Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

Active Learning for Network Estimation

Shotaro Akaho
Neuroscience Research Institute
AIST
Tsukuba, Ibaraki 305-8568 Japan
Email: s.akaho@aist.go.jp

Kenji Fukumizu
Institute of Statistical Mathematics
Tokyo 106-8569 Japan
Email: fukumizu@ism.ac.jp

***Abstract*— We address the problem of estimating the structure of networks described as a system of differential equations. In each experiment, the network's steady state is measured as an output depending on a controllable input. Due to the high cost of experiments, it is crucial to actively design the inputs for accurate estimation. Although standard active learning methods are designed to minimize the entropy of parameter distributions, it is very unstable to estimate the entropy of network structure. Therefore, we propose the two step algorithm as follows: first, the most uncertain link is chosen, and then the input is designed so as to minimize the variance of system equation parameter instead of network structure. Our method is tested in simulation experiments of gene networks following Yeung et al., PNAS (2002). We show that our algorithm gives stable and computationally effective solution.**

I. Introduction

Learning and inference on networks are important issues in various fields such as biological data analysis[1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. We discuss the problem of estimating a network structure based on identification of dynamical systems. We focus especially on the estimation of gene regulatory networks from microarray experiments. Experiments involved in the gene network identification are very time and cost consuming, while we need a number of experiments for an accurate estimate of network architecture. Thus, it is important to develop an efficient strategy for the design of experiments to achieve higher accuracy in a smaller number of experiments.

Active learning is a useful method of providing an effective design for statistical inference. It aims at optimizing controllable variables to maximize the usefulness of the next samples in learning. Active learning methodology has been successfully used for neural networks[13], [14], [15], in which the purpose is to design an input point x_{new} to obtain a new pair of training sample (x_{new}, y_{new}), which is expected to be the most effective for reducing the generalization error. In statistical literatures, active learning is called optimal experimental design[16], and has been used mainly for linear regression and simple nonlinear regression models.

The purpose of this paper is to propose a method of active learning for network estimation, which is motivated by inference of a gene regulatory network. Our method is based on statistical modeling of Gardner's method[17], which is based on the system identification assuming the model of a linear differential equation for the gene network. The method perturbs the system by providing an extraneous control input in each experiment, and observes the steady state of the system after convergence. The pairs of the control input and the steady state are used for training samples. Our active learning methods, thus, aims at providing effective control inputs in the sequential experiments repeating the process of design, experiment, and estimation.

In traditional active learning [13], the principle of choosing the input is to minimize the sum of uncertainty of all parameters. Typically, the uncertainty is measured by entropy. However, in network estimation with a small number of examples, it is not a good idea because of several reasons. One reason is that the entropy estimation from small samples is inaccurate for network structure as shown later. Another reason is that the traditional approach attempts to reduce the uncertainty uniformly over all parameters by just one input, which is very difficult in general. In practice it is effective to *focus* on the most uncertain parameters and reduce their uncertainty. We propose a new method to identify the parameters of high uncertainty and select a new input that is optimal for them. We use the entropy estimation of network structure just for focusing, and we minimize the entropy of system equation parameters instead. Our method was tested in numerical experiments using standard simulation models of gene networks[18], and we obtained promising results for applications toward real biological systems.

II. Dynamical model and network estimation problem

We briefly review the problem of estimating a network based on a linear system model[17], [19] (Figure 1). In fact, the network dynamics is highly nonlinear, but a linear approximation around the stationary point is often used in practice. To be specific, we will describe the problem based on the gene network, but it is applicable for other types of networks such as biochemical networks[19].

Suppose we have n species of genes, we approximate dynamic interaction between genes by a linear differential equation,

$$\frac{d\boldsymbol{x}(t)}{dt} = A\boldsymbol{x}(t) + \boldsymbol{u}(t) + \boldsymbol{\xi}(t), \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the concentration of the mRNAs that reflect the expression levels of the genes, $\boldsymbol{u}(t) \in \mathbb{R}^n$ is an external input, and $\boldsymbol{\xi}(t) \in \mathbb{R}^n$ is a noise. The $n \times n$ transition

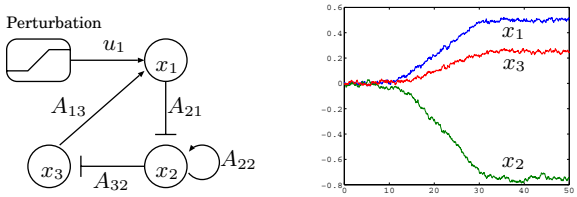

Fig. 1. Schematic figure of dynamical system of gene networks. Left: example of three gene network. Right: dynamics (outputs are measured after they converged to the steady states)

matrix A representing interactions between the genes is determined from gene network as follows: Let $V = \{1, \ldots, n\}$ be a set of nodes representing genes. Any directed graph over V can be represented by $n \times n$ binary matrix G called connectivity matrix, where $G_{ai} = 1$ when there is an edge from node i to node a, and $G_{ai} = 0$ otherwise. A transition matrix A associated with G is an $n \times n$ matrix such that $A_{ai} = 0$ for $G_{ai} = 0$. The set of transition matrices associated with G is denoted by $\mathcal{M}_G$.

Assume that the network is at a stationary state. To get information about the network, external inputs are raised from zero to specific values $\boldsymbol{u}_\nu$ and kept constant for a while. Then, the network moves to another stationary state (Figure 1). After convergence, mRNA conservations are measured as $\boldsymbol{x}_\nu$. Repeating the experiment N-times, we have the following linear equations,

$$A\boldsymbol{x}_\nu + \boldsymbol{u}_\nu \approx 0 \qquad (\nu = 1, \ldots, N). \tag{2}$$

because the network is at the steady state ($d\boldsymbol{x}(t)/dt \approx 0$). Our goal is to estimate the gene network G from X^N and U^N as accurately as possible, where we introduced the notation $X^N = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^\top, U^N = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N)^\top$. Especially, we design U^N by an active learning scheme.

III. Active learning framework for network estimation

In this section, we describe an active learning framework that we apply to network estimation. In the subsections below, first we introduce an entropy minimization criterion as an ideal but intractable one. Instead of the entropy criterion, we introduce a variance criterion which corresponds to the entropy of system parameter A_{ai}. We also propose a method to focus on a specific link to make the active learning effectively. Overall form of the algorithm will be summarized in Fig.3.

A. Entropy minimization

When we have already had N pairs of samples U^N and X^N, the basic procedure of active learning is designing a new input $\boldsymbol{u}$. One of the reasonable criteria is to choose $\boldsymbol{u}$ that is expected to minimize an uncertainty of the target variable G. Using entropy as a measure of uncertainty, we can define a naive form of cost function,

$$F_0(\boldsymbol{u}) = \int H(G; X^N, U^N, \boldsymbol{x}, \boldsymbol{u}) p(\boldsymbol{x} \mid X^N, U^N, \boldsymbol{u}) d\boldsymbol{x} \tag{3}$$

which should be minimized with respect to $\boldsymbol{u}$, where $H(G; Z)$ denotes the entropy of G with Z fixed, $H(G; Z) = -\sum_G p(G \mid Z) \log p(G \mid Z)$. This criterion is equivalent to maximize the expected mutual information between G and a new pair of sample $(\boldsymbol{u}, \boldsymbol{x})$.

To define the entropy precisely, let us define the generative model of the network. Elements of graph G are generated independently from binomial distribution, $G_{ai} = 0$ in probability p_0 and $G_{ai} = 1$ otherwise. Next, each element A_{ai} is generated from Gaussian distribution depending on $G_{ai} \in \{0, 1\}$,

$$p(A_{ai} \mid G_{ai} = g) = \exp\left\{-A_{ai}^2/(2\tau_g^2)\right\} / \sqrt{2\pi\tau_g^2},$$

where τ_0^2, τ_1^2 are underlying parameters ($\tau_0^2 \ll \tau_1^2$). Since the active learning algorithm does not rely on the detailed form of the generative model, we can modify the model depending on specific knowledge of target problems.

Next, we model the relation Eq. (2) statistically by

$$X_\nu + BU_\nu \sim N(0, \sigma^2 I_n), \tag{4}$$

where σ^2 is an underlying parameter representing noise level, and $B = A^{-1}$ under the assumption that A is invertible (in order to deal with the case that this assumption is violated, we take generalized inverse and further apply a regularization technique in estimation). Although it may seem to be natural that the left hand side of Eq. (2) is modeled by Gaussian distribution without introducing B, we would face difficulty in dealing with the distribution of $\boldsymbol{x}$ given A and $\boldsymbol{u}$. The formulation Eq. (4) makes it much easier.

Here we approximate $F_0(\boldsymbol{u})$ by assuming that conditional distributions given pairs of samples depend on the samples only through the estimation of A, i.e., in Eq. (3),

$$p(G \mid X^N, U^N, \boldsymbol{x}, \boldsymbol{u}) \approx p(G \mid \widehat{A}(X^N, U^N, \boldsymbol{x}, \boldsymbol{u})), \tag{5}$$

and

$$p(\boldsymbol{x} \mid X^N, U^N, \boldsymbol{u}) \approx p(\boldsymbol{x} \mid \widehat{A}(X^N, U^N), \boldsymbol{u}), \tag{6}$$

where $\widehat{A}(X^N, U^N, \boldsymbol{x}, \boldsymbol{u})$ is A estimated by using $X^N, U^N, \boldsymbol{x}, \boldsymbol{u}$. This assumption means that the posterior distribution of A given samples, when we regard A as a random variable, is peaky enough and is replaced by the plugin estimation. The asymptotic statistics theory tells us that this assumption is true as the number of samples increases.

Under the assumption, the entropy can be factorized into sum of piecewise functions,

$$\begin{aligned} F_0(\boldsymbol{u}) &\approx \sum_{a,i} F_{a,i}(\boldsymbol{u}) \\ &= \sum_{a,i} \int H(G_{ai}; \widehat{A}_{ai}(X^N, U^N, \boldsymbol{x}, \boldsymbol{u})) \\ &\qquad \times p(x \mid \widehat{A}(X^N, U^N), \boldsymbol{u}) dx, \end{aligned} \tag{7}$$

which reduces computational complexity significantly. The entropy $H(G_{ai}; A_{ai})$ is given from the generative model by

$$H(G_{ai}; A_{ai}) = -q_0 \log q_0 - (1 - q_0) \log(1 - q_0), \tag{8}$$

where $q_0 = p(G_{ai} = 0 \mid A_{ai})$ can be computed from p_0 and $p(A_{ai} \mid G_{ai} = g)$ by Bayes rule.

However, $F_0(\boldsymbol{u})$ is still not tractable in practice because the distributions appearing in $F_0(\boldsymbol{u})$ have very complicated forms, and further they depend largely on unknown parameters whose estimation is unreliable when the number of observations is small and the model assumption is violated. Therefore, we apply some approximations and derive a variance minimization criterion that is easy to compute and is robust against deviation of parameter estimation. Before explaining the approximation, we introduce focusing technique below.

B. Focusing

We can see that the minimization of $F_0(\boldsymbol{u})$ attempts to reduce the uncertainty averaged over all links in the network. In this paper, we focus on a single link in one active learning step, i.e., we optimize only one term $F_{ai}(\boldsymbol{u})$ of Eq. (7) by choosing the most uncertain link (a, i) in G. This focusing technique improves performance significantly in network estimation as shown later.

Intuitively, the focusing technique is justified as follows. In the sequential design of active learning, a single input is chosen so that it reduces the uncertainty as much as possible. However, the power of single input is not enough strong to reduce the uncertainty for all links of the network, and the effect of active learning may become unclear in particular when the estimation of entropy is unreliable. For example, suppose there are 4 links and $\boldsymbol{u}_1$ reduces the entropy by 0.1 for all samples and $\boldsymbol{u}_2$ reduces the entropy by 0.3 for just one link. Although the entropy criterion without focusing chooses $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ is more preferable when there are noise ≈ 0.1 in the estimation of entropy values.

The basic idea to choose a link (a, i) is that A_{ai} locates close to the classification boundary between $G_{ai} = 1$ and 0. However, exploration is important in active learning to prevent repeated choice of very similar data points. We use probabilistic choice of a link: First calculate the entropy $H(G_{ai}; A_{ai})$ for all (a, i) pairs by Eq. (8), and define a probability on all the links by softmax function

$$Q_{ai} = \frac{\exp \beta H(G_{ai}; A_{ai})}{\sum_{b,j} \exp \beta H(G_{ai}; A_{bj})}, \tag{9}$$

where we took $\beta = 1$ in the simulation, and then generate (a, i) following the probability Q_{ai}.

In order to calculate the entropy value for the choice of uncertain link, we need parameter values τ_0^2, τ_1^2 and p_0. A set of components of current estimate $\widehat{A}$ can be regarded as n^2 independent samples from Gaussian mixtures of $N(0, \tau_0^2)$ and $N(0, \tau_1^2)$ with mixing proportion p_0 and $1 - p_0$. We apply the EM algorithm to estimate those parameters. When n is large, the number of samples increases, which is generally good for estimation. However, the quality of estimation of $\widehat{A}$ becomes worse for large n, thus there is a trade-off.

Even when the quality of estimation of τ_0, τ_1 and p_0 is not so good, the focusing is rather robust. One reason is that it is not so important which component is selected in focusing. We need information for all components at the end. Therefore the active learning will work if we select columns sequentially (note that only the choice of column make sense for the design). Entropy criterion using τ_0 and p_0 only determines roughly priority of the sequence, which will improve the performance slightly.

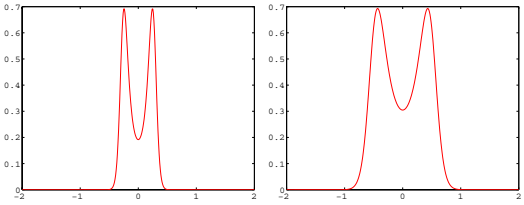

Fig. 2. Entropy function $H(G_{ai}; A_{ai})$ as a function of A_{ai}. Left: $\tau_0 = 0.1, \tau_1 = 2, p_0 = 0.5$, Right: $\tau_0 = 0.2, \tau_1 = 2, p_0 = 0.5$. The form of function is sensitive against the change of τ_0.

C. Variance minimization as an approximation of entropy minimization

The entropy minimization is difficult to implement directly. One reason is that the expected value of entropy is not analytically computable. Another reason lies in the shape of the entropy function. Let us consider the form of $H(G_{ai}; A_{ai})$ as a function of A_{ai}. Since $p(A_{ai} \mid G_{ai} = 0)$ is a peaky Gaussian and $p(A_{ai} \mid G_{ai} = 1)$ is a broader Gaussian, the form of function H is like mixture of two peaky Gaussian distributions with peaks around the Bayes optimal decision boundary $\{A_{ai} \mid p(G_{ai} = 0 \mid A_{ai}) = p(G_{ai} = 1 \mid A_{ai})\}$ (Fig.2).

The problem is that the two peaks depend on unknown parameters τ_0^2, τ_1^2, p_0 which are difficult to estimate from small samples, while $F_{ai}(\boldsymbol{u})$ is very sensitive against the change of the peaks. Therefore, it is necessary to replace $F_{ai}(\boldsymbol{u})$ by a qualitatively equivalent but more robust method.

In this paper, we use the criterion minimizing the variance of $\widehat{A}_{ai}(X^N, U^N, \boldsymbol{x}, \boldsymbol{u})$ instead of $F_{ai}(\boldsymbol{u})$. It is qualitatively justified as follows: The variance represents uncertainty of $\widehat{A}_{ai}$ while F_{ai} represents the uncertainty of G_{ai}. If the true value of A_{ai} is not close to peaks of H function (decision boundary), the uncertainty of G_{ai} decreases when the uncertainty of $\widehat{A}_{ai}$ is reduced (Mathematically, it can be seen from the fact that H is a locally convex function of $\widehat{A}_{ai}$). Therefore, in this case the entropy criterion can be replaced by the variance criterion. On the other hand, when the true A_{ai} is close to the peaks, as the uncertainty of A_{ai} is reduced, we are certain that A_{ai} is on the boundary, which means G_{ai}'s uncertainty increases (i.e., the uncertainty of G_{ai} cannot be essentially reduced in this case). Thus in this case, the variance criterion contradicts the entropy criterion. However, since the estimation of peaks is unreliable and peaks are located in a very narrow region, we assume that this case is negligible.

a) Derivation of the cost function: Although we can use any kind of estimator as $\widehat{A}$, it is preferable from computational viewpoint that the estimator is written in an analytic form in

order to evaluate the variance of $\widehat{A}(X^N, U^N, \boldsymbol{x}, \boldsymbol{u})$. In this paper, we use the ridge regression as an estimator of B, and $\widehat{A}$ is obtained by its inverse. Hereafter, we just write $\widehat{A}(\boldsymbol{x}, \boldsymbol{u}) = \widehat{A}(X^N, U^N, \boldsymbol{x}, \boldsymbol{u})$ and $\widehat{A} = \widehat{A}(X^N, U^N)$ and so on, unless confusing. We derive an asymptotic variance of $\widehat{A}$, i.e.,

$$E_{\boldsymbol{x}|\widehat{A},\boldsymbol{u}}[(\widehat{A}_{ai}(\boldsymbol{x}, \boldsymbol{u}) - \widehat{A}_{ai})^2].$$

The ridge regression estimation $\widehat{B}$ is given by

$$\widehat{B} = -\Sigma_{XU}\Sigma_{UU}^{-1}, \tag{10}$$

where

$$\Sigma_{XU} = (X^N)^\top U^N, \quad \Sigma_{UU} = (U^N)^\top U^N + \lambda I_n,$$

$\lambda > 0$ is a regularization constant. If $\lambda \to 0$, $\widehat{B}$ becomes the maximum likelihood solution that is an ordinary linear regression from $\boldsymbol{u}$ to $\boldsymbol{x}$ for Eq. (4). $\widehat{A}$ is given by $\widehat{B}^{-1}$ under the invertibility assumption of $\widehat{B}$.

By the argument of the asymptotic expansion theory in statistics, we have an approximation

$$\begin{aligned} &E_{\boldsymbol{x}|\widehat{A},\boldsymbol{u}}[(\widehat{B}(\boldsymbol{x}, \boldsymbol{u}) - \widehat{B})_{jb}(\widehat{B}(\boldsymbol{x}, \boldsymbol{u}) - \widehat{B})_{kc}] \\ &\quad \approx \sigma^2\delta_{jk}(\Sigma_{UU} + \boldsymbol{u}\boldsymbol{u}^\top)^{-1}_{bc}. \end{aligned} \tag{11}$$

The cost function to choose $\boldsymbol{u}$ is defined by $E_{\boldsymbol{x}|\widehat{A},\boldsymbol{u}}[(\widehat{A}(\boldsymbol{x}, \boldsymbol{u})_{ai} - \widehat{A}_{ai})^2]$. Given that the perturbation caused by $(\boldsymbol{x}, \boldsymbol{u})$ is small, we can use an approximation

$$\begin{aligned} \widehat{A}(\boldsymbol{x}, \boldsymbol{u}) - \widehat{A} &= \widehat{B}^{-1}(\boldsymbol{x}, \boldsymbol{u}) - \widehat{B}^{-1} \\ &\approx \widehat{B}^{-1}(\widehat{B}(\boldsymbol{x}, \boldsymbol{u}) - \widehat{B})\widehat{B}^{-1}, \end{aligned}$$

and from Eq. (11) we get

$$\begin{aligned} &E_{\boldsymbol{x}|\widehat{A},\boldsymbol{u}}[\{(\widehat{A}(\widehat{B}(\boldsymbol{x}, \boldsymbol{u}) - \widehat{B})\widehat{A})_{ai}\}^2] \\ &\quad \approx \sigma^2(\widehat{A}\widehat{A}^\top)_{aa}(\widehat{A}^\top(\Sigma_{UU} + \boldsymbol{u}\boldsymbol{u}^\top)^{-1}\widehat{A})_{ii}. \end{aligned} \tag{12}$$

With the equation

$$(\Sigma_{UU} + \boldsymbol{u}\boldsymbol{u}^\top)^{-1} = \Sigma_{UU}^{-1} - \frac{\Sigma_{UU}^{-1}\boldsymbol{u}\boldsymbol{u}^\top\Sigma_{UU}^{-1}}{1 + \boldsymbol{u}^\top\Sigma_{UU}^{-1}\boldsymbol{u}},$$

we obtain the objective function of active learning,

$$F_1(\boldsymbol{u}) = \min_{\|\boldsymbol{u}\|=1} -\frac{\left(\boldsymbol{a}_i^\top\Sigma_{UU}^{-1}\boldsymbol{u}\right)^2}{1 + \boldsymbol{u}^\top\Sigma_{UU}^{-1}\boldsymbol{u}}, \tag{13}$$

where $\boldsymbol{a}_i$ is the i-th column of $\widehat{A}$, i.e., $\widehat{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$. Interestingly, the cost function only depends on i not a, thus the effect of active learning covers all components of the column i rather than a single component.

The closed form of criterion $F_1(\boldsymbol{u})$ is a computational advantage of our approach against direct Bayesian framework, in which Monte Carlo or another kind of computer intensive methods are necessary.

1) Start-up samples: Generate random U_ν, and get X_ν by experiments $(\nu = 1, \ldots, N_{\text{init}})$
2) Estimation: Estimate $\widehat{B}$ (, $\widehat{A}$, and $\widehat{G}$) from the sample set (Eq. (10))
3) Focusing: Choose an uncertain link (sec.III-B)
4) Active learning: Choose $\boldsymbol{u}$ (sec.III-D)
5) Experiment: Get $\boldsymbol{x}$ from $\boldsymbol{u}$
6) Add $(\boldsymbol{u}, \boldsymbol{x})$ to the sample set and goto step 2

Fig. 3. Active learning algorithm (with focusing)

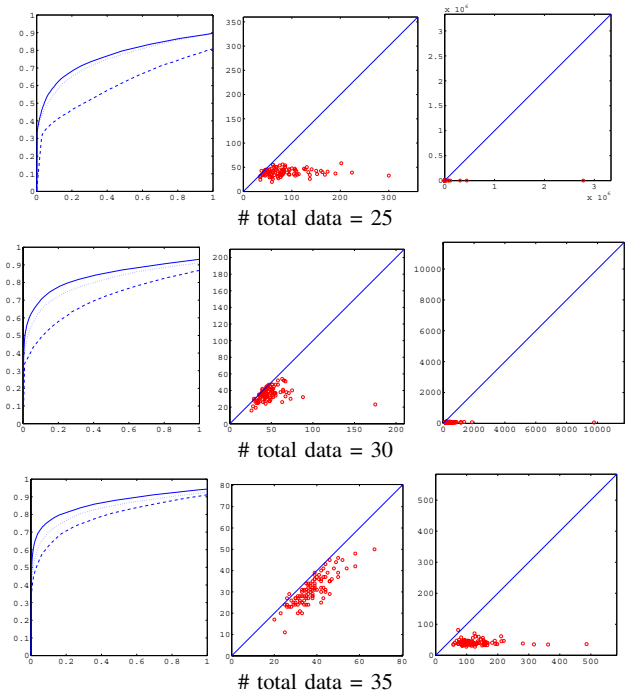

Fig. 4. Experimental results for linear model ($n = 20$). Left: ROC curves (solid:active with focusing, dotted:active without focusing, dashed:passive). Middle & Right:scatter plots of Error_G & Error_A (horizontal:passive, vertical:active with focusing).

D. Choice of $\boldsymbol{u}$

We assume the input $\boldsymbol{u}$ has an unit length vector $\|\boldsymbol{u}\| = 1$, because the cost function $F_1(\boldsymbol{u})$ decreases as $\|\boldsymbol{u}\|$ becomes large. In our modeling, the amplitude of $\boldsymbol{u}$ does not have an important role compared to its direction.

Exploration is important also in the case of choosing $\boldsymbol{u}$ to avoid concentration of sample points. For that purpose, one method is to choose $\boldsymbol{u}$ from random samples: First, generate N_{trial} samples randomly from uniform distribution from $\{\|\boldsymbol{u}\| = 1\}$, then choose $\boldsymbol{u}$ that minimizes $F_1(\boldsymbol{u})$. N_{trial} should not be too large for exploration while it should not be too small for effective active learning.

In Fig.3, the overall form of the active learning algorithm is summarized.

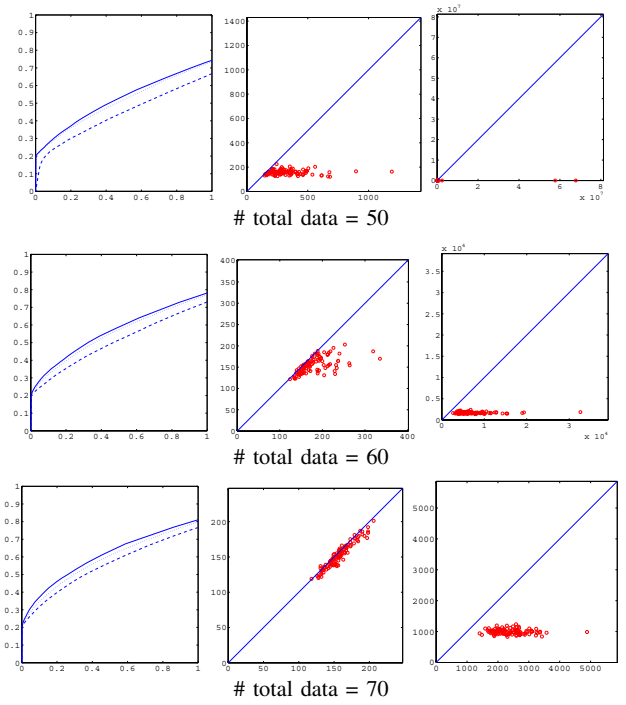

Fig. 5. Experimental results for linear model ($n = 40$). Left: ROC curves (solid:active with focusing, dotted:active without focusing, dashed:passive). Middle & Right:scatter plots of Error_G & Error_A (horizontal:passive, vertical:active with focusing).

N	P/A	Error_G	Error_A	ρ_G	ρ_A
25	P	84.58 (45.03)	4.0e4 (2.8e5)		
	A-nof	40.72 (7.43)	230.20 (81.23)	0.58	0.15
	A	39.65 (7.44)	175.31 (62.49)	0.57	0.11
30	P	47.58 (17.08)	506.11 (976.49)		
	A-nof	38.08 (7.59)	115.40 (36.87)	0.84	0.38
	A	35.65 (7.33)	70.63 (15.02)	0.79	0.23
35	P	37.76 (7.97)	126.81 (61.83)		
	A-nof	35.89 (7.48)	71.07 (16.61)	0.96	0.63
	A	31.13 (7.01)	41.12 (8.52)	0.83	0.37

TABLE I

AVERAGE VALUES (AND S.D.) OF $\mathrm{Error}_G, \mathrm{Error}_A$ AND AVERAGE ERROR RATIO ρ_G, ρ_A FOR LINEAR MODEL. NETWORK SIZE $n = 20$. A-nof STANDS FOR ACTIVE LEARNING WITHOUT FOCUSING

N	P/A	Error_G	Error_A	ρ_G	ρ_A
50	P	334.19 (160.84)	1.4e6 (8.8e6)		
	A-nof	163.97 (21.66)	5.7e3 (1.8e3)	0.58	0.16
	A	158.39 (19.65)	4.3e3 (845.61)	0.56	0.12
60	P	180.49 (37.99)	7.1e3 (4.2e3)		
	A-nof	155.96 (19.00)	2.5e3 (401.27)	0.88	0.43
	A	154.11 (17.98)	1.7e3 (191.07)	0.87	0.30
70	P	157.42 (18.70)	2.4e3 (540.72)		
	A-nof	154.07 (18.03)	1.5e3 (196.33)	0.98	0.64
	A	152.63 (17.89)	1.0e3 (82.65)	0.97	0.44

TABLE II

AVERAGE VALUES (AND S.D.) OF $\mathrm{Error}_G, \mathrm{Error}_A$ AND AVERAGE ERROR RATIO ρ_G, ρ_A FOR LINEAR MODEL. NETWORK SIZE $n = 40$. A-nof STANDS FOR ACTIVE LEARNING WITHOUT FOCUSING

IV. EXPERIMENTAL RESULTS

A. Linear Network

We performed experiments in standard numerical experiment settings described in [18]. First, we examined a simple linear gene network setting, which is a little more realistic than the generative model we have assumed. The dynamics is just the same as Eq. (1), and transition matrix A is separated into two parts, $A = -\Lambda + W$, where Λ is a constant diagonal matrix that determines the rate of self-degradation, and W represents the interaction between genes and is determined row by row as follows: First we choose a random integer k, $0 \leq k \leq k_{\max}$ from power law distribution, $p(k) \propto (k+1)^{-\eta}$, where $k_{\max} < n$ is a cutoff and η is a parameter and set to 0.5 in this experiment. We then choose k components and assign them a nonzero value from uniform distribution on $[-1, 0) \cup (0, 1]$. The chosen components form entries of graph edges ($G_{ai} = 1$) and other components are 0. Λ is set to $\Lambda = k_{\max} I$ in this experiment.

We performed experiments with the network size n =20 and 40 and various total numbers of training data. Noise term $\boldsymbol{\xi}(t)$ in Eq. (1) is a white noise with standard deviation 0.1. Cutoff parameter $k_{\max}$ is set to 5 and 10 for $n = 20, 40$ respectively. For each input, we simulated the differential equation from random initial points and stopped at $t = 30.0$ at which $\boldsymbol{x}$ converges to a steady state in most cases. In active learning, we prepared 10 and 20 passive samples for $n = 20, 40$ respectively as start-up data. This number is equal to a half of the network size and the covariance matrix becomes singular, thus it is the case the regularization is essentially necessary. We mean passive samples by pairs of random inputs and their outputs. In this paper, a random input is generated from the uniform distribution on the sphere $\{\|\boldsymbol{u}\| = 1\}$. In order to investigate the effect of focusing, we also performed active learning without focusing, where the cost function $F_1(\boldsymbol{u})$ is summed up for all i. For each setting of parameters, we performed $N_{\mathrm{exp}} = 100$ simulations with different true graphs. When we generate new input $\boldsymbol{u}$, we compare $N_{\mathrm{trial}} = 20n$ samples and choose the sample that minimizes the cost function.

b) Performance measure: Due to the linear equation setting, the optimal A is given by $A = W - \Lambda$. Therefore we can evaluate the error of A as well as G. Let $\widehat{A}$ and $\widehat{G}$ denote the estimated value of G and A in each simulation, and A^* and G^* be their true values. In order to estimate $\widehat{G}$, we set a threshold t and $\widehat{G}_{ai} = 1$ when $|\widehat{A}_{ai}| > t$ and $\widehat{G}_{ai} = 0$ otherwise. For evaluation of performance, it is useful to show ROC curves for G. We define the true positive rate by the frequency of $G^*_{ai} = 1$ components that are correctly estimated ($\widehat{G}_{ai} = 1$ and $\mathrm{sgn}[\widehat{A}_{ai}] = \mathrm{sgn}[A^*_{ai}]$), and the false positive rate

by the frequency of $G^*_{ai} = 0$ components that are correctly estimated ($\widehat{G}_{ai} = 0$). By changing the threshold t, we have a number of pairs of false and true positive rates. We get ROC curve by the scatter plot of the pairs. Note that the value of y axis (false positive) is not necessarily equal to 1.0 even when the value of x axis (true positive) is 1.0, because the error is defined with the sign of $\widehat{A}_{ai}$ taken into account.

Additionally we evaluated the performance for Bayes optimal decision rule. The Bayes decision boundary is obtained by solving $\{A_{ai} \mid p(G_{ai} = 0 \mid A_{ai}) = p(G_{ai} = 1 \mid A_{ai})\}$, which is given by

$$t_{\text{Bayes}} = \sqrt{\frac{2\left\{\log\left(\tau_1^2/\tau_0^2\right) + \log\left(p_0/(1-p_0)\right)\right\}}{1/\tau_0^2 - 1/\tau_1^2}},$$

where values of τ_0^2, τ_1^2, p_0 are estimated by the EM algorithm. We define the error index as follows: For A, we simply define the squared error by

$$\text{Error}_A = \sum_{a,i} (\widehat{A}_{ai} - A^*_{ai})^2.$$

However, the situation is not so simple about the graph G. If the link is correctly recovered ($\widehat{G}_{ai} = G^*_{ai} = 1$) but the effect of interaction (strengthen or weaken) is opposite ($\text{sgn}[\widehat{A}_{ai}] \neq \text{sgn}[A^*_{ai}]$), it should not be evaluated as the correct one. Therefore, we define the error measure for G by

$$\text{Error}_G = \sum_{a,i} (\widehat{G}_{ai}\text{sgn}[\widehat{A}_{ai}] - G^*_{ai}\text{sgn}[A^*_{ai}])^2.$$

c) Results: In figure 4 and 5, ROC curves and scatter plots of Error_G and Error_A are presented for the case of n =20 and 40 respectively. In the ROC curves, the average curves with the same t are shown. The figures show that active learning is effective in estimating the graphs and the transition matrix A. It is interesting to see that the active learning is more effective for smaller number of samples. In passive learning, estimation of A seems unstable giving large estimation error particularly for small sample size, while the active learning gives reasonable estimators.

The average errors and standard deviation over 100 simulations are shown in Tables I ($n = 20$) and II ($n = 40$). Index ρ_G for comparison between passive and active errors is defined by ratio of the errors averaged over all graphs G_i ($i = 1, \ldots, N_{\text{exp}}$),

$$\rho_G = \frac{1}{N_{\text{exp}}} \sum_{i=1}^{N_{\text{exp}}} \frac{1 + \text{Error}_{G_i}(\text{active})}{1 + \text{Error}_{G_i}(\text{passive})},$$

which indicates how much the error is reduced by active learning, i.e, the smaller ρ_G indicates the effectiveness of active learning. Index ρ_A for the errors of A is similarly defined,

$$\rho_A = \frac{1}{N_{\text{exp}}} \sum_{i=1}^{N_{\text{exp}}} \frac{\text{Error}_{A_i}(\text{active})}{\text{Error}_{A_i}(\text{passive})}.$$

In this experiment, we see a clear effect of focusing. In particular Error_A significantly decreases.

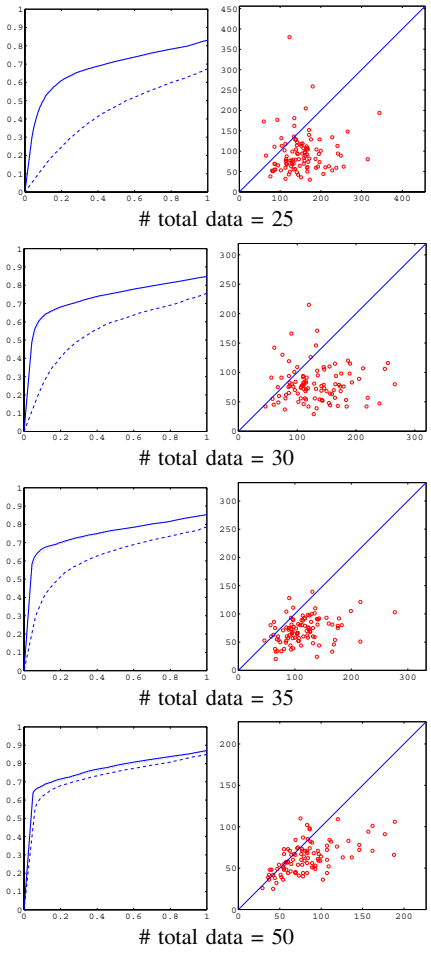

Fig. 6. Experimental results for nonlinear model ($n = 20$). Left: ROC curves (solid:active, dotted:active without focusing, dashed:passive). Right: scatter plots of Error_G (horizontal:passive, vertical:active).

B. Nonlinear gene network

Next, we examined a nonlinear network setting, which is more realistic and the linear model assumption is not satisfied any more. The graph G is generated randomly in the same way as in the linear case, and then links are classified into positive link or negative link with probability 1/2. The dynamical system of this setting is given by

$$\frac{dx_i(t)}{dt} = -\lambda_i x_i(t) + \frac{\alpha_i + \sum_{j\in\mathcal{A}_i} x_j(t)^{\gamma_{ij}}}{1 + \sum_{j\in\mathcal{A}_i} x_j(t)^{\gamma_{ij}} + \sum_{j\in\mathcal{R}_i} x_j(t)^{\beta_{ij}}} + \xi_i(t), \quad (14)$$

where $\mathcal{A}_i$ and $\mathcal{R}_i$ denote sets of positive and negative links connected to node i respectively, γ_{ij}, β_{ij} correspond to their strength, λ_i is a self-degradation factor, and α_i is the synthesis rate of the i-th node, $\xi_i(t)$ is noise. In this setting, we observe the behavior around the steady state. First, we measure the steady state $\boldsymbol{x}(\infty)$ by updating the differential equation for a

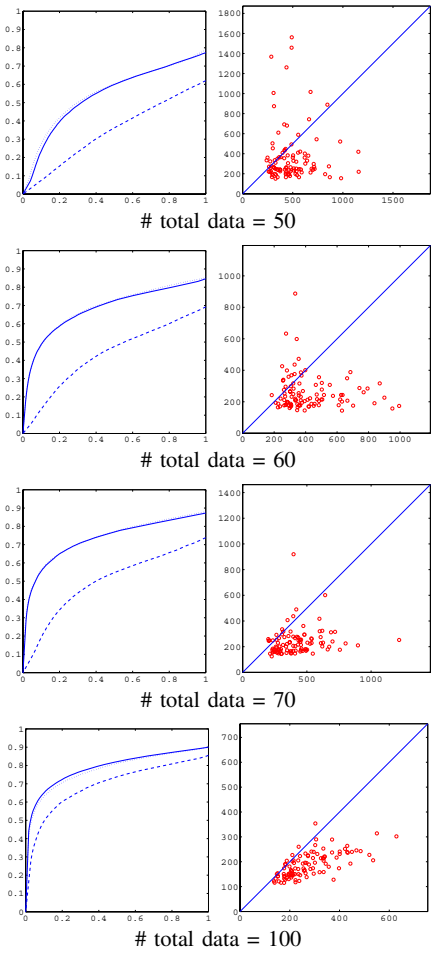

Fig. 7. Experimental results for nonlinear model ($n = 40$). Left: ROC curves (solid:active, dotted:active without focusing, dashed:passive). Right: scatter plots of Error_G (horizontal:passive, vertical:active).

sufficiently long time. Then for each input $\boldsymbol{u}$, we start from an initial state that is slightly perturbed with $\boldsymbol{u}$, i.e., $\boldsymbol{x}(0) = \boldsymbol{x}(\infty) + \epsilon\boldsymbol{u}$ and measure $\boldsymbol{x}(T)$ where T is some transient time before $\boldsymbol{x}$ is completely converged to the steady state. In this paper, we set $T = 2.0$, $\epsilon = 0.1$, $\mathrm{std}(\xi_i(t)) = 0.01$, and other parameters are set to constants: $\lambda_i = 1.0, \alpha_i = 0.01, \beta_{ij} = \gamma_{ij} = 1.0$.

d) Results: In this setting, since the optimal A is not tractable, we only evaluate the performance for the graph G (We can calculate Error_G because we know $\mathrm{sgn}[A_{ai}]$). In Figures 6($n = 20$) and 7($n = 40$), ROC curves and scatter plots of Error_G are presented. The average errors and standard deviation over 100 simulations are shown in Tables III($n = 20$) and IV($n = 40$). Although in general active learning scheme is considered to fail when the model assumption is violated, in this case active learning outperforms passive learning just as in the linear case. In this case, active learning without focusing also provided good results, because the parameter estimation of Gaussian mixture is pretty hard in the nonlinear setting.

N	P/A	Error_G	ρ_G
25	P	153.43 (50.32)	
	A-nof	92.59 (42.12)	0.65
	A	94.41 (49.51)	0.68
30	P	131.04 (46.36)	
	A-nof	80.26 (27.69)	0.68
	A	79.92 (30.62)	0.69
35	P	114.01 (38.66)	
	A-nof	72.31 (21.09)	0.68
	A	71.18 (22.65)	0.67
50	P	83.08 (33.50)	
	A-nof	64.75 (18.01)	0.85
	A	62.35 (18.58)	0.81

TABLE III

AVERAGE VALUES (AND S.D.) OF Error_G AND AVERAGE ERROR RATIO ρ_G FOR NONLINEAR MODEL. NETWORK SIZE $n = 20$. A-nof STANDS FOR ACTIVE LEARNING WITHOUT FOCUSING

N	P/A	Error_G	ρ_G
50	P	488.39 (191.85)	
	A-nof	324.04 (215.41)	0.76
	A	374.49 (276.82)	0.87
60	P	421.73 (176.75)	
	A-nof	259.91 (83.47)	0.69
	A	257.44 (110.86)	0.71
70	P	420.17 (169.02)	
	A-nof	229.12 (64.69)	0.61
	A	236.55 (104.68)	0.62
100	P	278.25 (102.56)	
	A-nof	201.19 (53.85)	0.77
	A	188.87 (49.78)	0.72

TABLE IV

AVERAGE VALUES (AND S.D.) OF Error_G AND AVERAGE ERROR RATIO ρ_G FOR NONLINEAR MODEL. NETWORK SIZE $n = 40$. A-nof STANDS FOR ACTIVE LEARNING WITHOUT FOCUSING

However, we can observe the effect of focusing when the number of samples is large.

V. CONCLUSION

We have proposed to apply an active learning framework to the network estimation problem, and have shown through some numerical experiments that the variance minimization criterion and focusing technique is effective for this problem. Main advantage of our approach is small computation cost and the robustness against parameter estimation. In future works, we would like to apply the method to real gene or biochemical networks. In such a real situation, we might need more constraint about the problem, for example, sparsity of network link.

REFERENCES

[1] T. Chen, H. He, and G. Church, "Modeling gene expression with differential equations," in *Proc. Pac. Symposium on Biocomputing*, vol. 4, 1999, pp. 29–40.

[2] T. Akutsu, S. Miyano, and S. Kuhara, "Algorithms for identifying boolean networks and related biological networks based on matrix multiplication and fingerprint function," *J. Comp. biol.*, vol. 7, pp. 331–344, 2000.

[3] P. D'haeseller, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[4] H. Toh and K. Horimoto, "Inference of a genetic network by a combined approach of cluster analysis and graphical modeling," *Bioinformatics*, vol. 18, pp. 287–297, 2002.

[5] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young, "Combining location and expression data for principled discovery of genetic regulatory network models," in *Proc. Pac. Symposium on Biocomputing*, vol. 7, 2002, pp. 437–449.

[6] M. De Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano, "Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations," in *Proc. Pac. Symposium on Biocomputing*, vol. 8, 2003, pp. 17–28.

[7] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alché Buc, "Gene networks inference using dynamic bayesian networks," *Bioinformatics*, vol. 19, no. Suppl. 2, pp. ii138–ii148, 2003.

[8] J. Tegnér, M. Yeung, J. Hasty, and J. Collins, "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 10, pp. 5944–5949, 2003.

[9] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano, "Combining microarrays and biological knowledge for estimating gene networks via bayesian networks," *J. of Bioinformatics and Computational Biology*, vol. 2, no. 1, pp. 77–98, 2004.

[10] R. Peeters and R. Westra, "On the identification of sparse gene regulatory networks," in *Proc. of the 16th Intern. Symp. on Mathematical Theory of Networks and Systems (MTNS2004)*, 2004.

[11] P. Brazhnik, "Inferring gene networks from steady-state response to single-gene perturbations," *J. of Theoretical Biology*, vol. 237, pp. 427–440, 2005.

[12] M. Bansal, G. Gatta, and D. di Bernardo, "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles," *Bioinformatics*, vol. 22, no. 7, pp. 815–822, 2006.

[13] D. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[14] D. Cohn, "Neural network exploration using optimal experiment design," *Neural Networks*, vol. 9, no. 6, pp. 1071–1083, 1996.

[15] K. Fukumizu, "Active learning in multilayer perceptrons," *Advances in NIPS 8*, pp. 295–301, 1996.

[16] V. Fedorov, *Theory of Optimal Experiments*. New York: Academic Press, 1972.

[17] T. Gardner, D. di Bernardo, D. Lorenz, and J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, pp. 102–105, 2003.

[18] M. Yeung, J. Tegnér, and J. Collins, "Reverse engineering gene networks using singular value decomposition and robust regression," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 9, pp. 6163–6168, 2002.

[19] H. Schmidt, K.-H. Cho, and E. Jacobsen, "Identification of small scale biochemical networks based on general type system perturbations," *FEBS Journal*, vol. 272, pp. 2141–2151, 2005.