The Inference method of the Gene Regulatory Network using RBF Networks

Naoyuki Kizaki† and Hiroaki Kurokawa†

†School of Computer Science, Tokyo University of Technology
1404-1, Katakura, Hachioji, Tokyo, Japan
Email: hkuro@bs.teu.ac.jp

Abstract—In this study, we propose the inference method of the Gene Regulatory Network(GRN) using the Radial Basis Function(RBF) network. The method using the RBF network improves inference time, but it does not show a good performance on the accuracy of the inference. Then, in our proposed method, a majority rule is applied to improve the accuracy. Our proposed method is evaluated by trials of the inference of artificially defined gene regulatory networks and a practical network.

1. Introduction

The regulatory interactions among gene expressions are well known fundamental mechanism in biological system. The model that shows the regulatory interactions among genes is known as the Gene Regulatory Network(GRN). A lot of models and inference methods of the GRN have been proposed recently. The differential equation model is one of the major GRN models that can represent the time series dynamics of the gene expressions. In general, the differential equation model is described by following equations.

$$\frac{dX_i}{dt} = G_i(X_1, X_2, \ldots, X_N). \tag{1}$$

Where X_i is the gene expression of gene i and the G_i is an unknown function.

In order to infer the GRN that described by this differential equations model, the function approximation of the unknown function G_i is required. Where we assumed that the time course data of the gene expressions are used for the function approximation of G_i. The time course data can be easily obtained by DNA microarray recently.

A lot of GRN inference methods using the differential equation model had been proposed[1][2][3]. In these conventional methods, the method using the neural networks for the function approximation[1] is known as the one of the effective method for the GRN inference. In the conventional method using the neural network, the learning algorithm of the neural network requires iteration procedures for the accurate function approximation. Also, the accuracy and the time for the function approximation depend on the convergence dynamics of the learning algorithm.

On the other hand, the Radial Basis Function (RBF) network has been also applied to the problem of the function approximation. The advantage of the RBF network is that it takes comparatively short time for the function approximation because the learning algorithm of the RBF network does not requires iteration procedures.

In this study, we propose the GRN inference method using the RBF network in order to improve the time for the inference. Also, we apply the policy of a majority rule[4] to our proposed method to improve the inference accuracy. In the simulation, we test our proposed method using the artificial GRN models defined by the S-system. We also test our proposed method using experimental data[7] as a practical case of the application of the method.

2. The GRN inference method using the neural network

In this section, we describe the conventional GRN inference method using neural networks[1]. Where we assume that the time course gene expression data of all genes in the GRN are obtained.

The method has two-step procedure. In the first step, the unknown functions G_i in equation 1 are approximated using neural networks with experimental gene expression data. The N inputs and 1 output neural network is used and the number of the hidden layer neurons is appropriately decided. In the conventional method, the genetic local search with distance independent diversity control (GLSDC) [1] is used as the learning algorithm of the neural network.

In the second step, the regulatory interactions of the GRN is estimated by analyzing the approximated functions G_i. Where the sensitivity coefficient S_{ij} is defined by following equation.

$$S_{ij} = \frac{\partial}{\partial X_j}\left(\frac{X_i}{dt}\right) = \frac{\partial G_i(X_1, X_2, \cdots, X_N)}{\partial X_j} \tag{2}$$

The sensitivity coefficient S_{ij} represents an influence of the jth gene expression X_j to the ith gene expression X_i.

In general, these sensitivity coefficients are calculated as time varying values, therefore, the averaging process[1] is applied to them to determine the static regulatory interactions in the GRN, *i.e.*, the regulation interaction is positive, negative or independent. Where the averaging algorithm proposed in the conventional study requires some parameters. In this study, we use same parameters used in the conventional study.

3. The GRN inference method using the RBF Network

In this study, we propose the GRN inference method using the RBF network. The RBF network is a kind of a multi-layer neural network that uses the radial basis function as the activation function of the neurons in the hidden layer. In our proposed algorithm, we simply use the RBF network for the function approximation instead of the neural network. A Reduction of the calculation amount is expected by using the RBF network, because the learning algorithm for the RBF network does not requires iteration procedures.

The RBF network used in this study[6] is given by,

$$O(X) = \sum_{j=1}^{M} w_j h_j(\boldsymbol{X}, \boldsymbol{C}_j, d_j), \tag{3}$$

where h_j is given by,

$$h_j = e^{-\frac{(X-C_j)^2}{{d_j}^2}}. \tag{4}$$

Where $\boldsymbol{X}$ is the input data for the network, and $\boldsymbol{C}_j$ and ${d_j}^2$ are the center vector and the variance for the basis function of neuron j, respectively.

For the function approximation using the RBF network, all the parameters of $\boldsymbol{C}_j$, d_j and w_j have to be determined. In this study, the center vector $\boldsymbol{C}_j$ is determined by following clustering method.

1. select the input vector $\boldsymbol{X}_t$ from gene expression data.
2. find a nearest center vector $\boldsymbol{C}_j$ to $\boldsymbol{X}_t$.
3. update the nearest center vector $\boldsymbol{C}_j$ by,

$$\boldsymbol{C}_j(n+1) = \boldsymbol{C}_j(n) + \eta(\boldsymbol{X}_t - \boldsymbol{C}_j). \tag{5}$$

4. repeat the 1∼3 steps for all input vectors.
5. repeat the step 4 with decreasing η until it becomes zero.

Also, the parameter d_j is determined by calculating the distance between the center vector $\boldsymbol{C}_j$ and its nearest neighbor in the other center vectors.

Then, using the parameters $\boldsymbol{C}_j$ and d_j obtained above, we find optimum $\boldsymbol{w}$ to minimize following function E by the least squares method.

$$E = \sum_{t=1}^{T} (y_t - O(\boldsymbol{X_t}))^2 \tag{6}$$

The GRN inference algorithm of our proposed method is the same as the conventional method using the neural network except for the method of the function approximation. That is, as shown in previous section, the sensitivity coefficients S_{ij} are calculated using approximated function by the RBF network and regulatory interactions are estimated by using the sensitivity coefficients.

To evaluate our proposed method, we test our method using three GRN models. Two of them are artificial 10-gene and 30-gene GRN defined by the S-system and another one is 6-gene practical network. The detailed description of each model is described in an appendix. Note that, all the calculations are executed on the PCs that have Core i7 2600K and equal configurations. The sensitivity(S_n) and the specificity(S_p) are used for the evaluation of the estimation accuracy. These criteria are defined as $S_n = \frac{TP}{TP+FN}$, $S_p = \frac{TN}{FP+TN}$, where, TP, FN, TN and FP are the numbers of regulations that is true-positive, false-negative, true-negative and false positive, respectively.

Table 1: An estimation accuracy and a calculation time for the inference of 10-gene network

method	sensitivity	specificity	time
RBFN	0.8667	0.3758	0.678s
NN	0.8362	0.5584	1.55h

Table 2: An estimation accuracy and a calculation time for the inference of 30-gene network

method	sensitivity	specificity	time
RBFN	0.7610	0.4444	6.74s
NN	0.9060	0.4326	12.2h
NN+MI	0.9659	0.9902	-

The results of 10-gene network is shown in Table 1. In the simulation, our proposed method and the conventional method are tested through 100 trials. We use the random initial values of the synaptic weights in conventional method and the number of the neurons in the hidden layer is varied from two to 101 in our proposed method. Each result in Table 1 is the average of these 100 trials. From the results, the conventional method has an advantage in the estimation accuracy, however, the time for the inference is exceedingly improved.

The results of the 30-gene network and 6-gene practical network are also shown in Table 2 and Table 3, respectively. All the results shown in the tables are average of 100 trials as similar to the simulation of the 10-gene network. Where, we show the results of the method considering the sparseness of the GRN (NN+MI) that enable to improve the estimation accuracy to the conventional method using neural network[1]. In this method, a modified objective function of the neural networks is used to consider the maximum indegree of the GRN. Where the maximum indegree of the

Table 3: An estimation accuracy and a calculation time for the inference of SOS network

method	sensitivity	specificity	time
RBFN	0.9300	0.3365	0.154s
NN	0.8975	0.5251	25.0m
NN+MI	0.7517	0.5368	-

Table 4: An estimation accuracy and a calculation time for the inference of 10-gene network with a majority rule

method	sensitivity	specificity	time
RBFN	0.8095	0.9385	1.13m
NN	0.8095	0.9777	153h

GRN is experimentally decided. From these results, similarly to the 10-gene network, the conventional method has an advantage in the estimation accuracy, however, the time for the inference is exceedingly improved.

4. The method applied the majority rule

In our previous study[4], we had proposed the inference method using neural network with a majority rule and we showed that it improved the accuracy of the GRN inference. In order to improve the estimation accuracy, we try to apply a majority rule to our proposed method. The basic strategy of the inference method with a majority rule is that the frequently estimated regulatory interactions in the results of large number of trials are chosen as a proper estimation. This strategy is based on the characteristics that the result of the heuristic method such as neural network varies depend on the initial condition or other configuration.

In this study, we assumed that the number of the trials of inference is 100. In the conventional method using the neural network, we use the random initial values of the synaptic weights for the 100 trials. Also, in our proposed method using the RBF network, the number of the neurons in the hidden layer is varied from two to 101.

In order to use the method with a majority rule, we have to decide the boundary of the frequency to judge the frequently estimated regulatory interactions. The method to decide the optimum boundary had been discussed in our previous study[5].

The results of the conventional and proposed methods with a majority rule are shown in Table 4, Table 5 and Table6. Similarly to the simulations shown in the previous section, the target network is 10-gene and 30-gene artificially defined GRN and practical 6-gene GRN. These results show that the estimation accuracy of our proposed method is improved to almost equal to the conventional method. Also, our proposed method achieves good performance in the inference time comparing to the conventional method that does not apply a majority rule. Therefore we can conclude that the adopting a majority rule is effective to the GRN inference method using RBF network.

4.1. Conclusion

In this study, we proposed the GRN inference method using RBF network. A majority rule was applied to the proposed method to improve the estimation accuracy. The results show that the estimation accuracy is almost equal to the conventional method and the time for the inference is exceedingly improved.

Table 5: An estimation accuracy and a calculation time for the inference of 30-gene network with a majority rule

method	sensitivity	specificity	time
RBFN	0.8382	0.8689	11.2m
NN	0.8088	0.9434	1220h
NN+MI	1.000	0.9919	-

Table 6: An estimation accuracy and a calculation time for the inference of SOS network with a majority rule

method	sensitivity	specificity	time
RBFN	0.9167	0.6667	15.4s
NN	0.9167	0.6333	41.7h
NN+MI	0.8333	0.5667	-

References

[1] S. Kimura, K.Sonoda, S. Yamane, K. Matsumura and M. Hatakeyama, "Function approximation approach to the inference of neural network models of genetic networks" *IPSJ Transactions on Bioinformatics*, Vol.48, No.SIG5(TBIO2), pp.9-19, 2007.

[2] M. Kabir, N. Noman and H. Iba, "Reverse engineering gene regulatory network from microarray data using linear time-variant model," *BMC bioinformatics*, 11(Suppl 1):S56, 2010.

[3] B. S. Mondal, A.K. Sarkar, M.M. Hasan and N. Noman, "Reconstruction of gene regulatory networks using differential evolution," *Proc. of ICCTT2010*, pp.440-445, 2010.

[4] Y. Hirai, M. Kikuchi, and H.kurokawa, "Inferring method of the gene regulatory networks using neural networks adopting a majority rule," *Proc. of IJCNN2011*, pp. 2936 - 2943, 2011.

[5] Y. Hirai, N. Kizaki, H. Yoshino and H.kurokawa, "The parameter optimization in the inference of gene regulatory network by neural networks adopting majority rule," *Proc. of IJCNN2011*, pp. 2936 - 2943, 2011.

[6] Philip D.Wasserman, Advanced Methods in Neural Computing, John Wiley & Sons, Inc., 1993

[7] "http://www.weizmann.ac.il/mcb/UriAlon/", Accessed on 2011

Appendix

In this study, three GRN models are used in the simulation. Two of them are the artificially defined GRN. Figure 1 and 2 show the 10-gene network and 30-gene network, respectively. These models are defined by S-system that is given by,

$$\frac{dX_i}{dt} = \alpha_i \Pi_{j=1}^{N} X_j^{g_{i,j}} - \beta_i \Pi_{j=1}^{N} X_j^{h_{i,j}} (i = 1, 2, ..., N). \quad (7)$$

Where the parameters in equation 7 for 10-gene network and 30-gene network are summarize in Table 7 and 8, respectively.

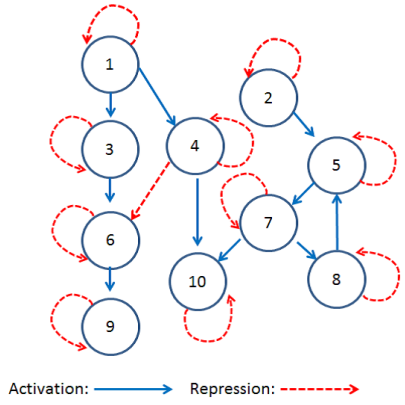

Figure 1: The 10-gene network

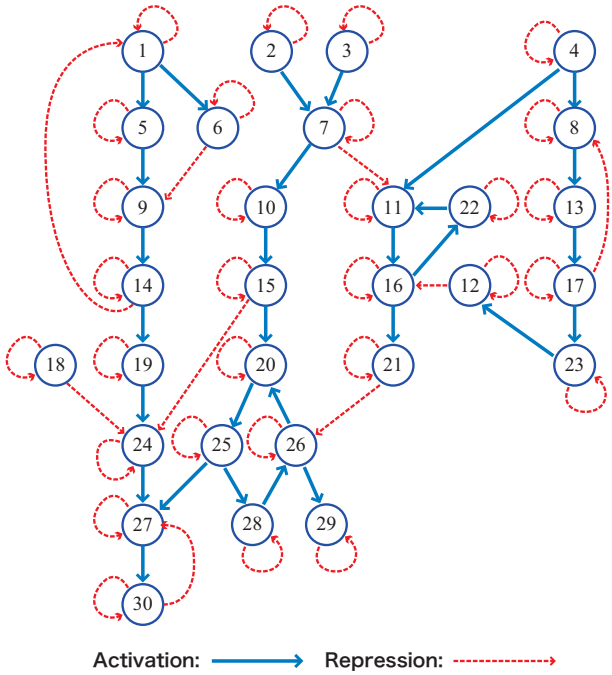

Figure 2: The 30-gene network

Table 7: The parameters for 10-gene network

α_i	1.0
β_i	1.0
$g_{i,j}$	$g_{3,1} = -0.4$, $g_{4,1} = 0.3$, $g_{5,2} = -0.3$, $g_{6,3} = 0.7$, $g_{10,4} = 0.6$, $g_{7,5} = 0.4$, $g_{9,6} = 0.5$, $g_{10,7} = -0.4$, $g_{7,8} = 0.3$, other $g_{i,j} = 0.0$
$h_{i,j}$	$g_{6,4} = -1.0$,other 1,0 if i=j, 0.0 otherwise.

Table 8: The parameters for 30-gene network

α_i	1.0
β_i	1.0
$g_{i,j}$	$g_{1,14} = -0.1$, $g_{5,1} = 1.0$, $g_{6,1} = 1.0$, $g_{7,2} = 0.5$, $g_{7,3} = 0.4$, $g_{8,4} = 0.2$, $g_{8,17} = -0.2$, $g_{9,5} = 1.0$, $g_{9,6} = -0.1$, $g_{10,7} = 0.3$, $g_{11,4} = 0.4$, $g_{11,7} = -0.2$, $g_{11,22} = 0.4$, $g_{12,23} = 0.1$, $g_{13,8} = 0.6$, $g_{14,9} = 1.0$, $g_{15,10} = 0.2$, $g_{16,11} = 0.5$, $g_{16,12} = -0.2$, $g_{17,13} = 0.5$, $g_{19,14} = 0.1$, $g_{20,15} = 0.7$, $g_{20,26} = 0.3$, $g_{21,16} = 0.6$, $g_{22,16} = 0.5$, $g_{23,17} = 0.2$, $g_{24,15} = -0.2$, $g_{24,18} = -0.1$, $g_{24,19} = 0.3$, $g_{25,20} = 0.4$, $g_{26,21} = -0.2$, $g_{26,28} = 0.1$, $g_{27,24} = 0.6$, $g_{27,25} = 0.3$, $g_{27,30} = -0.2$, $g_{28,25} = 0.5$, $g_{29,26} = 0.4$, $g_{30,27} = 0.6$, other $g_{i,j} = 0.0$
$h_{i,j}$	1,0 if i=j, 0.0 otherwise.

We also use the practical example called SOS network of the Escherichia coli bacterium that is shown in Figure 4.1. Where the regulatory interactions of SOS network that described in [3] are used. Also, we use the experimental gene expression data on the web site[7] in the simulation.

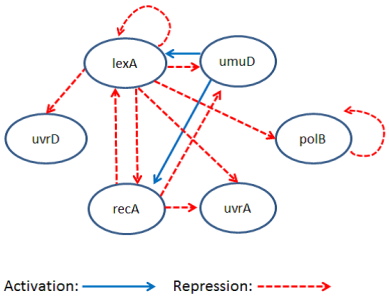

Figure 3: The practical example of the GRN