Inference of Differential Equation Models by Genetic Programming

Hitoshi Iba
Grad. School of Frontier Informatics,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, 113-8656, Japan

Erina Sakamoto
Grad. School of Inf. and Comm. Eng.,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, 113-8656, Japan

Abstract

This paper describes an evolutionary method for identifying a causal model from the observed time series data. We use a system of ordinary differential equations (ODEs) as the causal model. This approach is well-known to be useful for the practical application, e.g., bioinformatics, chemical reaction models, controlling theory etc. To explore the search space more effectively in the course of evolution, the right-hand sides of ODEs are inferred by Genetic Programming (GP) and the least mean square (LMS) method is used along with the ordinary GP. We apply our method to several target tasks and empirically show how successfully GP infers the systems of ODEs.

1 Introduction

Ordinary differential equations (ODEs) are one of the easiest media for modeling complex systems, where basic differential relationships are known between the system components. Solving a set of differential equations to produce their equivalent functions is relatively easy so as to obtain useful time-series data. On the other hand, the inverse problem, i.e., the inference of the system of ODE from the observed time-series data, is not necessarily easy, although very important for many fields. This is because there is no knowing the appropriate form, i.e., the order and terms of ODEs, beforehand.

In this paper, we deal with an arbitrary form in the right-hand side of the system of ODEs to allow the flexibility of the model. More precisely, we consider the following general form:

$$\frac{dX_i}{dt} = f_i(X_1, X_2, \ldots, X_n) \quad (i = 1, 2, \ldots, n), \tag{1}$$

where X_i is the state variable and n is the number of the observable components.

For the sake of identifying the system, we use Genetic Programming (GP) to evolve the ODEs from the observed time series. Although GP is effective in finding the suitable structure, it is sometimes difficult to optimize the parameters, such as constants or coefficients of the polynomials. This is because the ordinary GP searches for them simply by combining randomly generated constants. To avoid this difficulty, we introduce the least mean square (LMS) method.

There have been several studies for identifying differential equation models by means of EAs (Evolutionary Algorithms). For instance, GP was used to find a function in a symbolic form, which satisfies the differential equation and initial conditions [Koza92]. Cao and his colleagues used hybrid evolutionary modeling algorithms [Cao00]. The main idea was to embed GA in GP, where GP was employed to discover and optimize the structure of a model, while GA was used to optimize its parameters, i.e., coefficients. [Babovic00] also applied GP to approximate several ODEs from the domain of ecological modeling, e.g., Lotka-Volterra and logistic equations. They showed that the GP-based approach introduced numerous advantages over the most available modeling methods. In our previous researches [Sakamoto and Iba00] and [Sakamoto and Iba01], we proposed another integrated scheme, in which the least mean square (LMS) method is used along with GP. In this scheme, some individuals were created by the LMS method at some intervals of generations and they replaced the worst individuals in the population.

In this paper, we extend our previous approach so as to achieve the inference of the ODEs more effectively. More precisely, we empirically show the following points:

- The success in the acquisition of ODEs, which are close to the observed time series.
- The inference of the exact equation form, i.e., the exact causal relationship.
- The effectiveness of the LMS method.
- The superiority of our approach over the previous methods.

The rest of this paper is organized as follows. In Section 2, we describe the details of our method, i.e., how GP and LMS methods are integrated to work in the course of evolution. Three examples are used to examine the effectiveness of our method. Their experimental results are shown in Section 3. Then, we discuss the results in Section 4 and give some conclusion in Section 5.

2 Integration of GP and LMS

We use GP to identify a causal model in the form of the system of ODEs. Though GP is capable of finding a desirable structure effectively, it cannot always be effective in finding the proper coefficients because GP uses the combination of randomly selected ones. We have chosen the least mean square method (LMS) to tackle this defect of the ordinary GP. For this purpose, coefficients are not included in the terminal set for a GP individual tree. The coefficients of each term of a GP tree are calculated by the LMS method and a table of them composes a GP individual along with a tree.

2.1 Inference of the form of equations using GP

We use GP to identify the form of the system of differential equations. For this purpose, we encode right-hand sides of ODEs into a GP individual. Each individual contains a set of n trees, i.e., an n-tuple of trees$(f_1, \ldots, f_n)$. For example, consider the two trees in Fig.1. This shows the following system of ODEs:

$$\left\{ \begin{array}{ll} \dot{X_1} = & aX_1X_2^2 + b \\ \dot{X_2} = & cX_1X_2 + dX_2, \end{array} \right. \tag{2}$$

where the coefficients a, b, c, d, are derived by LMS described later. Note that the constant term b is added to the right hand side of the first equation, because of the constant terminal, i.e., 1. Thus, each equation uses a distinct program. A GP individual maintains multiple branches, each of which serves as the right-hand side of a differential equation.

Crossover operations are restricted to the correspondent branch pairs. Actually, each tree, i.e., each right hand side of the ODE sytem, is evolved independently in parallel.

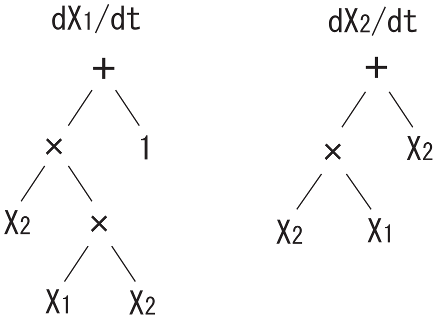


Figure 1: Example of a GP individual.

2.2 Optimization of models using LMS method

Coefficients of a GP individual is derived by the LMS method described below. Assume that we want to acquire the approximate expression in the following form:

$$y(x_1, \ldots, x_L) = \sum_{k=1}^{M} a_k F_k(x_1, \ldots, x_L), \tag{3}$$

where $F_k(x_1, \ldots, x_L)$ is the basis function, $x_1, \ldots, x_L$ are the independent variables, $y(x_1, \ldots, x_L)$ is the dependent variable, and M is the number of the basis functions. Let $\mathbf{a}$ be the vector of coefficients, i.e., $(a_1, \ldots, a_M)$. Then, our purpose is to minimize χ^2 described in (4) to acquire $\mathbf{a}$.

$$\chi^2 = \sum_{i=1}^{N} \Big(y(i) - \sum_{k=1}^{M} a_k F_k(x_1(i), \ldots, x_L(i)) \Big)^2, \tag{4}$$

where $x_1(i), \ldots, x_L(i)$ and $y(i)$ are data given for the LMS method and N is the number of data points. Let $\mathbf{b}$ be the vector of $(y(1), \ldots, y(N))$ and $\mathbf{A}$ be the $N \times M$ matrix described below:

$$\begin{pmatrix} F_1(x_1(1), \ldots, x_L(1)) & \ldots & F_M(x_1(1), \ldots, x_L(1)) \\ F_1(x_1(2), \ldots, x_L(2)) & \ldots & F_M(x_1(2), \ldots, x_L(2)) \\ \vdots & \ddots & \vdots \\ F_1(x_1(N), \ldots, x_L(N)) & \ldots & F_M(x_1(N), \ldots, x_L(N)) \end{pmatrix}$$

Then, (5) should be satisfied to minimize χ^2.

$$(\mathbf{A^T} \cdot \mathbf{A}) \cdot \mathbf{a} = \mathbf{A^T} \cdot \mathbf{b} \tag{5}$$

Thus, **a** can be acquired by solving this equation.

When applying to the time-series problem, $y(i)$ for the jth equation of the system of differential equations is calculated according to the following discrete difference of the time-series $x_j(t)$:

$$y(i) = \dot{X_j}|_{t=t_i} = \frac{x_j(t_i + \Delta t) - x_j(t_i - \Delta t)}{2\Delta t}, \tag{6}$$

where t_i is the time of the ith selected data point. For example, consider the first ODE ($\dot{X_1}$) of the system (2), in which the number of the components is two ($L = n = 2$). In this case, we are using two basis functions, i.e., $M = 2$ and (F_1, F_2)=$(X_1 X_2^2, 1)$. Then, the ith row of the matrix $\mathbf{A}$ is determined as $(x_1(t_i)x_2(t_i)^2$,1).

The coefficients in the approximate expressions of the right-hand sides of the equations can be derived by using $\mathbf{A}$ and $\mathbf{b}(y(1), \ldots, y(N))$ acquired above.

2.3 Fitness definition

The fitness of each individual is defined as the sum of the squared error and the penalty for the degree of the equations:

$$fitness = \sum_{i=1}^{n} \sum_{k=0}^{T-1} (x_i'(t_0+k\Delta t) - x_i(t_0+k\Delta t))^2 + a \cdot m, \tag{7}$$

$$\left(\begin{array}{ccl} t_0 & : & \text{the starting time} \\ \Delta t & : & \text{the stepsize} \\ n & : & \text{the number of the observable components} \\ T & : & \text{the number of the data points} \end{array} \right)$$

where $x_i(t_0 + k\Delta t)$ is the given target time series ($k = 0, 1, \cdots, T-1$). $x_i'(t_0 + k\Delta t)$ is the time series acquired by calculating the system of ODEs represented by a GP individual. All these time series are calculated by using the forth-order Runge-Kutta method. m is the number of terms and a is the weight constant. In other words, the individual which has a smaller number of terms and is closer to the target time series has the higher possibility to be selected and inherited to the next generation. This fitness derivation is based on the MDL (Minimum Description Length) criterion, which has been often used in GP (see [Iba94], [Zhang95] and [Nikolaev and Iba01] for examples). When calculating the time series, some individuals may go overflow. In this case, the individual's fitness value gets so large that it will be weeded out from the population.

	Exp.1	Exp.2	Exp.3	Exp.4
Population size	1000	1000	1000	3000
Generation	100	100	100	100
Crossover rate	0.80	0.80	0.80	0.80
Mutation rate	0.10	0.10	0.10	0.10
# time series	1	1	3	3
Stepsize	0.01		0.01	0.01
# data points	100		40	30

Table 1: GP and LMS parameters for experiments.

We use several sets of time series as the training data for GP. This is to acquire the equations as close to the target as possible. Each data set was generated from the same target by using different initial values.

3 Experimental results

We have prepared three different tasks to test the effectiveness of our method. Experimental parameters are summarized in Table 1. Function and terminal sets F and T are as follows:

$$\begin{array}{rl} F = & \{+, -, *\} \\ T = & \{X_1, \ldots, X_n, 1\} \end{array}$$

3.1 Example 1 : Chemical reaction model

The reaction between formaldehyde (X_1) and carbamide in the aqueous solution gives methylol urea (X_2) which continues to react with carbamide and form methylene urea (X_3) (see [Cao00] for details). The reaction equations are described as below:

$$HCHO + (NH_2)_2CO \xrightarrow{k_1} H_2N \cdot CO \cdot NH \cdot CH_2OH \tag{8}$$

$$H_2N \cdot CO \cdot NH \cdot CH_2OH + (NH_2)_2CO \xrightarrow{k_1} (NH_2CONH)_2CH_2 \tag{9}$$

As a kind of typical consecutive reaction, the concentrations of the three components in the system satisfy the following system:

$$\begin{cases} \dot{X_1} = & -1.4000X_1 \\ \dot{X_2} = & 1.4000X_1 - 4.2000X_2 \\ \dot{X_3} = & 4.2000X_2 \end{cases} \tag{10}$$

Experimental parameters for this task are shown in Table 1. By applying our method, we have acquired the system of eq.(11), which gave the sums of square errors as $(X_1, X_2, X_3) = (0.000, 2.082 * 10^{-11}, 1.883 * 10^{-11})$. The time series generated by this system is shown in Fig.2 along with that of the target.

$$\begin{cases} \dot{X_1} = & -1.4000X_1 \\ \dot{X_2} = & 1.4004X_1 - 4.2006X_2 \\ \dot{X_3} = & 4.1998X_2 \end{cases} \tag{11}$$

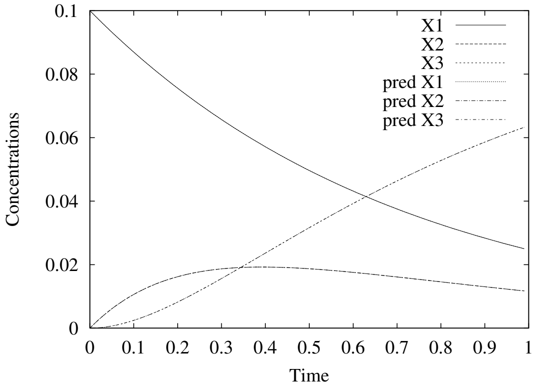


Figure 2: Time series of the acquired model for chemical reaction.

The best kinetic model acquired in [Cao00] was as follows:

$$\begin{cases} \dot{X_1} = & -1.400035X_1 \\ \dot{X_2} = & 1.355543(X_1 + t) - 4.482911X_2 \\ \dot{X_3} = & 4.069420X_2 + t - 0.002812 \end{cases} \tag{12}$$

where the sums of square errors were $(X_1, X_2, X_3) = (1.600*10^{-11}, 3.240*10^{-8}, 3.025*10^{-9})$. Note that the terminal set used in [Cao00] included the time variable t.

3.2 Example 2 : Three-species Lotka-Volterra model

The Lotka-Volterra model describes interactions between two or more species, i.e., predators and preys, in an ecosystem [Takeuchi96]. The following DOEs represent a three-species Lotka-Volterra model:

$$\begin{cases} \dot{X_1} = & (1 - X_1 - X_2 - 10X_3)X_1 \\ \dot{X_2} = & (0.992 - 1.5X_1 - X_2 - X_3)X_2 \\ \dot{X_3} = & (-1.2 + 5X_1 + 0.5X_2)X_3 \end{cases} \tag{13}$$

This system models the introduction of the third species, i.e., a predator, into a two-species system of competition, i.e., preys. More precisely, X_1 and X_2 are the number of preys competing with each other, whereas X_3 represents the number of predators.

The GP and LMS parameters we used are shown in Table 1. As a result of experiments, the following DOEs were acquired in a typical run:

$$\begin{cases} \dot{X_1} = & -10.001X_1X_3 - 1.000X_1X_2 - 0.999X_1^2 + 1.000X_1 \\ \dot{X_2} = & 0.992X_2 - 1.500X_1X_2 - 0.996X_2X_3 - 1.000X_2^2 \\ \dot{X_3} = & 4.998X_1X_3 + 0.500X_2X_3 - 1.200X_3 \end{cases} \tag{14}$$

Note that the two systems of DOEs, i.e., eqs.(13) and eqs.(14), are almost identical except for slightly different coefficients. In all runs, we have succeeded in acquiring almost the same DOEs. The MES (Mean Square Error) of the above DOEs are very small $(4.78 * 10^{-11})$.

We have conducted the further experiments with this Lotka-Volterra model to compare the performances of the following methods:

- Standard GP
- Old version of GP with LMS
- Proposed method of GP with LMS

As mentioned in Section 1, in our previous papers [Sakamoto and Iba01] and [Sakamoto and Iba00], we used the least mean square (LMS) method along with GP in a different way, i.e., some individuals were created by the LMS method at some intervals of generations and they replaced the worst individuals in the population. We compared the performance of the old version to see the effectiveness of the approach proposed in this paper.

The experimental results are given in Table 2. The table shows the MSE data and hit percentages, i.e., the ratios of successes in acquiring the target DOEs, averaged over ten runs. As clearly shown in the table, GP with LMS performed better than GP alone (standard GP), in view of MSE values. Moreover, the superiority of the proposed approach over the old version has been confirmed by the hit percentage.

3.3 Example 3 : E-cell simulation

We have conducted the experiment on the data of a metabolic network that consists of three substances.

	MSE	Hit(%)
Standard GP	$4.47 * 10^{-5}$	0%
Old version	$2.85 * 10^{-7}$	0%
Proposed method	$4.78 * 10^{-11}$	100%

Table 2: Comparision of Three methods.

This target network is a part of the biological phospholipid pathway. The data were derived from the E-cell simulation model. E-cell Simulation Environment (E-CELL SE) is a software package for cellular and biochemical modeling and simulation (see [Tomita99] for details of bioinformatics). This network can be approximated as (15).

$$\begin{cases} \dot{X}_1 = & -k_1 X_1 X_3 \\ \dot{X}_2 = & k_1 X_1 X_3 - k_2 X_2 \\ \dot{X}_3 = & -k_1 X_1 X_3 + k_2 X_2 \end{cases} \tag{15}$$

Note that the parameters k_1, k_2, and k_3 are unknown for the simulation experiment.

Three sets of time series generated by E-cell with a different initial value were used for the training of GP. Experimental parameters are shown in Table 1. By applying our method, we have acquired the following equations in a typical run:

$$\begin{cases} \dot{X}_1 = & -10.3176 X_1 X_3 \\ \dot{X}_2 = & 9.7149 X_1 X_3 - 17.5084 X_2 \\ \dot{X}_3 = & -9.7018 X_1 X_3 + 17.4766 X_2 \end{cases} \tag{16}$$

When we compare the two systems, i.e., eq.(16) and eq.(15), we can confirm the success in acquiring the almost identical model to the target ODEs. The time series generated by eq.(16) is shown in Fig.3 along with that of the target. The average MSE (Mean Square Error) of 10 runs was $2.545 * 10^{-3}$.

We have also conducted a comparative experiment without the LMS method to confirm its effectiveness (in this case, coefficients are added to the terminal set). The average MSE of 10 runs is $5.328 * 10^{-3}$, whereas that of the experiment with the LMS method is $2.545 * 10^{-3}$. Besides, the correct form of ODEs was not always acquired without the LMS method. For example, in no runs, the correct ODE for X3 was acquired without the LMS method.

3.4 Example 4 : S-system model

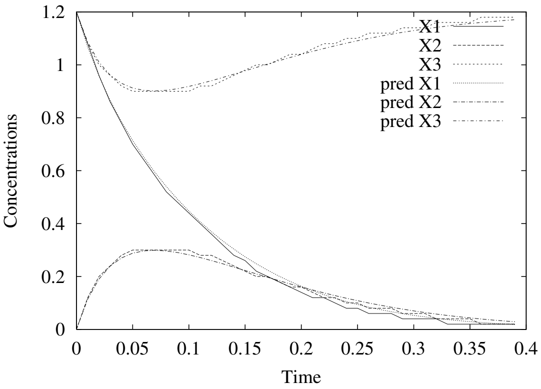


Figure 3: Time series of the acquired model for E-cell simulation.

S-system is a type of power-law formalism and has been proposed for the causality model. The concrete form of S-system is given as follows:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} (i = 1, 2, \ldots, n),$$

where X_i is a state variable. The first term represents all influences that increase X_i, whereas the second term represents all the influences that decrease X_i [Savageau76]. S-system is commonly used in many fields and its parameters were optimized by using GA [Tominaga00].

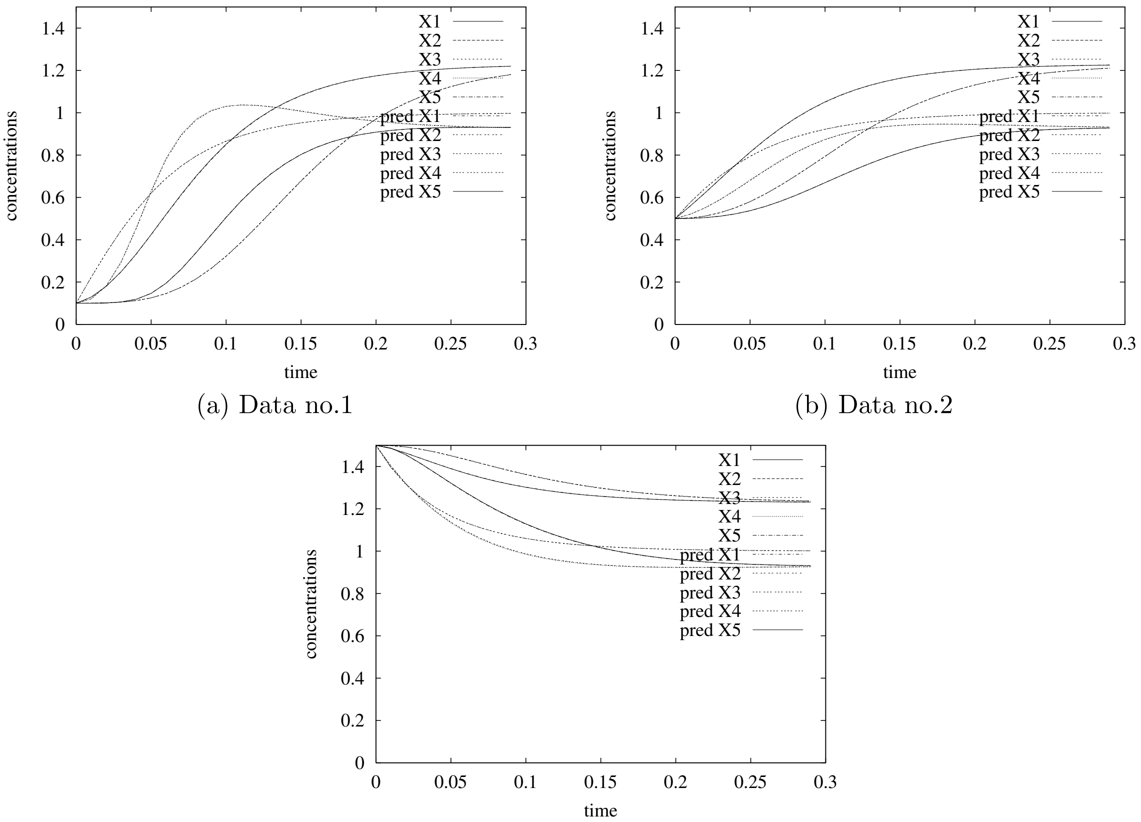
We tested on the gene regulatory network which consists of five nodes and had been generated from the S-system. This causality model can be approximated as follows (see [Tominaga00] for details):

$$\begin{cases} \dot{X}_1 = & 15.0 X_3 X_5^{-0.1} - 10.0 X_1^{2.0} \\ \dot{X}_2 = & 10.0 X_1^{2.0} - 10.0 X_2^{2.0} \\ \dot{X}_3 = & 10.0 X_2^{-0.1} - 10.0 X_2^{-0.1} X_3^{2.0} \\ \dot{X}_4 = & 8.0 X_1^{2.0} X_5^{-1.0} - 10.0 X_4^{2.0} \\ \dot{X}_5 = & 10.0 X_4^{2.0} - 10.0 X_5^{2.0} \end{cases} \tag{17}$$

Three sets of time series with a different initial value were used for the training of GP. Experimental parameters are shown in Table 1. To cope with the real-valued power of the component variables, we used the following terminal set:

$$T = \{X_1, X_1^{-1}, X_1^{0.1}, X_1^{-0.1}, X_2, X_2^{-1}, X_2^{0.1}, X_2^{-0.1}, \\ \cdots, X_5, X_5^{-1}, X_5^{0.1}, X_5^{-0.1}\} \tag{18}$$

By applying our method, we have acquired the following equations in a typical run:



(a) Data no.1

(b) Data no.2

(c) Data no.3

Figure 4: Acquired and target time series data of S-system.

$$\begin{cases} \dot{X}_1 = & 14.926X_3X_5^{-0.1} - 9.941X_1^{2.0} \\ \dot{X}_2 = & 9.950X_1^{2.0} - 9.938X_2^{2.0} \\ \dot{X}_3 = & 10.010X_2^{-0.1} - 10.005X_2^{-0.1}X_3^{2.0} \\ \dot{X}_4 = & 7.880X_1^{2.0}X_5^{-1.0} - 9.826X_4^{2.0} \\ \dot{X}_5 = & 9.935X_4^{2.0} - 9.919X_5^{2.0} \end{cases} \qquad (19)$$

Note that two systems, i.e., eq.(19) and eq.(17), are almost identical. The acquired and the given target time series are shown in Fig.4. As can be seen, the acquired time series is quite close to the target one.

For the above task, the average MSE (Mean Square Error) of 10 runs was $4.532 * 10^{-6}$. On the other hand, that of the experiment without the LMS method was $6.145 * 10^{-4}$. The equations of the correct forms were acquired in 92% of the runs with LMS, whereas in no runs the correct form of equations was acquired without the LMS method. Fig.5 shows the fitness transitions for both methods in typical cases. Thus, we can confirm that the search became more effective by using GP along with LMS method.

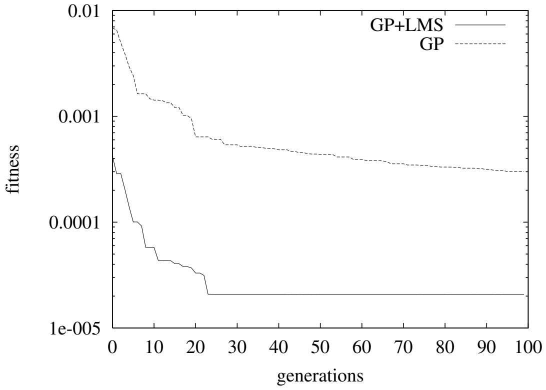
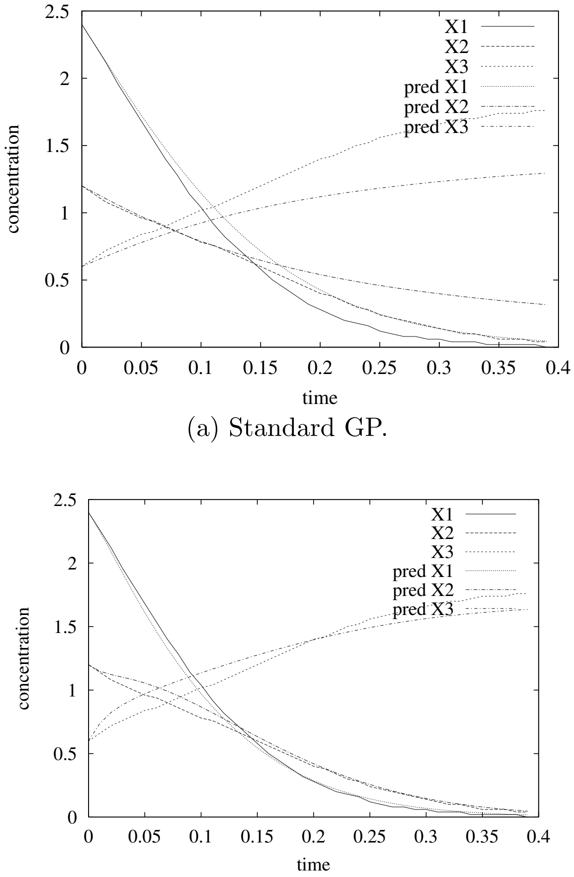


Figure 5: Typical case of the evolution for S-sytem.

4 Discussion

Although the above section shows the effectiveness of our approach in acquiring the exact form which is very close to the target observed data, there is another factor to be considered, i.e., the robustness. To test the robustness of our method to the real noisy world, we conducted the E-cell experiment (i.e., Exp.2) with noise-added data sets. 5% and 10% random noises were added to the target time series. The acquired time series are plotted in Fig.6 with the target data. The MSE values and the success ratios averaged over 15 runs are shown in Table 3. The table compares these values by our approach and the standard GP, in which the right hand sides of ODEs are evolved in a similar way to the symbolic regression (see [Koza92] for details). We can observe that the proposed method worked effectively to acquire the better individual with noisy environments than the standard GP.



(a) Standard GP.

(b) Proposed method.

Figure 6: Acquired time series of noisy data.

As can be seen in Section3.1, the proposed approach is superior to the traditional method [Cao00]. Remember that the coefficients of ODEs were derived by means of GA in Cao's scheme, whereas we used LMS for this purpose. Therefore, the superiority of our approach can also be confirmed when we consider the difference of computational burden of these techniques.

As with many other proposed models, the solution which fits the given time series quite well is not necessarily determined uniquely. In other words, there may exist more than one solution which behave consistently with the target. Therefore, even if one system of ODEs is acquired as a solution, we cannot disregard other candidates. Our aim is to obtain the candidates scattered in the huge search space and to propose to users the possible causal relationship among the observable components. Therefore, as future works, we will concentrate on the construction of the interactive system, which proposes the possible solutions and tells users what kinds of data are needed to determine the relationship among the components (see [Mimura and Iba02] for details).

5 Conclusion

We have proposed the inference method of the system of ODEs from the observed time series by using GP along with the LMS method. We showed how successfully our method can infer the causal model by several experiments. More precisely, we succeeded in acquiring the system of ODEs which is very close to the observed time series and inferring the exact equation form. The effectiveness of the LMS method and the superiority of our approach over the previous method were confirmed by comparative experiments.

As a future research, we will apply our approach to some real-world tasks. For this purpose, we are working on the development of the interactive inference system, in which users will be able to pick up the correct equations or discard the meaningless equations from the suggested ones. We are trying to solve some of the real biological problems by using this system [Mimura and Iba02].

References

[Babovic00] Babovic, V., and Keijzer, M., Evolutionary algorithms approach to induction of differential equations, in Proc. of *Genetic and Evolutionary Computation Conference (GECCO2000)*, pp.251-258, 2000.

[Cao00] Cao, H., Kang, L., Chen, Y., Yu, J., Evolutionary Modeling of Systems of Ordinary Differential Equations with Genetic Programming, *Genetic Programming and Evolvable Machines*, vol.1, pp.309-337, 2000.

[Iba94] Iba, H., deGaris, H., Sato, T., Genetic Programming using a Minimum Descrip-

Noise	0%		5%		10%	
	MSE	Success ratio	MSE	Success ratio	MSE	Success ratio
Our method	2.54^{-03}	100%	2.83^{-03}	100%	8.04^{-03}	67%
Standard GP	1.25^{-02}	40%	1.44^{-02}	33%	1.52^{-02}	33%

Table 3: The comparison of two methods in noisy environments.

tion Principle, in *Advances in Genetic Programming*, Kinnear,K.Jr. (ed.), MIT Press, pp.265-284, 1994.

[Koza92] Koza,J., Genetic Programming, MIT Press, 1992.

[Mimura and Iba02] Mimura,A. and Iba,H., Inference of a Gene Regulatory Network by Means of Interactive Evolutionary Computing, in Proc. of *Fourth Conference on Computational Biology and Genome Informatics*, (CBGI-02), 2002.

[Nikolaev and Iba01] Nikolaev,N. and Iba,H., Regularization Approach to Inductive Genetic Programming, *IEEE Transaction on Evolutionary Computation*, vol.5, no.4, 2001.

[Sakamoto and Iba00] Sakamoto, E., Iba, H., Identifying Gene Regulatory Network as Differential Equation by Genetic Programming, in Proc. of *11th Genome Informatics Workshop*, University Academy Press, 2000.

[Sakamoto and Iba01] Sakamoto, E., Iba, H., Inferring a System of Differential Equations for a Gene Regulatory Network by using Genetic Programming, in Proc. of *the 2001 Congress on Evolutionary Computation (CEC2001)*, 2001.

[Savageau76] Savageau, M.A., Biochemical Systems analysis: a study of function and design in molecular biology, Addison-Wesley, Reading, 1976.

by Y Takeuchi

[Takeuchi96] Takeuchi, Y., Global Dynamical Properties of Lotka-Volterra Systems ,World Scientific, 1996.

[Tominaga00] Tominaga, D., Koga, N., Okamoto, M., Efficient Numerical Optimization Algorithm Based
on Genetic Algorithm for Inverse Problem, Proc. of *Genetic and Evolutionary Computation Conference (GECCO2000)*, pp.251-258, 2000.

[Tomita99] Tomita,M., Hashimoto,K., Takahashi,K., Shimizu,T.S., Matsuzaki,Y., Miyoshi,F., Saito,K., Tanida,S., Yugi,K., Venter,J.C., and Hutchison,C.A.,3rd, E-CELL: software environment for whole-cell simulation, *Bioinformatics,* vol.15,no.1, pp.72-84, 1999.

[Zhang95] Zhang, B.-T. and Muehlenbein, H., Balancing accuracy and parsimony in genetic programming, *Evolutionary Computation*, vol.3, no.1, pp.17–38, 1995.