A STATE SPACE REPRESENTATION OF VAR MODELS WITH SPARSE LEARNING FOR DYNAMIC GENE NETWORKS

KANAME KOJIMA[1] kaname@ims.u-tokyo.ac.jp
RUI YAMAGUCHI[1] ruiy@ims.u-tokyo.ac.jp
SEIYA IMOTO[1] imoto@ims.u-tokyo.ac.jp
MAI YAMAUCHI[1] cyowako@ims.u-tokyo.ac.jp
MASAO NAGASAKI[1] masao@ims.u-tokyo.ac.jp
RYO YOSHIDA[2] yoshidar@ism.ac.jp
TEPPEI SHIMAMURA[1] shima@ims.u-tokyo.ac.jp
KAZUKO UENO[1] uepi@ims.u-tokyo.ac.jp
TOMOYUKI HIGUCHI[2] higuchi@ism.ac.jp
NORIKO GOTOH[1] ngotoh@ims.u-tokyo.ac.jp
SATORU MIYANO[1] miyano@ims.u-tokyo.ac.jp

[1] *Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan*
[2] *Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo, 106-8569, Japan*

We propose a state space representation of vector autoregressive model and its sparse learning based on L1 regularization to achieve efficient estimation of dynamic gene networks based on time course microarray data. The proposed method can overcome drawbacks of the vector autoregressive model and state space model; the assumption of equal time interval and lack of separation ability of observation and systems noises in the former method and the assumption of modularity of network structure in the latter method. However, in a simple implementation the proposed model requires the calculation of large inverse matrices in a large number of times during parameter estimation process based on EM algorithm. This limits the applicability of the proposed method to a relatively small gene set. We thus introduce a new calculation technique for EM algorithm that does not require the calculation of inverse matrices. The proposed method is applied to time course microarray data of lung cells treated by stimulating EGF receptors and dosing an anticancer drug, Gefitinib. By comparing the estimated network with the control network estimated using non-treated lung cells, perturbed genes by the anticancer drug could be found, whose up- and down-stream genes in the estimated networks may be related to side effects of the anticancer drug.

Keywords: state space model; VAR model; LASSO; dynamic gene networks.

1. Introduction

For statistical estimation of gene networks from time course microarray data, vector autoregressive (VAR) models have widely been used [4, 6, 8], but have some drawbacks. One is due to the experimental design of time course microarray data. That is, by considering the behaviors of gene expressions after dosing some shock, e.g., drug of interest, it is a usual case that the observed time points are not equally spaced [1, 11]. The assumption of VAR models does not fit this common

property of time course microarray data, i.e., VAR models assume equal time intervals in time course data. Also, VAR models do not distinguish observation noise and systems noise in their model equation. To extract reliable information from time course gene expression data, we need to separate observation noise from the gene expressions, otherwise the detection power of the method for gene regulations decreases. To overcome the drawbacks of VAR models, the use of state space model (SSM) was proposed [5, 12]. However, SSM considers the modularity of gene networks to reduce the dimension of the data to achieve stable parameter estimation, the modularity assumption is sometimes not suitable for estimation of gene networks; this decreases prediction power of the network structure when the true network does not have module property.

We propose to combine VAR model and SSM to utilize each advantage and cover each disadvantage and define a new statistical model called VAR-SSM. To use the advantage of VAR model, a simply derived EM algorithm for estimating VAR-SSM requires the computation of inverse matrices whose sizes are equal to the number of genes analyzed. This implies that the applicability of VAR-SSM is limited to small number of genes. Therefore, we also establish a new computational technique for EM algorithm to avoid the computation of the inverse matrices. This technique is essential to increase the effectiveness of VAR-SSM. In many cases, structure learnings of graphical models like Bayesian networks are computationally hard problems [2]. For learning VAR-SSM, we employ a sparse learning strategy based on L1 regularization [10]; this transfers the problem of structure learning into the choice of the value of hyperparameter, which can be optimized based on an information criterion.

The proposed method is used for a case-control study of human lung cells. As case cells, we use normal human lung cells treated by EGF and Gefitinib; the latter is an anticancer drug and known as a selective inhibiter of EGF receptors. As control cells, we use those cells without the treatment. Since Gefinitib is known as a selective inhibiter of EGF receptors, case condition ideally must be the same as control condition. However, gene expression profiles in those conditions are actually different due to the existence of off-targets of Gefitinib. The proposed method is applied for estimating gene regulatory networks from time course data in control and case cells, respectively, and we compare these two estimated networks. By focusing on the hub genes, we could unexpectedly find perturbed genes by Gefitinib, which have different expression profiles between two types of cells in early response time.

2. Methods for Estimating Dynamic Gene Networks

2.1. *Existing Methods*

Vector autoregressive model Given gene expression profile vectors of p genes during T time points $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\}$, the first order vector autoregressive (VAR(1)) model at time point t is given by:

$$\boldsymbol{y}_t = A\boldsymbol{y}_{t-1} + \boldsymbol{\varepsilon}_t,$$

where A is a $p \times p$ autoregressive coefficient matrix, and $\boldsymbol{\varepsilon}_t$ is observation noise at time t which is normally distributed with $N(0, \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2))$. The (i, j)th element of A, a_{ij}, indicates a temporal regulation from the jth gene to the ith gene, and thus, if $a_{ij} = 0$, no regulation is considered from the jth gene to the ith gene. By estimating A and checking if a_{ij} is zero for all i, j, regulatory network can be constructed. However, due to the assumption of the model, time points in the time course data must be equally spaced, which is unrealistic in actual biological data.

State space model Unlike VAR model, state space model (SSM) can deal with non equally spaced time course data. In SSM, the set of equally spaced entire T time points are defined as $\mathcal{T}$ and the set of time points where gene expressions are measured or observed are denoted as $\mathcal{T}_{obs}$.

Given gene expression profile vectors of p genes and $|\mathcal{T}_{obs}|$ time points $(\boldsymbol{y}_t)_{t \in \mathcal{T}_{obs}}$, SSM is represented by two equations: system model and observation model. System model is given by

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + \boldsymbol{w}_t,$$

where $\boldsymbol{x}_t$ is a k dimensional hidden variable vector at time t and A is a $k \times k$ autoregressive coefficient matrix. Observation model is given by

$$\boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{u}_t, t \in \mathcal{N}_{obs},$$

where C is a $p \times k$ matrix which is called observation matrix. $\boldsymbol{w}_t \sim N(0, R)$ and $\boldsymbol{u}_t \sim N(0, Q)$ are called observation noise and system noise, respectively, where Q is $\mathrm{diag}(q_1, \ldots, q_k)$ and R is $\mathrm{diag}(r_1, \ldots, r_p)$. In SSM, the system model describes the progress of the dynamic system in equally spaced time interval, while the observation model represents the generative process of observation data at intermittent time points, by which non-equally spaced time course data can be properly handled. In addition, although system noise and observation noise are distinguished in SSM, these are merged and cannot be distinguished in VAR model.

State space model as modularity network For estimating gene regulatory networks, the number of genes p is order of 10^2 to 10^4, but the currently available time course consists of < 100 time points. In this case, conventional parameter estimation methods for VAR, e.g., maximum likelihood estimation, may fail due to the overfitting. On the other hand, parameters of SSM can be estimated with a maximum likelihood estimation method by setting k to be much smaller than p. In that case, SSM is used as a dimension-reduction method. In the dimensional reduction, gene expressions $\boldsymbol{y}_t$ are regulated by a few latent factors $\boldsymbol{x}_t$ and the dynamic system is explained by $\boldsymbol{x}_t$. From the biological view point, these latent factors may represent unobserved activities of transcription factors which regulate transcriptions of downstream target genes, and genes regulated by the same latent factor are considered as a transcriptional module. Temporal regulatory relationships among the transcriptional modules are expressed by the system model, and thus

the matrix A represents a module network [5, 12]. Genes belonging to a module can be detected by statistical test or other machine learning techniques such as $L1$ penalization or greedy search based on several information criteria. Not only a module network, but also we can estimate a gene regulatory network from the parameters of SSM through a VAR expression of SSM given by

$$R^{-1/2}(\boldsymbol{y}_t - \boldsymbol{u}_t) = D'C'R^{-1}C\Lambda AD R^{-1/2}(\boldsymbol{y}_{t-1} - \boldsymbol{u}_{t-1}) + R^{-1/2}C\boldsymbol{w}_t, \qquad (1)$$

where $D = C'R^{-1}C^{-1}A'R^{-1/2}$. For details of the derivation of the VAR expression, see Hirose *et al.* [5]. However, accuracy of an estimated gene regulatory network probably becomes poor if module structures are not clearly embedded in true one. That property of SSM with dimension reduction may limit its applicability.

Here, Table 1 summarizes advantageous points and drawbacks of gene network estimation using VAR model and SSM, in which check marks indicate the advantageous ones; the both models have different and compensatory characteristics. In the next section, we propose a new probabilistic model that overcomes those drawbacks by combining the two models.

Table 1: Advantageous points and drawbacks of VAR model and SSM.

	Gene Network	Data	System noise and observation noise
VAR	√	Equal interval	Not distinguished
SSM	Modularity is assumed	√	√
Proposed	√	√	√

2.2. *State Space Representation of VAR Model (VAR-SSM)*

To address the drawbacks of VAR model and SSM given in Table 1, we propose a new probabilistic model, state space representation of VAR model (VAR-SSM) and develop its parameter estimation method with L1 regularization. VAR-SSM combines preferable properties of VAR and SSM as shown in Table 1. Like a usual SSM, VAR-SSM is given by the following system model and observation model:

$$\begin{aligned} \boldsymbol{x}_t &= A\boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \\ \boldsymbol{y}_t &= \boldsymbol{x}_t + \boldsymbol{u}_t, t \in \mathcal{T}_{obs}. \end{aligned}$$

A notable difference between usual SSM (i.e., SSM with dimensional-reduction) and VAR-SSM is that matrix C in observation model is set to be an identity matrix. Thus, the hidden variable vectors $\boldsymbol{x}_t$ are p dimensional and the size of A in the system model increases from $k \times k$ to $p \times p$. VAR-SSM can deal with non-equally spaced time course data and distinguish system noise and observation noise. In addition, no modularity is assumed in the network to be estimated. However, due

to the increase of the size of A, the estimation of A may fail in a usual manner. We thus consider sparse learning for the estimation of A.

2.3. *Parameter Estimation by EM algorithm with L1 Regularization*

An L1 regularized log likelihood of VAR-SSM is given by

$$\log \int \prod_{t=2}^{T} \frac{|Q|^{-1/2}}{\sqrt{2\pi}^p} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_t - A\boldsymbol{x}_{t-1})' Q^{-1}(\boldsymbol{x}_t - A\boldsymbol{x}_{t-1})\right\} \prod_{t\in\mathcal{T}_{obs}} \frac{|R|^{-1/2}}{\sqrt{2\pi}^p} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_t - \boldsymbol{x}_t)' R^{-1}(\boldsymbol{y}_t - \boldsymbol{x}_t)\right\} d\boldsymbol{x}_1 \ldots d\boldsymbol{x}_T + \sum_{i=1}^{p}\sum_{j=1}^{p} \lambda_i |a_{ij}|, \tag{2}$$

where A, Q, and R are parameters to be optimized, and λ_i is the L1 regularization parameter. Since it is difficult to estimate parameters by optimizing Equation (2), we use EM algorithm to estimate the parameters of VAR-SSM model. In EM algorithm, E-step and M-step are repeated to update the values of the parameters so that the log-likelihood converges.

E-step Following expectations are required for the parameter estimation in M-step.

$$\begin{aligned} E[\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T] & \\ E[\boldsymbol{x}_t\boldsymbol{x}_t'|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T] &= Var[\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T] \\ &\quad + E[\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]E[\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]' \\ E[\boldsymbol{x}_t\boldsymbol{x}_{t-1}'|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T] &= Cov[\boldsymbol{x}_t\boldsymbol{x}_{t-1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T] \\ &\quad + E[\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]E[\boldsymbol{x}_{t-1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]'. \end{aligned} \tag{3}$$

M-step The elements in A are estimated by solving the following equation:

$$\arg\min_{\boldsymbol{a}_i} \left\{ \boldsymbol{a}_i' U \boldsymbol{a}_i + \boldsymbol{z}'\boldsymbol{a}_i + \lambda_i \sum_{j=1}^{p} |a_{ij}| \right\},$$

where $\boldsymbol{a}_i$ is transpose of the ith row vector of A, i.e., $A = (\boldsymbol{a}_1,\ldots,\boldsymbol{a}_p)'$ and $U = \sum_{t=1}^{T-1} E[\boldsymbol{x}_t\boldsymbol{x}_t'|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]$, and $\boldsymbol{z}$ is the ith row vector of $-2\sum_{t=2}^{T} E[\boldsymbol{x}_t\boldsymbol{x}_{t-1}'|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]$. The ith diagonal element of Q, q_i can be obtained by

$$q_i = \frac{1}{T-1}\left(\hat{\boldsymbol{a}}_i' U \hat{\boldsymbol{a}}_i + \boldsymbol{z}'\hat{\boldsymbol{a}}_i + v\right),$$

where $\hat{\boldsymbol{a}}_i$ is the estimated $\boldsymbol{a}_i$ in the above equation, and v is the (i,i)th element of $\sum_{t=2}^{T} E[\boldsymbol{x}_t\boldsymbol{x}_t'|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]$. Similarly, the ith diagonal element of R, r_i can be obtained by the (i,i) th element of

$$\frac{1}{T}\sum_{t\in\mathcal{T}_{obs}} \left[\boldsymbol{y}_t\boldsymbol{y}_t' - 2\boldsymbol{y}_t E[\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]' + E[\boldsymbol{x}_t\boldsymbol{x}_t'|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T]\right].$$

2.3.1. *Kalman filter for E-step*

In E-step, the expectation values in Equation (3) are calculated by Kalman Filter efficiently through the following three steps: prediction, filtering, and smoothing. Here, $Var[\boldsymbol{x}_t]$ given $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ is denoted by $\Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_{1:s}}$, $E[\boldsymbol{x}_t]$ given $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ is denoted by $\boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_{1:s}}$, and $Cov[\boldsymbol{x}_t, \boldsymbol{x}_{t-1}]$ given $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ is denoted by $\Sigma_{\boldsymbol{x}_t,\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:s}}$ Prediction, filtering, and smoothing are calculated by the following recursive formulas:

- Prediction:

$$\begin{aligned}\Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}} &= A\Sigma_{\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}}A' + Q,\\ \boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}} &= A\mu_{\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}}.\end{aligned}$$

- Filtering:

$$\begin{aligned}\Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t}} &= (C'R^{-1}C + \Sigma^{-1}_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}})^{-1},\\ \boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t}} &= \mu_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}} + \Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t}}C'R^{-1}(\boldsymbol{y}_t - C\boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}}).\end{aligned} \tag{4}$$

- Smoothing:

$$\begin{aligned}J_t &= \Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_t}A'\Sigma^{-1}_{\boldsymbol{x}_{t+1}|\boldsymbol{y}_1:\boldsymbol{y}_t},\\ \Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_T} &= \Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_t} + J_t(\Sigma_{\boldsymbol{x}_{t+1}|\boldsymbol{y}_1:T} - \Sigma_{\boldsymbol{x}_{t+1}|\boldsymbol{y}_1:\boldsymbol{y}_t})J'_{t-1},\\ \Sigma_{\boldsymbol{x}_t,\boldsymbol{x}_{t-1}|\boldsymbol{y}_1:\boldsymbol{y}_T} &= \Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_t}J'_{t-1} + J_t(\Sigma_{\boldsymbol{x}_{t+1},\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_T} - A\Sigma_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_t})J'_{t-1},\\ \boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_T} &= \boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_t} + J_t(\boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_T} - A\boldsymbol{\mu}_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_t}),\end{aligned} \tag{5}$$

  where $\Sigma_{\boldsymbol{x}_T,\boldsymbol{x}_{T-1}|\boldsymbol{y}_1:\boldsymbol{y}_T}$ is initialized by

$$\Sigma_{\boldsymbol{x}_T,\boldsymbol{x}_{T-1}|\boldsymbol{y}_1:\boldsymbol{y}_T} = (I - \Sigma_{\boldsymbol{x}_T|\boldsymbol{y}_1:\boldsymbol{y}_T}C^TR^{-1}C)A\Sigma_{\boldsymbol{x}_{T-1}|\boldsymbol{y}_1:\boldsymbol{y}_{T-1}}.$$

Equations (4) and (5) require calculation of $p \times p$ inverse matrix, which is computationally heavy task and numerically unstable as p is order of 10^2 to 10^4. In order to avoid these inverse matrix calculations, we derived a recursive formula in the next section.

2.3.2. *Recursive Formula*

Given $n \times n$ matrix A, $m \times n$ matrix B, and $m \times m$ diagonal matrix $\Delta = \mathrm{diag}(\delta_1, \ldots, \delta_m)$, we consider a matrix formula $(A^{-1} + B^T\Delta B)^{-1}$ for computing. Let $A_0 = A$. By calculating the recursive formula:

$$A_{i+1} = A_i - \frac{1}{1/\delta_{i+1} + \boldsymbol{b}'_{i+1}A_i\boldsymbol{b}_{i+1}}A_i\boldsymbol{b}_{i+1}\boldsymbol{b}'_{i+1}A'_i,$$

where $\boldsymbol{b}_i$ is the ith row vector of B, and $(A^{-1} + B'\Delta B)^{-1}$ is given by A_n.

Proof. From blockwise matrix inversion theorem in Matrix Analysis for Statistics [7] p. 9, we have

$$\begin{aligned} A_{i+1} &= A_i - \frac{1}{1/\delta_{i+1} + \boldsymbol{b}'_{i+1} A_i \boldsymbol{b}_{i+1}} A_i \boldsymbol{b}_{i+1} \boldsymbol{b}'_{i+1} A'_i \\ &= (A_i^{-1} + \delta_{i+1} \boldsymbol{b}_{i+1} \boldsymbol{b}'_{i+1})^{-1} \\ &= (A_i^{-1} + \sum_{j=1}^{i+1} \delta_j \boldsymbol{b}_j \boldsymbol{b}'_j)^{-1}. \end{aligned}$$

Since $B'\Delta B = \sum_{i=1}^{n} \delta_i \boldsymbol{b}_i \boldsymbol{b}'_i$, $(A^{-1} + B'\Delta B)^{-1}$ is given by A_n. □

Since $\Sigma^{-1}_{\boldsymbol{x}_{t+1}|\boldsymbol{y}_1:\boldsymbol{y}_t}$ in Equation (5) can be given by

$$\Sigma^{-1}_{\boldsymbol{x}_{t+1}|\boldsymbol{y}_1:\boldsymbol{y}_t} = A'(Q^{-1} - Q^{-1}A(A'Q^{-1}A + \Sigma^{-1}_{\boldsymbol{x}_t|\boldsymbol{y}_1:\boldsymbol{y}_t})^{-1}A'Q^{-1}),$$

Equations (4) and (5) can be calculated by the recursive formula without inverse matrix calculation.

2.3.3. *L1 Regularized Estimation in M-Step*

In M-step, parameters in A are estimated by minimizing

$$\arg\min_{\boldsymbol{a}_i} \left\{ \boldsymbol{a}'_i U \boldsymbol{a}_i + \boldsymbol{z}_i \boldsymbol{a}_i + \lambda_i \sum_{j=1}^{p} |a_{ij}| \right\}, \tag{6}$$

where $A = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p)'$ and $U = \sum_{t=1}^{T-1} E[\boldsymbol{x}_t \boldsymbol{x}'_t | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T]$, and $\boldsymbol{z}_i$ is transpose of the ith row vector of $-2\sum_{t=2}^{T} E[\boldsymbol{x}_t \boldsymbol{x}'_{t-1} | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T]$. For the better performance of the variable selection, weights $w_1, \ldots, w_p$ are given to the regularization term, which is known as weighted LASSO [9]:

$$\arg\min_{\boldsymbol{a}_i} \left\{ \boldsymbol{a}'_i U \boldsymbol{a}_i + \boldsymbol{z}' \boldsymbol{a}_i + \lambda_i \sum_{j=1}^{p} w_j |a_{ij}| \right\}. \tag{7}$$

We set the weight w_j as $\log(|a_{ij}| + 1 + \zeta)$, where ζ is some small value. Note that by using $W = \mathrm{diag}(w_1, \ldots, w_p)$, optimization problem in Equation (7) can be given as

$$\arg\min_{\boldsymbol{a}_i} \left\{ \boldsymbol{a}'_i W^{-1} U W^{-1} \boldsymbol{a}_i + \boldsymbol{z}' W^{-1} \boldsymbol{a}_i + \lambda_i \sum |a_{ij}| \right\}. \tag{8}$$

Since the form of Equation (8) is the same as that of Equation (6), for brevity, optimization of Equation (6) is shown in the next section.

2.3.4. *LARS-LASSO for VAR-SSM*

In a linear regression problem $y = X\beta + \varepsilon$, where y is the vector of responses, X is the design matrix and ε is the noise. For the estimation of β, LASSO [10] solves

the following optimization problem:

$$\arg\min_{\boldsymbol{\beta}} \left\{ (\boldsymbol{y} - X\boldsymbol{\beta})'(\boldsymbol{y} - X\boldsymbol{\beta}) + \lambda_i \sum_i |\beta_i| \right\}. \tag{9}$$

The exact solution of Equation (9) can be obtained by the LARS-LASSO algorithm [3] in an efficient manner, but the LARS-LASSO algorithm is limited to the equation given by Equation (9) and thus cannot solve the optimization of Equation (6) for VAR-SSM. Thus, we derived a new LARS-LASSO algorithm for M-step of VAR-SSM. We first define a function $f(\boldsymbol{a}_i)$ as

$$f(\boldsymbol{a}_i) = \boldsymbol{a}_i' U \boldsymbol{a}_i + \boldsymbol{z}' \boldsymbol{a}_i + \lambda_i \sum_j |a_{ij}|.$$

Let $\mathcal{A}$ be an active set of indices, i.e., the set of indices whose corresponding coefficient is not zero. From Karush-Kuhn-Tucker conditions, we have

$$\frac{d}{d\boldsymbol{a}_{i,\mathcal{A}}} f(\boldsymbol{a}_i) = 2U_{\mathcal{A}} \boldsymbol{a}_{i,\mathcal{A}} + \boldsymbol{z}_{\mathcal{A}} + \lambda_i \boldsymbol{s}_{\mathcal{A}} = \boldsymbol{0}, \tag{10}$$

where $\boldsymbol{a}_{i,\mathcal{A}} = (a_{ij})'_{j\in\mathcal{A}}$, $\boldsymbol{z}_{\mathcal{A}} = (z_j)'_{j\in\mathcal{A}}$, $U_{\mathcal{A}} = (u_{jk})_{j,k\in\mathcal{A}}$, and $\boldsymbol{s}_{\mathcal{A}} = (\mathrm{sgn}(a_{ij}))'_{j\in\mathcal{A}}$, and thus $\boldsymbol{a}_{i,\mathcal{A}}$ can be given by

$$\boldsymbol{a}_{i,\mathcal{A}} = -\frac{1}{2} U_{\mathcal{A}}^{-1} \left(\boldsymbol{z}_{\mathcal{A}} + \lambda_i \boldsymbol{s}_{\mathcal{A}} \right).$$

As far as $\mathcal{A}$ does not change, Equation (10) holds if λ changes continuously. Thus, if λ_i decreases by γ and $\mathcal{A}$ does not change, $\boldsymbol{a}_i$ is given by

$$\boldsymbol{a}_{i,\mathcal{A}}(\lambda_i + \gamma) = \boldsymbol{a}_{i,\mathcal{A}}(\lambda_i) + \frac{\gamma}{2} U_{\mathcal{A}}^{-1} \boldsymbol{s}_{\mathcal{A}}. \tag{11}$$

Thus, when decreasing λ_i from λ_{max}, we need to focus on the point where $\mathcal{A}$ changes. We suppose that $\mathcal{A}$ changes at $\tilde{\lambda}_j$. From Equation (10), the following elementwise equation also holds:

$$|2U_{\mathcal{A}} \boldsymbol{a}_{i,\mathcal{A}} + \boldsymbol{z}_{\mathcal{A}}| = \lambda_i. \tag{12}$$

If the index $j \notin \mathcal{A}$ is added when $\tilde{\lambda}_j$ decreases by γ_j, from Equation (12), $|z_j| = \tilde{\lambda}_j - \gamma_j$ holds. On the other hand, if the index $j \in \mathcal{A}$ is deleted, i.e., $a_j(\tilde{\lambda}_i - \gamma_j) = 0$, from Equation (11), $2a_{ij}/o_j = \gamma_j$ holds, where o_j is an element of $(1/2) \cdot U_{\mathcal{A}}^{-1} \boldsymbol{s}_{\mathcal{A}}$ corresponding to the index j. Since $\mathcal{A}$ actually changes at the closest point satisfying the above cases to $\tilde{\lambda}_i$, the point where $\mathcal{A}$ changes is $\tilde{\lambda}_i - \gamma$, where γ is defined by

$$\gamma = \min\{\gamma_{na}, \gamma_a\},$$

where γ_{na} is given by

$$\gamma_{na} = \min_{j\notin\mathcal{A}} \gamma_j > 0 \quad \text{s.t.} \quad |z_j| = \tilde{\lambda}_i - \gamma_j,$$

and γ_a is given by

$$\gamma_a = \min_{j\in\mathcal{A}} \frac{2a_{ij}}{o_j} > 0.$$

From Equation (12), the maximum value $\tilde{\lambda}_i$ can take is given by $\max_j |z_j|$ when $\mathcal{A} = \emptyset$. Hence, beginning from $\mathcal{A} = \emptyset$ and $\tilde{\lambda}_i = \max_j |z_j|$, we can obtain $\boldsymbol{a}_i$ by computing the above procedure until $\tilde{\lambda}_i$ reaches some specified λ_i.

3. Numerical Experiments

We apply VAR-SSM, VAR model, and SSM with dimension reduction to simulated datasets to evaluate performances of the proposed method. To generate simulation datasets, a directed scale free network of 100 nodes and 150 edges is generated and self regulation edges are added to root nodes of the network. Finally, we obtain the network of 181 edges which includes 31 self regulation edges. The AR coefficients are uniformly assigned to the edges from $\{-0.9, -0.8, -0.7, -0.6, -0.5, 0.5, 0.6, 0.7, 0.8, 0.9\}$. System noise $\boldsymbol{w}_t$ and observation noise $\boldsymbol{u}_t$ are set to follow $N(0, I)$ and $N(0, 0.2I)$, respectively. In the following experiments, the three network estimation methods are applied to 100 datasets generated from the above network and averaged results are used for comparison.

3.1. *Comparison to VAR*

We compare performances of VAR-SSM and VAR model by applying to simulated datasets in which non-equally spaced time course data are generated by the following procedures:

- Prepare equally spaced time course data of 100 time points from the scale free network model.
- The first 40 points are directly used, every two points is left in the next 30 points, and every three point is left in the final 30 points. Thus, no-equally spaced time course data of 65 time points are used.

The numbers of true positives, false positives, true negatives, and false negatives for estimated networks by VAR model and VAR-SSM are summarized in Table 2. Since the elements of the coefficient $p \times p$ matrix in VAR model should be estimated and determined whether zero or not, like VAR-SSM we use LASSO for sparse learning.

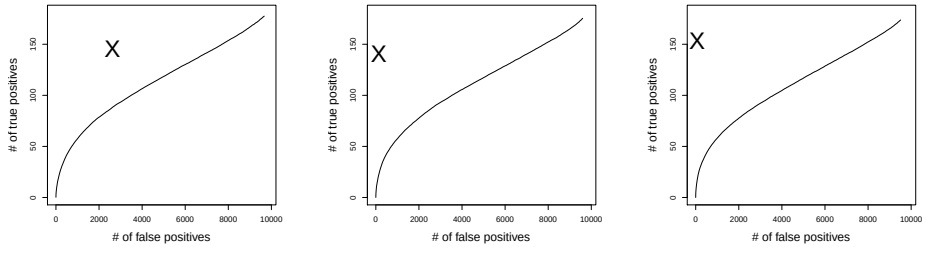
Table 2: The numbers of true positives (# TP), false positives (# FP), true negatives (# TN), and false negatives (# FN) of VAR and VAR-SSM.

	# TP	# FP	# TN	# FN
VAR (LASSO)	138	1851	43	7968
VAR-SSM	111	47	70	9772

3.2. *Comparison to Usual SSM*

We next compare the performances of VAR-SSM and SSM with dimension reduction. To estimate a gene regulatory network from usual SSM, we utilize the VAR representation of usual SSM given by Equation (1). Although Hirose *et al.* [5] applied permutation test to detect significant edges in gene regulatory networks, it is computationally heavy task. Thus, instead of the permutation test, we select edges by a threshold value for absolute values of the estimated AR coefficients in Equation (1). Thus, by changing the threshold value, ROC like curves can be drawn, in which vertical axis indicates the number of true positive edges, while horizontal axis indicates the number of false positive edges.

Figure 1 gives the ROC like curves of SSM for time course data of 40, 60, 100 time points. A point represented by cross in each ROC like curve indicates the results of VAR-SSM. Since, in the ROC like curves, left upper area indicates better performance, the results suggest that VAR-SSM is better than usual SSM. In addition, for longer time course data, false positives of VAR-SSM are drastically reduced, while its true positives seem saturated and do not change a lot. Also the results of SSM are essentially unchanged if the number of time points increases. We guess the reason is that the performance of SSM was saturated when 40 time points were used. This saturation might be due to the true network structure; the true network structure does not follow the modularity assumption in SSM.



40 time points 60 time points 100 time points

Fig. 1: ROC like curves of usual SSM for time course data of 40, 60, and 100 time points. The numbers of true positives and false positives of VAR-SSM are indicated by cross

3.3. *Real Data Application*

We apply VAR-SSM to time course gene expression data of normal human Small Airway Epithelial Cells (SAECs), which are lung cells. In lung cancer cells, EGF receptors (EGFRs) are often overexpressed. By stimulating EGFRs in SAEC with EGF, lung cancer cells might be simulated. An anticancer drug, Gefitinib is known

as a selective inhibiter of EGFRs. Thus we can expect that SAECs dosed with both EGF and Gefitinib are ideally in the same state of SAECs without the stimulation. However, we observed that some gene expression patterns were different between the two conditions. These genes may represent unknown action mechanisms of Gefitinib and relate to side effects. Gene regulatory networks constructed from these data probably give some insights about the above points. We, thus, employ SAECs dosed with EGF and Gefitinib as case cells and SAECs without the stimulation as control cells and consider a case-control study based on network viewpoint.

We first estimate gene regulatory networks by applying VAR-SSM to time course gene expression data of the case and those of the control for 100 genes, respectively. These 100 genes are selected as follows:

- 500 genes with large variation coefficients are learnt by SSM.
- Well learnt 100 genes with respect to p-value defined in [11] are selected.

The time course data consist of 19 time points during 48 hours and are not equally spaced. We next focus on hub genes in estimated networks and analyze how many edges connected to each of the genes in both networks. Hereafter, the number of edges connected to a node is referred to as degree. We calculate how much degree from the control network to the case network increases for each of the genes and rank the genes with respect to the increased degree. Top six genes in the ranking are listed in Table 3. Differences of expression profiles of the top genes between the case and control cells are shown in Figure 2. Many of the top genes are differentially expressed between control cell and case cell in the beginning stage and are considered as perturbed genes by Gefitinib. We consider that genes around these perturbed genes in the estimated case network may be off-targets of Gefitinib and related to the side effects of Gefitinib.

Table 3: Ranking of genes with respect to increased degree

Gene	Degree in Control	Degree in Case	Increased Degree
FOS	3	12	9
SOCS3	2	10	8
LRFN1	21	29	7
KLF2	4	11	7
SNX26	2	9	7
BMF	3	9	6

4. Conclusion

In this study, we proposed a new probabilistic model, VAR-SSM and showed its parameter estimation method using a sparse learning technique. Efficient calculation technique for recursive formula in E-step and a new LARS algorithm for M-step enable the parameter estimation. Simulation studies using time course data from a

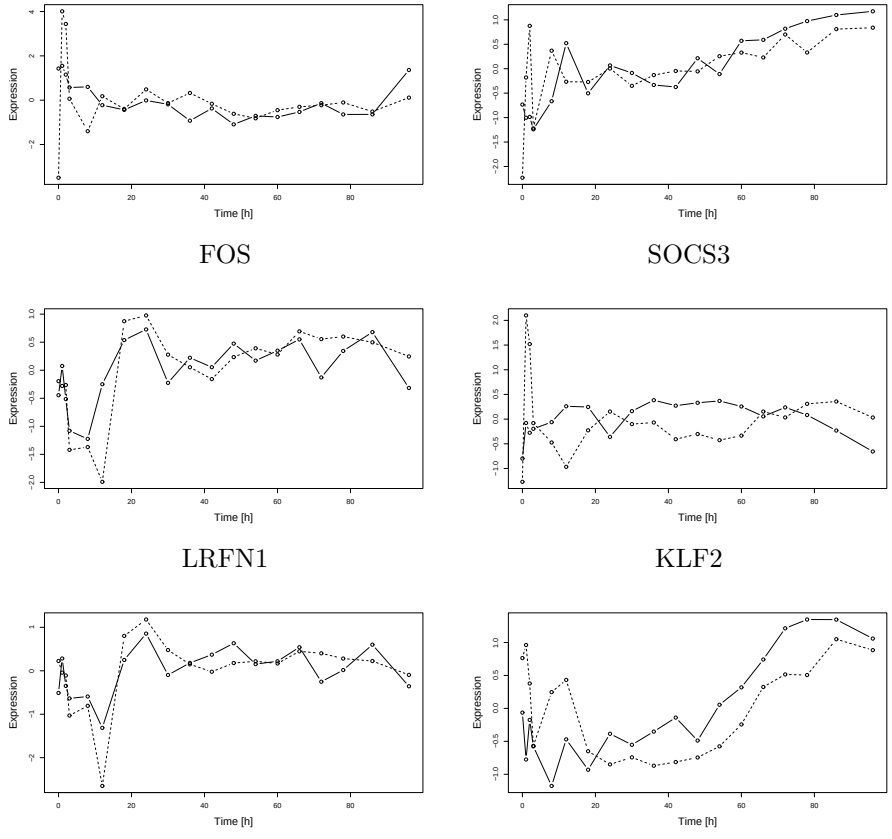


Fig. 2: Expression profiles of top genes in ranking given by Table 3. Expression profiles from control cell and case cell are indicated by solid and dashed lines, respectively.

scale free network shows that the proposed model outperforms VAR model and SSM with dimension reduction. For a real data application, VAR-SSM is used to estimate gene regulatory networks using time course data from human normal lung cells treated by EGF and Gefitinib and non-treated cells. From the estimated networks, we could find candidates of perturbed genes by Gefitinib. By investigating genes around candidate perturbed genes in the estimated networks, we would like to elucidate factors of the side effect of Gefitinib and their mechanisms in future work.

References

[1] Bar-Joseph, Z., Gerber, G., Simon, I., Gifford, D., Jaakkola, T., Comparing the continuous representation of time-serise expression profiles to identify

differentially expressed genes, *Proceedings of the National Academy of Science*, 100(18):10146–10151, 2003.

[2] Chickering, D.M., Heckerman, D., Meek, C., Large-sample learning of Bayesian neworks is NP-hard. *Journal of Machine Learning Research*, 15:1287–1330, 2004.

[3] Efron, B., Hastie, T., Johnstone, J., Tibshirani, R., Least angle regression, *Annals of Statistics*, 32(2):407–499, 2004.

[4] Fujita, A., Sato, J.R., Garay-Malpartida, H.M., Morettin, P.A., Yamaguchi, R., Miyano, S., Sogayar, M.C., Ferreira, C.E., Modeling gene expression regulatory networks with the sparse vector autoregressive model, *BMC Systems Biology*, 1:39, 2007.

[5] Hirose, O., Yoshida, R., Imoto, S., Yamaguchi, R., Higuchi, T., Stephen, D., Chamock-Jones, C., Miyano, S., Statistical inference of transcriptional module-based gene networks fro time course gene expression profiles by using state space models, *Bioinformatics*, 24(7):932–942, 2008.

[6] Kojima, K., Fujita, A., Shimamura, T., Imoto, S., Miyano, S., Estimation of nonlinear gene regulatory networks via L1 regularized NVAR from time series gene expression data, *Genome Informatics*, 20:37–51, 2008.

[7] Schott, J.R., *Matrix Analysis for Statistics*, John Wiley & Sons, Inc., 1997.

[8] Shimamura, T., Imoto, S., Yamaguchi, R., Fujita, A., Nagasaki, M., Miyano, S., Recursive elastic net for inferring large-scale gene networks from time course microarray data, *BMC Systems Biology*, 3:41, 2009.

[9] Shimamura, T., Imoto, S., Yamaguchi, R., Miyano, S., Weighted lasso in graphical Gaussian modeling for large gene network estimation based on microarray data, *Genome Informatics*, 19:142-153, 2007.

[10] Tibshirani, R., Regression shrinkage and selection via the lasso, *Journal of Royal Statistical Scociety: B*, 5(1):267–288, 1996.

[11] Yamaguchi, R. Imoto, S., Yamauchi, M., Nagasaki, M, Yoshida, R., Shimamura, T., Hatanaka, Y., Ueno, K., Higuchi, T., Gotoh, N., Miyano, S., Predicting differences in gene regulatory systems by state space models, *Genome Informatics*, 21:101-113, 2008.

[12] Yamaguchi, R. Yoshida, R., Imoto, S. Higuchi, T. Miyano, S., Finding module-based gene networks with state-space models-Mining high-dimensional and short time-course gene expression data, *IEEE Signal Processing Magazine*, 24(1):37-46, 2007.