time derivatives of the systems variables through numerical differentiation and then search for algebraic equations. This simple approach had two major drawbacks: large errors were introduced by numerical differentiation and the space of equations considered was too large.

Its successor Lagramge (Todorovski, 1998; Todorovski & Džeroski, 1997) allows the user to explicitly define the space of possible equations using a context free grammar. This is a more general and powerful formalism for reducing the space of possible equations than the one used in SDS (Washio & Motoda, 1997). The problem of large errors introduced by numerical differentiation is also addressed: numerical integration is used instead of differentiation for the highest-order derivatives. Lagramge can also use different search methods (exhaustive and beam search) as well as heuristics for more efficient search. The SSE (sum of squared errors) heuristic is equal to the sum of squared differences between the measured values and the values obtained by the discovered equation. The MDL (minimal description length) heuristics adds a penalty for equation complexity (expressed in number of terms) to the sum of squared errors.

Lagrange and Lagramge are strongly related to system identification methods, used for building models of dynamic systems based on measured data. However, mainstream system identification methods work under the assumption that the structure of the model, i.e., the form of the equations, is known and are concerned with determining the values of the constant parameters in the model (Ljung, 1993). Equation discovery systems, on the other hand, do not assume a single prescribed model structure, but rather explore a space of possible equation structures. They aim at identifying both an appropriate structure of the equations and appropriate values of the parameters.

3. A First Approach to PDE Structure Discovery

3.1 The Approach

Our first approach to re-discovering models in the form of PDEs mirrors the one taken in Lagrange (Džeroski & Todorovski, 1995). It is summarized in Table 1. We first calculate (numerically) the partial derivatives with respect to the dimensional variables and thus extend the original data set. We then attempt to discover algebraic equations on the extended data set by calling the procedure FindAlgEqn. At present, we employ Lagramge (Todorovski, 1998; Todorovski & Džeroski, 1997) for this purpose. In this simple approach to PDE structure discovery, other equation discovery systems, such as SDS (Washio & Motoda, 1997), can be also used. These would disregard the grammar G.

The input to PDE-SD is a set of measurements along each of the system variables/functions ($\mathcal{F}$) and dimension variables ($\mathcal{D}$). For the first-order wave equation $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$, $\mathcal{F} = \{u\}$ and $\mathcal{D} = \{t, x\}$. In addition, a grammar is provided which specifies the form of algebraic equations to be considered. A grammar specifying polynomials was used for the experiments described in this section.

Table 1. A simple algorithm for discovering the structure of partial differential equations.

```
procedure PDE-SD($\mathcal{V} = \mathcal{F} \cup \mathcal{D}$, $G$)
  $\mathcal{P}$ = set of numerically calculated
    partial derivatives of variables in $\mathcal{F}$
    with respect to dimensions in $\mathcal{D}$
  foreach function/variable $V \in \mathcal{F}$ do
    FindAlgEqn($\mathcal{V} \cup \mathcal{P}, G, V$)
  endfor
endprocedure
```

Numerical calculation of the partial derivatives is performed by first constructing a multi-dimensional polynomial for each function in $\mathcal{F}$ in terms of the variables in $\mathcal{D}$, then calculating the derivatives of the polynomial along each of the dimensions. Polynomials of degree 5 were used in our experiments.

3.2 Experiments

3.2.1 Structure of Equations

The focus of the paper is on reconstructing the *structure* of PDEs, rather than the values of the constant parameters appearing in the equations. In order to evaluate the discovered equations, we extract what we term the "structure of the equation" (defined below) and match it against the structure of the original equations. While we use this criterion in the exploratory experiments described here, it cannot be used in real-world domains, where the structure of the model equations is not known in advance. This criterion should be replaced with more objective ones, based on simulating the discovered equations and matching the simulation against the training and/or testing data. To this end, standard parameter estimation techniques used for PDEs should be included within the proposed methods, as opposed to the crude ones currently used in LAGRAMGE.

We obtain the structure of an equation by (1) rewriting it in a canonical form such that the right hand side is 0; and (2) abstracting the constant parameters in the left hand side to generic constants. Thus, the structure of the equation $\partial u/\partial t = -0.657043 \times u\ \partial u/\partial x$ is $\partial u/\partial t + c_1 \times u\ \partial u/\partial x = 0$. We do not explore here a logical semantics for this generalization, except to note that two equations will be said to have the same structure if there is a trivial rewrite of the abstracted coefficients that makes the structures identical. For example, $\partial u/\partial t = -0.657043 \times u\ \partial u/\partial x$ and $\partial u/\partial t = u\ \partial u/\partial x$ have the same structure.

3.2.2 Re-discovery of Simple PDEs

PDE-SD successfully recovers the structure of textbook equations, e.g., the non-linear first-order wave equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

and the second-order wave equation

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0$$

from simulated data.[2]

3.2.3 Predator-Prey (PP) Model

We next test PDE-SD on a slightly more complex task. The predator-prey model describes situations such as the population of rabbits and foxes on an island, where foxes prey on rabbits and rabbits have an unlimited supply of food. Variable u is the dimensionless population of the prey, v is the dimensionless population of the predator. This model allows for spatial variations so that the predators have to move to catch the prey, and the prey move to evade the predator:

$$\begin{aligned}\frac{\partial u}{\partial t} &= u(1-u-v) + 0.1\frac{\partial^2 u}{\partial x^2}\\ \frac{\partial v}{\partial t} &= v(u-0.02) + 0.1\frac{\partial^2 v}{\partial x^2}.\end{aligned}$$

The training data set was generated with a simple simulation method for PDEs, using numerical approximations of the partial derivatives. The simulation step size in the t-direction is 10^{-5} and the step size in the x-direction is 0.5. The small time step is needed for stability of the numerical approximation to the PDE. The numerical solutions for u and v were then saved at 201 values of x equally spaced between -50 and 50, and 35 values of t, equally spaced between 0 and 34, giving a total of $201 \times 35 = 7035$ data points. No external noise was added to the numerical solution. However, the training data set is not completely noise free, due to the numerical error of the simulation method and saving the simulation results at coarse mesh.

In experiments with LAGRAMGE, both heuristic functions, SSE and MDL, were used in combination with beam search (width 25) through the space of multivariate polynomial equations. In both cases, the structure of the 10 best equations differs from the structure of the original PP equations. Exhaustive search could not be used in this domain, because of the search space size (7.5×10^{11} equation structures, see Table 7).

PDE-SD thus fails to recover the correct structure of the predator-prey equations. To understand better why, let us consider several dimensions of difficulty of the PDE discovery task.

3.3 On the Complexity of PDE Discovery

The PDE discovery problem becomes more difficult if:

1. we have sparser measurements (coarser mesh),
2. higher-order derivatives are involved, and
3. the degree (of non-linearity) of terms in the equations is higher.

[2] The actual forms in which the equations were rediscovered are

$$\frac{\partial u}{\partial t} = -0.657043\, u\, \frac{\partial u}{\partial x}$$

and

$$\frac{\partial^2 u}{\partial t^2} = -1.05818 \times 10^{-5} + 1.00098\,\frac{\partial^2 u}{\partial x^2}$$

The error in the coefficient for the first-order wave equation is large. In general, numerical calculation of partial derivatives introduces large errors, just like numerical calculation of ordinary derivatives, especially if the measurements are sparse (taken on a coarse mesh).

Items 1 and 2 are related to numerical differentiation errors, while items 2 and 3 are related to the size of the space of possible models. The coarser the mesh and the higher the derivatives order, the larger the errors. The higher the derivatives order and the degree of terms, the larger the space of possible equations.

Two difficulties arise when we have a large space of possible equations: (1) it takes a lot of time to search this space; and (2) it is more difficult to select the appropriate equation structure. Given the same data, the more models we consider, the more likely we are to find models that fit the data by chance rather than true regularities. The first difficulty is addressed, and can be partly overcome, using different search strategies (e.g., beam search). The second difficulty can be overcome by restricting the space of possible equations.

In the case of the PP model, the size of the space of polynomial equations that contains the target equations is of the order 10^{11} (see Table 7); we can identify this as the reason for the failure of our simple approach. Greedy (beam) search considers only a fraction of the equations, but misses the original equation structure. We therefore need to restrict the space of possible models/equations. In the following section, we propose a two-stage method for PDE structure discovery: we learn how to restrict the space of possible equations in the first stage, and search the restricted space in the second stage.

4. Two-Stage Discovery of Complex PDE Models

4.1 Padles

A two-stage PDE structure discovery method is presented in Table 2. In the first stage (lines 1-9), the problem of discovering PDEs is simplified to the problem of discovering ODEs. In the second stage (lines 10-14), the most common of the discovered ODE structures are used to define a grammar and thus reduce the space of PDEs which is then searched.

In the first stage, the idea is to take slices of the training data for fixed values of all but one of the dimensional variables (time) and search for ODEs in these slices. E.g., for the second PP equation for the function $v(t,x)$, we take a fixed point in space $x = x_0$ and observe the behavior of the function $v_0(t) = v(t, x_0)$ over time. In the slice of the data set where $x = x_0$, we try to discover an ODE describing $v_0(t)$. We repeat the process for all different points in space (x_i) and obtain a set of ODEs which approximate the structure of the original PDE ($\partial v/\partial t + c_1 \times uv + c_2 \times v + c_3 \times \partial^2 v/\partial x^2 = 0$). We expect that the most common ODE structures in this set will be very close to the structure of the original PDE. For the second PP equation, the most common ODE structure is $\partial v/\partial t + c_1 \times uv + c_2 \times v = 0$, which is equivalent to the original PDE structure if we disregard the partial derivative $\partial^2 v/\partial x^2$.

Table 2. Two-stage algorithm for discovering the structure of partial differential equations.

```
     procedure PADLES($\mathcal{V} = \mathcal{F} \cup \mathcal{D}$, $G$)
 1     foreach tuple of values $x$ of $\mathcal{D} - \{t\}$ do
 2       $\mathcal{F}_x = \{s \in \mathcal{F} : \mathrm{values}(s, \mathcal{D} - \{t\}) = x\}$
 3     endfor
 4     $\mathcal{P}$ = set of numerically calculated
         partial derivatives of variables in $\mathcal{F}$
         with respect to dimensions in $\mathcal{D}$
 5     foreach variable $V \in \mathcal{F}$ do
 6       foreach $\mathcal{F}_x$ do
 7         $\mathcal{E}_{V,x}$ = the set of 20 best equations
             from FINDODE($\mathcal{F}_x \cup \{t\}, G, V$)
 8       endfor
 9     endfor
10     foreach variable $V \in \mathcal{F}$ do
11       Let $\mathcal{S}_V$ be the set of most-frequent
           equation structures in $\cup_x \mathcal{E}_{V,x}$
12       Let $G_V$ be the grammar built upon $\mathcal{S}_V$
13       FINDALGEQN($\mathcal{V} \cup \mathcal{P}, G_V, \partial V/\partial t$)
14     endfor
     endprocedure
```

In the first step of the algorithm (lines 1-3 in Table 2), the data set $\mathcal{F}$ is partitioned into slices $\mathcal{F}_x$ with constant values of all dimensions in $\mathcal{D}$ except time. For each system variable in $\mathcal{V}$, ODEs are discovered in each data slice $\mathcal{F}_x$ (lines 6-9). Again, we use LAGRAMGE to discover ODEs: note that for discovering first-order ODEs LAGRAMGE does not use numerical calculation of the time derivatives of system variables (Todorovski, 1998). If we calculate the time derivatives numerically, an arbitrary equation discovery system can be used in the first stage.

The 20 best ODEs (10 best according to the SSE heuristic and 10 best according to the MDL heuristic) are kept ($\mathcal{E}_{V,x}$) for each data slice. Abstracting the values of the constant parameters in the discovered equations (see Section 3.2.1) the structures of the equations are extracted. All structures, obtained on different data slices, are merged and their frequencies are calculated. Only the most frequent ones are used in the grammar that determines the PDE structures to be considered in the second stage (line 11). In particular, the equation structures with frequencies within the interval $[f_{max}/2, f_{max}]$ (f_{max} being the frequency of the most frequent structure) constitute the set $\mathcal{S}_V$.

Table 3. An example G_V grammar.

E	$\rightarrow$	$const * V_p + S \mid S$
V_p	$\rightarrow$	$\partial u/\partial x \mid \partial^2 u/\partial t^2 \mid \partial^2 u/\partial x \partial t \mid \partial^2 u/\partial x^2$
V_p	$\rightarrow$	$\partial v/\partial x \mid \partial^2 v/\partial t^2 \mid \partial^2 v/\partial x \partial t \mid \partial^2 v/\partial x^2$
S	$\rightarrow$	$const * u * v + const * v \mid const * u * v \mid \ldots$

In the second stage, a grammar G_V is formed using the equation structures from $\mathcal{S}_V$ (line 12). It has three nonterminal symbols: the start symbol E, defining the restricted space of PDEs; S, reflecting the most common ODE structures, discovered in the first-stage experiments; and V_p, used to denote partial derivatives with respect to at least one variable other than time. An example grammar G_V used to discover the second PP equation is presented in Table 3. On the right hand sides of the productions for the non-terminal symbol S are the equation structures from the set $\mathcal{S}_V$ (only the two most common ones given in Table 3). The grammar can be easily extended to include an arbitrary number of dimensional and system variables with adding new productions to the nonterminal symbol V_p.

The class of PDEs defined by the grammar G_V consists of equations in which the partial derivatives with respect to at least one dimension other than time are linearly coupled with other (non-linear) terms. Note that this restriction on the class of discovered PDEs is entirely due to the grammar, and can be lifted by using alternative productions for the symbol E.

Finally, we use an equation discovery method to discover algebraic equations on the extended data set (which includes the partial derivatives) within the restricted equation space defined by the grammar G_V (line 13). Again, LAGRAMGE is the actual engine behind the call FINDALGEQN. Note that in the second stage, we have to use an equation discovery system that is able to consider different spaces of possible equations, specified by user defined grammars.

4.2 Experiments

4.2.1 PREDATOR-PREY (PP) MODEL

Here we apply the procedure PADLES to the example behavior generated by simulating the predator-prey model described in Section 3.2.3. Disregarding the partial derivatives, the structure for the first equation is $\partial u/\partial t + c_1 \times u + c_2 \times uv + c_3 \times u^2 = 0$ and the structure for the second equation is $\partial v/\partial t + c_1 \times v + c_2 \times uv = 0$. We will refer to these as the true ODE structures in the tables below. The true PDE structure takes into account the partial derivatives $\partial^2 u/\partial x^2$ and $\partial^2 v/\partial x^2$. In the first-stage experiments, the true ODE structures for both PP equations were ranked best: they were the most frequent among the ODE structures found.

Table 4. Ranks of the true PDE structures of the Predator-Prey equations after the second-stage experiments.

	EQUATION FOR	
HEURISTIC	$\partial u/\partial t$	$\partial v/\partial t$
MDL	2	1
SSE	1	8

The results of the second stage experiments are given in Table 4. The ranks listed here are the ranks of the true PDE structure among the 10 best equations returned by LAGRAMGE according to the error heuristic selected. According to the MDL heuristic, the true PDE structure of the equation for $\partial v/\partial t$ is ranked best, while the true PDE structure of the equation for $\partial u/\partial t$ is ranked second best. We can thus say that the structure of both PDEs was successfully rediscovered.

Having successfully discovered a moderately complex PDE model using the two-stage approach, we now turn to the task of discovering a more complex and practically relevant PDE model.

4.2.2 FITZHUGH-NAGUMO (FHN) MODEL

FitzHugh (1961) and Nagumo et al. (1962) independently derived simplified versions of the Hodgkin-Huxley equations which retain the most important biological features. The form of the FHN equations is

$$\begin{aligned} \frac{\partial v}{\partial t} &= \frac{\partial^2 v}{\partial x^2} + v(v-a)(1-v) - w \\ \frac{\partial w}{\partial t} &= b(v - dw) \end{aligned}$$

where a, b, and d, are constant parameters, and v and w are functions of time t and distance x. For a given initial condition (say, a narrow Gaussian pulse), this system might display any one of three types of behavior: simple decay; a single traveling wave solution; or multiple traveling wave solutions, determined by the values of the three parameters. These might correspond, respectively, to a nerve stimulus being inadequate to initiate axon firing; a nerve stimulus being sufficient to initiate a single nerve impulse; and repeated nerve firing such as occurs in the sinus node in the heart.

Table 5. Ranks of the true ODE structures of the FHN equations after the first-stage experiments.

	DATA SET			
EQUATION FOR	1	2	3	1+2+3
$\partial v/\partial t$	1	7	6	1
$\partial w/\partial t$	1	1	1	1

Three behavior traces (data sets) were generated using numerical simulation of the model. The first data set was generated using the following values of the constant parameters: $a = -0.02$, $b = 0.005$ and $d = 3$. The other two data sets were generated using two different initial conditions and the following values for the constant parameters: $a = 0.02$, $b = 0.005$ and $d = 3$. The equations were simulated using the same method as in the PP experiments. The numerical solutions for v and w were then saved at 201 values of x equally spaced between -100 and 100, and 13 values of t, equally spaced between 0 and 120, giving a total of $201 \times 13 = 2613$ data points in each data set. No external noise was added to the numerical solution.

Disregarding the partial derivative, the structure of the first FHN equation is $\partial v/\partial t + c_1 \times w + c_2 \times v + c_3 \times v^2 + c_4 \times v^3 = 0$. The structure of the second FHN equation is $\partial w/\partial t + c_1 \times w + c_2 \times v = 0$. We will refer to these as the true ODE structures of the FHN equations in the tables below. Since all three data sets were obtained using the same model structure, we can use the union of equation structures obtained on each of the three data sets for calculating ODE structure frequencies (row 1+2+3 in the tables) in the first-stage experiments.

The results of the first-stage experiments are presented in Table 5. The figures in this table represent the ranks of the true ODE structure among all ODE structures discovered in the first stage. Taking the union of equations, discovered for all three training data sets, the true ODE structures of both equations are the most-frequent ones. From the table, we can conclude that if we have more than one training data set for the domain, it is more robust to merge the ODE structures discovered on different training data sets.

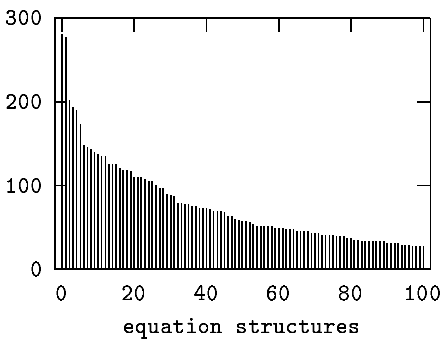

Figure 1. The frequencies of the 100 most-frequent equation structures after the first-stage experiments for the first FitzHugh-Nagumo equation.

The frequencies of the 100 most-frequent structures after the first-stage experiments for the first FitzHugh-Nagumo equation are depicted in Figure 1. It can be seen from the figure that equation structures' frequency decays exponentially. The most frequent structure was discovered 281 times in the first-stage experiments, and there are nine equation structures with frequencies within the interval $[281, 281/2 = 141.5]$, used in the second-stage experiments.

Table 6. Ranks of the true PDE structure of the FitzHugh-Nagumo equations the second-stage experiments.

		Data Set		
	Heuristic	1	2	3
$\partial v/\partial t$	MDL	*5	*5	*2
	SSE	6	*5	1
$\partial w/\partial t$	MDL	1	1	1
	SSE	*	*	*

The summary of the second-stage experiments is presented in Table 6. The figures in the table represent the rank of the true PDE equation structure (including the partial derivatives). The *N means that the true PDE structure was not among the 10 best as evaluated by the respective error heuristic (MDL or SSE), but an otherwise identical structure with one missing term had rank N. The true PDE structure of the first equation is discovered in the experiments with the third (and first) data set using the SSE heuristic. The second PDE structure is discovered for all data sets using the MDL heuristic.

In summary, Padles successfully discovers the structure of a complex and practically important PDE model.

Table 7. Reduction of the equation space complexity.

	Number of Equation Structures	
Model	PDE-SD	Padles
PP	7.5049×10^{11}	161
FHN	3.01872×10^{17}	189

To understand why Padles works, consider the reduction of the PDE space from PDE-SD to Padles, illustrated in Table 7. Column one gives the number of PDEs that need to be considered by the one-stage approach using a polynomial grammar, while column two gives the actual number of PDEs considered in the second stage of Padles. In both domains, the latter approach considers less than 200 PDEs and the PDE space is reduced by a factor of over 10^9.

5. Concluding Remarks

This paper describes the first steps towards the discovery of an important class of PDEs, mathematical models that has very wide ranging applications. While the use of PDE models abound in the physical sciences and engineering, our motivation has been predominantly the modeling of biological systems. The need for quantitative models for biological processes is growing rapidly, and we expect it to play a significant role in establishing the kind of mathematical understanding sought from enterprises like the Human Physiome Project. It is our belief that an automated model discovery method of the form proposed here will greatly assist the analysis of the large quantities of data expected to be available for the project.

In this paper, we have presented evidence that "easy" PDE structures can be extracted using a straightforward adaptation of an existing methods for finding equations. It also appears feasible to extract more complex PDE models using a two-stage procedure that first obtains ordinary differential equations in order to restrict the space of possible PDEs.

Much work needs to be done to make the method acceptable to mathematical biologists. In the short term, this takes the form of further experiments with more models and with truly observational data. The simulated data used here introduces some errors (due to the numerical error of the PDE simulation method), which are of a different nature to experimental data. We would need to establish that the method works robustly under both conditions. In the longer term, the technique needs to be properly integrated into the data analysis and simulation environment in routine use in the field. This will enable standard techniques for parameter estimation and sensitivity analysis to be used in conjunction with model-discovery to yield a proper scientific assistant.

The work presented in the paper is concerned with assisting the empirical approach to scientific discovery, where different models are constructed on the trial-and-error basis to fit observed data. This is in contrast with the theoretical approach, where the primitive physical processes involved in the modeled system are first identified, the model being a deductive consequence of the basic principles about these processes. Allowing the definition of the space of possible equations in equation discovery systems is a step towards incorporating theoretical knowledge into the empirical approach. Still, much work remains to be done to bring equation discovery systems closer to the theoretical approach to scientific discovery.

Acknowledgements

A. Srinivasan is supported by a Nuffield Trust Research Fellowship at Green College, Oxford. J. Whiteley is supported by the Engineering and Physical Sciences Research Council, UK. D. Gavaghan is supported by a Research Fellowship of the Medical Research Council, UK.

References

Džeroski, S., & Todorovski, L. (1995). Discovering dynamics: From inductive logic programming to machine discovery. *Journal of Intelligent Information Systems*, *4*, 89–108.

Falkenheiner, B., & Michalski, R. (1990). Integrating quantitative and qualitative discovery in the ABACUS system. In Y. Kodratoff and R. Michalski (Eds.), *Machine learning: An artificial intelligence approach*, 153–190. San Mateo, CA: Morgan Kaufmann.

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, *1*, 445–466.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, *117*, 500–544.

Langley, P., Simon, H., & Bradshaw, G. (1987). Heuristics for empirical discovery. In L. Bolc (Ed.), *Computational models of learning*, 21–54. Berlin: Springer.

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

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

Nagumo, A. S., Arimoto, S., & Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the Institute of Radio Engineers*, *50*, 2061–2071.

Nobelstiftelsen (1972). *Nobel lectures: Physiology or medicine 1963-1970.* Amsterdam: Elsevier.

Todorovski, L. (1998). *Declarative bias in equation discovery.* MSc Thesis, Faculty of computer and information science, University of Ljubljana, Ljubljana, Slovenia.

Todorovski, L., & Džeroski, S. (1997). Declarative bias in equation discovery. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 376–384). San Mateo, CA: Morgan Kaufmann.

Washio, T., & Motoda, H. (1997). Discovering admissible models of complex systems based on scale-types and identity constraints. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (pp. 810–817). San Mateo, CA: Morgan Kaufmann.

Żytkow, J., & Zhu, J. (1991). Application of empirical discovery in knowledge acquisition. *Proceeding of Fifth European Working Session on Learning* (pp. 101–117). Berlin: Springer.