CLASSIFICATION OF GAUSSIAN TRAJECTORIES WITH MISSING DATA IN BOOLEAN GENE REGULATORY NETWORKS

Alireza Karbalayghareh, Ulisses Braga-Neto, and Edward R. Dougherty

Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX, USA
{alireza.kg, ulisses}@tamu.edu, edward@ece.tamu.edu

ABSTRACT

This paper studies the classification of gene regulatory networks (GRNs) modeled by probabilistic Boolean networks (PBNs). After observing Gaussian expression values of n genes at m consecutive time points, with consideration of missing data, an algorithm based on expectation maximization (EM) is proposed to estimate the parameters and infer the unknown parts of the networks in the maximum likelihood (ML) sense. Then the estimated values are plugged in to the Bayes classifier, which is optimal, and the performance of the classifier is investigated through various simulations.

Index Terms— Trajectory classification, missing trajectory data, Bayes classifier, probabilistic Boolean network, gene regulatory network, expectation maximization

1. INTRODUCTION

Phenotypic classification using genomic data has been a staple of genomic signal processing [1] for fifteen years, the salient technologies first being expression microarrays and then RNA-seq. Both technologies measure gene expression over collections of cells and thus do not capture regulatory timing. Unless there is synchronization, which in practice there is not, the cells in a collection will be in different states, so that the expression measurement is actually an average of expressions over these states. For instance, it is typically the case that the expression level of a gene will oscillate cyclically when in the steady state, so that a microarray or RNA-seq measurement is an expression average over the cycle. This averaging masks differences in gene activity in different phenotypes, thereby degrading classification. With the advent of new single-cell technologies (still formative), one can envision classification based on gene-expression time trajectories, which will be far more sensitive to phenotypic changes [2, 3, 4].

In [5] we considered classification via trajectories generated by gene regulatory networks: given a network, in this case a binary probabilistic Boolean network (PBN) [6], and a second network resulting from mutations in the original network, the problem is to classify an observed time trajectory of expression vectors as to which network it belongs. In [5] we considered optimal classification (the Bayes classifier) based on complete observation of the trajectory. Here we again assume an underlying $0-1$ PBN; however, we assume a Gaussian observational model, in which the expression level of each gene given its state (hidden) follows a normal density with some unknown mean and variance, and we observe the Gaussian expression values of n genes in m consecutive time points. Moreover, we allow missing observations, which reflects the practical situation in which the sampling rate for expression measurement is insufficient to capture all expression state changes. Specifically, for any point in a trajectory, there is a probability of missing the measurement.

Upon observing a trajectory, we estimate the unknown network parameters as well as the unknown connections of the networks that are partially known. To achieve maximum likelihood estimation and inference, we use Expectation Maximization (EM) to estimate network parameters and plug in the estimated parameters and the inferred networks to the Bayes classifier [7]. We consider the effect of different parameters on average classification error over many random networks to investigate the ability to classify healthy and cancerous networks over a range of mutations using different length trajectories and amounts of missing data.

2. BOOLEAN NETWORKS

For a binary Boolean network (BN) on n genes, a truth table gives the functional relationships between the genes [8]. Each gene value $x_i \in \{0,1\}$, for $i = 1, \cdots, n$, at time $k+1$ is determined by the values of some predictor genes at time k via a Boolean function $f_i : \{0,1\}^n \rightarrow \{0,1\}$ in the truth table. In practice, f_i is a function of the small number of genes, K_i, which is called the input degree of the gene x_i in the network. Given a truth table, a gene network can be represented as a graph with vertices representing genes and edges representing regulations.

For BNs with perturbations (BNps), perturbation is introduced with a probability p by which the current state of the network can be randomly changed. Implicitly, we assume that there is an independent identically distributed (i.i.d.) random

perturbation vector at each time k, denoted by $\mathbf{n}_k \in \{0,1\}^n$, where the i-th gene flips at time k if the i-th component of $\mathbf{n}_k$ is equal to 1. Therefore, the dynamical model of the states can be expressed as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \oplus \mathbf{n}_{k+1}, \quad k = 0, 1, 2, \cdots, \tag{1}$$

where $\mathbf{x}_k = [x_1(k), x_2(k), \cdots, x_n(k)]^T$ is a binary state vector, called a *gene activity profile* (GAP), at time k, in which $x_i(k)$ indicates the expression level of the ith gene at time k (either 0 or 1); $\mathbf{f}(\mathbf{x}_k) = [f_1, f_2, \cdots, f_n]^T : \{0,1\}^n \to \{0,1\}^n$ is the vector of the network functions, in which f_i shows the expression level of the ith gene at time $k+1$ when the system lies in the state $\mathbf{x}_k$ at time k; $\mathbf{n}_k = [n_1(k), n_2(k), \cdots, n_n(k)]^T$ is the perturbation vector at time k, in which $n_1(k), n_2(k), \cdots, n_n(k)$ are i.i.d. Bernoulli random variables for every k with the parameter $p = P(n_i(k) = 1)$; and $\oplus$ is component-wise modulo 2 addition. The existence of perturbation makes the corresponding Markov chain of a BNp irreducible.

We assume the following Gaussian observation model

$$f(y_j(k)|x_j(k)) \sim \mathcal{N}(\mu_j(k), \sigma_j^2), \quad j = 1, 2, \cdots, n, \tag{2}$$

where $x_j(k)$ is the hidden binary state (0 or 1) of the j-th gene, and $y_j(k)$ is the observed value of the j-th gene drawn from the Gaussian pdf in (2). We assume that the variance σ_j^2 of the j-th gene is constant over time, but its mean varies over time and is defined as $\mu_j(k) = \lambda + \delta_j x_j(k)$. This shows that when the j-th gene is non-activated and activated, its observed expression values come respectively from a normal distribution with the means of λ and $\lambda + \delta_j$ and with the same variance of σ_j^2, respectively. We denote the observed expression values of the n genes at time k by the vector $\mathbf{y}_k = [y_1(k), \cdots, y_n(k)]^T$.

3. MAXIMUM LIKELIHOOD PARAMETER ESTIMATION AND NETWORK INFERENCE

We assume that we partially know the network and do not know the network parameters p, δ_j, and σ_j^2. Using D observed trajectories $\mathbb{Y} = \{\mathcal{Y}^{(1)}, \mathcal{Y}^{(2)}, \cdots, \mathcal{Y}^{(D)}\}$, we aim to infer the unknown connections of the network as well as the parameters. This observed data set $\mathbb{Y}$ may be incomplete, meaning that it may include missing data. Without missing data, each trajectory $\mathcal{Y}^{(d)}$, for $d = 1, \cdots, D$, is supposed to have the expression measurements of the n genes in m consecutive time points. However, we assume that each gene at each time point has the probability p_{miss} of being missed. As a result, each observed trajectory has the form $\mathcal{Y}^{(d)} = \left[\mathbf{y}_{i_1}^{(d)}, \cdots, \mathbf{y}_{i_{m(d)}}^{(d)}\right]$, where $T_{obs}^{(d)} = \{i_1, \cdots, i_{m(d)}\}$ is the set of the observed time points in which at least one gene is observed. It is obvious that $T_{obs}^{(d)} \subseteq \{1, 2, \cdots, m\}$ for any $d = 1, \cdots, D$.

For the maximum likelihood (ML) problem, our search space consists of both discrete and continuous. The space of the functions is discrete, and that of the parameters is continuous. Suppose $\mathbf{F} = \{\mathbf{f}^1, \mathbf{f}^2, \cdots, \mathbf{f}^M\}$ is the space of M possible network functions. Suppose our parameter space is defined as $\theta = [p, \delta_1, \cdots, \delta_n, \sigma_1^2, \cdots, \sigma_n^2]$. For any given network function $\mathbf{f}^i$, $i = 1, \cdots, M$, we employ the EM algorithm to find the optimal parameters θ:

$$\hat{\theta}_i = \underset{\theta}{\operatorname{argmax}}\, f(\mathbb{Y}|\mathbf{f}^i, \theta), \tag{3}$$

The ML inferred network and estimated parameters are then derived as,

$$(\hat{\mathbf{f}}, \hat{\theta}) = \arg \max_{(\mathbf{f},\theta)\in\{(\mathbf{f}^1,\hat{\theta}_1),\cdots,(\mathbf{f}^M,\hat{\theta}_M)\}} f(\mathbb{Y}|\mathbf{f}, \theta). \tag{4}$$

3.1. EM algorithm for finding θ

In (3), the network function is given, and we are supposed to find the ML estimation for θ. Suppose the given network function in (3) is $\mathbf{f}$. The EM algorithm can be described simply as repeating the following steps until convergence:
1- Expectation: $Q(\theta, \theta^s) = \sum_{\mathbb{X}} \log[f(\mathbb{X}, \mathbb{Y}|\theta)] P(\mathbb{X}|\mathbb{Y}, \theta^s)$,
2- Maximization: $\theta^{s+1} = \operatorname{argmax}_\theta Q(\theta, \theta^s)$,
where $\mathbb{X} = \{\mathcal{X}^{(1)}, \cdots, \mathcal{X}^{(D)}\}$ are the hidden variables corresponding to D observed trajectories, and $\mathcal{X}^{(d)} = [\mathbf{x}_1^{(d)}, \cdots, \mathbf{x}_m^{(d)}]$ are the hidden variables (states) of the d-th trajectory from time 1 to m. Since the observations are i.i.d., we have

$$\log[f(\mathbb{X}, \mathbb{Y}|\theta)] = \sum_{d=1}^{D} \log[f(\mathcal{X}^{(d)}, \mathcal{Y}^{(d)}|\theta)], \tag{5}$$

$$f(\mathcal{X}^{(d)}, \mathcal{Y}^{(d)}|\theta) = \pi_{\mathbf{x}_1^{(d)}} \prod_{k=1}^{m-1} P(\mathbf{x}_{k+1}^{(d)}|\mathbf{x}_k^{(d)}) \prod_{k\in T_{obs}^{(d)}} f(\mathbf{y}_k^{(d)}|\mathbf{x}_k^{(d)}), \tag{6}$$

where the transition probability matrix (TPM) is

$$P(\mathbf{x}_{k+1}^{(d)}|\mathbf{x}_k^{(d)}) = p^{\mathbf{d}(\mathbf{x}_{k+1}^{(d)}, \mathbf{f}(\mathbf{x}_k^{(d)}))} (1-p)^{n - \mathbf{d}(\mathbf{x}_{k+1}^{(d)}, \mathbf{f}(\mathbf{x}_k^{(d)}))}, \tag{7}$$

where $\mathbf{d}(\mathbf{x}_{k+1}^{(d)}, \mathbf{f}(\mathbf{x}_k^{(d)}))$ denotes the Hamming distance between the two vectors $\mathbf{x}_{k+1}^{(d)}$ and $\mathbf{f}(\mathbf{x}_k^{(d)})$, and the conditional density of the observation given the state is

$$f(\mathbf{y}_k^{(d)}|\mathbf{x}_k^{(d)}) = \prod_{j\in G_k^{(d)}} f(y_j^{(d)}(k)|x_j^{(d)}(k)), \tag{8}$$

where $G_k^{(d)}$ is the set of observed genes at time k of the d-th trajectory. Note that $G_k^{(d)} \subseteq \{1, 2, \cdots, n\}$. From (2) and (8), we have,

$$f(\mathbf{y}_k^{(d)}|\mathbf{x}_k^{(d)}) = (2\pi)^{-\frac{1}{2}\left|G_k^{(d)}\right|} \times \prod_{j\in G_k^{(d)}} \frac{1}{\sigma_j} \exp\left[\frac{-\left(y_j^{(d)}(k) - \lambda - \delta_j x_j^{(d)}(k)\right)^2}{2\sigma_j^2}\right]. \tag{9}$$

We assume a uniform initial distribution, that is, $\pi_{\mathbf{x}_1^{(d)}} = \frac{1}{2^n}$ for every value of $\mathbf{x}_1^{(d)}$. Using (5)-(9) we can write $Q(\theta, \theta^s)$ as in (10) (on top of the next page). Now we take the derivative of $Q(\theta, \theta^s)$ with respect to θ and make it equal to zero. Without the loss of generality, we assume that $\sigma_j^2 = \sigma^2$ and $\delta_j = \delta$ for $j = 1, \cdots, n$, and also λ, the baseline level of expression, is known. Taking the derivative of $Q(\theta, \theta^{(s)})$ with respect to p, δ, and σ^2 leads to their estimates in (11), (12), and (13), respectively, at the $(s+1)$-th step. We define the following posterior probabilities which have appeared in (11)-(13):

$$\gamma_i^{d,s}(k) = P(\mathbf{x}_k^{(d)} = i | \mathcal{Y}^{(d)}, \theta^s), \tag{14}$$

$$\xi_{i,j}^{d,s}(k) = P(\mathbf{x}_k^{(d)} = i, \mathbf{x}_{k+1}^{(d)} = j | \mathcal{Y}^{(d)}, \theta^s). \tag{15}$$

The posterior probabilities in (14) and (15) can be efficiently computed using the forward-backward algorithm whose complexity is linear in m. It can be shown that the posteriors in (14) and (15) can be written as

$$\gamma_i^{d,s}(k) = \frac{\alpha_i^{d,s}(k)\beta_i^{d,s}(k)}{\sum_{r=1}^{2^n} \alpha_r^{d,s}(k)\beta_r^{d,s}(k)}, \tag{16}$$

$$\xi_{i,j}^{d,s}(k) = \frac{\alpha_i^{d,s}(k) A_{i,j}^{(s)} \beta_j^{d,s}(k+1) b_j^{d,s}(k+1)}{\sum_{r=1}^{2^n} \alpha_r^{d,s}(m)}, \tag{17}$$

where $A_{i,j}^{(s)} = P(\mathbf{x}_{k+1} = j | \mathbf{x}_k = i)$ is the transition matrix assuming $\theta = \theta^s$, and $b^{d,s}(k)$ is a $2^n \times 1$ vector at time k, whose j-th entry is defined as $b_j^{d,s}(k) = f(\mathbf{y}_k^{(d)} | \mathbf{x}_k^{(d)} = j)$ assuming $\theta = \theta^s$, which can be computed using (9). Furthermore, $\alpha_i^{d,s}(k)$ and $\beta_i^{d,s}(k)$ are respectively the forward and backward parameters. We can write α, β, γ, and ξ in the vector-matrix form. Define the vectors $\alpha^{d,s}(k) = [\alpha_1^{d,s}(k), \cdots, \alpha_{2^n}^{d,s}(k)]^T$ and $\beta^{d,s}(k) = [\beta_1^{d,s}(k), \cdots, \beta_{2^n}^{d,s}(k)]^T$. As a result, we have the following recursions in the vector-matrix form,

$$\begin{aligned} \alpha^{d,s}(1) &= \frac{1}{2^n} \mathbf{1}_{2^n} \circ b^{d,s}(1), \\ \alpha^{d,s}(k+1) &= \left[{A^{(s)}}^T \alpha^{d,s}(k) \right] \circ b^{d,s}(k+1), \end{aligned} \tag{18}$$

and

$$\begin{aligned} \beta^{d,s}(m) &= \mathbf{1}_{2^n}, \\ \beta^{d,s}(k) &= A^{(s)} \left[\beta^{d,s}(k+1) \circ b^{d,s}(k+1) \right], \end{aligned} \tag{19}$$

where $\mathbf{1}_{2^n}$ is the all-one column vector of length 2^n, and $\circ$ denotes the Hadamard product (or component-wise product). Now suppose that $\gamma^{d,s}(k)$ and $\xi^{d,s}(k)$ are respectively a $2^n \times 1$ vector and $2^n \times 2^n$ matrix whose entries are given in (16) and (17). Therefore, we have

$$\gamma^{d,s}(k) = \frac{\alpha^{d,s}(k) \circ \beta^{d,s}(k)}{\| \alpha^{d,s}(k) \circ \beta^{d,s}(k) \|_1}, \tag{20}$$

$$\xi^{d,s}(k) = \frac{\left[\alpha^{d,s}(k) \beta^{d,s}(k+1)^T \right] \circ A^{(s)} \circ C^{d,s,k}}{\| \alpha^{d,s}(m) \|_1}, \tag{21}$$

where $C^{d,s,k}$ is a $2^n \times 2^n$ matrix defined as

$$C^{d,s,k} = \left[b^{d,s}(k+1), \cdots, b^{d,s}(k+1) \right]^T. \tag{22}$$

4. PLUG-IN CLASSIFIER

Once we have derived the ML estimates for the parameters of both BNps using the EM algorithm, we can plug them in to the Bayes classifier. Suppose $\hat{\theta} = \{\hat{p}, \hat{\delta}, \hat{\sigma^2}\}$ and $\hat{\tilde{\theta}} = \{\hat{\tilde{p}}, \hat{\tilde{\delta}}, \hat{\tilde{\sigma}^2}\}$ are respectively the ML estimates of the parameters of the original and mutated BNps resulted from the EM algorithm, and $\hat{\mathbf{f}}$ and $\hat{\tilde{\mathbf{f}}}$ are respectively the inferred original and mutated networks. Hence, the plug-in classifier for any given observation $\mathcal{Y} = [\mathbf{y}_{i_1}, \cdots, \mathbf{y}_{i_m}]$ (with missing data) is defined as

$$\psi_D(\mathcal{Y}) = \begin{cases} 1, & \hat{p}_1 f(\mathcal{Y} | \hat{\tilde{\mathbf{f}}}, \hat{\tilde{\theta}}) \geq \hat{p}_0 f(\mathcal{Y} | \hat{\mathbf{f}}, \hat{\theta}) \\ 0, & \hat{p}_1 f(\mathcal{Y} | \hat{\tilde{\mathbf{f}}}, \hat{\tilde{\theta}}) < \hat{p}_0 f(\mathcal{Y} | \hat{\mathbf{f}}, \hat{\theta}) \end{cases}, \tag{23}$$

where $\hat{p}_0$ and $\hat{p}_1$ are the estimated values of the prior probabilities of the original and mutated networks respectively. These values can be estimated by the number of observations for each network divided by the total number of observations, but this estimate is not so reliable, especially in the small sample scenarios. As a result, we assume the commonly used equiprobable case, that is, $\hat{p}_0 = \hat{p}_1 = \frac{1}{2}$, and in the simulation part, we generate the equal number of observations for each network as well. The densities $f(\mathcal{Y} | \hat{\tilde{\mathbf{f}}}, \hat{\tilde{\theta}})$ and $f(\mathcal{Y} | \hat{\mathbf{f}}, \hat{\theta})$ in (23) can be computed using the forward-backward algorithm, as previously demonstrated. To do so, we only need the forward computations, and they can be written by

$$f(\mathcal{Y} | \hat{\mathbf{f}}, \hat{\theta}) = \| \hat{\alpha}(m) \|_1, \tag{24}$$

$$f(\mathcal{Y} | \hat{\tilde{\mathbf{f}}}, \hat{\tilde{\theta}}) = \| \hat{\tilde{\alpha}}(m) \|_1, \tag{25}$$

where $\hat{\alpha}(m)$ and $\hat{\tilde{\alpha}}(m)$ can be computed by the forward computations, as in (18), as follows

$$\begin{aligned} &\hat{\alpha}(1) = \frac{1}{2^n} \mathbf{1}_{2^n} \circ \hat{b}(1), \quad \hat{\alpha}(k+1) = \left[\hat{A}^T \hat{\alpha}(k) \right] \circ \hat{b}(k+1), \\ &\hat{\tilde{\alpha}}(1) = \frac{1}{2^n} \mathbf{1}_{2^n} \circ \hat{\tilde{b}}(1), \quad \hat{\tilde{\alpha}}(k+1) = \left[\hat{\tilde{A}}^T \hat{\tilde{\alpha}}(k) \right] \circ \hat{\tilde{b}}(k+1). \end{aligned} \tag{26}$$

5. SIMULATION RESULTS

In this part, we have studied the classification error averaged over many randomly generated BNps. In the simulations, we assume that the maximum input degree is $K = 2$, and the bias probability is $p_{bias} = 0.5$, meaning that each gene i has

$$Q(\theta,\theta^s)=-nD\log 2+\sum_{d=1}^{D}\sum_{k=1}^{m-1}\sum_{\mathbf{x}=1}^{2^n}\sum_{\mathbf{x}'=1}^{2^n}\Big[\mathbf{d}(\mathbf{x}',\mathbf{f}(\mathbf{x}))\log p+[n-\mathbf{d}(\mathbf{x}',\mathbf{f}(\mathbf{x})]\log(1-p)\Big]P(\mathbf{x}_k^{(d)}=\mathbf{x},\mathbf{x}_{k+1}^{(d)}=\mathbf{x}'|\mathcal{Y}^{(d)},\theta^s)$$

$$-\frac{1}{2}\log 2\pi\sum_{d=1}^{D}\sum_{k\in T_{obs}^{(d)}}\left|G_k^{(d)}\right|+\sum_{d=1}^{D}\sum_{k\in T_{obs}^{(d)}}\sum_{\mathbf{x}=1}^{2^n}\sum_{j\in G_k^{(d)}}\left[-\frac{1}{2}\log\sigma_j^2-\frac{\left(y_j^{(d)}(k)-\lambda-\delta_j x_j\right)^2}{2\sigma_j^2}\right]P(\mathbf{x}_k^{(d)}=\mathbf{x}|\mathcal{Y}^{(d)},\theta^s), \quad (10)$$

$$p^{s+1}=\frac{\sum_{d=1}^{D}\sum_{k=1}^{m-1}\sum_{\mathbf{x}=1}^{2^n}\sum_{\mathbf{x}'=1}^{2^n}\mathbf{d}(\mathbf{x}',\mathbf{f}(\mathbf{x}))P(\mathbf{x}_k^{(d)}=\mathbf{x},\mathbf{x}_{k+1}^{(d)}=\mathbf{x}'|\mathcal{Y}^{(d)},\theta^s)}{nD(m-1)}, \quad (11)$$

$$\delta^{s+1}=\frac{\sum_{d=1}^{D}\sum_{k\in T_{obs}^{(d)}}\sum_{\mathbf{x}=1}^{2^n}\sum_{j\in G_k^{(d)}}x_j\left(y_j^{(d)}(k)-\lambda\right)P(\mathbf{x}_k^{(d)}=\mathbf{x}|\mathcal{Y}^{(d)},\theta^s)}{\sum_{d=1}^{D}\sum_{k\in T_{obs}^{(d)}}\sum_{\mathbf{x}=1}^{2^n}\sum_{j\in G_k^{(d)}}x_j P(\mathbf{x}_k^{(d)}=\mathbf{x}|\mathcal{Y}^{(d)},\theta^s)}, \quad (12)$$

$$\sigma^{2^{s+1}}=\frac{\sum_{d=1}^{D}\sum_{k\in T_{obs}^{(d)}}\sum_{\mathbf{x}=1}^{2^n}\sum_{j\in G_k^{(d)}}\left(y_j^{(d)}(k)-\lambda-\delta^{s+1}x_j\right)^2P(\mathbf{x}_k^{(d)}=\mathbf{x}|\mathcal{Y}^{(d)},\theta^s)}{\sum_{d=1}^{D}\sum_{k\in T_{obs}^{(d)}}\left|G_k^{(d)}\right|}. \quad (13)$$

the probability 0.5 of being 0 or 1 for any of its 2^{K_i} inputs. Furthermore, we have considered single gene mutations, such that the mutated network is derived from the healthy network after flipping the value of a random gene in one random configuration of its inputs. We also assume that the output of one random gene is unknown in one random configuration of its inputs in the both BNps.

Fig. 1-(a) shows the error versus the number of sample trajectories D. We can see that when D increases, the classification error decreases, the reason being that the parameters are estimated, and the networks are inferred more accurately when having more training trajectories. When there is enough sample trajectories, the error converges to the Bayes error, as the ML estimates are consistent. Fig. 1-(b) represents the error versus m for different values of n. We see that the error has a decreasing trend as m grows, which shows us that having longer trajectories leads to a better classification of the networks. We also see from Fig. 1-(b) that the networks with $n = 6$ genes have reached the lowest error, which can help us in network reduction of larger networks. Figs. 1-(c) and (d) show the error plots versus m for different values of p_{miss} and σ^2, respectively. As expected, when p_{miss} increases, the classification error increases. The increase of σ^2 also results higher error, since σ^2 plays the role of observational noise, that is, when σ^2 grows, the observed expression values of genes get closer in ON and OFF situations.

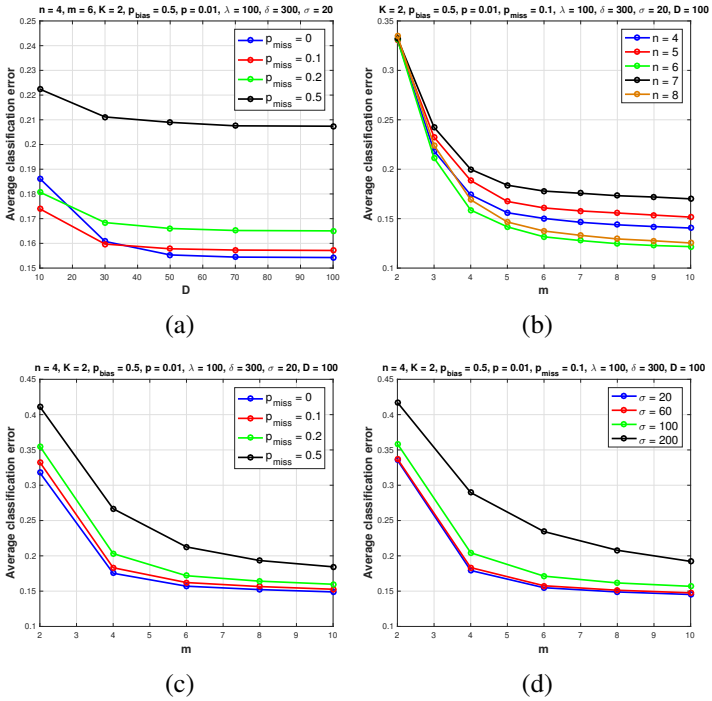

Fig. 1: (a) Average classification error vs. D for $p_{miss} = 0, 0.1, 0.2, 0.5$, (b) Average classification error vs. m for $n = 4, \cdots, 8$, (c) Average classification error vs. m for $p_{miss} = 0, 0.1, 0.2, 0.5$, (d) Average classification error vs. m for $\sigma = 20, 60, 100, 200$.

6. CONCLUSION AND FUTURE WORK

In this paper, we studied the classification of the Boolean gene regulatory networks upon the observation of the gene expression trajectories in the presence of missing data. We proposed an algorithm to estimate the parameters and infer the networks based on the observed training trajectories, and used the plug-in Bayes classifier to classify the healthy and cancerous networks. Due to the lack of space, we only provide part of analysis here. However, we are performing more thorough analysis of this scenario in our full version paper and comparing it with RNA-seq which reveals expression values averaged over unsynchronized multiple-cell scenarios.

7. REFERENCES

[1] I. Shmulevich and E. R. Dougherty, "Genomic Signal Processing," *Princeton University Press*, Princeton, 2007.

[2] A. R. Wu, N. F. Neff, T. Kalisky, P. Dalerba, B. Treutlein, M. E. Rothenberg, F. M. Mburu, G. L. Mantalas, S. Sim, M. F. Clarke, and S. R. Quake, "Quantitative assessment of single-cell RNA-sequencing methods," *Nature Methods*, vol. 11, pp. 41-46, 2014.

[3] A. Stahlberg and M. Kubista "The workflow of single-cell expression profiling using quantitative real-time PCR," *Expert Rev Mol Diagn*, vol. 14, pp. 323-331, 2014.

[4] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature Biotechnology*, vol. 32, pp. 381-386, 2014.

[5] A. Karbalayghareh, U. Braga-Neto, J. Hua, and E. R. Dougherty, "Classification of State Trajectories in Gene Regulatory Networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. PP, no. 99, pp. 11, 2016.

[6] I. Shmulevich and E. R. Dougherty, "Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks," *SIAM Press*, New York, 2010.

[7] L. Devroye, L. Gyorfi, and G. Lugosi, "A Probabilistic Theory of Pattern Recognition," *Springer*, New York, 1996.

[8] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," Journal of Theoretical Biology, vol. 22, pp. 437-467, 1969.