Proceedings of the 10th WSEAS International Conference on SYSTEMS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp727-738)

Finding an Optimal DE Model for Biological Network Inference

RUYE WANG
Engineering Department, Harvey Mudd College
Claremont, CA 91711, United States of America
-

ZHAOHUA TANG
Joint Science Department, Claremont Colleges
Claremont, CA 91711, U.S.A

Abstract

A variety of biological processes can be modeled by a network composed of many interacting component units (e.g. genes, proteins, neurons). It is of great biological interest to learn the interactions among these units involved in a biological process by network inference methods. However, one main obstacle in model-based network inference is the high dimensionality of the corresponding mathematical problem due to the large scale of the network model of the biological system under study. In particular, for those models based on differential equations, the large number of model parameters that need to be optimally estimated pose a major challenge due to the high computational costs in the optimization search process. This paper presents a new algorithm that decomposes the task of global search for the optimal parameters for the entire network model into a set of sub-tasks each searching for a subset of parameters associated with one unit in the network. (Such an algorithm can also be easily implemented in parallel by a multiprocessor computer system with each processor responsible for the computation for one or more of the units in the network.) The improved algorithm makes it possible to address realistic problems involving large number of components. The implementation details of the algorithm are discussed and the simulation results based on synthetic data are presented.

Keywords: biological network inference; differential equation model

1 Introduction

In various fields in science and engineering, there may be the need to model a system under study as a network composed of a set of component units, and to find how these units interact with each other. Network (or circuit) inference is a general approach that could be used to address such needs. In particular, various biological processes may be modeled as a network, such as gene regulation networks [1] [2] [3], protein networks [4] [5] [6] [7], and neural networks in the nervous system [8] [9] [10] [11] [12]. In any of these cases, the component units (the genes, proteins, or neurons) can be activated (turned on, excited) or deactivated (turned off, inhibited) to varying degrees at different times by other units in the network. For example, when a gene is turned on it is transcribed to produce messenger RNA (mRNA) which is subsequently translated into protein molecules. Some of these proteins are transcription factors which can bind to specific sites (promoter regions) of the DNA and affect the corresponding genes to turn them on or off. As another example, large number of neurons in the brain form networks responsible for various neural functions. A neuron may be either excited or inhibited by the synaptic inputs received by its dendrites from many other neurons, while its response to these inputs is in turn sent through its axon to interact with still other neurons. The interplay of these component units in such networks can be described as a network model, which controls the relevant biological processes, such as gene expression, signal transduction, or neural signal processing. It is therefore of great biological interest to learn how the component units in the network of a specific biological process interact with each other, based on the observed time course data (gene expressions, protein activities, neural activations, etc.), by various network inference methods.

Modeling and network inference can be considered as a reverse engineering problem. Based on the understanding of the biological process under study, a mathematical or computational model is to be developed to produce output data consistent with the observed biological data, such as the expression levels of genes. A model should be able to simulate the behaviors of the gene or neural networks and thereby reveal the interactions between the genes or neurons. The goal of such modeling efforts is to gain insights into and better understand the biological process under study.

Various types of models for network inference have been proposed, such as the Boolean network models (Akutsu et al [13], Liang et al [14]) which simulate the genes by a set of binary nodes interacting with each other following some simple logical operations; the linear and quasi-linear models (D'Haeseleer [15], van Someren [16]) based on the assumption that the genes are all linearly related in the network; the Bayesian networks methods (Friedman et al [17]) which treat the gene interactions as complex stochastic processes and recover the interactions by tools for learning Bayesian networks; and the differential equation (DE) models (Cohen and Grossberg [18], Hopfield [19], Mjolsness and Rienitz [20], Chen [21], Mjolsness et al [22]) which simulate the dynamic interaction of components in a biological network by a set of differential equations and find the interaction of the units in the network qualitatively. Compared to other model types, the differential equation models are more realistic biologically, but are also most challenging computationally, not only because a large set of differential equations (one for each component in the network) need to be solved, but also, a large number of parameters, typically proportional to the square of the number of differential equations, need to be optimally estimated to fit the observed data.

Due to the complexity of most of the biological systems of interest, the more realistic models such as those based on differential equations have only been able to address small scale problems with a network containing a small number of units. it is highly desirable to extend this general modeling approach of network inference to address large scale problems of many more units. In particular in order to analyze the huge amount of data produced by the latest biotechnologies, such as the gene expression data massively produced by microarray technology based on DNA hybidization, it is necessary to develop more robust and efficient algorithms. The method presented in this paper is just the result of such an effort to improve upon the existing network inference methods and to address larger scale problems in biological network inference.

2 Differential Equation Gene Network Models

Models based on differential equations are inspired biologically (neural networks, gene regulation networks) with the assumption that the dynamics of the interaction among the units in the network can be approximated by a set of first order differential equations [18] [19] [20]. Let the time varying variable (time course data) associated with the ith unit of the network with n units be denoted by $v_i(t)$, then how the units in the network interact with this ith unit can be described by the following differential equation:

$$\tau_i \frac{d}{dt} v_i(t) + \lambda_i v_i(t) = g[\sum_{j=1}^{n} T_{ij} v_j(t) + h_i] \qquad (i = 1, \cdots, n) \tag{1}$$

Among all the parameters τ_i, λ_i, h_i and T_{ij} of this differential equation model, the n by n matrix $\boldsymbol{T}$= $[T_{ij}]_{n\times n}$ is most important, as it represents specifically how the n units in the network interact with each other. Specifically, the ijth component T_{ij} of the matrix represents how the jth unit interacts with the ith unit. If $T_{ij} > 0$, the jth units activates (up-regulates, excites) the ith unit, whereas if $T_{ij} < 0$, the jth unit represses (down-regulates, inhibits) the ith unit. The two units are not related if $T_{ij} = 0$. Note that these n differential equations, one for each unit in the network, are coupled together by the T matrix, therefore they have to be solved simultaneously.

The $g(x)$ in the equation is a nonlinear monotonic sigmoidal function, typically defined as

$$g(x) = \frac{e^x - 1}{e^x + 1}, \quad \text{or} \quad g(x) = \frac{x}{\sqrt{1 + x^2}}$$

to represent the overall effect (activation or suppression) of all n units in the network asserted on the ith unit. In certain network models a constant needs to be added to the function $g(x)$ to make it non-negative (e.g., the gene expression level in gene regulation networks is always positive), while in other models it should have both polarities (e.g., the excitation and inhibition of neurons in recurrent neural networks).

The problem of network inference based on the differential equation network model described above can be formulated as below: given a set of n time course data $v_i(t)$ $(i = 1, \cdots, n)$ each for one of the n component units in the biological process at k sampling times $t = 1, 2, \cdots, k$, we want to find the parameters in the differential equation system in Equation 1, including, τ_i, h_i, λ_i and T_{ij} for all n units $(i, j = 1, \cdots, n)$, which are optimal in the sense that when the differential equation system is solved based on such parameters, its solution, the reconstructed data, will fit the observed time course data with minimum error. Once found, these optimal parameters, in particular, the matrix $\boldsymbol{T}$= $[T_{ij}]$, will reveal the interaction among the component units in the network. However, the simultaneous search for these $n(n + 3)$ optimal parameters poses a major computational challenge.

3 global optimization

The parameters in the differential equation model can be estimated by an optimization process [22] that searches through an $n(n + 3)$ dimensional parameter space spanned by all the unknown parameters to find the optimal parameters including $\boldsymbol{h}$= $[h_1, \cdots, h_n]$, $\boldsymbol{\tau}$=

$[\tau_1, \cdots, \tau_n]$, $\boldsymbol{\lambda}= [\lambda_1, \cdots, \lambda_n]$, as well as $\boldsymbol{T}= [T_{ij}]_{n\times n}$, so that the cost function defined below is minimized:

$$S(\boldsymbol{T}, \boldsymbol{h}, \boldsymbol{\tau}, \boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{t} [\hat{v}_i(t, \boldsymbol{T}, \boldsymbol{h}, \boldsymbol{\tau}, \boldsymbol{\lambda}) - v_i(t)]^2 \tag{2}$$

Here $\boldsymbol{h}$, $\boldsymbol{\tau}$, $\boldsymbol{\lambda}$, are vectors each containing the n corresponding parameters for the n units in the network. Whenever a set of estimated parameters is available, the differential equations of the model can be solved to obtain the solutions $\hat{v}_i(t, \boldsymbol{T}, \boldsymbol{h}, \boldsymbol{\tau}, \boldsymbol{\lambda})$, $(i = 1, 2, \cdots, n)$, i.e., the reconstructed data points, and the cost function, the error between the reconstructed and observed time course data, can be found to represent how well the differential equation network model fits the biological reality. Moreover, more terms may be added to this cost function for various reasons. For example, a second term of $\sum_{i,j=1}^{n} T_{ij}^2$ could be added to encourage a sparse matrix T, as in general only a small number of units in a biological network have direct links to other units. Other terms may be included in the cost function to impose additional constraints, such as the upper bounds for the parameters. However, for simplicity, all these additional terms are to be omitted in the following discussion.

The general approach of this kind of optimization problem is to iteratively search the parameter space from an initial guess to find the optimal solution that minimizes the cost function, i.e., the global minimum of the cost function in the parameter space. However, this search process presents a major challenge due to the high dimensionality of the parameter space. Moreover, to obtain the reconstructed value $\hat{v}_i(t,\ \boldsymbol{T},\ \boldsymbol{h}, \boldsymbol{\tau}, \boldsymbol{\lambda})$, needed in the cost function to be minimized, the n differential equations which are coupled by matrix T have to be solved simultaneously and repeatedly for the iterations. Consequently, the parameters for all n units will have to be searched simultaneously. In general, there are $O(n^2)$ unknown parameters to search for a network of n unit, including n^2 components of the interaction matrix $\boldsymbol{T}$, together with n sets of additional parameters one for each unit $(\{h_i, \lambda_i, \tau_i\}, i = 1, \cdots, n)$. For a network model of moderate size of $n = 10$ units, a 130-dimensional parameter space has to be searched.

Further more, due to the many possible local minima in the parameter space, usual search methods such as gradient descent does not guarantee the optimal solution corresponding to the global minimum in the parameter space. For this reason, simulated annealing search algorithms (with the capability of getting out of a local minimum) have been used for this optimization problem [20]. However, due to the extremely high computational cost and the algorithm's very slow rate of convergence, only some relatively small scale problems can be practically solved within a reasonable time.

It is practically impossible to solve a network inference problem of realistic size involving hundreds or even more units without some major improvement in both the search algorithm and the computational implementation. New effective algorithms are needed to address the difficulty of high dimensionality (the number of units in the network and the parameters in the model). In the following, we will present a new network inference algorithm as such an effort.

4 New Approach for Solving DE Model

Our algorithm is carried out in a sequence of steps as discussed below.

4.1 Interpolation of raw data

The time course data $v_i(t)$ obtained biologically are usually highly constrained due to various limitations in the laboratory setting. For example, the data are inevitably contaminated by noise of various sorts, and the number of time samples is likely to be small, due to the cost of the data collection process. To clean up the data, and to ensure the availability of enough time samples of the data required by the algorithm (more time samples than number of units in network, as discussed later), the first step is to interpolate the collected data samples by, for example, cubic spline interpolation. From this point on, we can assume the availability of the analytical expressions of the time course data $v_i(t)$. The immediate benefit of this interpolation is not only the arbitrary number of data points usable, but, more importantly, the fact that the time derivatives of the data $v_i'(t) = dv_i(t)/dt$ can be obtained analytically, rather than numerically, without computational error. This turns out to be an essential advantage based on which the following algorithm is developed.

4.2 Decomposition of the network in the search process

Searching the high dimensional space ($O(n^2)$) for the optimal parameters of the model is the most difficult part in the minimization of the cost function S defined by Equation 2 above. In general, the chance of finding the global minimum in such a high dimensional parametric space by any search-based algorithm is slim, due to the local minimum problem and the tremendously high computational cost. The hope for such a search problem is the reduction of the dimensionality of the parameter. This becomes possible if we use an alternative cost function based on the time derivatives $v_i'(t)$ of the observed time course data $v_i(t)$ and the estimated data generated by the model:

$$\begin{aligned} & S(\boldsymbol{T}, \boldsymbol{h}, \boldsymbol{\tau}, \boldsymbol{\lambda}) \\ = & \sum_{i=1}^{n} S_i(\boldsymbol{T_i}, h_i, \tau_i, \lambda_i) = \sum_{i=1}^{n} \sum_{t} [\frac{d}{dt}\hat{v}_i(t, \boldsymbol{T_i}, h_i, \tau_i, \lambda_i) - \frac{d}{dt}v_i(t)]^2 \end{aligned} \tag{3}$$

where

$$S_i(\boldsymbol{T_i}, h_i, \tau_i, \lambda_i) \triangleq \sum_{t} [\frac{d}{dt}\hat{v}_i(t, \boldsymbol{T_i}, h_i, \tau_i, \lambda_i) - \frac{d}{dt}v_i(t)]^2 \tag{4}$$

is the cost function for the ith unit, representing the difference between the time derivatives of the observed data $dv_i(t)/dt$ and the derivatives $d\hat{v}_i(t)/dt$ of the reconstructed data from the model, which can be obtained directly from the model as a function of the parameters, τ_i, λ_i, h_i and $\boldsymbol{T_i}$. Here $\boldsymbol{T_i}$ is the ith row of the n by n matrix $\boldsymbol{T}$, representing how the ith unit is affected by other units (as well as by itself if $T_{ii} \neq 0$).

The time derivative $dv_i(t)/dt$ in the cost function S_i can be found analytically from the interpolation of the observed data $v_i(t)$, while the time derivative of the reconstructed data $d\hat{v}_i(t)/dt$ based on the estimated parameters is readily available in the differential equation model as the right-hand side of Equation 1:

$$\frac{d}{dt}\hat{v}_i(t) = \frac{1}{\tau_i} g[\sum_{j=1}^{n} T_{ij} v_j(t) + h_i] - \frac{\lambda_i}{\tau_i} v_i(t) \tag{5}$$

Note that the derivative of the reconstructed data $d\hat{v}_i(t)/dt$ is given directly as the right-hand side of the above expression, i.e., finding this derivative no longer requires solving the differential equation system.

Two major advantages can be gained from this method associated with the alternative cost function in Equation 3. First, Equation 1 for the network model is no longer a set of differential equations, because now the derivative of the time course data $dv_i(t)/dt$ is known as well as the time course data $v_i(t)$, and the equations become algebraic, instead of differential. Consequently, the effort of repeatedly solving the differential equation system is avoided altogether. Second, just because of the avoidance of solving the differential equation systme, the task of finding the optimal parameters for the model can be completely decoupled so that the n sets of parameters can be estimated *independently* for each of the n units.

The second advantage above is most essential as the problem of searching an $O(n^2)$ dimensional space to minimize $S = \sum_{i=1}^{n} S_i$ can now be divided into n independent sub-problems each for searching an $O(n)$ dimensional space to minimize S_i. To see the improvement, assume there are L discrete values to consider along each dimension of the parameter space. If we need to minimize the cost function n Equation 2, we have to search a $O(n^2)$ dimensional parameter space of size $O(L^{n^2})$ for all n units. However, if the cost function in Equation 3 is used, the problem is decoupled so that we will solve n subproblems each for searching a $O(n)$ dimensional subspaces of size $O(L^n)$. Obviously, with the second method, both the complexity and computational cost are tremendously reduced, as the computational cost is linearly related to the size of the network model.

4.3 Solving the algebraic subproblems

To find the optimal parameters for each of the n subproblems, say the ith one, we first consider an arbitrary point in the 2D space spanned by the two parameters τ_i and λ_i on the left-hand side of Equation 1. For this particular point (τ_i, λ_i), together with the known time derivative $dv_i(t)/dt$ as well as the data point $v_i(t)$, we can obtain a specific value $x = \tau_i dv_i(t)/dt + \lambda_i v_i(t)$, and then its inverse function $g^{-1}(x)$, which is also the value for the right-hand side of the equation:

$$g^{-1}(x) = g^{-1}[\tau_i \frac{dv_i(t)}{dt} + \lambda_i v_i(t)] = \sum_{j=1}^{n} T_{ij} v_j(t) + h_i, \quad (t = 1, \cdots, k)$$

This is a linear algebraic equation system with $n+1$ unknowns (h_i, $T_{ij}, j = 1, \cdots, n$) and k equations each for one of the k data samples at time $t = 1, \cdots, k$. If we assume $k > n$, i.e., there are more time samples than components in the network, this equation system can be solved for the $n+1$ unknown parameters by the least square method, with a certain error. This process is repeated for each of the points in the 2D space spanned by (τ_i, λ_i) (with a reasonable range and resolution), and the solution corresponding to the smallest least square error can chosen as the optimal parameters for the network model. Moreover, we could also keep the solutions associated to the lowest few mean square errors during the search process. If these errors are similar to the absolute lowest error, the corresponding solutions can also be kept as the sub-optimal parameters, which may provide a few alternative interaction patterns of the network model, potentially meaningful for understanding the biological process. However, if the number of such sub-optimal solutions is to large, we will need to consider taking more data points to increase the number of constraining equations.

4.4 Meaningfulness of solutions

It is in general impossible to determine the existence and uniqueness of the solutions of a nonlinear algebraic system such as the one discussed above. However, when using least square method, if we can have significantly more constraining equations than unknown variables, i.e., the number of time samples k is much larger than the number of component units n in the network, we can still obtain one or a small number of meaningful solutions for the algebraic equation system.

In the case where there are a large number of units (e.g., number of genes in microarray data) but limited time samples (due to constraints imposed by the biological experiments), clustering analysis can be carried out to combine similarly behaving units (e.g. co-expressing genes) to reduce the number of independent units in the network.

Since the estimated parameters of the mode are constants instead of time variables, the network model describes a biological process that is stable, i.e., the way the units in the network interact does not change over time. However, for an unstable process, the time-varying characteristics of the process can also be captured by a time window that shifts along in time, generating the model parameters for the corresponding time pariod along the way.

5 Simulation Results

The algorithm discussed above was tested by synthetic data. Based on a set of parameters $h_i, \tau_i, \lambda_i, T_{ij}$ randomly selected, the differential equation system in Equation 1 was solved to generate a set of $n = 8$ time functions $v_i(t), \quad (i = 1, \cdots, n = 8)$ each containing $k = 50$ time samples, to simulate the time course values of eight component units in the network. Also four sets of data were generated to simulate the data collected from the same biological process repeated four times. Next the analytical derivatives $dv_i(t)/dt$ of data were obtained, and, together with $v_i(t)$, fed into the program of the algorithm to estimate the network model parameters.

Based on the cost function S_i in Equation 4, the parameters for each of the eight units were estimated separately, and then compared to the actual parameters used to generate the synthetic data. The estimated parameters obtained by the optimal parameter search algorithm for h_i, τ_i, λ_i $(i = 1, \cdots, n)$ were almost identical to those actually used to generate the data, as shown in the tables below:

	Parameters used to generate the data							
h	2.10	4.60	-5.50	10.50	3.40	-6.60	-7.90	-7.60
τ	0.90	24.90	24.90	19.20	25.20	6.60	20.70	26.40
λ	0.60	0.90	1.00	0.60	0.50	0.30	0.90	1.10
	Estimated parameters							
$\hat{h}$	2.11	4.49	-5.48	10.49	3.35	-6.60	-7.93	-7.59
$\hat{\tau}$	0.91	24.75	24.88	19.20	25.17	6.60	20.68	26.39
$\hat{\lambda}$	0.60	0.89	1.00	0.60	0.50	0.30	0.90	1.10

The two tables below show the $\boldsymbol{T}$ matrix used for generating the data and the estimated $\hat{\boldsymbol{T}}$ matrix obtained by the search algorithm. As can be seen, the estimated $\hat{\boldsymbol{T}}$ matrix closely resemble the actual $\boldsymbol{T}$ matrix. The estimation errors for most of the matrix components are small (within 10 percent), although there may be some individual entries with larger errors.

Matrix $\boldsymbol{T}$ used to generate the data							
0.40	2.00	-0.80	-0.40	-2.00	-0.40	1.20	0.40
-3.00	1.60	0.00	0.00	0.40	1.60	0.40	-0.40
-1.60	-1.20	0.40	-10.00	-0.80	1.20	-1.20	0.40
0.80	-1.20	5.00	0.40	-0.40	0.40	-1.60	-2.00
1.60	-0.40	0.40	-2.00	-1.60	5.00	0.40	1.20
-0.40	-0.80	0.00	-1.20	-4.00	-0.80	0.40	0.00
1.20	-0.40	-2.00	-1.60	-1.20	-0.40	1.60	7.00
0.40	-2.00	0.40	-1.60	0.40	1.60	-6.00	1.20
Estimated matrix $\hat{\boldsymbol{T}}$							
0.40	2.01	-0.80	-0.40	-2.00	-0.40	1.20	0.40
-2.87	1.57	-0.01	-0.01	0.33	1.51	0.42	-0.36
-1.58	-1.21	0.41	-9.85	-0.81	1.15	-1.16	0.41
0.80	-1.20	5.00	0.40	-0.40	0.40	-1.60	-2.00
1.57	-0.37	0.37	-1.97	-1.57	4.93	0.41	1.20
-0.40	-0.80	0.00	-1.20	-4.00	-0.80	0.40	-0.00
1.19	-0.43	-1.99	-1.59	-1.17	-0.41	1.58	6.98
0.40	-2.00	0.40	-1.60	0.40	1.60	-5.99	1.20

Moreover, to further validate the estimated parameters obtained above, the differential equation system was re-solved based on the estimated parameters to reconstruct the data $\hat{v}_i(t)$, which were then compared with the original data $v_i(t)$, as shown in the two figures below. These two figures each contain eight curves $v_i(t)$, $(i = 1, \cdots, n = 8)$ colored differently for the $n = 8$ units. The horizontal axis is divided into four segments each containing 50 time points for the four sets of data of the same model but under different initial conditions. As shown in the figures above, the difference between the two sets of curves for $\hat{v}_i(t)$ and $v_i(t)$ is hardly noticeable.

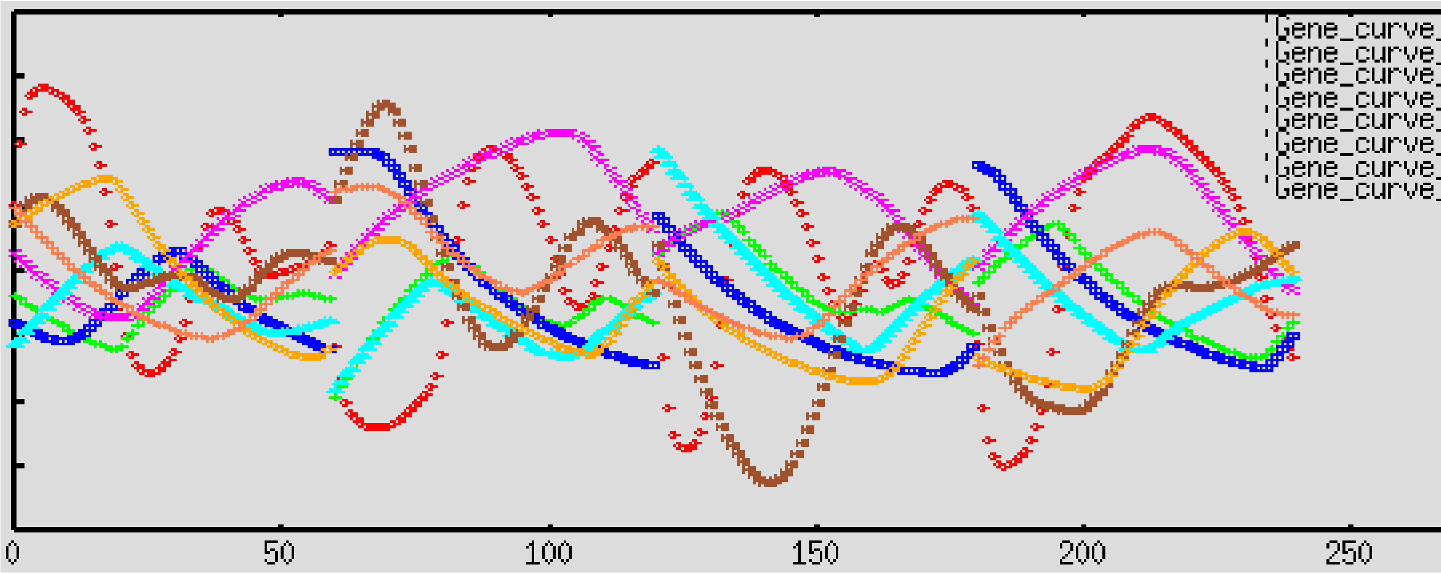

Figure 1: The simulation of observed data

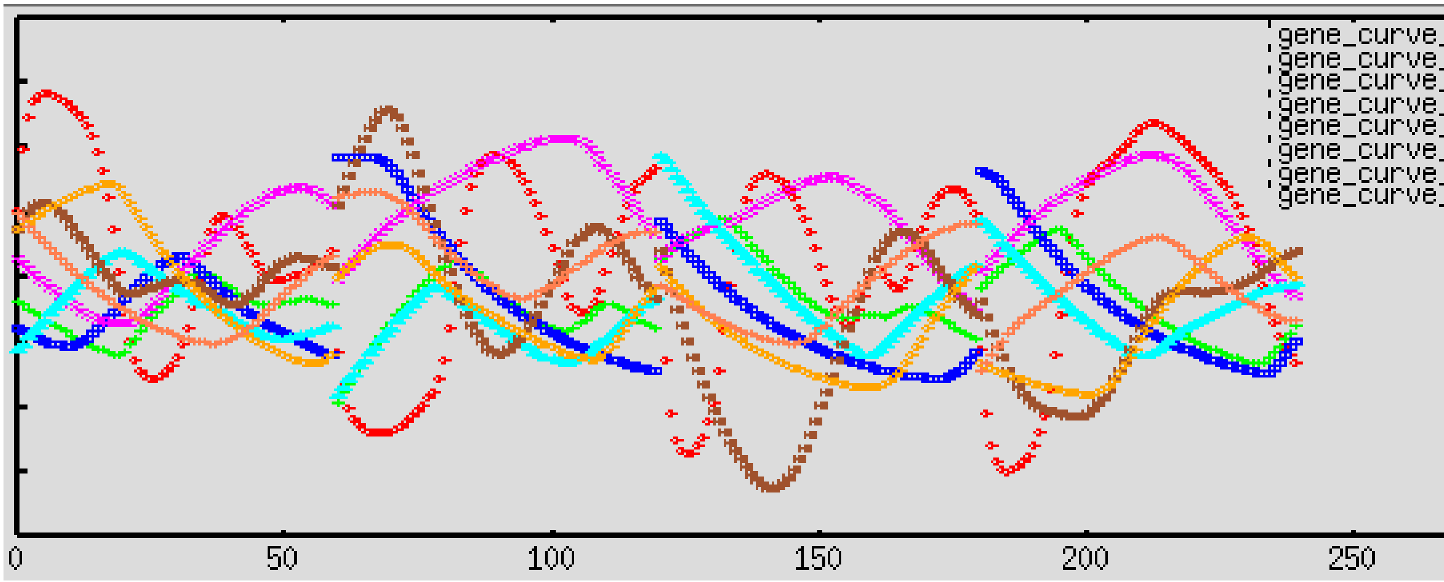

Figure 2: The reconstructed data based on estimated parameters

6 Conclusion and Future Work

We have presented a new parameter estimation algorithm by which the differential equation network model can be converted to a set of algebraic equations which can be further decomposed into a set of sub-problems each addressing one individual unit in the network. The computational cost of this algorithm for the estimation of the network model parameters is reduced to be linearly related to the number of units in the network model, and the optimal search in a high dimensional parameter space commonly encountered in such problems and the associated computational cost (typically growing exponentially as the number of units in the network increases) can be totally avoided. Consequently it becomes more likely to be able to address larger scale network problems in realistic biological systems with tens or even hundreds of units.

Moreover, as the algorithm presented here decomposes the multi-component system linearly into a set of independent sub-problems, it is highly suitable to be parallelized and implemented by a multiprocessor computer system with each processor responsible for the computation for one or more of the units in the network. With such a system, the computational task of the algorithm can be significantly sped up.

The main goal of the continuation of this research is to apply the algorithm to realistic biological problems. Inevitably various challenges will be encountered. For example, the number of samples of the time course data collected by the biological experiments is likely to be quite limited, possibly smaller than the number of units in the network model for the biological system, e.g, the number of genes in microarray data. In such cases, there will be more unknown parameters than constraining equations, consequently many sets of solutions could be obtained but revealing little relevant information of the biological process. In this case, clustering analysis methods can be used to combine similarly behaving units (e.g. co-expressing genes) to reduce the number of independent units in the network. Yet another possible challenge is that sometimes the time course data are not available in the linear scale, for example, the gene expression data are commonly collected as the ratio to some reference level. In such cases, some conversion process is needed before the algorithm discussed above can be applied.

References

[1] D'haeseleer P, Liang S, and Somogyi R, Genetic Network Inference: From Co-Expression Clustering to Reverse Engineering, *Bioinformatics* 16(8):707-26, 2000.

[2] Yu J, Smith VA, Wang PP, Hartemink AJ, Jarvis ED. Advances to Bayesian Network Inference for Generating Causal Networks from Observational Biological Data *Bioinformatics* Vol. 20, No. 18: 3594-3603 (2004)

[3] Tong A. et al, Global mapping of the yeast genetic interaction network, *Science*, 303:808-813, 2004.

[4] Bowers PM, et al. Use of logic relationships to decipher protein network organization, *Science*, 306:2246-2249, 2004.

[5] Yamanishi Y, et al. Protein network inference from multiple genomic data: a supervised approach, *Bioinformatics ISMB 2004 Supplement*, 20(S1):i363-i370, 2004.

[6] Bader JS, Greedily building protein networks with confidence, *Bioinformatics*, 19(15):1869-1874, 2003

[7] Ueta P et al A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae, *Nature* 403, 623-7, 2000.

[8] Poznanski R, Ed. *Modeling in the Neurosciences*, ISBN 9057022842, Taylor & Francis, 1999

[9] Amit D, *Modeling brain function: the world of attractor neural networks*, ISBN 0521421241, Cambridge University Pres, 1992.

[10] Wang R. A hybrid learning network for shift, orientation, and scaling invariant pattern recognition, *Network: Computation in Neural Systems*, 12, 2001, pp 493-512

[11] Wang R. A network model of motion processing in area MT of primates, *Journal of Computational Neuroscience*, 4, 1997.

[12] Wang R. A network model for the optic flow computation of the MST neurons *Journ-Neural Networks*, Volume 9, Number 3, pp 411-426, 1996.

[13] Akutsu T, Miyano S, Kuhara S. Identification of genetic networks form a small number of gene expression patterns under the Boolean network model, *Proc. of Pacific Symposium on Biocomputing, 1999* pp.17-28.

[14] Liang S, Fuhrman S, and Somogyi R, Reveal a general reverse engineering algorithm for inference of genetic network architectures *Pacific Symposium on Biocomputing 1998* 3:18-29.

[15] D'Haeseleer P, Wen X, Fuhrman S, and Somogyi R, Linear modeling of mrna expression levels during cns development and injury *Pacific Symposium on Biocomputing 1999* 4: 41-52.

[16] van Someren EP, Wessels LFA, and Reinders MJT, Linear Modeling of Genetic Networks from Experimental Data *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB00)*, 355-366, La Jolla, California, 2000.

[17] Friedman N, Linial M, Nachman I, Peer D, Using Bayesian networks to analyze expression data. *Journal of Computeraional Biology* Aug 2000, Vol. 7, No. 3-4.

[18] Cohen MA, Grossberg S, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans. system, Man and Cybernetics*, 13, pp 815-826, 1983.

[19] Hopfield JJ, Neurons with graded responses have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, USA 81 pp 3088-3092, 1984.

[20] Mjolsness E, and Reinitz J, A connectionist model of development, *Journal of Theoretical Biology*, 152, 1991.

[21] Chen T, He H, and Church G, Modeling gene expression with differential equations, *Pacific Symposium on Biocomputing* 4: 29-40, 1999.

[22] Mjolsness E, Mann T, Castano R, Wold B, From co-expression to coregulation: an approach to inferring transcriptional regulation among gene classes from large-scale expression data, in *Advances in neural information processing systems 12*, 2000.