[S11-5]

Methods for Learning the Structure of Regulatory Networks from Time Course Expression Data via Linear Models

Dougu Nam

Division of Industrial Mathematics, National Institute for Mathematical Sciences, Yuseong, Daejeon

1. Introduction

Inferring genetic regulatory networks from time course gene expression data has been of general interest. To this aim, many mathematical models have been employed such as dynamic Bayesian networks, Boolean networks, system of linear equations, and so on [1].

One major challenge with the problem is that most time course data have only a small number of data points which makes it difficult to monitor the full dynamics of the gene expression process. In this regard, system of linear equations has been most widely used, because they are relatively simple and are easily learned from short time-course expression data.

To cope with the dimensionality problem in linear models, several algorithms or techniques have been applied such as interpolation between time points [2], subset selection for multiple linear regression [3], and singular value decomposition (SVD) [4]. Subset selection [5] is a typical way to deal with the dimensionality problem, but easily causes over-fitting of the data because most of the time-series data have a small number of data points [6]. SVD is a useful alternative that robustly captures the network features even for a small number of data points. However, prediction by SVD fluctuates largely depending on the number of principal components chosen.

One crucial problem in these methods that has easily been overlooked is that we cannot estimate the parameters of the model exactly from sparsely sampled short time-course data. Recently, a new system identification technique what the author called *ensemble learning* [7] aimed to reconstruct only the network structures (connections and their relative strengths) from time-course expression data.

Ensemble learning amalgamates the 'signs' of estimated parameters from multiple likely models instead of using the single most likely model. The method does not estimate the exact parameters, but provides more accurate information on network structures.

In this article, we review the ensemble method for learning the structure of regulatory networks and investigate the performance of its variation. We test the algorithms on the SOS system of *E. coli*.

2. System and Methods

Here, we briefly describe the ensemble learning method and its variation. See [7] for a detailed description of the method.

2.1 Model Description

Suppose we have m time-series measurements of the mRNA levels of n genes. Let g_i be the ith gene or its expression level, and k be the maximum number of regulators for each target gene. We only have the restriction on the number of measurements that $m \geq k$.

We model the regulatory networks by a system of linear differential equations as follows:

$$\frac{dg_i(t)}{dt} = \gamma_i + \sum_{j=1}^{n} \lambda_{ij} g_j, \quad t \geq 0, \ i=1, \ \mathrm{K}, \ n,$$

where we call g_j in the right hand side of the equation *regulator* of g_i and the coefficient λ_{ij}, *regulating factor* (*r-factor* in short). If $\lambda_{ij} > 0$, we interpret that g_j activates g_i, and if $\lambda_{ij} < 0$, g_j represses g_i. We assume regulatory networks are very sparse so that we set most of the r-factors λ_{ij} to be zero. In the vector form, the system reads

$$\frac{dG(t)}{dt} = \Lambda \bullet G(t) + \Gamma, \quad t \geq 0, \tag{1}$$

where $G = (g_1, \ \mathrm{K}, \ g_n)^{\mathrm{T}}$, $\Lambda = [\lambda_{ij}]_{1 \leq i, j \leq n}$, and $\Gamma = (\gamma_{1,\mathrm{K}}, \gamma_n)^{\mathrm{T}}$. We call $R = [\Lambda \ \Gamma]$ the *regulation matrix*. The system (1) is approximated by the difference equations as follows:

$$\Delta G(t_j) = (\Lambda \bullet G(t_j) + \Gamma)\Delta t_j + \mathrm{E}(t_j), \quad j = 0, \ldots, m-1, \tag{2}$$

where $\Delta G(t_j) = G(t_{j+1}) - G(t_j)$, $\Delta t_j = t_{j+1} - t_j$ and $\mathrm{E}(t) = (e_1(t_j), \ldots e_n(t_j))$ represents error. We will estimate the parameters in (2) from discretely observed data.

2.2 Algorithms

ssLMS: subset selection for Least Mean Square error

The least mean square error estimation is the most popularly used method for recovering the linear (difference) equations, but the number of variables (regulators) should not exceed that of data points. Thus, we search for a subset of variables that minimizes the LMS error under the assumption of the network

sparseness.

Let R_i be the *i*th row of R. For our localized algorithm, we consider a fixed gene g_i and the *i*th equation of (2)

$$\Delta g_i(t_j) = (R_i \bullet [G(t_j)\ 1]^{\mathrm{T}})\Delta t_j + e_i(t_j), \quad j = 0,\ldots,m-1. \tag{3}$$

Let $G_i = (g_1^{(i)},\ldots,g_{r_i}^{(i)})$ be a collection of regulators for g_i with non-zero r-factors $C_i = (\lambda_1^{(i)},\ldots,\lambda_{r_i}^{(i)})$. We let the last element $g_{r_i}^{(i)}$ in G_i be the self regulator g_i and $\lambda_{r_i}^{(i)}$ be its r-factor. G_i should include the self regulator g_i to represent the auto-regulation.

Calculation of r-factors for known regulators In this Subsection, we assume that we know all of its regulators $g_1^{(i)},\ldots,g_{r_i}^{(i)}$, but not their r-factors. By assuming the same amount of error at each time step, we constitute the least mean square error from (3) as follows:

$$\hat{\varepsilon}_i^2 = \frac{1}{m}\sum_{j=0}^{m-1}(\Delta g_i(t_j) - [C_i \cdot G_i(t_j)^{\mathrm{T}}]\Delta t_j)^2 . \tag{4}$$

By differentiating (4) with respect to the row vector C_i, we obtain the optimal estimates of r-factors $\hat{C}_i$ that minimizes (4). Now, we search for non-zero r-factors using subset selection in the next step.

Searching for regulators To identify the key regulators, we exhaustively search the possible combinations of regulators for one that has the smallest LMS estimate (4). From the assumption of network sparseness, it is assumed that $k \le 4$.

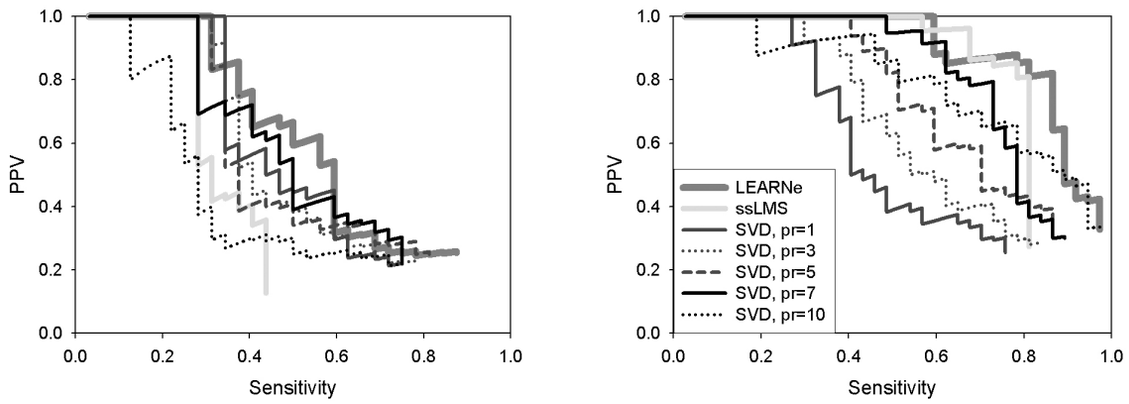
LEARNe: *Ensemble algorithm*

Step 1. For each fixed target gene g_i, we calculate $\hat{\varepsilon}_i^2$ (4) for every possible combination of k regulators: If the number of gene is *n*, the number of possible combinations of regulators is ${}_nC_k$ taking into account the drift term and the self regulator.

Step 2. Among the ${}_nC_k$ number of $\hat{\varepsilon}_i^2$'s, we take the smaller (more likely) μ% of the estimates: Each estimate provides a model (a combination of regulators and corresponding r-factors) for regulating g_i.

Step 3. Set the 'voting' regulation matrix Θ of size $n \times (n+1)$ initialized with zero elements. For each of the μ% likely models, we accumulate signs (votes) of the corresponding r-factors on Θ.

Step 4. Repeat *Step 1~3* for all *i*'s to complete Θ.



(A) (B)

Fig. 1. ROC curves for (a) 10 and (b) 50 data points.

See [7] on how to choose the optimal parameter μ.

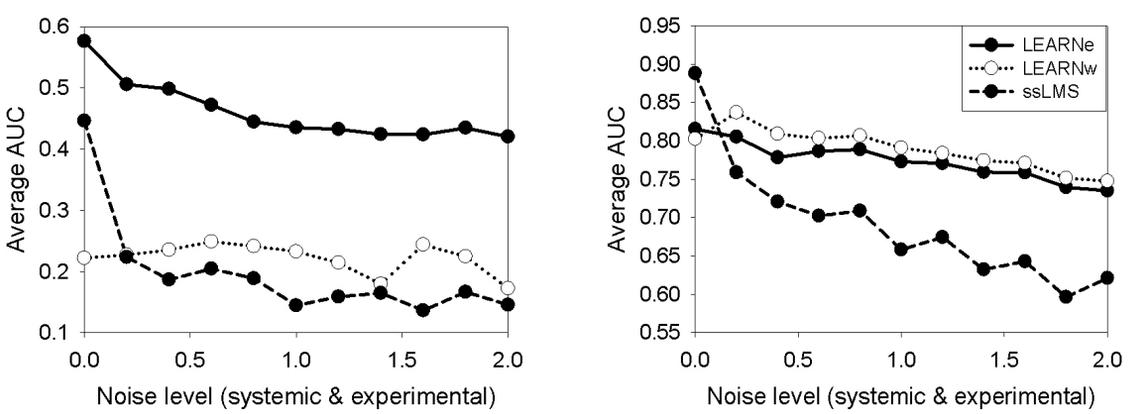
LEARNw: *A variation of LEARNe*

Only the Step 3 is replaced by Step 3'.

Step 3'. Set the 'voting' regulation matrix Θ of size $n \times (n+1)$ initialized with zero elements. For each of the μ% likely models, we accumulate the corresponding r-factors themselves on Θ.

3. Test of the algorithms

Here, we compare the performances of three algorithms, ssLMS, LEARNe, LEARNw and SVD by simulation tests. We generated time-series data from randomly constructed stable linear systems that incorporate both systemic (biological) and experimental noise. From the noisy data, we reconstructed the underlying networks using each of the four algorithms. We repeated the test for 40 randomly generated stable systems in each setting and we compared AUC for each algorithm. See [7] for a detailed explanation for the measure of performance.



(A) (B)

Fig. 2. Average AUC for (a) five and (b) 30 data points.

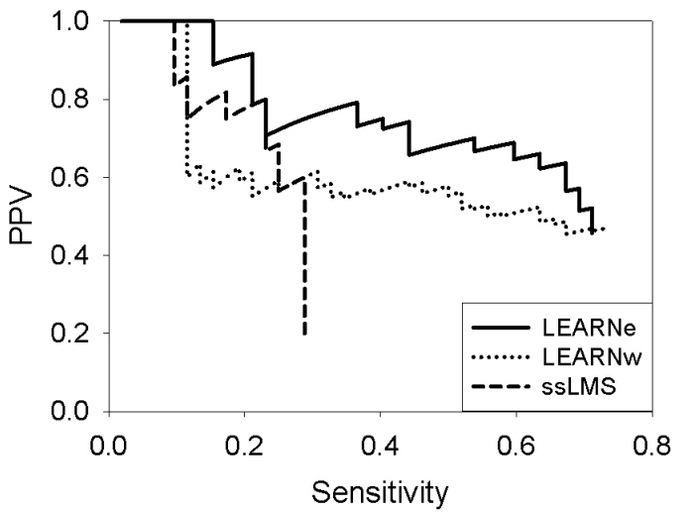


Fig. 3. ROC curves for SOS regulatory network of *E. coli.*

Typical performance plots (ROC) are shown in Fig. 1. LEARNe significantly improved the prediction power of the naïve subset selection method (ssLMS), and also outperformed SVD for ten or more number of data points. The average-case test results are shown in Fig. 2. We show ten dimensional cases, but similar patterns are observed for higher dimensional networks. Fig. 2A demonstrates why ensemble methods are required over the least square method (ssLMS). For this small number of data points, μ=100% worked best for both ensemble methods. For this small number of data points (five), accumulating the estimated coefficients (LEARNw) performed a little better than ssLMS. However, accumulating only the signs of the estimated coefficients much better identified the network structure (LEARNe). This implies the individual models used for the ensemble learning do not provide reliable estimates for small number of data points. On the other hand, when we used 30 data points, LEARNw outperformed LEARNe. For five data points, SVD performed better than ensemble methods, but for thirty data points, both the ensemble methods outperformed SVD (data not shown). See also [7] for further test results.

Lastly, we tested the methods on a time series microarray data set to reconstruct the SOS regulatory network of *E. coli.* We used the time series data used Bansal *et al*. (2006). With this real data set, accumulating the signs of coefficients (LEARNe) still performed clearly better than accumulating the coefficients themselves (LEARNw) for a small number of data points.

References

1. H. de Jong. (2002) Modeling and simulation of genetic regulatory systems: a literature review, *J. Comput. Biol.*, **9**, 67-103.
2. P. D'Haeseleer, P. *et al.* (1999) Linear modeling of mRNA expression levels during CNS development and injury, *Pac. Symp. Biocomput.*, 41-52.
3. T.S. Gardner, *et al.* (2003) Inferring genetic networks and identifying compound mode of action via expression profiling, *Science*, **301**, 102-105.
4. M. Bansal, *et al.* (2006) Inference of gene regulatory networks and compound mode of action from time course gene expression profiles, *Bioinformatics*, **22**, 815-822.
5. A. Miller (2002) *Subset selection for regression*. Chapman&Hall/CRC.
6. J. Ernst, *et al.* (2005) Clustering short time series gene expression data, *Bioinformatics*, **21 Suppl 1**, i159-168.
7. D. Nam, *et al*. (2007) Ensemble learning of genetic networks from time-series expression data, *Bioinformatics* **23**, 3225-3231.