Ensemble learning of genetic networks from time-series expression data

Dougu Nam¹,²,*, Sung Ho Yoon¹ and Jihyun F. Kim¹, *

¹Korea Research Institute of Bioscience and Biotechnology (KRIIBB), PO Box 115, Yuseong, Daejeon 305-600 and ²National Institute for Mathematical Sciences (NIMS), Yuseong, Daejeon 305-340, Republic of Korea

Received on July 12, 2007; revised on September 12, 2007; accepted on October 7, 2007
Advance Access publication October 31, 2007
Associate Editor: Olga Troyanskaya

ABSTRACT

Motivation: Inferring genetic networks from time-series expression data has been a great deal of interest. In most cases, however, the number of genes exceeds that of data points which, in principle, makes it impossible to recover the underlying networks. To address the dimensionality problem, we apply the subset selection method to a linear system of difference equations. Previous approaches assign the single most likely combination of regulators to each target gene, which often causes over-fitting of the small number of data.

Results: Here, we propose a new algorithm, named LEARNe, which merges the predictions from all the combinations of regulators that have a certain level of likelihood. LEARNe provides more accurate and robust predictions than previous methods for the structure of genetic networks under the linear system model. We tested LEARNe for reconstructing the SOS regulatory network of Escherichia coli and the cell cycle regulatory network of yeast from real experimental data, where LEARNe also exhibited better performances than previous methods.

Availability: The MATLAB codes are available upon request from the authors.

Contact: dunam@nims.re.kr or jfk@kribb.re.kr

1 INTRODUCTION

Cells adapt to specific environmental conditions by controlling the expression of genes, which is mediated by complex intracellular regulatory networks. To elucidate key biological processes, transcriptome profiles of DNA microarray have been analyzed for a wide range of species (Arbeitman et al., 2002; Spellman et al., 1998; Yoon et al., 2003). In many cases, expression of genes is indicative of the activities of gene products. Based on this knowledge, many computational models have been developed to reverse-engineer the genetic regulatory networks from mRNA expression profiles (see the review by de Jong, 2002).

Among the models, linear systems are of the most widely used for unraveling genetic networks from time-series expression data (Bansal et al., 2006; Chen et al., 2005; de Hoon et al., 2003; Gardner et al., 2003; Guthke et al., 2005; Radde et al., 2006; van Someren et al., 2000; Weaver et al., 1999; Yeung et al., 2002). In particular, stochastic differential equations have been used to model biological variations (Chen et al., 2005; de Hoon et al., 2003). Other kinds of models for time-series data include Boolean networks (Kauffman, 1969; Martin et al., 2007; Nam et al., 2006) and dynamic Bayesian networks (Murphy and Mian, 1999; Ong et al., 2002). Boolean networks derive logical functional relationships among expression profiles and usually consider two expression levels, on and off. However, such simplification causes significant information loss. Dynamic Bayesian networks provide a sophisticated framework for deriving stochastic relationships of gene expression, but they also suffer from information loss from the simplification of data and require a considerable number of data points for a reliable estimation that limits its utilization. Linear system models are based on the simplification of linearity, but are easily learned from short time-series data with the assumption of the sparseness of network structures. As linear systems are the first-order approximation of the underlying non-linear regulatory networks, they have been successfully applied to some well-organized sub-networks of genes (Bansal et al., 2006; Gardner et al., 2003; Tegner et al., 2003).

A common problem for using linear system models is that the number of genes of our interest often far exceeds the small number of temporal data points. To address the problem, several algorithms or techniques have been applied such as interpolation between time points (D’Haeseleer et al., 1999), subset selection for multiple linear regression (Gardner et al., 2003) and singular value decomposition (SVD) (Bansal et al., 2003; Holter et al., 2001; Yeung et al., 2002). Subset selection (Miller, 2002) 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 (Ernst et al., 2005). 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.

In this article, we provide a new algorithm, dubbed LEARNe, that recovers the structure of sparse linear system model from time-series data. Our motivation is that the optimal model learned from short time-series data may not be reliable, and multiple likely models taken together may provide a better explanation of the data. Based on the subset selection method,
we merged the predictions from all the combinations of regulators that have a certain level of likelihood instead of using the single most likely combination of regulators. This approach considerably improved the prediction power of the subset selection method, and it also surpassed SVD especially for larger numbers of data points. This method resembles the ensemble learning technique in the machine-learning field (Dietterich, 2000), the principle of which is that a predictor composed of multiple models often shows a better prediction power than any single model. We compare LEARNe with the two most widely used methods, i.e. subset selection for Least Mean Square (ssLMS) error and SVD for reconstructing genetic networks for both simulated and real data, which demonstrates that LEARNe is a prominent method for identifying the sparse structure of genetic networks from time-series data.

2 SYSTEM AND METHODS

2.1 Model description

Suppose we have $m$ time-series measurements of the mRNA levels of $n$ genes. Let $g_i$ be the $i$th 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:

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

where we call $g_i$ in the right-hand side of the equation regulator of $g_i$ and the coefficient $\lambda_{ij}$ 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 let ‘$\gamma$’ denotes the drift regulator and $\gamma_i$ be its $r$-factor, which represents the effect of outer perturbations acting on $g_i$. We assume regulatory networks are very sparse so that we set most of the $r$-factors $\lambda_{ij}$ to be zero. In the vector form, the system reads

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

where $G = (g_1, \ldots, g_n)^T$, $\Lambda = [\lambda_{ij}]_{n \times n}$, and $\Gamma = (\gamma_1, \ldots, \gamma_n)^T$. We call $R = [\Lambda \quad \Gamma]$ the regulation matrix. The system (1) is approximated by the difference equations as follows:

$$\Delta G(t_j) = (\Lambda \cdot G(t_j) + \Gamma) \Delta t_j + E(t_j), \quad j = 0, \ldots, m - 1,$$

where $\Delta G(t_j) = G(t_{j+1}) - G(t_j)$, $\Delta t_j = t_{j+1} - t_j$, and $E(t) = (e_1(t), \ldots, e_n(t))$ represents error. We will estimate the smaller (more likely) $\mu^\%$ of the estimates: Each estimate provides a "$r$"-factor for known regulators, we take the smaller $r$-factors (more likely) $\mu^\%$ of the estimates: Each estimate provides a model (a combination of regulators and corresponding $r$-factors) for regulating $g_i$.

2.2 Algorithms

2.2.1 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)

$$\Delta g_i(t_j) = \left(R_i \cdot \left[ G(t_j) - G_0 \right] \right)^T \Delta t_j + e_i(t_j), \quad j = 0, \ldots, m - 1.$$  \hspace{1cm} (3)

Let $G_i = (g_{i1}, \ldots, g_{in})$ be a collection of regulators for $g_i$, with non-zero $r$-factors $C_i = (\lambda_{i1}, \ldots, \lambda_{in})$. We let the last element $g_0$ in $G_i$ be the self-regulator $g_i$ and $\lambda_{i0}$ be its $r$-factor. $G_i$ should include the self-regulator $g_i$ to represent the auto-regulation.

2.2.2 Calculation of $r$-factors for known regulators

In this Subsection, we assume that we know all of its regulators $g_{i1}, \ldots, g_{im}$, 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:

$$\hat{\lambda}_{ij} = \frac{1}{m-1} \sum_{j=0}^{m-1} (\Delta g_i(t_j) - [C_i \cdot G_i(t_j)] \Delta t_j)^2.$$  \hspace{1cm} (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). Yet, another important problem is to know which genes have non-zero $r$-factors for each target gene. We use the subset selection method in the next step.

2.2.3 Searching for regulators

If we have chosen right regulators, we might have a small estimate of the noise variance (4) and obtain the corresponding $r$-factors as described above. To identify the key regulators (regulators with non-zero $r$-factors) for a target gene, we exhaustively search the possible combinations of regulators for one that has the smallest LMS estimate (4). Smaller error estimate indicates the corresponding regulators are more likely ones for the target gene. From the assumption of network sparseness, it is assumed that $k \leq 4$. We may exhaustively search all the cases $k = 2, 3$ and 4, but it would be sufficient to consider $k = 4$ for simplicity because the zero $r$-factors, if included in $C_i$, will be estimated to be zero.

We note that another system identification method (maximum-likelihood estimation) proposed by De Hoon and coworkers (De Hoon et al., 2003) is virtually the same as ssLMS for equidistant time steps. Now, we describe our main algorithm as follows:

2.2.4 LEARNe

Step 1. For each fixed target gene $g_i$, we calculate $\hat{\lambda}_{ij}$ (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 $\hat{\lambda}_{ij}$’s, we take the smaller (more likely) $\mu^\%$ 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 $\Theta$ of size $n \times (n+1)$ initially filled with zero elements in the $i$th row. For each of the $\mu^\%$ likely models, we accumulate sums (votes) of the corresponding $r$-factors on $\Theta$: $\Theta$ corresponds to the regulation matrix $R$, but is composed of integer values. A large absolute integer value in a row of $\Theta$ indicates the corresponding regulator was often included in the ‘likely’ regulator combinations and its $r$-factors exhibited a consistent sign in each model considered.

Step 4. Repeat Steps 1–3 for all $i$’s to complete $\Theta$.

The optimal value of $\mu$ is inferred from simulation tests before applying the algorithm to real data. It mainly depends on the number of measurements (see Supplementary Material). For example, with 10 data points, the algorithm shows best performance with $\mu = \sim 40$–60% across different noise levels (see Supplementary Fig. 2b). $\Theta$ contains the information for the connectivity and the regulatory effects among genes. If one wants to estimate the exact kinetics, one can re-calculate them by ssLMS utilizing the connectivity information in $\Theta$.

2.2.5 Singular value decomposition

We used the SVD algorithm presented by di Bernardo and coworkers (Bansal et al., 2006). But, we modified it to have the ratio of increments in the left side of the following Equation (4), which enabled to apply the algorithm to varying time intervals and fixed ones as well. From the Equation (2),
we input the observed data into \( \Delta G(t_k) \) and \( G(t_k) \) and apply SVD to the rightmost matrix of the following equation such that

\[
W = \begin{bmatrix}
\frac{\Delta G(t_k)}{\Delta t_k} \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
\Lambda & \Gamma
\end{bmatrix}
\begin{bmatrix}
G(t_k) \\
1 \ldots 1
\end{bmatrix}
\]

where \( U \) and \( V \) are matrices of left and right singular vectors, respectively and \( D \) is a diagonal matrix of singular values arranged in descending order. Choosing the first \( k \) singular values in \( D \), say \( D_k \), we have

\[
W \cdot P_k = \begin{bmatrix}
\Lambda & \Gamma
\end{bmatrix} \cdot U_k,
\]

where \( P_k \) is the pseudo-inverse of \( D_k \cdot V^T \) and \( U_k \) is composed of the first \( k \) columns of \( U \). Then, extending both sides of the equation with a zero matrix and taking \( U^{-1} \), we have

\[
[ W \cdot P_k \ 0] \cdot U^{-1} = [ \Lambda \quad \Gamma] \cdot [U_k \ 0] \cdot U^{-1}
\]

where \( 0 \) is an \( (n + 1) \times (n + 1 - k) \) zero matrix. Then the left-hand side is the canonical solution for the regulation matrix \( \begin{bmatrix} \Lambda & \Gamma \end{bmatrix} \).

3 RESULTS

3.1 Comparison of ssLMS, LEARNe and SVD

Here, we compare the performances of three algorithms, ssLMS, LEARNe and SVD for various settings. In each test, we compared multiple results of SVD for different choices of principal components (PCs) with the single result of LEARNe for the optimally chosen \( \mu \) (%).

3.1.1 Data generation and network reconstruction We generated a sparse and stable regulation matrix as well as a drift vector. Using this system, we generated time-series data incorporating both systemic (biological) and experimental noise using independent Gaussian distributions. We set the same variance for the systemic and experimental noise. We perturbed \( \sim30\% \) of the genes in the network using the drift term (Bansal et al., 2006). From the noisy data, we reconstructed the underlying networks using each of the three algorithms. We repeated the test for 40 randomly generated stable systems in each setting and averaged the following measure of performance (performance area). We compared the average values for each algorithm.

3.1.2 Measure of performance For a regulation matrix predicted by each algorithm, we ordered the elements of the matrix according to their absolute values. We counted how many non-zero \( r \)-factors were caught with the correct signs in a set of elements with high absolute values. This provided Sensitivity (TP/(TP + FN)) versus PPV [positive predictive value; TP/(TP + FP)] plots (say performance plot). Wider area under the plotted line (say performance area) indicates better performance.

3.1.3 Test results for simulated data Typical performance plots are shown in Figure 1. LEARNe significantly improved the prediction power of the naïve subset selection method (ssLMS), and also outperformed SVD. ssLMS was the least sensitive, and SVD exhibited varying performances depending on the number of PCs. When the numbers of data are increased to 30 and 50 (Fig. 1b and c), the performances of both ssLMS and LEARNe were improved considerably. For this larger numbers of data, LEARNe clearly outperformed SVD and showed the highest sensitivity.

The average-case test results are summarized in Figure 2. We show 10-dimensional cases, but similar patterns are observed for higher dimensional networks. For five time
as good as the best case of SVD; but from eight data points, LEARNe began to surpass the other algorithms (data not shown). The test results for 10 and 20 data points are demonstrated in Figure 2b and c, respectively. LEARNe clearly outperformed the other methods as the number of data increased (see also Fig. 1 b and c). ssLMS was still better than a random prediction (a line below 0.1, data not shown), though it was worst among the tested algorithms.

Overall, LEARNe is clearly better than ssLMS. It is also clearly better than SVD for larger numbers of data points, but is just a little better than SVD for small numbers of data points (from eight). However, LEARNe has another important advantage over SVD in the robustness of predictions as shown in the following:

### 3.2 Test of robustness

Here, we compare the robustness of LEARNe and SVD by varying their free parameters, $\mu$ and the number of PCs, respectively. We tested two cases for high-dimensional networks (Fig. 3a) and higher rates of experimental noise (Fig. 3b). We plotted multiple results of LEARNe and SVD simultaneously, which highlighted the superiority of LEARNe in both performance and robustness. In both cases, LEARNe provided similar good performances for every choice of $\mu$, while SVD varied relatively largely depending on the choice of the PCs. The best case of LEARNe was just a little better than that of SVD, but on average, LEARNe was clearly better.

### 3.3 Test of the algorithms for real data

We tested the algorithms for reconstructing two real regulatory networks. In the first example, we used the SOS regulatory network of *Escherichia coli* comprising nine genes (*lexA*, *recA*, *recF*, *rpoS*, *rpoD*, *rpoH*, *umuDC*, *dinI* and *SsB*). We used the five time-point data used by di Bernardo and coworkers (Bansal et al., 2006), which were generated with Norfloxacin treatment. For the detailed description of the SOS system and the experiments, see Gardner et al. (2003) and Bansal et al. (2006). For the 43 reported regulatory relationships examined by di Bernardo and coworkers, we measured the performances of the three algorithms (Fig. 4a). We used the raw data and did not apply interpolation or smoothing (Bansal et al., 2006).

Unlike the simulation tests shown above, we are not aware of the exact drift terms of real regulatory networks, and hence, we considered only $\Lambda$ that represents the 43 known connections and self-degradations (diagonal terms) for our validation. SVD with one PC exhibited good specificity, but was less sensitive than that with three PCs. According to the test results in Supplementary Figure 2, we chose $\mu = 100\%$ for LEARNe. In a simulation test, SVD is the best method for nine genes and five data points (data not shown), but LEARNe performed as well as SVD in this real network. LEARNe seems to perform even better than the average of the two SVD plots (Fig. 4a). ssLMS still performed worst among the algorithms.

In the second example, we reconstructed the cell cycle regulatory network of *Saccharomyces cerevisiae*. We used the regulatory network model provided by Young and coworkers (Simon et al., 2001), which is composed of 24 genes including eight transcription factors (see Fig. 4 of Simon et al., 2001).

---

**Figure 2.** Performance areas averaged upon 40 randomly generated linear systems of (a) 10 dimensions for 5 data points. Those of 10-dimensional networks for (b) 10 and (c) 20 time point data are also shown.
For our task, we used the 18 time points of the cell cycle expression data (Spellman et al., 1998) from yeast cultures that were synchronized by alpha factor arrest. We only considered genes included in the /C2480 cell cycle-regulated genes identified by Botstein and coworkers (Spellman et al., 1998), which removed four transcription factors from our list of genes. As a result, we have four transcription factors (ace2, fkh1, swi4 and swi5) and 16 target genes (sic1, cln3, far1, cln2, cln1, clb6, clb5, gin4, swe1, clb4, clb2, clb1, tem1, apc1 and cdc20) for our test.

The regulatory interactions between the transcription factors and their target genes were retrieved from the YEASTRACT database (Teixeira et al., 2006). From the reconstructed regulation matrix by each method, we captured the four-column sub-matrix corresponding to the four transcription factors and compared with known regulatory connections (see Table 1 for example).

As shown in Figure 4b, LEARNe performed best with \( \mu = 70 \sim 80\% \) and SVD performed best with eight PCs. In this example, however, the optimal value of \( \mu \) differed from the expected optimum (\( \sim 20–30\% \) for 18 time points) that is inferred from simulation tests. One possible reason for such a deviation is that the time-series data we used represent two periods of the cell cycle process, and hence nearly the same information is duplicated in the 18 time points. Indeed, when we used only the first nine data points, LEARNe performed best with \( \mu = 70\% \), which deviates from the expected optimum (60% for nine data points) only by 10%. In general, the optimal parameter for real data also depends on the unknown levels of noise and non-linearity such as varying time-delays among the regulatory relationships. This is why the robustness of the prediction on free parameters is important (Fig. 3). Notwithstanding, we want to emphasize that the best case of LEARNe was clearly better than that of SVD. We note that SVD is also encountered with the same problem and there is no established method for determining the optimal number of PCs in analyzing real data (Bansal et al., 2006).

Table 1 shows the reconstructed network with PPV = 0.7 and Sensitivity = 0.54, which means 30% of the connections are false positives and 46% of the true connections are not included...
4 DISCUSSION

We described a new system identification method for time-series expression data based on a difference equation model and an ensemble application of subset selection method. Ensemble learning is a very powerful machine-learning technique especially for classification problems (Dietterich, 2000). We developed a similar approach to multiple linear regression problems through voting the information from multiple likely models.

LEARNe exhibits clear advantages over previously used methods in the following aspects. First, it performs far better than the standard optimization technique by least mean square method. It also surpasses the widely used SVD except for the case of small number of data points (six or smaller). Its superiority is clear and consistent for higher dimensional networks and larger numbers of data points. This is a desired feature in the perspective of systems biology that aims to decipher the large intracellular networks thoroughly by analyzing large-scale data. Second, it provides more robust predictions than SVD with respect to noise. The performance of LEARNe slowly declines with the increase of noise while that of SVD with a large number of PCs falls rapidly. Third, it provides another kind of robustness with respect to the choice of free parameter. SVD and LEARNe have free parameters, the number of PCs and μ (the percentage of likely models to be merged), respectively. LEARNe is less affected in performance by the parameter than SVD especially for larger numbers of genes and higher levels of experimental noise (Fig. 3). This is an important advantage as a reverse-engineering algorithm, because the optimal parameter is hardly known a priori for real data.

As we used only expression profiles, false positive and true negative predictions are inevitable. False positives can be considerably reduced by incorporating biological knowledge for transcription networks. However, we admit the existence of true negatives because every regulatory relationship is not turned on under specific experimental conditions. Combined with biological knowledge, it would be an interesting future work to identify ‘activated’ regulatory relationships using reverse-engineering algorithms. With improved accuracy for many data points and the higher level of robustness to noise and parameter values, we expect LEARNe will provide useful predictions on key genetic relationships from time-series expression data.

ACKNOWLEDGEMENTS

The authors thank Drs D. di Bernardo and M. Bansal (Bansal et al., 2006) for providing their microarray data. This work was financially supported by the 21C Frontier Microbial Genomics and Applications Center Program, Ministry of Science and Technology, Republic of Korea. D.N. was partially supported by the Korea Research Council of Fundamental Science & Technology (KRCF), Grant No. C-RESEARCH-07-08-NIMS.

Conflict of Interest: none declared.

REFERENCES


