2014

Induction Of Governing Differential Equations From Hydrologic Time Series Data Using Genetic Programming

Jayashree Chadalawada

Vladan Babovic

INDUCTION OF GOVERNING DIFFERENTIAL EQUATIONS FROM HYDROLOGIC TIME SERIES DATA USING GENETIC PROGRAMMING

JAYASHREE CHADALAWADA (1), VLADAN BABOVIC (2)
(1): Department of Civil and Environmental Engineering, National University of Singapore, EW1-02-05, 2 Engineering Drive 2, 117577, Singapore
(2): NUS Deltares, Block E1-08-24, No: 1 Engineering Drive, 117576, Singapore

This contribution describes an evolutionary method for identifying causal model from the observed time-series data. In the present case, we use a system of Ordinary Differential Equations (ODEs) as the causal model. Usefulness of the approach is demonstrated on real-world time series of hydrologic processes and the unknown function of governing factors are determined. To explore the evolutionary search space more effectively, the right hand sides of ODEs are inferred by genetic programming (GP). The importance of different fitness criteria, as well as introduction of background knowledge about underlying processes are also being discussed and assessed. The method is being applied to real world datasets in order to empirically demonstrate how successfully GP infers the systems of ODEs.

INTRODUCTION

Hydrologic phenomena depend on several governing parameters and a single independent variable, time. A system of differential equations can be used to describe such phenomena. The nonlinear relationship between the parameters makes it impossible to solve the system of differential equations analytically. Therefore, our goal is to design a heuristic approach to estimate the coefficients and function of the parameters without determining the exact solution to the nonlinear system.

LOTKA-VOLTERRA MODEL

The data analyzed in the present contribution are synthesized using Lotka-Volterra Model Lotka[1] and Volterra[2] which consists of a pair of first order nonlinear differential equations known as the predator-prey equations that describe the dynamics of biological systems in which the two species one as predator and the other as prey interact.

$$\frac{dx}{dt} = x(A(1) - A(2)y)$$
$$\frac{dy}{dt} = -y(A(3) - A(4)x) \quad (1)$$

In Eq (1), x is the number of prey, y is the number of predators and A (1), A (2), A (3) and A (4) are the parameters describing the interaction of the two species. The assumptions of Lotka-Volterra model are (i) The prey population has an unlimited food supply, (ii) Rate of predation is proportional to the rate at

which the predators and prey meet and (iii) The time spent by the predator consuming the prey is negligible.

There are many heuristics that can provide approximate solutions to the optimization problems. Out of which, Nedler-Mead simplical heuristic (local search technique) and Genetic Programming (Global search technique) are chosen for the present study.

Nedler-Mead direct search also called a simplex search algorithm is a popular heuristic for approximate minimization which requires only function evaluations and not derivatives. It has been reported in the earlier studies Abebe [3] that this algorithm provides a rapid reduction in function values and terminates with bounded level sets that contain possible minimum points. This algorithm has been implemented in the current study using "fminserach. m" Matlab function. The efficiency of Nedler-Mead simplical heuristic in estimating the parameters on the RHS of Eq(1) is tested using the synthesized dataset.

As a next step, the contribution of evolutionary algorithms that are based on the concepts of inheritance, variability and selection, is to be analyzed. In Genetic Programming paradigm, the evolutionary force is directed to produce a closed form mathematical expression describing the available data Babovic *et al.* [4]. The variation operators used in Genetic Programming include subtree mutation and subtree crossover. The functions within the tree structure include arithmetic and logical operators. Earlier studies focused on symbolic regression and evolution of algebraic expressions. Therefore, in this study, an attempt is being made to describe the time series data using models that take the form of differential equations by means of Genetic Programming.

DATA GENERATION

The two equations that constitute the Lotka - Volterra model form a system of ordinary differential equations. Given the initial conditions, these differential equations can be solved over the time domain of interest (Initial Value Problem). In order to solve this Initial Value Problem (IVP), the following numerical methods Mathews *et al.* [5] are used with a step size of $h= 10^{-1}$.

(i) Euler Explicit method
(ii) Heuns' method
(iii) Runge-Kutta fourth order method (RK4)
(iv) Predictor-Corrector methods, namely Adam-Bashforth Moulton, Milne Simpsons' and Hammings' methods

Every numerical method attempts to generate the solution by starting with the initial condition, approximating the value at the next step, and then continuing that process for several iterations Averill [6]. They are all based on the Taylor Series expansion. To analyze the accuracy of the approximations according to the above mentioned methods and due to the difficulty in arriving at the exact solution of Lotka-Volterra Model, the solution using RK4 with a step size of $h=10^{-4}$ is regarded as the exact solution of the model.

Numerical Models for Lotka Volterra Equation

For Lotka Volterra Equation (1), the following parameters are chosen to solve this IVP over the time domain of 100 time steps. Initial population densities are chosen for prey as x_0=35 and predator as y_0=4 and parameters are set as A (1) = 0. 5, A (2) = 0.02, A (3) = 0.8 and A (4) =0.02. Data is generated using RK4 with a step size of $h=10^{-4}$ as shown in Figure 1.

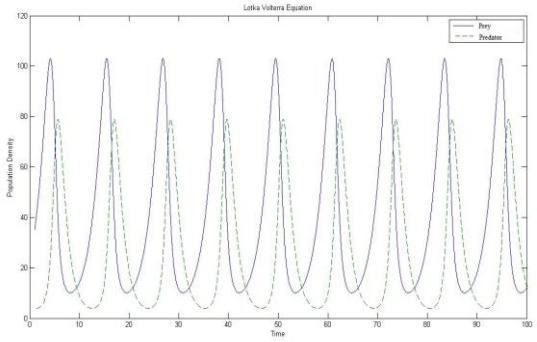

Figure 1. RK4 solution to Lotka Volterra Equation with x_0=35, y_0=4 and parameters A (1) = 0. 5, A (2) = 0.02, A (3) = 0.8 and A (4) =0.02.

The parameters are estimated using Nedler-Mead direct search method. The estimates are validated by checking each model against the generated data and the results are shown in the Table 1. As Nedler Mead simplex search is a local search method, its performance is highly sensitive to the initial guess values assigned to the four parameters. The guess values of the parameters are estimated by central difference derivative approximation of second order accuracy with step size h=1, without finding the exact solution to the system of equations.

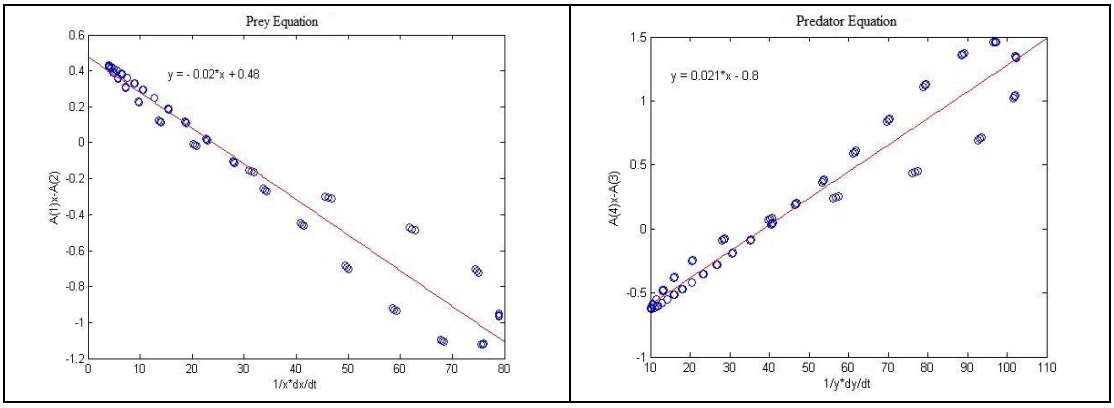

Figure 2a and 2b. Initial Guess for predator and prey equations. Therefore, A (1) = 0.48, A (2) = 0.02, A (3) = 0.8 and A(4) = 0.021

Table 1. Validation of Numerical Models

Sno.	Numerical Method (Model)	Order of Accuracy	Stepsize (h)	Initial values		Parameters				Remarks
				x_0	y_0	A(1)	A(2)	A(3)	A(4)	
	RK-4	5	$10^{(-4)}$	35	4	0.5	0.02	0.8	0.02	Exact Solution
1	Euler Explicit	2	$10^{(-1)}$	35	4	0.4935	0.0198	0.8115	0.0202	Not a good approximation
2	Heuns'	3	$10^{(-1)}$	35	4	0.4999	0.02	0.7999	0.02	Close to the best
3	RK-4	5	$10^{(-1)}$	35	4	0.5	0.02	0.8	0.02	Best
4	Adam Bashforth Moulton	5	$10^{(-1)}$	35	4	0.5	0.02	0.79	0.02	Best
5	Milne Simpsons	5	$10^{(-1)}$	35	4	0.5	0.02	0.8	0.02	Best
6	Hammings	5	$10^{(-1)}$	35	4	0.5	0.02	0.8	0.02	Best

Out of the four best numerical approximations listed in Table 1, one method is selected and subjected to further analysis. In order to make a choice, a small time domain of ten steps, step size of h=0.1 are chosen and data is generated based on each of the four numerical models. The maximum error between the numerical methods and the exact solution is found in the case of prey and predators. The results are shown in the Table 2. RK4 is found to be the closest to the exact solution.

Table 2. Maximum error calculations for the four numerical approximation methods for h=0.1

Method	Prey	Predator
RK4	5.03E-07	5.07E-07
Hammings	1.12E-06	1.47E-06
Milne Simpsons	1.57E-06	1.65E-06
Adam Bashforth Moulton	9.83E-06	7.25E-06

Therefore, in the subsequent sections, the performance of the RK4 numerical model with constant step size h=0. 1 is tested under two conditions:

(i) Solving the system of differential equations by discretization of the time domain of 100 steps into subdomains of 5 steps each.

(ii) Adding different levels of Gaussian white noise to the data generated over time domain of 100, 50, 25 and 10 steps.

DISCRETIZATION

In this section, the data generation using RK4 numerical approximation with step size h=0. 1 is not carried out at once over the entire time domain of 100 steps. Instead, the time domain is split into subdomains of 5 steps each. This means that for each subdomain, new initial conditions are defined and the solution is generated only for 5 subsequent time steps based on those initial conditions. This exercise is carried to verify if the small forecast domain and continuous update process, i.e. updating as and when an observation is recorded in the field, contribute to the improvement in the accuracy of the numerical solution. The data generated with and without discretization are compared with that of exact solution and the results are shown in the Table 3. The results confirm the improvement in accuracy in the case of a discretized time domain.

Table 3. Improvement in accuracy by dicretization of the time domain of 100 steps

Method	Stepsize (h)	Time Domain Size	Number of Subdomains	Prey	Predator
RK4	$10^{(-1)}$	100	0	8.87E-06	1.12E-05
RK4	$10^{(-1)}$	100	20	1.97E-06	3.31E-06

NOISE SENSITIVITY ANALYSIS

Data is generated using Eq 1 by means of RK4 numerical approximation with a constant step size h=0.1 over time domain of 100, 50, 25 and 10 steps, adding Gaussian white noise with different standard deviation values say, 0.05, 0.25, 0.5, 1, 2, 3, 5, 10, 15, 20 and 25,

(i) Only to the initial values x_0=35 and y_0=4.
(ii) Both to the initial values and the values generated at every time step.

The parameters of the equations are estimated using Nedler-Mead direct search method for each of the above mentioned settings. The estimates are compared with that of the exact solution and the results are shown in the Table 4 and 5 for cases (i) and (ii) respectively.

In both the cases (i) and (ii), the error in prediction increases with the increase in standard deviation (sigma). As expected, the deviation from the original parameters is the highest for case (ii) in comparison with case (i) for the same sigma value and time domain size. As the size of the time domain decreases, the tolerance to Gaussian white noise increases. Therefore, prediction results over the time domain of 10 steps is better than that of 25 steps which is better than 50 steps which in turn is better than that of 100 steps.

For example, as per Table 4, in case (i), with sigma=10, the overall percentage deviation from the original parameters is 8% and 4% in case of 100 and 10 steps time domains respectively. As given in Table 5, in case(ii), the sum of squared errors between the exact solution and noisy data with a sigma value of 5 are 51163.40 and 5523.51 for 100 and 10 steps time domains respectively. Figure 3a and 3b show the

predicted results of the case (ii) scenario in which Gaussian white noise with sigma value 5 is incorporated in the data generated over a time domain of 100 steps and 10 steps respectively. The noise affects the 100 steps time domain more than the 10 steps time domain.

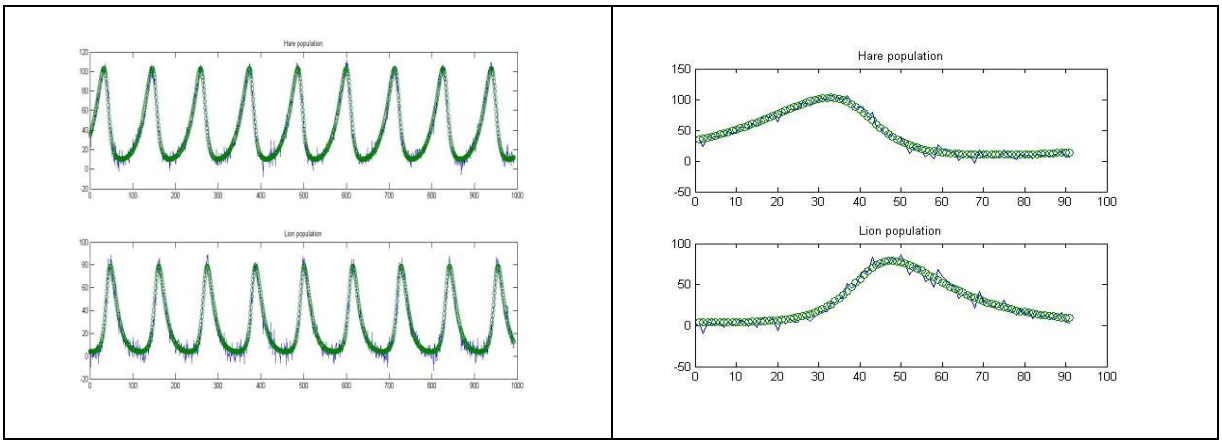

Figure 3a and 3b. The parameter estimation result of the case (II) scenario with noise of sigma=5 over time domain of 100 and 10 steps respectively.

SUMMARY AND WORK IN PROGRESS

In the present study, the performance of the numerical models and parameter estimation through direct search optimization technique in different scenarios say, in time domains of varying sizes, with and without discretization and in the absence and presence of varying levels of Gaussian white noise have been discussed in detail. In the subsequent study, it is intended to employ a global search technique based on Genetic Programming to evolve differential equations describing the given data.
The data used in the analysis so far has been synthesized from the Lotka - Volterra model. In the future, the technique is to be tested using real-world, hydrologic time series data set.

REFERENCES

[1] Lotka A. J., "*Elements of physical biology*", Williams and Wilkins, Baltimore, (1925).

[2] Volterra V., "*Animal ecology*", McGraw-Hill, New York, (1926), pp 409-448.

[3] Geletu A., "*Solving Optimization Problems using the Matlab Optimization Toolbox - A Tutorial*", Faculty of Mathematics and Natural Sciences, TU Ilmenau, (2007).

[4] Babovic V. and Keijzer M., "Evolutionary Algorithms Approach to Induction of Differential Equations", *Proceedings of the Fourth International Conference on Hydroinformatics"*, Iowa City, USA, (2000).

[5] Mathews J.H. and Fink K.D., "*Numerical Methods using MATLAB*", 4 th edition, Pearson Prentice Hall, New Jersy, (2004).

[6] Averill M., "*Numerical Approximation Methods applied to the Lotka-Volterra Model of Predator Prey Interactions*", Department of Mathematics, University of North Carolina.

Table 4. Noise sensitivity analysis with Gaussian white noise incorporated only in the initial values

Sno	Name of the Method	Order of Accuracy	Stepsize (h)	Initial values		Sigma	A (1)	A (2)	A (3)	A (4)	Error	Remarks
	RK-4	5	10^(-4)	35	4	-	0.5	0.02	0.8	0.02	-	Exact Solution
1	RK-4	5	10^(-1)	35	4	0.05	0.4999	0.02	0.8004	0.02	0.16	time domain=100
						0.25	0.4993	0.02	0.8019	0.0201	4.01	
						0.5	0.4986	0.02	0.8038	0.0201	16.13	
						1	0.4973	0.02	0.8076	0.0202	65.10	
						2	0.4946	0.02	0.8155	0.0205	265.37	
						3	0.4918	0.0201	0.8236	0.0207	608.84	
						5	0.4863	0.0201	0.8406	0.0212	1761.62	
						10	0.472	0.0203	0.8891	0.0227	7900.31	
						15	0.4566	0.0206	0.9497	0.0246	20417.30	
						20	0.0248	0.002	5.1286	0.1816	1580220.00	
						25	0.4541	0.0305	0.3415	0.0182	1697280.00	
2	RK-4	5	10^(-1)	35	4	0.05	0.4999	0.02	0.8004	0.02	0.08	time domain=50
						0.25	0.4993	0.02	0.8019	0.0201	1.93	
						0.5	0.4986	0.02	0.8039	0.0201	7.73	
						1	0.4973	0.02	0.8078	0.0202	31.22	
						2	0.4946	0.02	0.8158	0.0205	127.23	
						3	0.4918	0.0201	0.824	0.0207	291.81	
						5	0.4863	0.0201	0.8414	0.0212	843.80	
						10	0.4719	0.0203	0.8908	0.0227	3778.09	
						15	0.4565	0.0206	0.9525	0.0246	9746.53	
						20	0.4392	0.0211	1.0355	0.0271	20642.60	
						25	0.0089	0.000828	23.1443	0.7458	896219.00	
3	RK-4	5	10^(-1)	35	4	0.05	0.499	0.02	0.8004	0.02	0.04	time domain=25
						0.25	0.4994	0.02	0.8018	0.0201	0.96	
						0.5	0.4988	0.02	0.8037	0.0201	3.85	
						1	0.4975	0.02	0.8075	0.0202	15.55	
						2	0.495	0.0201	0.8152	0.0205	63.32	
						3	0.4925	0.0201	0.8231	0.0207	145.15	
						5	0.4874	0.0202	0.8398	0.0212	419.14	
						10	0.4742	0.0204	0.8873	0.0227	1869.87	
						15	0.4598	0.0208	0.9465	0.0245	4804.22	
						20	0.4436	0.0213	1.0262	0.027	10127.70	
						25	0.4236	0.0222	1.1489	0.0308	20161.10	
4	RK-4	5	10^(-1)	35	4	0.05	0.4999	0.02	0.8	0.02	0.01	time domain=10
						0.25	0.4996	0.02	0.8002	0.02	0.33	
						0.5	0.4992	0.02	0.8004	0.0201	1.32	
						1	0.4984	0.0201	0.8008	0.0201	5.33	
						2	0.4967	0.0202	0.8015	0.0203	21.62	
						3	0.4951	0.0203	0.8021	0.0204	49.35	
						5	0.4917	0.0204	0.8028	0.0207	141.18	
						10	0.483	0.021	0.8022	0.0215	613.54	
						15	0.4734	0.0218	0.796	0.0224	1526.88	
						20	0.4627	0.0229	0.7801	0.0234	3088.52	
						25	0.4497	0.0248	0.7446	0.0248	5791.59	

Table 5. Noise sensitivity analysis with Gaussian white noise added to all the data points

Sno	Name of the Method	Order of Accuracy	Stepsize (h)	Initial values		Sigma	A (1)	A (2)	A (3)	A (4)	Error	Remarks
				x	y							
	RK-4	5	10^(-4)	35	4	-	0.5	0.02	0.8	0.02	-	Exact Solution
1	RK-4	5	10^(-1)	35	4	0.05	0.4999	0.02	0.8003	0.02	5.10	time domain=100
						0.25	0.4995	0.02	0.8017	0.0201	127.42	
						0.5	0.499	0.02	0.8034	0.0201	509.77	
						1	0.4979	0.02	0.8068	0.0202	2039.82	
						2	0.4958	0.0201	0.8138	0.0204	8165.53	
						3	0.4937	0.0201	0.821	0.0207	18387.10	
						5	0.4895	0.0202	0.8361	0.0211	51163.40	
						10	0.4783	0.0205	0.8794	0.0225	205716.00	
						15	0.4661	0.0209	0.9338	0.0242	466125.00	
						20	0.025	0.0022	6.2823	0.191	2766914.38	
						25	0.0216	0.0019	6.2699	0.2218	2968308.23	
2	RK-4	5	10^(-1)	35	4	0.05	0.4999	0.02	0.8003	0.02	2.60	time domain=50
						0.25	0.4997	0.02	0.8014	0.02	64.94	
						0.5	0.4994	0.02	0.8028	0.0201	259.79	
						1	0.48	0.02	0.81	0.021	1039.52	
						2	0.4974	0.0202	0.8116	0.0204	4161.21	
						3	0.496	0.0202	0.8178	0.0206	9370.13	
						5	0.4932	0.0204	0.8308	0.021	26072.30	
						10	0.4854	0.0209	0.8687	0.0222	104819.00	
						15	0.476	0.0216	0.9178	0.0237	237459.00	
						20	0.4641	0.0224	0.9862	0.0258	426371.00	
						25	0.0106	0.00096	19.0623	0.6154	1582750.00	
3	RK-4	5	10^(-1)	35	4	0.05	0.5	0.02	0.8002	0.02	1.28	time domain=25
						0.25	0.4999	0.02	0.8012	0.02	32.04	
						0.5	0.4997	0.201	0.8025	0.0201	128.17	
						1	0.4994	0.0201	0.8049	0.0202	512.89	
						2	0.4988	0.0202	0.8101	0.0203	2053.34	
						3	0.4981	0.0203	0.8155	0.0205	4624.24	
						5	0.4966	0.0206	0.827	0.0208	12870.20	
						10	0.4919	0.0212	0.8609	0.0219	51779.80	
						15	0.4853	0.022	0.9055	0.0232	117406.00	
						20	0.476	0.023	0.9688	0.0251	211050.00	
						25	0.4618	0.0245	1.0717	0.0282	335911.00	
4	RK-4	5	10^(-1)	35	4	0.05	0.5	0.02	0.7997	0.02	0.55	time domain=10
						0.25	0.5001	0.0201	0.7981	0.02	13.79	
						0.5	0.5001	0.0201	0.7963	0.02	55.18	
						1	0.5002	0.0202	0.7925	0.02	220.73	
						2	0.5005	0.0204	0.7847	0.02	883.11	
						3	0.5007	0.0207	0.7766	0.02	1987.45	
						5	0.5013	0.0212	0.7589	0.02	5523.21	
						10	0.503	0.0226	0.7058	0.0199	22116.70	
						15	0.5062	0.0246	0.6343	0.0196	49806.60	
						20	0.511	0.0277	0.5422	0.0194	88606.80	
						25	0.5135	0.0324	0.4427	0.0196	138716.00	