MAXIMUM LIKELIHOOD ADAPTIVE FILTER USING METAHEURISTIC OPTIMIZATION FOR STOCHASTIC NONLINEAR DYNAMIC SYSTEMS

A Dissertation

by

YUKUN TAN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Ulisses M.Braga-Neto
Committee Members,	Xiaoning Qian
	Aniruddha Datta
	Ivan Ivanov
Head of Department,	Aniruddha Datta

May 2021

Major Subject: Electrical Engineering

ABSTRACT

Nonlinear state-space models have long been used in the statistical signal processing community as a powerful tool for modeling and forecasting the behavior of dynamical systems. However, in the case of a large system with many unknown parameters, which is a common scenario in real-world applications, the computational complexity of most methods becomes intractable, especially when the system parameters contain both discrete and continuous components.

This dissertation is focused on efficient state and parameter estimation in nonlinear dynamical systems with applications in biochemical regulatory networks and epidemic models. First, we present PALLAS, a practical nonlinear state-space method for gene regulatory network (GRN) and protein-protein interaction network (PPI) inference from real-world time-series data, which employs penalized maximum likelihood and particle swarms for optimization. PALLAS is based on the Partially-Observed Boolean Dynamical System (POBDS) model and thus does not require ad-hoc binarization of the data. The penalty in the likelihood is a LASSO regularization term, which encourages the resulting network to be sparse. PALLAS is able to scale to networks of realistic size under no prior knowledge, by virtue of a novel continuous-discrete Fish School Search particle swarm algorithm for efficient simultaneous maximization of the penalized likelihood over the discrete space of networks and the continuous space of observational parameters. The accuracy and efficiency of PALLAS are demonstrated by a comprehensive set of experiments using synthetic data generated from real and artificial networks, as well as real time-series microarray and RNA-seq data, where it is compared to several other well-known methods.

In addition, we developed a similar state-space method to model the outbreak of the COVID 19. Mathematical models are widely recognized as an important tool to help people better understand the epidemic, predict its future trends, explore intervention scenarios and ultimately control the epidemic, such as lock-down or vaccination. We proposed a sophisticated spatial-temporal nonlinear state-space model based on a discrete-time susceptible - exposed - infected - recovered - deceased (SEIRD) model, which can estimate the hidden states and parameters from a noisy, incomplete, time series of reported epidemiological data, by applying Unscented Kalman Filter (UKF), Maximum

Likelihood (ML) adaptive filtering and Broyden–Fletcher–Goldfarb–Shanno (BFGS)/metaheuristic optimization. A comprehensive set of experiments, including simulations with different parameters set on the state model, and estimations by using the synthetic dataset, demonstrate our model can not only effectively simulate the different scenarios of the epidemic, such as spread with complex contagion patterns, lock-down patterns, and vaccination scenario, but also accurately estimate the unknown states and parameters which can be used for the prediction of the future trend.

DEDICATION

This dissertation is gratefully dedicated to my parents.

ACKNOWLEDGMENTS

First and foremost, I would like to gratefully thank my academic advisor, Dr. Ulisses M. Braga-Neto, not only for his guidance, unconditional encouragement and support in my research, but also the advice in my daily life over the years.

I would like to thank Dr. Fernando Buarque de Lima-Neto and Dr. Martial L. Ndeffo-Mbah for providing useful suggestions and guidance in my research.

I am also grateful to my committee members, Dr. Xiaoning Qian, Dr. Ivan Ivanov and Dr. Aniruddha Datta, for their constructive comments and constant support.

Special thanks goes to my parents who give me unwavering encouragement, even from hundreds of miles away. All of my success and achievements are undoubtedly because of their endless support.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee of Dr. Braga-Neto and Dr. Qian and Dr. Datta of the Department of Electrical & Computer Engineering and Dr. Ivanov of Department of Veterinary Medicine & Biomedical Sciences. All work for the dissertation was completed by the student, under the advisement of Dr. Braga-Neto of the Department of Electrical & Computer Engineering.

Funding Sources

This work was supported by National Science Foundation, through NSF awards CCF-1718924.

NOMENCLATURE

GRN	Gene Regulatory Network
PPI	Protein-Protein Interaction Network
POBDS	Partially-Observed Boolean Dynamical System
ML	Maximum Likelihood
PML	Penalized Maximum Likelihood
UKF	Unscented Kalman Filter
EKF	Extended Kalman Filter
NGS	Next Generation Sequencing
LC-MS	Liquid-Chromatography Mass-Spectrometry
APF	Auxiliary Particle Filter
BKF	Boolean Kalman Filter
GPS	Global Position System
SARS	Severe Acute Respiratory Syndrome
HPAI	Highly Pathogenic Avian Influenza
COVID 19	Coronavirus Disease
SIR model	Susceptible - Infected - Recovered Model
SEIRD	Susceptible - Exposed - Infected - Recovered- Decease
BNp	Boolean Networks with perturbation
PBN	Probabilistic Boolean Network
BCN	Boolean Control Network
ABC-SMC	Approximate Bayesian Computation Based on Sequential Monte Carlo Samplin

MMSE	Minimum Mean Square Error
SMC	Sequential Monte Carlo
SIS	Sequential Importance Sampling
SIR algorithm	Sampling Importance Resampling Algorithm
MC	Monte Carlo
APF-BKF	Auxiliary Particle Filter Implementation of the Boolean Kalman Filter
SIR-BKF	Sampling Importance Resampling Implementation of the Boolean Kalman Filter
EM	Expectation Maximization
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
FSS	Fish School Search
DFSS	Discrete Fish School Search
MFSS	Mixed Fish School Search
TPR	Sensitivity/True Positive Rate
SPC	Specificity/True Negative Rate
PPV	Precision/Positive Predictive Value

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Nonlinear state space model, which has long been considered in the statistic literature [1], has become a powerful tool for modeling and forecasting real-world dynamic systems. Such model, which has the ability to analyze and understanding hidden system behaviour through noisy indirect measurements, in conjunction with various variants Kalman filter, has been used in a wide range of applications, including Global position system (GPS) [2], target tracking [3, 4, 5], biological processes [6] and more. Usually, the state model can be derived from some prior knowledge about the process, but the system parameters, which cannot be calculated with established laws of nature, can only be inferred from the indirectly observed noisy time-series data. That is to say, most of the work in this area can be summarized as the problem of state and parameter estimation. However, with the number of the unknown system parameter increases, especially containing both discrete and continuous parameters, the computation complexity problem of parameter estimation is still largely unsolved. This dissertation is mainly focused on efficiently state and parameter estimation with applications on the area of computation biology.

1.1 Biological Networks

Biological networks such as gene regulatory networks (GRN), protein-protein interaction networks (PPI), can help build a picture of complex interactions that occur in cells. Identifying the structure and the parameters of these networks from time series data is undoubtedly one of the most important works in the system biology. With the advancement of high-throughput experimental technologies, such as next generation sequencing (NGS), Liquid-Chromatography Mass-Spectrometry (LC-MS), massive amounts of data make the inference of GRN and PPI networks possible. In this dissertation, we present PALLAS, a practical method for parametric GRN and PPI network inference based on partially-observed Boolean dynamical system (POBDS) model, using penalized maximum likelihood (PML) and particle swarms for optimization. PALLAS is a sophisticated state-space method that can detect edge directionality and activation/inhibition sta-

tus, without any prior knowledge, in addition to being capable of working directly on expression data, without the need for ad-hoc binarization. The penalty in the likelihood score is a L_1-norm LASSO regularization term [7], which encourages the resulting network to be sparse, i.e., contain a small number of edges between genes; its value can be adjusted by the user to obtain a desired level of sparsity. The likelihood itself is calculated efficiently by an auxiliary particle filter (APF) implementation of the Boolean Kalman Filter [8, 9]. Another novel feature of PALLAS is the application to Boolean models of a particle swarm method: a new mixed continuous-discrete version of the Fish School Search algorithm [10, 11], for efficient simultaneous maximization of the penalized likelihood over the discrete space of networks and the continuous space of observational parameters.

1.2 Mathematical Epidemiology Model

Infectious disease outbreaks remain a major threat to global health. This is especially the case for highly pathogenic and transmissible diseases with pandemic potential. These global threats were recently exemplified by the 2009 swine flu outbreak and the ongoing COVID-19 pandemic caused by the novel Severe Acute Respiratory Syndrome coronavirus 2 (SARS-CoV-2). To effectively mitigate and control the spread of a disease epidemic, it is paramount for public health decision-making to be informed by an accurate understanding of the dynamics of the epidemic and the potential impact of intervention measures. To this end, epidemic models have become an important tool to help people understand and take measures to prevent disease. Modern mathematical epidemiology models can be broadly divided into two main types: compartmental models and agent-based models. By virtue of simplicity and fairly scalable of the model, we propose a nonlinear state-space framework motivated by the compartmental model. The history of the compartment model can be traced back to the beginning of twentieth century, the most famous work by [12] whose susceptible - infected - recovered (SIR) model was used for modeling the cholera (London 1865) and plague (London 1665-1666, Bombay 1906) epidemics [13]. It is a deterministic model which is simple and well-understandable, however, there is a poor agreement between the observation and data because of the nonignorable randomness of the observation. In addition, many models cannot comprehensively

depict the complex contagion patterns in the real world which induced by modern transportation [14]. Furthermore, the most existing models assume all the parameters are known which is not the case in real world. To handle these issues, we came up with a sophisticated spatial-temporal stochastic dynamic model based on the nonlinear state space model combined with the well-known susceptible - exposed - infected - recovered - deceased model (SEIRD - a variant of SIR model), and the state and parameter estimation problems have been solved by using Unscented Kalman Filter (UKF), which has been widely used for nonlinear system, and Maximum Likelihood (ML) adaptive filtering combing with optimization (Broyden–Fletcher–Goldfarb–Shanno (BFGS)/metaheuristic) algorithms.

2. PALLAS: PENALIZED MAXIMUM LIKELIHOOD AND PARTICLE SWARMS FOR INFERENCE OF GENE REGULATORY NETWORKS AND PROTEIN-PROTEIN INTERACTION NETWORKS FROM TIME SERIES DATA*

2.1 Overview

PALLAS is a practical method for parametric GRN and PPI network inference based on partially-observed Boolean dynamical system (POBDS) model, using penalized maximum likelihood (PML) and particle swarms for optimization. The algorithm has two main components: 1) efficient computation of a penalized log-likelihood cost function, shown in Section 2.4; 2) maximization of the previous cost function using a novel particle swarm method, namely, a mixed discrete-continuous fish school search procedure, shown in Section 2.5. PALLAS is a fully-fledged program, written in python, and available on GitHub (`https://github.com/yukuntan92/PALLAS`).

2.1.1 Gene Regulatory Network

Inference of gene regulatory networks (GRN) from gene expression time-series data is a problem of critical importance in Bioinformatics [15]. Many mathematical models have been proposed in the literature to address this problem, including linear models [16, 17], Bayesian networks [18, 19], neural networks [20], differential equations [17, 21] and information theory based approaches [22, 23]. The Boolean network (BN) model [24], is an effective model for GRNs due to its ability to describe temporal patterns of gene activation and inactivation and its comparatively small data requirement for inference [25, 26, 27, 28, 29]. Several extensions of the BN model have been proposed, including Random Boolean Networks [24], Boolean Networks with perturba-

*Reprinted with permission from "Inference of gene regulatory networks by maximum-likelihood adaptive filtering and discrete fish school search." by Tan, Yukun, Fernando B. Lima Neto, and Ulisses-Braga Neto, 2018. IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Copyright 2018 by IEEE.

*Reprinted with permission from "Inference of Protein-Protein Interaction Networks from Liquid-Chromatography Mass-Spectrometry Data by Approximate Bayesian Computation-Sequential Monte Carlo Sampling." by Tan, Yukun, Fernando B. Lima Neto, and Ulisses Braga-Neto, 2020. IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), Copyright 2020 by IEEE.

*Reprinted with permission from "PALLAS: Penalized mAximum LikeLihood and pArticle Swarms for inference of gene regulatory networks from time series data." by Tan, Yukun, Fernando Lima Neto, and Ulisses Braga-Neto, 2020. IEEE/ACM Transactions on Computational Biology and Bioinformatics, Copyright 1969 by IEEE.

tion (BNp) [30], and Probabilistic Boolean Networks (PBN) [31], and Boolean Control Networks (BCN) [32, 33]. However, all of those models assume that the system Boolean states are completely observable. This is a significant drawback, since all practical methods for the inference of Boolean networks must include a step of ad-hoc binarization of the gene expression data. The Partially-observed Boolean dynamical system (POBDS) model [8] addresses this problem in a principled way, by postulating separate Boolean state and general observation processes. The time-series gene expression data, whether microarray or RNA-seq data, is modeled by the observation process, while the Boolean states are hidden. This allows the optimal inference of the sequence of Boolean states, as well as system parameters, from the time series data.

The GRN function of PALLAS is an extension of the adaptive filtering method proposed in [9]. The latter performs maximization of the likelihood function by exhaustive search over the space of networks and expectation maximization over the space of parameters of the observational model for each candidate network. It is well suited if there is prior knowledge about the network, e.g., most of the edges are known and only a few putative edges are being sought, given the prohibitive computational cost of exhaustively searching the space of all networks. As shown in Section 2.4, in the absence of prior knowledge about the gene interactions, the number of networks is given by $3^{d^2} \times 2^d$, where d is the number of genes. With only $d = 4$ genes, there are a total of 688,747,536 Boolean network models to be searched, and with $d = 10$ genes this number is larger than 10^{50}, rendering exhaustive search completely unfeasible. PALLAS differs from the method in [9] in using penalized maximum likelihood and particle swarms for optimization, which allows it to handle networks of realistic size in the absence of any prior knowledge.

The performance of PALLAS is demonstrated by a comprehensive set of experiments. Using synthetic data generated from both real and artificial GRNs, which allows computation of performance metrics, we compare PALLAS to regression-based methods, e.g. GENIE3 [34, 35], TIGRESS [36, 35]; Bayesian Networks methods, e.g. Banjo [37]; and Boolean network methods, e.g. Best-Fit algorithm [38], REVEAL [39], GABNI [40], and FBNNet [41]. Using real time series microarray data from the SOS DNA Repair System in E. Coli, we compare PALLAS to the meth-

ods in [42, 43, 44]. We also illustrate the performance of PALLAS in recovering known regulatory links in the E. Coli Biofilm Formation Pathway using time series RNA-Seq data.

2.1.2 Protein-Protein Interaction Network

Reconstructing the molecular networks underlying the functioning of a living cell is one of the main goals of biology and medicine. In this respect, protein-protein interaction (PPI) networks play a major role in most cellular processes. Inference of PPI networks from protein expression data is essential for understanding the structure, function, and dynamics of the cell [45]. With the advancement of high-throughput experimental technologies, such as Liquid-Chromatography Mass-Spectrometry (LC-MS), massive amounts of proteomics data make the reconstruction of PPI networks possible. Many methods for inference of PPI networks have been developed, including experimental [46, 47] and computational techniques [45, 48, 49, 50]. The Boolean network (BN) model [24] is an effective model which is widely used for inference gene regulatory networks (GRN) due to its ability to describe temporal patterns of gene activation and inactivation and its comparatively small data requirement.

The PPI inference function is developed based on GRN inference by applying Approximate Bayesian Computation based on Sequential Monte Carlo sampling (ABC-SMC) method [51] to handle the issue when the likelihood function is intractable in proteomics data. The performance of the proposed approach is assessed by numerical experiments based on a prototype immunomic network.

2.2 Partially-Observed Boolean Dynamical Systems (POBDS)

We have introduced the signal model of partially-observed Boolean dynamical systems (POBDS) which is a general class of nonlinear state-space models which allows for uncertainty in Boolean state transitions and partial observation of the Boolean state variables through noise.

2.2.1 State Model

Consider a state process $\{\mathbf{X}_k; k = 0, 1, \ldots\}$, where $\mathbf{X}_k \in \{0, 1\}^d$ is a Boolean vector of size d, which evolves according to

$$\mathbf{X}_k \;=\; \mathbf{f}\,(\mathbf{X}_{k-1}) \,\oplus\, \mathbf{n}_k\,, \tag{2.1}$$

for $k = 1, 2, \ldots$ where $\mathbf{f} \;:\; \{0, 1\}^d \longrightarrow \{0, 1\}^d$ is called the *network function*, $\mathbf{n}_k \in \{0, 1\}^d$ is additive noise at time k, and "$\oplus$" indicates component-wise modulo-2 addition. The state and noise processes are assumed to be independent. The state model (2.1) can be suitably modified to include external inputs, if desired.

The noise random vector $\mathbf{n}_k$ models uncertainty in the state transition: if a component of $\mathbf{n}_k$ is 1, the corresponding component of $\mathbf{f}\,(\mathbf{X}_{k-1})$ is flipped. As long as all components of $\mathbf{n}_k$ have a nonzero probability of being 1, the state process is an ergodic Markov Chain, with a steady state distribution. But if the noise is too intense, i.e., the probability of 1's in $\mathbf{n}_k$ is too large, state evolution becomes chaotic. However, it is well known that important biological pathways are tightly regulated. Accordingly, each component of the noise vector is assumed here to be equal to 1 with a small probability value p, independently of the others. The user can select a fixed value for p or allow the algorithm to treat p as a parameter to be estimated from the data. Results in the experiment Section 2.6.2.1 indicate that there is not a big difference between using a small fixed value $p = 0.05$ and estimating p in an interval $[0.01, 0.1]$, for the p53-MDM2 Negative-Feedback Loop Gene Regulatory Network.

We assume a specific model for the network function. Let a sample state vector $\mathbf{x} \in \{0, 1\}^d$ and the network function $\mathbf{f}$ be expressed in component form as $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{f} = (f_1, \ldots, f_d)$, respectively. Each component $f_i \;:\; \{0, 1\}^d \to \{0, 1\}$ is given by

$$f_i(\mathbf{x}) \;=\; \begin{cases} 1, & \sum_{j=1}^{d} a_{ij} x_j + b_i > 0\,, \\ 0, & \text{otherwise,} \end{cases} \tag{2.2}$$

where $a_{ij} = +1$ if there is positive regulation (activation) from gene/protein j to gene/protein i;

$a_{ij} = -1$ if there is negative regulation (inhibition) from gene/protein j to gene/protein i; and $a_{ij} = 0$ if gene/protein j is not an input to gene/protein i, whereas $b_i = +1/2$ if gene/protein i is positively biased in the sense that an equal number of activation and inhibition inputs will produce activation; the reverse being the case if $b_i = -1/2$. The network model is depicted in Figure 2.1, where the threshold units are step functions that output 1 if the input is positive, and 0, otherwise. This model constraint reduces the number of parameters needed to specify $\mathbf{f}$ from 2^d to $d^2 + d$.

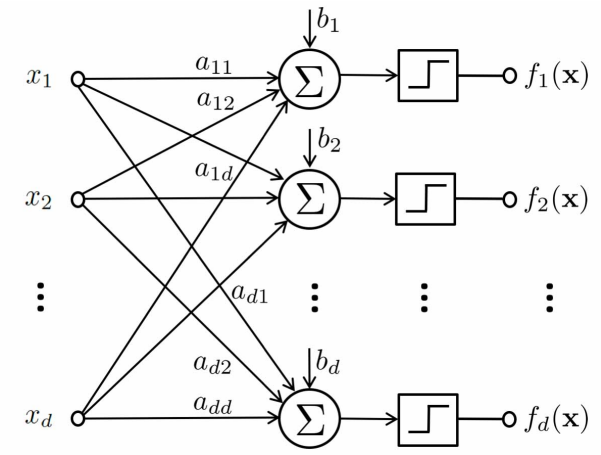

Figure 2.1: Schematic representation of the network function.

2.2.2 Observation Model

The sequence of states is observed indirectly through the process $\{\mathbf{Y}_k; k = 0, 1, \ldots\}$, where the measurement vector $\mathbf{Y}_k$ is a general nonlinear function of the state and observation noise:

$$\mathbf{Y}_k = \mathbf{h}(\mathbf{X}_k, \mathbf{v}_k) \tag{2.3}$$

for $k = 1, 2, \ldots$, where the noise vector $\mathbf{v}_k$ is assumed to independent of the state process and state transition noise process. We describe next the three observational models considered in the package, corresponding to two common gene expression modalities: RNA-Seq count data or microarray fluorescence data and one protein expression modality: liquid chromatography - mass spectrometry

(LC-MS) data. Observational models for other data modalities can be introduced, if desired.

2.2.2.1 RNA-Seq Observation Model

RNA-Seq data can be modeled with the Poisson distribution [52] or the negative binomial distribution [53, 54]. Here, we employ the latter, since it is able to address overdispersion in the count distributions. We assume that the transcript counts $\mathbf{Y}_k = (Y_{k1}, \ldots, Y_{kd})$ are related to the state $\mathbf{X}_k = (X_{k1}, \ldots, X_{kd})$ via

$$P(\mathbf{Y}_k = \mathbf{y} \mid \mathbf{X}_k = \mathbf{x}) = \prod_{i=1}^{d} P(Y_{ki} = y_i \mid X_{ki} = x_i), \tag{2.4}$$

and adopt the negative binomial model for each count,

$$\begin{aligned} &P(Y_{ki} = y_i \mid X_{ki} = x_i) = \\ &\quad \frac{\Gamma(y_i + \phi_i)}{y_i!\,\Gamma(\phi_i)} \left(\frac{\lambda_i}{\lambda_i + \phi_i}\right)^{y_i} \left(\frac{\phi_i}{\lambda_i + \phi_i}\right)^{\phi_i}, \end{aligned} \tag{2.5}$$

where Γ denotes the Gamma function, and $\phi_i, \lambda_i > 0$ are the real-valued inverse dispersion parameter and mean read count of transcript i, respectively, for $i = 1, \ldots, d$. The inverse dispersion parameter ϕ_i specifies the amount of observation noise: the larger it is, the less observation noise is present. We model the parameter λ_i in log-space as:

$$\log \lambda_i = \log s + \mu_i + \delta_i x_i, \tag{2.6}$$

where the parameter s is the sequencing depth, which depends on the instrument, $\mu_i \geq 0$ is the baseline level of expression in the inactivated transcriptional state, and $\delta_i > 0$ is the difference between read count as gene i goes from the inactivated ($x_i = 0$) to the activated ($x_i = 1$) state, for $i = 1, \ldots, d$.

2.2.2.2 *Microarray Observation Model*

A reasonable model for continuous microarray fluorescence data is a Gaussian linear model:

$$\mathbf{y} \;=\; \boldsymbol{\mu} + D\,\mathbf{x} + \mathbf{v}\,, \tag{2.7}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d) \geq 0$ is the vector of baseline expression levels corresponding to the "zero" or inactive state for each gene, $D = \mathrm{diag}\{\delta_1, \ldots, \delta_d\} > 0$ is a diagonal matrix containing differential expression values for each gene, and $\mathbf{v} \sim \mathcal{N}(0, \Sigma)$ is an uncorrelated zero-mean Gaussian noise vector, where $\Sigma = \mathrm{diag}\{\sigma_1^2, \ldots, \sigma_d^2\} > 0$. Notice that (2.4) is still satisfied here.

2.2.2.3 *Liquid Chromatography-Mass Spectrometry Observation Model*

In what follows, we consider the observational model for liquid chromatography-mass spectrometry (LC-MS) data proposed in [55], which we describe briefly next. The protein concentration can be modeled as a Gamma distribution [56],

$$\gamma_i \;=\; \Gamma(\kappa, \vartheta),\; i \;=\; 1, 2, \ldots, d, \tag{2.8}$$

where the shape κ and scale ϑ parameters are assumed to be uniform random variables, such that $\kappa \sim \mathrm{Unif}(\kappa_{low}, \kappa_{high})$ and $\vartheta \sim \mathrm{Unif}(\vartheta_{low}, \vartheta_{high})$. The multivariate Gaussian distribution is recommended as the model for protein concentration variations. In this paper, we assume that protein concentrations are mutually independent, so that

$$y_i = \gamma_i + \gamma_i(\zeta_i - 1)\, x_i + \upsilon_i, \tag{2.9}$$

for $i = 1, \ldots, d$, where γ_i is the baseline of protein concentration expression levels, ζ_i is the fold change when protein i is overexpressed, and υ_i is an uncorrelated zero-mean Gaussian noise for protein i which $\boldsymbol{\upsilon} \sim \mathcal{N}(0, \Sigma)$, where $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ and $\sigma_i^2 = \varphi \times \gamma_i^2$. The coefficient of variation φ is calibrated based on the observed data.

2.3 Boolean State Estimation

2.3.1 Boolean Kalman Filter

Given a time series of observations $\mathbf{Y}_{1:k} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_k\}$, we would like to find an estimator $\hat{\mathbf{X}}_k = \mathbf{g}(\mathbf{Y}_{1:k})$ of the state $\mathbf{X}_k$ that minimizes the conditional MSE:

$$\hat{\mathbf{X}}_k^{\text{MS}} = \underset{\hat{\mathbf{X}}_k \in \{0,1\}^d}{\operatorname{argmin}} E[\,||\hat{\mathbf{X}}_k - \mathbf{X}_k||^2 \mid \mathbf{Y}_{1:k}]\,. \tag{2.10}$$

For a vector $\mathbf{u} \in [0,1]^d$, define the threshold operator $\overline{\mathbf{u}} \in \{0,1\}^d$ as $\overline{\mathbf{u}}(i) = 1$ if $\mathbf{u}(i) > 1/2$ and 0 otherwise, for $i = 1, \ldots, d$, respectively. It was shown in [8, 9] that

$$\hat{\mathbf{X}}_k^{\text{MS}} = \overline{E\left[\mathbf{X}_k \mid \mathbf{Y}_{1:k}\right]}\,. \tag{2.11}$$

Assuming that all the parameters of the state and observation models *are known*, then the optimal MMSE filter in (2.11) can be calculated exactly by a recursive procedure called the Boolean Kalman filter (BKF) [8, 9], which is described briefly next.

Let $(\mathbf{x}^1, \ldots, \mathbf{x}^{2^d})$ be an arbitrary enumeration of all state vectors. Define the state conditional probability distribution vector $\mathbf{\Pi}_{k|k}$ with components

$$(\mathbf{\Pi}_{k|k})_i = P\left(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{Y}_{1:k}\right)\,,\ \ i = 1, \ldots, 2^d, \tag{2.12}$$

for $k = 0, 1, \ldots$ According to equation (2.11),

$$\hat{\mathbf{X}}_k^{\text{MS}} = \overline{E\left[\mathbf{X}_k \mid \mathbf{Y}_{1:k}\right]} = \overline{A\mathbf{\Pi}_{k|k}}\,,\ \ k = 1, 2, \ldots, \tag{2.13}$$

where $A = \left[\mathbf{x}^1 \cdots \mathbf{x}^{2^d}\right]$ is a matrix of size $d \times 2^d$.

The computation of $\mathbf{\Pi}_{k|k}$ can be performed recursively. First, we have

$$\mathbf{\Pi}_{k|k-1} = M_k \mathbf{\Pi}_{k-1|k-1}\,,\ \ k = 1, 2, \ldots \tag{2.14}$$

where M_k is the *transition matrix* of the Markov state process, with entries

$$\begin{aligned}(M_k)_{ij} &= P(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j) \\ &= P(\mathbf{n}_k = \mathbf{x}^i \oplus \mathbf{f}(\mathbf{x}^j)), \;\; i, j = 1, \ldots, 2^d.\end{aligned} \tag{2.15}$$

On the other hand,

$$\mathbf{\Pi}_{k|k} \propto T(\mathbf{Y}_k)\, \mathbf{\Pi}_{k|k-1}, \;\; k = 1, 2, \ldots \tag{2.16}$$

where "$\propto$" means that the result must be normalized to add up to 1, and the *update matrix* $T(\mathbf{Y}_k)$ is diagonal of size $2^d \times 2^d$, with diagonal elements:

$$\left(T_k(\mathbf{Y}_k)\right)_{ii} = p\left(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}^i\right), \;\; i = 1, \ldots, 2^d. \tag{2.17}$$

2.3.2 Auxiliary Particle Filter Implementation of BKF

When the network is large, however, the exact computation of the BKF is intractable since each transition matrix contain 2^{2d} elements which requires large computational and memory. In this case, approximate methods must be used, such as the Sequential Monte Carlo (SMC) method, also known as particle filter [57].

The sequential importance sampling (SIS) algorithm is the most basic Monte Carlo (MC) method but forms the basis of most SMC filters. SIS algorithm is based on importance sampling which is widely used when it is difficult to directly sample from the target distribution $p(x)$, but much easier to sample from the proposal distribution $q(x)$. The idea of SIS is to approximate the required posterior density function by a set of "particles" drawn from proposal distribution with corresponding weights, and recursively update these particles.

The mainly drawback of SIS is the degeneracy phenomenon, where after a few iterations, only a few of the particles will have a significant weight and all the others will have very small weights [58]. To address this problem, sampling importance resampling (SIR) algorithm has been proposed in which resampling approach is applied in each iteration [59]. The original particle filtering im-

plementation of the BKF in [60] was based on the SIR algorithm, which can be called the SIR-BKF algorithm.

However, the SIR filter is sensitive to outliers since the state space is explored without any knowledge of the observations. A more sophisticated implementation of the BKF based on the Auxiliary Particle Filter (APF) is proposed in [61] called APF-BKF, which will be used in our framework. APF is a look-ahead method that at time step $k-1$ tries to predict the location of particles with high probability at time k, with the purpose of making the subsequent resampling step more efficient. Without the look-ahead, the basic SIR algorithm blindly propagates all particles, even those in low probability regions.

The APF was introduced by [62], derived from the SIS algorithm by introducing an importance density $q(\mathbf{X}_k, \xi_k|\mathbf{Y}_{1:k})$ with auxiliary variable ξ_k which represents as the index of the particles at $k-1$. Particles can be drawn from $p(\mathbf{X}_k, \xi_k|\mathbf{Y}_{1:k})$ and omitting the auxiliary variable in pair $(\mathbf{x}_k, i)$ to produce a sample from the marginalized density $p(\mathbf{X}_k|\mathbf{Y}_{1:k})$. Given particles $\{\mathbf{x}_{k-1,i}\}_{i=1}^N$ at time $k-1$ and associated weights $\{W_{k-1,i}\}_{i=1}^N$, distribution $p(\mathbf{X}_k, \xi_k|\mathbf{Y}_{1:k})$ can be factored as:

$$\begin{aligned} & p(\mathbf{X}_k, \xi_k \mid \mathbf{Y}_{1:k}) \\ & \propto p(\mathbf{Y}_k \mid \mathbf{X}_k)\, p(\mathbf{X}_k, \xi_k \mid \mathbf{Y}_{1:k-1}) \\ & = p(\mathbf{Y}_k \mid \mathbf{X}_k)\, p(\mathbf{X}_k \mid \mathbf{X}_{k-1}, \xi_k)\, p(\mathbf{X}_{k-1}, \xi_k \mid \mathbf{Y}_{1:k-1}) \\ & = p(\mathbf{Y}_k \mid \mathbf{X}_k)\, p(\mathbf{X}_k \mid \mathbf{x}_{k-1,\xi_k})\, W_{k-1,\xi_k}, \end{aligned} \tag{2.18}$$

for $\xi_k = 1, \ldots, N$. The importance density used to draw samples is defined as:

$$q(\mathbf{X}_k, \xi_k \mid \mathbf{Y}_{1:k}) \propto p(\mathbf{X}_k \mid \mathbf{x}_{k-1,\xi_k})\, p(\mathbf{Y}_k \mid \mu_{k,\xi_k})\, W_{k-1,\xi_k}, \tag{2.19}$$

for $\xi_k = 1, \ldots, N$, where $\mu_{k,i}$ is the mode of $\mathbf{X}$ given $\mathbf{x}_{k-1,i}$ in our implementation:

$$\begin{aligned} \mu_{k,i} &= Mode[\mathbf{X}_k \mid \mathbf{x}_{k-1,i}] \\ &= Mode[\mathbf{f}(\mathbf{x}_{k-1,i}) \oplus \mathbf{n}_{k,i}] = \mathbf{f}(\mathbf{x}_{k-1,i}), \end{aligned} \tag{2.20}$$

for $i = 1, \dots, N$, where the noise is zero-mode required by APF.

Sampling from (2.19) is done in two steps. In the first step, $\{\mu_{k,i}\}_{i=1}^{N}$ is obtained from the particles $\{\mathbf{x}_{k-1,i}\}_{i=1}^{N}$ using (2.20) and the first-stage weights or the presampling probability $\{V_{k,i}\}_{i=1}^{N}$ can be presented as:

$$V_{k,i} = p(\mathbf{Y}_k \mid \mu_{k,i})\, W_{k-1,i}, \tag{2.21}$$

for $i = 1, \dots, N$. In the second step, the auxiliary variables $\{\xi_{k,i}\}_{i=1}^{N}$ are drawn with probability $V_{k,i}$ which is a discrete distribution. We denote this by $\{\xi_{k,i}\}_{i=1}^{N} \sim Cat(\{V_{k,i}\}_{i=1}^{N})$, where "$Cat$" stands for the categorical (discrete) distribution.

Finally, the new particles $\{\mathbf{x}_{k,i}\}_{i=1}^{N}$ and associated second-stage weights $\{\tilde{W}_{k,i}\}_{i=1}^{N}$ can be computed as:

$$\mathbf{x}_{k,i} = \mu_{k,\xi_{k,i}} \oplus \mathbf{n}_{k,i} \sim p(\mathbf{X}_k \mid \mathbf{x}_{k-1,\xi_{k,i}}), \tag{2.22}$$

$$\tilde{W}_{k,i} \propto \frac{p(\mathbf{X}_k, \xi_k \mid \mathbf{Y}_{1:k})}{q(\mathbf{X}_k, \xi_k \mid \mathbf{Y}_{1:k})} = \frac{p(\mathbf{Y}_k \mid \mathbf{x}_{k,i})}{p(\mathbf{Y}_k \mid \mu_{k,\xi_{k,i}})}. \tag{2.23}$$

Based on [63], the unbiased estimator of the unnormalized posterior probability at each time step is:

$$||\hat{\beta}_k||_1 = \left(\frac{1}{N}\sum_{i=1}^{N} V_{k,i}\right)\left(\frac{1}{N}\sum_{i=1}^{N} \tilde{W}_{k,i}\right). \tag{2.24}$$

This is needed in (2.29) when maximum-likelihood adaptive estimation is discussed.

Based on (2.11) and normalized second-stage weights $W_{k,i} = \tilde{W}_{k,i} / \sum_{i=1}^{N} \tilde{W}_{k,i}, i = 1, \dots, N$, we can write

$$\hat{\mathbf{X}}_k^{\mathrm{MS}} = \overline{E\left[\mathbf{X}_k \mid \mathbf{Y}_{1:k}\right]} \approx \overline{\sum_{i=1}^{N} W_{k,i}\mathbf{x}_{k,i}}. \tag{2.25}$$

2.4 Parameter Estimation

In the previous section, we have assumed that the system parameters are known and only the state needs to be estimated. However, in practical models, the parameters are unknown as well. The parameter estimation methods can be divided into two main categories: maximum-likelihood (ML) and Bayesian approaches. In the PALLAS, we are using the latter case and we apply ABC-

Algorithm 1 APF-BKF: Auxiliary Particle Filter implementation of the Boolean Kalman Filter

$\mathbf{x}_{0,i} \sim \Pi_{0|0}, W_{0,1} = 1/N$, for $i = 1, \ldots, N$.
for $k = 1, 2, \ldots,$ **do**:
  for $i = 1$ to N **do**
    $\mu_{k,i} = \mathbf{f}(\mathbf{x}_{k-1,i})$.
    $V_{k,i} = p(\mathbf{Y}_k|\mu_{k,i})W_{k-1,i}$.
  end for
  $\{\xi_{k,i}\}_{i=1}^N \sim Cat(\{V_{k,i}\}_{i=1}^N)$.
  for $i = 1$ to N **do**
    $\mathbf{x}_{k,i} = \mu_{k,\xi_{k,i}} \oplus \mathbf{n}_{k,i}$.
    $\tilde{W}_{k,i} = \frac{p(\mathbf{Y}_k|\mathbf{x}_{k,i})}{p(\mathbf{Y}_k|\mu_{k,\xi_{k,i}})}$.
  end for
  $||\hat{\beta}_k||_1 = (\frac{1}{N}\sum_{i=1}^N V_{k,i})(\frac{1}{N}\sum_{i=1}^N \tilde{W}_{k,i})$.
  $W_{k,i} = \tilde{W}_{k,i}/\sum_{j=1}^N \tilde{W}_{k,j}, i = 1, \ldots, N$.
  $\mathbf{z}_k = \sum_{i=1}^N W_{k,i}\mathbf{x}_{k,i}$.
  $\hat{\mathbf{X}}_k^{MS} = \overline{\mathbf{z}_k}$.
  $\text{MSE}(\hat{\mathbf{X}}_k^{MS}|\mathbf{Y}_{1:k}) = ||min\{\mathbf{z}_k, \mathbf{z}_k^c\}||_1$.
end for

SMC algorithm to estimate the likelihood when it is intractable with proteomics data. Expectation-maximization (EM) algorithm is always used for finding the maximum when the system parameters are continuous, however, when the system parameters contain both discrete and continuous, the computation complexity is too high. In this case, we proposed a new metaheuristic optimization method described in Section 2.5 which can get rid of EM algorithm and exhaustive search.

Let $\theta = (\theta_{\text{disc}}, \theta_{\text{cont}}) \in \Theta$, with $\theta_{\text{disc}} \in \Theta_{\text{disc}}$ and $\theta_{\text{cont}} \in \Theta_{\text{cont}}$, be the discrete and continuous unknown model parameters, where Θ, Θ_{disc} and Θ_{cont} are the corresponding parameter spaces, with $\Theta = \Theta_{\text{disc}} \times \Theta_{\text{cont}}$. Here, θ_{disc} contains the parameters of the network function in (2.2), namely the edge parameters $a_{ij} \in \{-1, 0, 1\}$, for $i, j = 1, \ldots, d$, and the regulation bias parameters $b_i \in \{-1/2, 1/2\}$, for $i = 1, \ldots, d$. Hence, $\Theta_{\text{disc}} = \{-1, 0, 1\}^{d^2} \times \{-1/2, 1/2\}^d$. This is a finite space, but its cardinality $|\Theta_{\text{disc}}| = 3^{d^2} \times 2^d$ increases extremely fast with the number of genes d. For example,

for a network with only $d = 4$ genes, $|\Theta_{\text{disc}}| = 688747536$, while if $d = 8$, then $|\Theta_{\text{disc}}| \approx 8.8 \times 10^{32}$. On the other hand, θ_{cont} contains the observational parameters: 1) GRN: the baseline expression levels $\mu_i > 0$ and the differential expression levels $\delta_i > 0$, for $i = 1, \ldots, d$, for both RNA-Seq and microarray data, the inverse dispersion parameters $\phi_i > 0$, for $i = 1, \ldots, d$, for RNA-Seq data, and the standard deviations $\sigma_i > 0$, for $i = 1, \ldots, d$, for microarray data (the sequencing depth parameter s is assumed known for a given RNA-seq assay, so it is not part of θ_{cont}). Hence, the dimensionality of θ_{cont} is $Q = 3d$ in both cases; 2) PPI: the shape κ and scale ϑ of the baseline of protein concentration expression level, the fold change ζ, and the coefficient of variation φ.

2.4.1 Penalized Maximum-Likelihood Adaptive Filtering

Suppose that the sample data consist of n independent time series $\mathbf{Y}_{1:k}^j = \{\mathbf{Y}_1^j, \ldots, \mathbf{Y}_k^j\}$ up to time k, for $j = 1, \ldots, n$. The penalized log-likelihood of model θ at time k is defined as

$$\begin{aligned} L_k(\theta) &= \frac{1}{kn} \log\, p_\theta(\mathbf{Y}_{1:k}^{(1)}, \ldots, \mathbf{Y}_{1:k}^{(n)}) - \eta \sum_{i,j=1}^{2^d} |a_{ij}| \\ &= \frac{1}{kn} \sum_{j=1}^{n} \log\, p_\theta(\mathbf{Y}_{1:k}^j) - \eta \sum_{i,j=1}^{2^d} |a_{ij}|\,, \end{aligned} \tag{2.26}$$

where $\eta > 0$ is a regularization parameter, which has a default value of $\eta = 0.01$ in our implementation. Hence, the penalized log-likelihood in (2.26) is the sum of the average log-likelihood per time series and a negative value times the number of edges in the model. Maximization of (2.26) thus encourages the model to both fit the data and be sparse, i.e., contain a small number of edges between genes, which is in agreement with biological knowledge. The value of η can be adjusted by the user to obtain a desired level of sparsity. Notice that

$$\begin{aligned} \log\, p_\theta(\mathbf{Y}_{1:k}^j) &= \log\big[\, p_\theta(\mathbf{Y}_k^j \mid \mathbf{Y}_{1:k-1}^j) p_\theta(\mathbf{Y}_{k-1}^j \mid \mathbf{Y}_{1:k-2}^j) \\ &\qquad \cdots\, p(\mathbf{Y}_2^j \mid \mathbf{Y}_1^j) p(\mathbf{Y}_1^j)\big] \\ &= \sum_{m=1}^{k} \log\, p_\theta(\mathbf{Y}_m^j \mid \mathbf{Y}_{1:m-1}^j)\,, \end{aligned} \tag{2.27}$$

where

$$
\begin{aligned}
& p_\theta(\mathbf{Y}_m^j \mid \mathbf{Y}_{1:m-1}^j) \\
&= \sum_{i=1}^{2^d} p_\theta(\mathbf{Y}_m^j \mid \mathbf{X}_m = \mathbf{x}^i,\, \mathbf{Y}_{1:m-1}^j)\; P_\theta(\mathbf{X}_m = \mathbf{x}^i \mid \mathbf{Y}_{1:m-1}^j) \\
&= \sum_{i=1}^{2^d} p_\theta(\mathbf{Y}_m^j \mid \mathbf{X}_m = \mathbf{x}^i)\; P_\theta(\mathbf{X}_m = \mathbf{x}^i \mid \mathbf{Y}_{1:m-1}^j)
\end{aligned}
\tag{2.28}
$$

With $(\beta_m^{\theta,j})_i = p_\theta(\mathbf{Y}_m^j \mid \mathbf{X}_m = \mathbf{x}^i)\; P_\theta(\mathbf{X}_m = \mathbf{x}^i \mid \mathbf{Y}_{1:m-1}^j)$, the penalized log-likelihood in (2.26) be written as

$$
L_k(\theta) \;=\; \frac{1}{kn} \sum_{j=1}^{n} \sum_{m=1}^{k} ||\beta_m^{\theta,j}||_1 \;-\; \eta \sum_{i,j=1}^{2^d} |a_{ij}| \,. \tag{2.29}
$$

When we do the GRN inference, the sequence of values $||\beta_m^{\theta,j}||_1$, for $j = 1, \ldots, n$ and $m = 1, \ldots, k$, can be computed by a BKF tuned to parameter θ applied to the time series $\mathbf{Y}_{1:k}^j$. As mentioned in the previous section, here we use the auxiliary particle filtering implementation of the BKF, for computational efficiency. On the other hand, when we do the PPI network inference, $p_\theta(\mathbf{Y}_k^j \mid \mathbf{Y}_{1:k-1}^j)$ can only be estimated by ABC-SMC algorithm shown in Section 2.4.2.

The maximum-likelihood estimator of parameter θ at time k is then given by

$$
\hat{\theta}_k^{\mathrm{ML}} \;=\; \arg\max_{\theta \in \Theta} L_k(\theta) \,. \tag{2.30}
$$

A state estimate $\hat{\mathbf{X}}_k^{\mathrm{ML}} = \hat{\mathbf{X}}_k(\hat{\theta}_k^{\mathrm{ML}})$ can be obtained, if desired, where $\hat{\mathbf{X}}_k(\theta)$ denotes the optimal state estimator produced by a BKF tuned to the parameter θ.

2.4.2 ABC-SMC Algorithm

In our previous work [64], we use an auxiliary particle filter algorithm to estimate the likelihood and obtain the unknown parameters by maximum the estimation. However, this method cannot be used for complex models, in which the conditional density $\mathbf{g}(\mathbf{Y}|\mathbf{X})$ is intractable or computationally expensive. Instead, we will assume that one may still able to obtain samples from this conditional likelihood for different values of the parameter θ, which leads to the Approximate Bayesian Computation (ABC) technique [51]. ABC replaces the calculation of the likelihood with a comparison

between the observed and sampled data to approximate the likelihood, namely, we generate M samples from $\mathbf{g}_\theta(\mathbf{Y}_k|\mathbf{X}_k)$ and the estimated likelihood can be calculated as

$$l_k = \frac{\sum_{j=1}^{M} \mathbb{1}(d(\hat{\mathbf{Y}}_k^j, \mathbf{Y}_k) \leq \epsilon_k)}{M} \tag{2.31}$$

for $k = 1, \ldots, T$, where ϵ_k is the precision tolerance, and $d(\cdot, \cdot)$ is the distance function between the observed and sampled data. Theoretically, the approximation obtained by ABC filtering is matched to true one when $\epsilon_k \approx 0$ and $M = \infty$.

However, a drawback of the ABC method is the low acceptance rate when stuck in a bad region. In order to improve the ABC performance, the use of Sequential Monte Carlo (SMC) sampling has been suggested [65, 66, 67]. In the SMC algorithm, $\frac{1}{N}\sum_{i=1}^{N} \mathbf{g}(\mathbf{Y}_k \mid \mathbf{X}_{k,i})$ is an approximation to the conditional likelihood $p(\mathbf{Y}_k \mid \mathbf{Y}_{1:k-1})$. Thus, with the estimated likelihood from the ABC algorithm, the full likelihood approximation $p(\mathbf{Y}_{1:k})$ can be generated.

In Algorithm 2 we present the ABC-SMC algorithm based on [65, 67]. The basic design elements are the number of particles N, the number of auxiliary observation samples M and the ABC precision tolerance ϵ. The vector $\Pi_{0|0}$ is the initial (prior) distribution of the states at time zero. The vector $\mathbf{W}$ gives the weight of the particles, which is initialized to $1/N$ for all particles. The resampling step is necessary when the effective sample size (ESS) is low. The resampling threshold E is commonly taken to be $N/2$ [68].

Based on [65, 67],

$$p_\theta(\mathbf{Y}_k^j \mid \mathbf{Y}_{1:k-1}^j) = \frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{W}}_{m,i}^{\theta,j}, \tag{2.32}$$

where (2.26) can finally be written as

$$L_k(\theta) = \frac{1}{kn}\sum_{j=1}^{n}\sum_{m=1}^{k} \log\left(\frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{W}}_{k,i}^{\theta,j}\right) - \eta \sum_{i,j=1}^{2^d} |a_{ij}|\,. \tag{2.33}$$

Algorithm 2 ABC-SMC

Initialize $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T > 0$ and
$\mathbf{X}_{0,i} \sim \Pi_{0|0}, \mathbf{W}_{0,i} = 1/N$, for $i = 1, 2, \ldots, N$
for $k = 1$ to T **do**:
  for $i = 1$ to N, **do**:
    $\mathbf{X}_{k,i} = \mathbf{f}(\mathbf{X}_{k-1,i}) \oplus \mathbf{n}_{k,i}$
    for $j = 1$ to M **do**:
      Generate $\hat{\mathbf{Y}}^j_{k,i} \sim \mathbf{g}_\theta(\cdot|\mathbf{X}_{k,i})$
    end for
    $\tilde{\mathbf{W}}_{k,i} = \frac{\sum_{j=1}^{M} \mathbb{1}(d(\hat{\mathbf{Y}}^j_{k,i}, \mathbf{Y}_{k,i}) \leq \epsilon_k)}{M}$
    $\mathbf{W}_{k,i} \propto \mathbf{W}_{k-1,i}\tilde{\mathbf{W}}_{k,i}$
  end for
  $\mathbf{W}_{k,i} = \mathbf{W}_{k,i} / \sum_{i=1}^{N} \mathbf{W}_{k,i}$
  If $ESS = [\sum_{i=1}^{N} (\mathbf{W}_{k,i})^2]^{-1} \leq E$:
    Resample $\mathbf{X}_{k,i}$ with weights $\mathbf{W}_{k,i}$
    Set $\mathbf{W}_{k,i} = 1/N$
end for

2.5 Metaheuristic Optimization

2.5.1 Overview

Metaheuristic methods have been widely used to tackle all kinds of optimization problems. The goal when using such algorithms is to find a solution that is good enough to solve the problem in a feasible amount of time, which would not be possible using exact methods due to the size and complexity of some problems. Algorithms, such as the particle swarm optimization (PSO)[69], ant colony optimization (ACO)[70] and fish school search (FSS)[71], use nature inspired mechanisms to guide particles through the search space aiming to find a position which represents a better solution for the problem at each iteration. We are using FSS as the foundation of our new proposed algorithm since the ability to switch automatically between exploration and exploitation modes and its modular concept.

The FSS algorithm [71] operates in a search space of continuous variables. Its main charac-

teristic is the ability of switching from exploration to exploitation and vice-versa automatically according to the state of the fish school. Furthermore, it incorporates the concept of weight for the fish enabling that more successful fish get heavier and therefore have more influence over the fish school. However, the original FSS is also only for the continuous parameters, so the discrete version of FSS (DFSS) has been proposed [72]. Combine original FSS and DFSS, we call it mixed fish school search algorithm (MFSS) which can be used to search discrete and continuous space simultaneously.

2.5.2 Proposed Mixed Fish School Search Algorithm

In this section, we describe in detail a novel particle-swarm optimization algorithm for discrete-continuous parameter search, called the mixed fish school search (MFSS) algorithm. One of the main novelties in the MFSS algorithm is the ability to operate on large continuous and discrete parameter spaces simultaneously, which is needed to infer the continuous noise parameters of the observation process, in addition to the discrete parameters of the GRN itself. As the original algorithm, MFSS has a few properties that are unique among most particle swarm optimization techniques, namely, the ability to switch automatically between exploration and exploitation modes and its modular concept.

In the MFSS algorithm, the objective is to find a model that maximizes a given score or fitness — in our present case, this is the penalized log-likelihood defined in the previous section. Each candidate model, i.e., each candidate parameter vector $\theta = (\theta_{\text{disc}}, \theta_{\text{cont}})$, corresponds to a particle or "fish." The length of θ is denoted by P. From the previous section, $P = d^2 + d + Q$. The fish school is an ensemble of S such particles in the parameter space $\Theta = \Theta_{\text{disc}} \times \Theta_{\text{cont}}$. The position of fish s at iteration r will be denoted by $\theta^s(r) = (\theta^s_{\text{disc}}(r), \theta^s_{\text{cont}}(r))$, for $s = 1, \ldots, S$, and $r = 0, \ldots, R$. The number of fishes S and the total number of iterations R are user-defined parameters (in practice, $S = 3 \times P$ and $R = 5000$ are found to be good values). In addition, each fish s has a weight $w_s(r)$ at iteration r, which reflects the quality of the solution.

Initialization. The initial position $\theta^s(0) = (\theta^s_{\text{disc}}(0), \theta^s_{\text{cont}}(0))$ of each fish is assigned randomly. The continuous vector $\theta^s_{\text{cont}}(0)$ is drawn from a uniform distribution over Θ_{cont}, but for the discrete

part, it is advantageous to use a non-uniform distribution to initialize the edge parameters, in such a way that $a_{ij}^s(0)$ is equal to -1 or 1 with probability $1/4$, and 0 with probability $1/2$, for $i, j = 1, \ldots, d$, which introduces a bias towards 0 over 1 and -1. This is in agreement with the biological observation that GRNs tend to be sparsely connected. The initial value $b_i^s(0)$ of the regulation bias parameter is chosen to be either $-1/2$ or $1/2$ with equal probabilities, for $i = 1, \ldots, d$.

Individual movement operator. This is an exploratory step, where each fish independently moves a short distance in a random direction, as long as this increases the fitness function. Let $\Delta\theta_{\text{ind}}^s(r) = (\Delta\theta_{\text{disc,ind}}^s(r), \Delta\theta_{\text{cont,ind}}^s(r))$ be the (candidate) individual displacement vector for fish s at iteration r. Vector $\Delta\theta_{\text{disc,ind}}^s(r)$ is drawn from a uniform distribution over the rectangular region $[-1, 1]^{d^2+d}$, while $\Delta\theta_{\text{cont,ind}}^s(r)$ is drawn from a uniform distribution over the rectangular region $[-\tau_1(r), \tau_1(r)] \times \cdots \times [-\tau_Q(r), \tau_Q(r)]$. The step size bounds $\tau_q(r)$, for $q = 1, \ldots, Q$, shrink linearly with r, in order to ensure convergence and emphasize exploitation over exploration at later iterations. In our implementation, the initial and final values $\tau_q(1)$ and $\tau_q(R)$ are set, respectively, to 10% and 0.01% of the range (i.e., the difference between upper and lower bounds) of the corresponding continuous parameter — these values can be modified by the user, if desired. Now, $\Delta\theta_{\text{disc,ind}}^s(r)$ needs to be quantized into the lattice $\{-1, 0, 1\}^{d^2+d}$ in order to be added to the discrete component of the current fish position. The quantization scheme we adopt here is a generalization of the method for binary parameters in [73]. We define two adaptive thresholds:

$$\begin{aligned} \text{thr}_{\text{pos}}^s(r) &= \max\nolimits_+(\Delta\theta_{\text{disc,ind}}^s(r)) \times \tfrac{r}{R}\,, \\ \text{thr}_{\text{neg}}^s(r) &= \min\nolimits_-(\Delta\theta_{\text{disc,ind}}^s(r)) \times \tfrac{r}{R}\,, \end{aligned} \tag{2.34}$$

where the operator $\max_+(\mathbf{v})$ is equal to the maximum of the components of vector $\mathbf{v}$ if at least one of them is positive, and equal to zero, otherwise; similarly, $\min_-(\mathbf{v})$ is equal to the minimum of the components of $\mathbf{v}$ if at least one of them is negative, and equal to zero, otherwise. The factor r/R increases the thresholds (in magnitude) with r, to favor exploitation over exploration at later iterations and ensure convergence. Exploitation could be understood as an analogy to tree depth-

first search as opposed to exploration, which would equate to tree breadth-first search. In exploration mode the algorithm widens the search in the parameter space, while in exploitation mode, the algorithm attempts to get a more accurate result in a small area of the parameter space.

The quantized discrete displacement vector is obtained by assigning 1 to a positive component if it is larger than $\text{thr}^s_{\text{pos}}(r)$, assigning -1 to a negative component if it is smaller than $\text{thr}^s_{\text{neg}}(r)$, and assigning 0 to all other components (no movement). Then the position of fish s is updated if the exploratory move causes an increase in fitness:

$$\theta^s_{\text{ind}}(r) = \begin{cases} \theta^s(r-1) + \Delta\theta^s_{\text{ind}}(r), \\ \quad \text{if } L_k(\theta^s(r-1)+\Delta\theta^s_{\text{ind}}(r)) > L_k(\theta^s(r-1)), \\ \theta^s(r-1), \qquad \text{otherwise.} \end{cases} \tag{2.35}$$

where L_k is the penalized log-likelihood of the model, defined in the previous section. An absorbing boundary condition is adopted, whereby each fish interrupts its movement at the boundary of the parameter space, at the point where it encounters it.

Feeding operator. The weights of all fish are updated based on the fitness improvement from the previous individual movement, if any:

$$w^s(r) = w^s(r-1) + \frac{L_k(\theta^s_{\text{ind}}(r)) - L_k(\theta^s(r-1))}{\max_s\{L_k(\theta^s_{\text{ind}}(r)) - L_k(\theta^s(r-1))\}}. \tag{2.36}$$

Collective instinctive movement operator. This operator makes the fish that had successful individual movements influence the collective direction of movement of the school. The position of each fish s is updated according to:

$$\theta^s_{\text{inst}}(r) = \theta^s_{\text{ind}}(r) + \frac{\sum_{s'=1}^{S} \Delta\theta^{s'}_{\text{ind}}(r)(L_k(\theta^{s'}_{\text{ind}}(r)) - L_k(\theta^{s'}(r-1)))}{\sum_{s'=1}^{S}(L_k(\theta^{s'}_{\text{ind}}(r)) - L_k(\theta^{s'}(r-1)))}. \tag{2.37}$$

The displacement in discrete parameter space is quantized following the same procedure adopted

to discretize the individual movement displacement vector.

Collective volitive movement operator. This is similar to the individual movement step, but now the fish move in concert, depending on whether the fish school is successful after the previous steps, i.e., its total weight increases, or not. If the fish school is successful, then it should contract, changing from exploration to exploitation mode. Otherwise, it should expand in order to explore the space more. This is accomplished by first defining the current fish school barycenter:

$$\mathbf{b}(r) \;=\; \frac{\sum_{s=1}^{S} w^s(r)\theta^s_{\text{inst}}(r)}{\sum_{s=1}^{S} w^s(r)}\,. \tag{2.38}$$

For each fish s, after the collective instinctive movement at iteration r, let $\boldsymbol{\xi}^s(r) = \theta^s_{\text{inst}}(r) - \mathbf{b}(r) = (\xi^s_1(r), \ldots, \xi^s_R(r))$ be the position vector with respect to the school barycenter. Let $\Delta\theta^s_{\text{vol}}(r) = (\Delta\theta^s_{\text{disc,vol}}(r), \Delta\theta^s_{\text{cont,vol}}(r))$ be the collective volitive displacement vector for fish s at iteration r. Vector $\Delta\theta^s_{\text{disc,vol}}(r)$ is drawn from a uniform distribution over the rectangular region $[0, \xi^s_1]\times\cdots\times[0, \xi^s_{d^2+d}]$ and quantized by the same process used in the individual move, while $\Delta\theta^s_{\text{cont,vol}}(r)$ is drawn from uniform distribution over the rectangular region $[0, 2\tau_1(r)\xi^s_{d^2+d+1}(r) \times \cdots \times [0, 2\tau_Q(r)\xi^s_{d^2+d+Q}(r)]$, where $\tau_1(r), \ldots, \tau_Q(r)$ are the same step sizes used in the individual movement step. If the school is successful, i.e., if $\sum_{s=1}^{S} w^s(r) > \sum_{s=1}^{S} w^s(r-1)$, then its radius should contract, and

$$\theta^s_{\text{vol}}(r) \;=\; \theta^s_{\text{inst}}(r) - \Delta\theta^s_{\text{vol}}(r)\,, \tag{2.39}$$

otherwise, the radius expands, so the school can escape a bad region, and

$$\theta^s_{\text{vol}}(r) \;=\; \theta^s_{\text{inst}}(r) + \Delta\theta^s_{\text{vol}}(r)\,, \tag{2.40}$$

The new position of the fish is $\theta^s(r) = \theta^s_{\text{vol}}(r)$.

2.6 Numerical Experiments

In this section, we present the result of a comprehensive set of numerical experiments, using both synthetic and real gene expression time series data, to assess the performance of PALLAS and compare it against that of other popular methods in the literature.

2.6.1 Performance Criteria

The problem of comparing networks is a nontrivial one; there is not a single metric that captures both the topological and dynamical properties of the networks [74]. Here we consider two classes of metrics, one based on the difference between the network functions, which takes into account the full regulatory relationships among genes, and the other based on edge-calling accuracy rates, which considers only the network topology.

2.6.1.1 Network Function Distance

Let $\mathbf{f} = (f_1, \dots, f_d)$ and $\hat{\mathbf{f}} = (\hat{f}_1, \dots, \hat{f}_d)$ be the network functions of the groundtruth and inferred networks, where the component functions f_i and $\hat{f}_i$ are Boolean functions on d variables, for $i = 1, \dots, d$; see (2.1). The performance criterion is the average number of disagreeing Boolean functions between the two networks

$$\varphi(\mathbf{f}, \hat{\mathbf{f}}) = \frac{1}{d \times 2^d} \sum_{i=1}^{d} \sum_{j=1}^{2^d} [f_i(\mathbf{x}^j) \oplus \hat{f}_i(\mathbf{x}^j)] . \tag{2.41}$$

This distance is related to the dynamical behavior of the networks, since it has to do with how the Boolean functions differ.

2.6.1.2 Edge-Calling Accuracy Rates

An *edge* in the groundtruth network represents a relationship between two genes. Here we consider directionality (an edge from gene i to gene j is distinct from an edge from gene j to gene i), but disregard activation/inhibition relationships (this is done because some of the methods to which PALLAS is compared in this section do not capture activation/inhibition). Let TP and FN be the total number of directional edges that are correctly detected (irrespective of inhibition/activation)

and incorrectly missed by the inference algorithm, respectively. Similarly, let FP and TN be the total number of directional edges that are incorrectly found and correctly missed, respectively. We define the following edge-calling accuracy rates:

(i) Sensitivity/True Positive Rate (TPR):

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.42}$$

(ii) Specificity/True Negative Rate (SPC):

$$\text{SPC} = \frac{\text{TN}}{\text{FP} + \text{TN}}. \tag{2.43}$$

(iii) Precision/Positive Predictive Value (PPV):

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.44}$$

2.6.2 Experiments with Synthetic Data

2.6.2.1 Experiment 1: P53-MDM2 Negative-Feedback Gene Regulatory Network

The experiments in this section use the well-known p53-MDM2 negative-feedback gene regulatory network [75], which is displayed in Figure 2.2. The gene interaction parameters a_{ij} can be read from Figure 2.2 (a), for example, p53 is activated by ATM, and is inhibited by WIP1 and MDM2. These interactions can be represented as $a_{21} = +1, a_{22} = 0, a_{23} = -1, a_{24} = -1$:

$$\begin{aligned}
&a_{11} = 0, \quad a_{12} = 0, \quad a_{13} = -1, \quad a_{14} = 0 \\
&a_{21} = +1, \quad a_{22} = 0, \quad a_{23} = -1, \quad a_{24} = -1 \\
&a_{31} = 0, \quad a_{32} = +1, \quad a_{33} = 0, \quad a_{34} = 0 \\
&a_{41} = -1, \quad a_{42} = +1, \quad a_{43} = +1, \quad a_{44} = 0
\end{aligned}$$

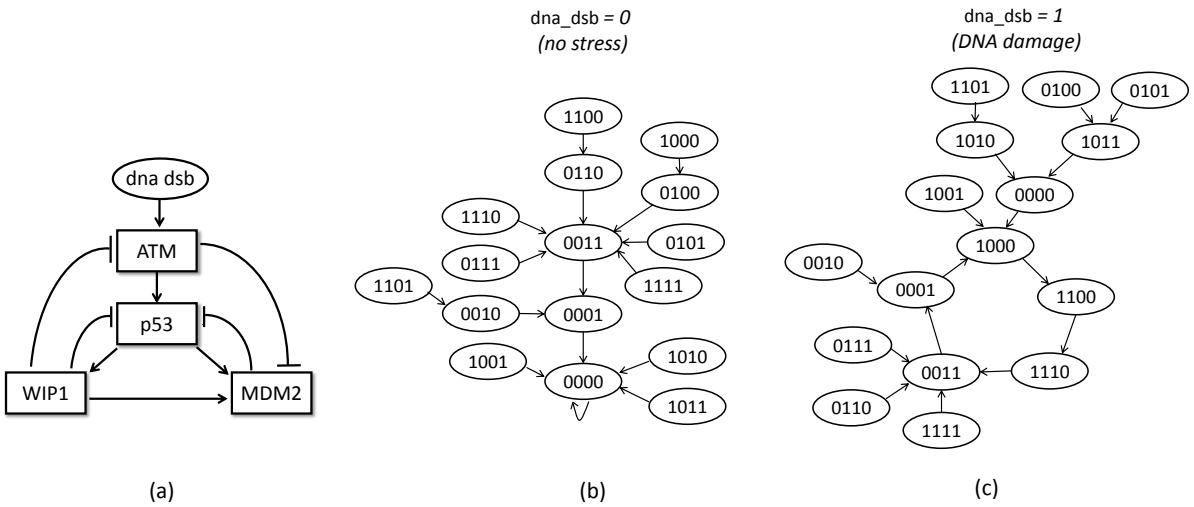

Figure 2.2: Activation/inhibition pathway diagram and state transition diagrams corresponding to a constant input dna_dsb = 0 (no-stress) and dna_dsb = 1 (DNA-damage) for the p53-MDM2 negative feedback loop gene regulatory network with negative regulation biases.

with all biases $b_i = -1/2$, $i = 1, 2, 3, 4$.

The state vector is $\mathbf{X}$ = (ATM, p53, Wip1, MDM2), while dna_dsb acts an external Boolean input that signals DNA damage (p53 is a master tumor-suppressing gene that activates DNA repair mechanisms). These two cases lead to the state transition diagrams displayed in Figure 2.2(b) and (c), respectively.

- **Incomplete Network Topology**:

  In this experiment, we assume the system parameters are known, except that the relationships between genes. Since these parameters are discrete only, we will use DFSS (MFSS without continuous searching space) to do the optimization.

  Average accuracy rates computed over 500 independently-generated time series of different length n, process noise intensity p, observation noise standard deviation $\sigma_i \equiv \sigma$, for $i = 1, \ldots, 4$, and the no-stress and DNA-damage conditions are displayed in Table 2.1. The accuracy rates correspond to the proportion of time all four gene interaction parameters are correctly identified at the time-series endpoint (i.e., an error occurs if at least one parameter is incorrectly identified). We can observe that performance increases monotonically with an

Table 2.1: Average accuracy rates for estimation of the gene interaction parameters.

		No-stress			DNA-damage		
n	p	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$
20	0.05	0.378	0.338	0.194	0.830	0.762	0.624
	0.1	0.446	0.388	0.208	0.738	0.616	0.462
	0.2	0.426	0.290	0.156	0.516	0.374	0.202
	0.3	0.230	0.192	0.086	0.238	0.138	0.074
50	0.05	0.528	0.426	0.312	0.954	0.908	0.838
	0.1	0.728	0.610	0.400	0.948	0.898	0.766
	0.2	0.808	0.628	0.322	0.838	0.666	0.490
	0.3	0.538	0.392	0.170	0.518	0.334	0.152
100	0.05	0.692	0.596	0.444	0.986	0.956	0.914
	0.1	0.900	0.786	0.528	0.996	0.976	0.896
	0.2	0.932	0.854	0.518	0.972	0.898	0.698
	0.3	0.780	0.630	0.296	0.756	0.634	0.324
200	0.05	0.902	0.732	0.486	1.000	0.992	0.964
	0.1	0.982	0.882	0.688	1.000	1.000	0.966
	0.2	0.996	0.958	0.742	1.000	0.980	0.902
	0.3	0.964	0.858	0.522	0.944	0.828	0.566

increasing time-series length and decreasing observation noise intensity, as expected. The behavior with respect to the process noise is more interesting: under no stress, performance exhibits peaking, whereby accuracy rates initially increases with increasing process noise but eventually decreases. The reason for this is that at low process noise levels, the system cannot escape its singleton attractor easily, visiting fewer states and decreasing performance. This is not an issue under DNA damage, which contains a large cyclic attractor. On the other hand,

large process noise intensity renders the system too chaotic, decreasing performance in all cases. Finally, we can see that accuracy rates are better under DNA damage than no stress, for a similar reason moderate process noise helps the inference process: under DNA damage the system contains a large cyclic attractor and thus, for a fixed time series length, tends to visit a larger portion of the state space than under no stress, when the system contains a singleton attractor. This can also be verified in Fig 2.3 that the estimated parameters convergence more quickly under DNA damage condition. In fact, performance can be quite poor under no stress, large process and observation noise and small time series length, while the opposite happens under DNA damage and small process and observation noise levels.

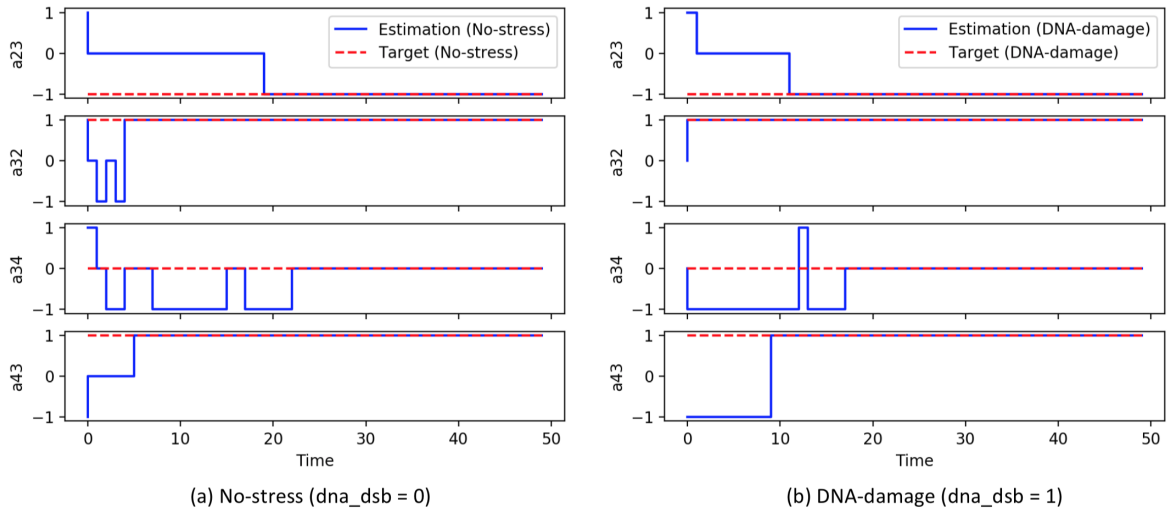

Figure 2.3: Estimated parameters versus time for (a) no-stress and (b) DNA-damage conditions.

Next we compare the performance of the ML-BKF and the DFSS-ML-BKF approaches. Since the former corresponds to an exhaustive search, it is expected to uniformly dominate in terms of accuracy. The question we would like to ask instead is how they compare in terms of computational effort at a high level of accuracy for the DFSS-ML-BKF, as the number of unknown parameters (i.e., the number of unknown gene interactions) increases. The parameters of the simulation are set to $n = 100$, $p = 0.1$, and $\sigma = 0.1$ under DNA damage.

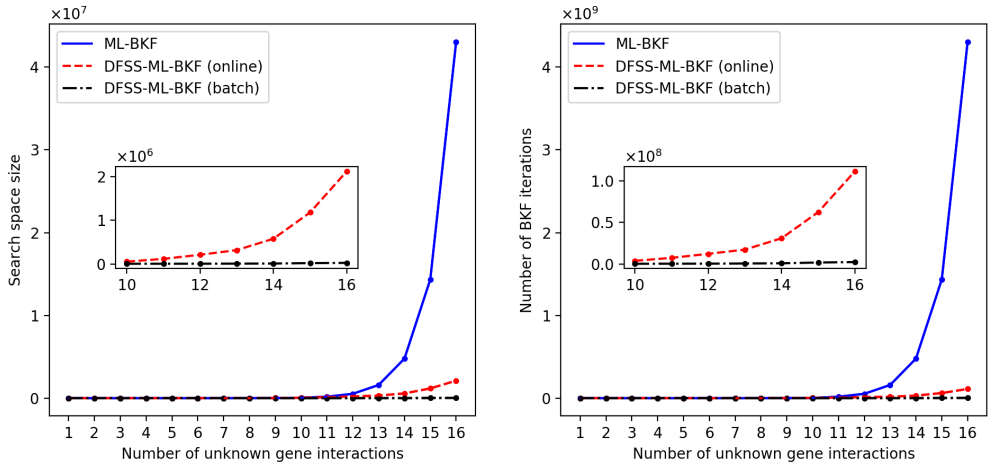

Figure 2.4: Comparison in computational effort among the various methods.

Two settings for the DFSS-ML-BKF are considered: "online," when the observations are presented one by one and "batch," when all 100 data points are presented at once. This does not make a difference for the ML-BKF algorithm, since it runs 3^M BKFs in parallel for a total of $3^M \times 100$ BKF iterations in either case. We increased the size of the fish school N and the maximum number of iterations $t_{\max}$ to make the DFSS-ML-BKF accuracy rate at least 97% throughout. The number of visited parameters in the search space and the number of BKF iterations against the number of unknown gene interactions are plotted in Figure 2.4. We can see that the two methods are very similar in computational effort for a small number of parameters, but DFSS-ML-BKF is much more efficient for a number of unknown parameters exceeding 11. We can also observe that the batch method is more efficient than the online method, since in the former case DFSS is only run once.

- **Incomplete Network Topology and Unknown Noise and Expression Parameters**

  In this experiment, no prior knowledge is used, i.e., all model parameters must be estimated. The dna_dsb input vector is held at a constant value 1, meaning that the system is constantly under DNA damage stress. We assume in this experiment negative regulation biases, $b_i = -1/2$, for $i = 1, 2, 3, 4$. The transition noise parameter p is selected randomly in the interval

[0.01, 0.1]. The microarray data model has parameters $\mu_i \equiv \mu = 30$, $\delta_i \equiv \delta = 20$, $\sigma_i^2 \equiv \sigma^2 = 49$, for $i = 1, \ldots, 4$.

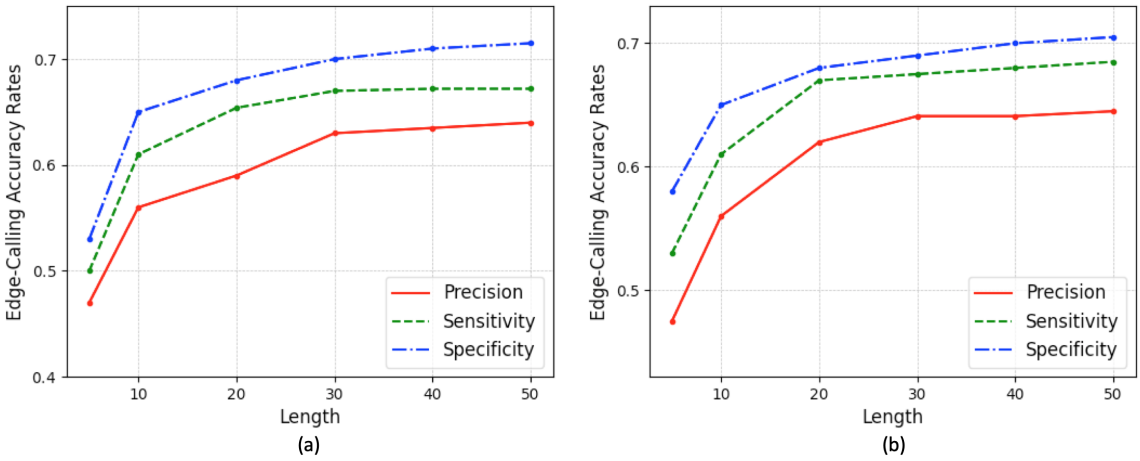

Figure 2.5: p53-MDM2 experiment edge-calling accuracy rate results as a function of time series: (a) search process noise; (b) fix process noise

Average edge-calling accuracy rates obtained by PALLAS over 20 repetitions of the experiment are displayed as a function of time series length in Figure 2.5. The difference between Figure 2.5 (a) and (b) is that we fix the process noise parameter to 0.05 in (b), but search for this parameter in the interval [0.01, 0.1] in (a). We can see that the performance is similar. In addition, in both cases, as the time series length increases, precision, sensitivity and specificity all increase. It can be seen also that performance improves quickly initially, but after the time series length exceeds 20 there is little additional improvement.

Table 2.2 displays execution times obtained on a MacBook Pro, with 2.5 GHz Quad-Core Intel Core i7 processor and 16 GB 1600 MHz DDR3 memory. Each set of parameters is applied on 5 different synthetic datasets with 30 time points and average results are calculated.

Table 2.2: Performance and execution time comparison with different parameters.

fish	**iterations**	**particles**	**execution time(s)**	**sensitivity**	**specificity**	**precision**
24	3000	8	1091	0.611	0.450	0.480
24	3000	16	1123	0.63	0.550	0.520
24	5000	8	1834	0.608	0.550	0.558
24	5000	16	1912	0.645	0.575	0.587
48	3000	8	2081	0.634	0.525	0.580
48	3000	16	2213	0.656	0.563	0.583
48	5000	8	3520	0.667	0.575	0.646
48	5000	16	3621	0.672	0.590	0.650
72	3000	8	3211	0.679	0.600	0.678
72	3000	16	3274	0.681	0.630	0.654
72	5000	8	5324	0.683	0.623	0.673
72	5000	16	5498	0.681	0.667	0.665

We can see from this table that computational complexity increases linearly with the number of fish and number of iteration in the MFSS algorithm, but less than linearly in the number of particles used in the particle filter to approximate the likelihood score. We can see that performance is quite sensitive to the number of fish used in the search (for a fixed number of iterations). Computation time can be reduced by decreasing the number of fish, at a cost of performance. However, increasing the number of fish beyond 48, in this case, does not improve performance significantly, indicating that there is a value of diminishing returns, as expected.

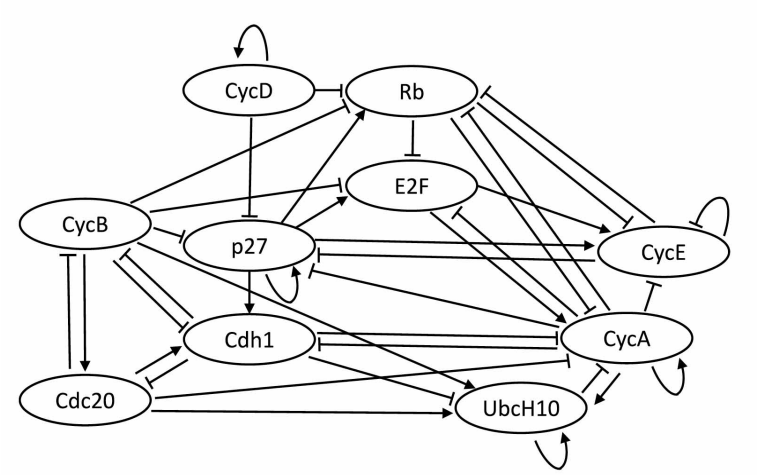

Figure 2.6: Mammalian cell cycle network.

2.6.2.2 Experiment 2: Mammalian Cell-Cycle Gene Regulatory Network

Here, we present results based on the well-known Mammalian Cell-Cycle network [76], which is displayed in Figure 2.6. (Results for a different GRN are presented in the Supplementary Material). The state vector is $\mathbf{X} =$ (CycD, Rb, p27, E2F, CycE, CycA, Cdc20, Cdh1, UbcH10, CycB). This is a large network, with a huge parameter space, for which the estimation problem is hard. The gene interaction parameters a_{ij} can be read from Figure 2.6 in the same way as in the p53-MDM2 network in the Supplementary Material. Once again, the regulation biases are set to $b_i = -1/2$, for $i = 1, \ldots, 10$. The transition noise parameter p is selected randomly in the interval $[0.01, 0.1]$. The RNA-Seq data model parameters are $\mu_i \equiv \mu = 0.1$, $\delta_i \equiv \delta = 3$, $\phi_i \equiv \phi = 5$, for $i = 1, \ldots, 10$. The sequencing depth is set to $s = 22.52$ (500K-550K reads) and the time series length is fixed at 50. Here we compare PALLAS with the GENIE3 [34], TIGRESS [36], and Banjo [37] algorithms. Like PALLAS, these algorithms can operate directly on the noisy time series, without a need for ad-hoc binarization. However, they do not estimate observational parameters or provide activation/inhibition information, so only the edge-calling accuracy rates in Section 2.6.1.2 are appropriate here. Average rates obtained over 20 repetitions of the experiment are displayed in Figure 2.7. One can see that with similar specificity, PALLAS displays higher sensitivity and pre-

cision than GENIE3 and TIGRESS. Although it was not possible to adjust the specificity of Banjo to the same levels, we can see that its sensitivity is quite low. In fact, Banjo returned a very small number of edges overall in this experiment. PALLAS also displayed the highest precision among all the algorithms.

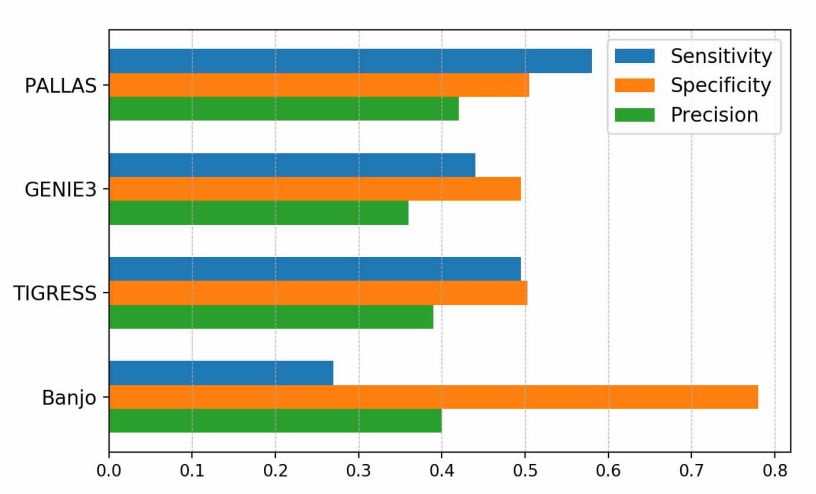

Figure 2.7: Mammalian cell cycle experiment results.

2.6.2.3 Experiment 3: Artificial Networks

In this section we report results obtained on an ensemble of 10 randomly generated networks with $d = 8$ genes, where each gene is regulated by 3 other genes on average. Edge connectivity, including activation and inhibition, as well as regulation biases, are randomly chosen. The transition noise parameter p is selected randomly in the interval $[0.01, 0.1]$. RNA-Seq synthetic data are generated with parameters $\mu_i \equiv \mu = 0$, $\phi_i \equiv \phi = 1$ or 5, for $i = 1, \ldots, 8$. In the first case, there is more observation noise, and the problem is harder. The parameters δ_i are allowed to vary uniformly over the intervals $[1, 2]$ or $[1, 5]$, for $i = 1, \ldots, 8$. In the first case, the problem is harder, since the differences in observed expression are smaller. Sequencing depth is set at $s = 22.52$ (500K-550K reads).

Here, we compare PALLAS with the Best-Fit [38], REVEAL [39], FBNNet[41], and GABNI

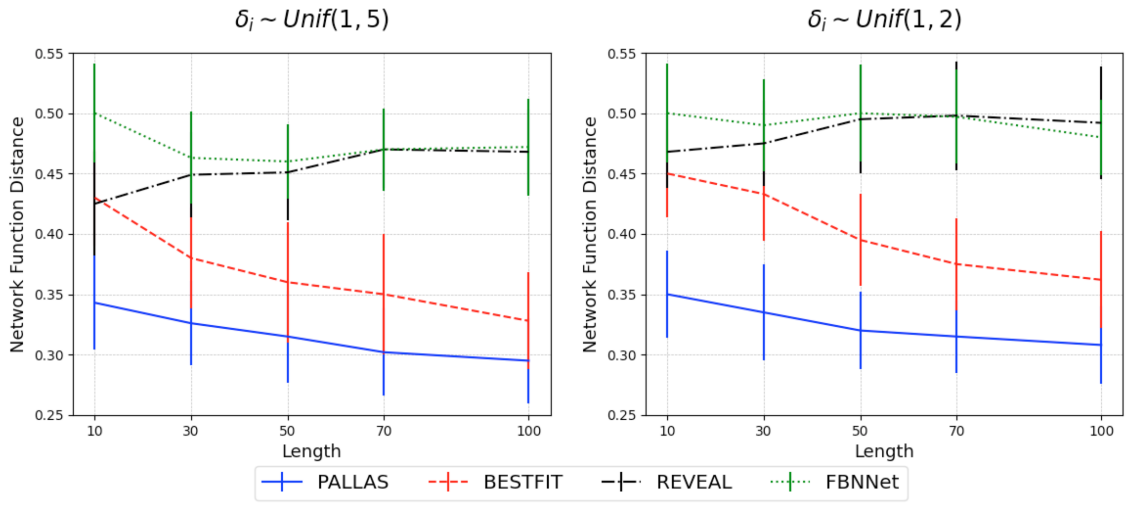

Figure 2.8: Comparison of network function distance among the PALLAS, Best-Fit, REVEAL, and FBNNet algorithms, under different δ ranges.

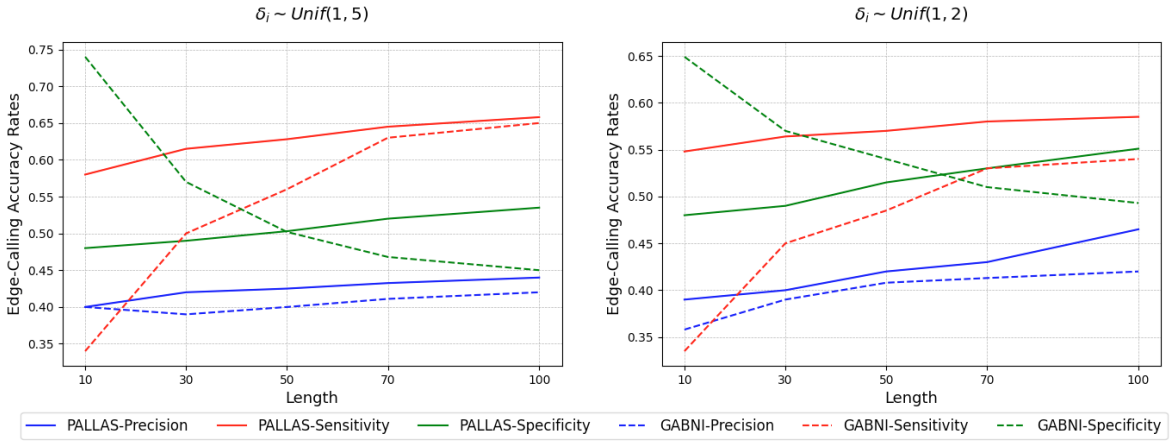

Figure 2.9: Comparison of edge-calling accuracy rates between the PALLAS and GABNI algorithms, under different δ ranges.

[40] algorithms. These methods apply to Boolean time series, so they need to employ ad-hoc binarization of the gene expression data. For the first two, [77] recommends the use of the KM3 binarization method, while for GABNI, [23] recommends the use of K-means binarization, as well as FBNNet; hence, we use those binarization methods here. The output of the Best-Fit, REVEAL and FBNNet algorithms are Boolean transition functions, for which the network function distance is appropriate. On the other hand, the output of GABNI consist of positive (activating) or negative (inhibitory) interactions, for which we use the edge-calling accuracy rates defined previously.

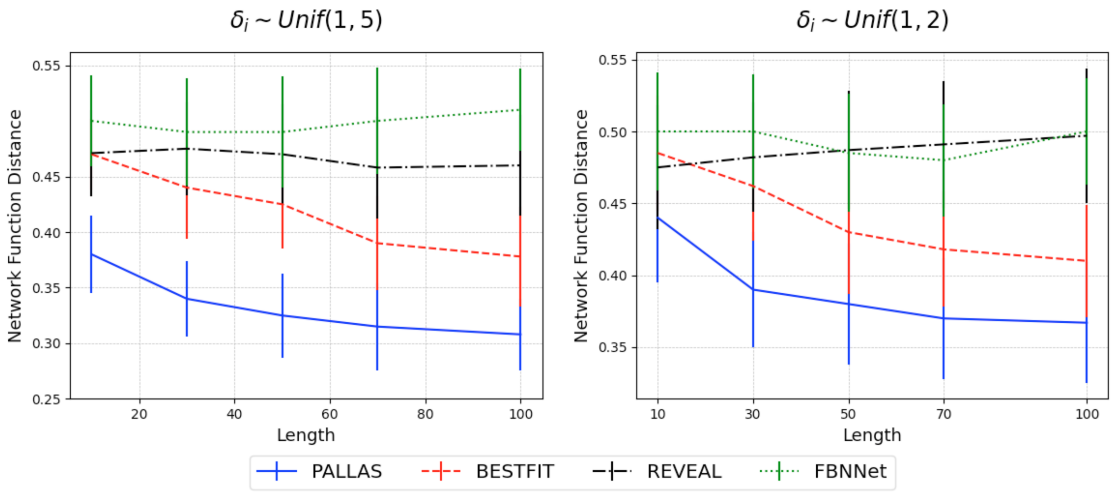

Figure 2.10: Comparison of network function distance among the PALLAS, Best-Fit, REVEAL, and FBNNet algorithms, under different δ ranges.

Average network function distances and edge-calling accuracy rates obtained over 20 repetitions of the experiment (2 for each of the 10 networks) with $\phi = 5$ are displayed in Figures 2.8 and 2.9, corresponding results for $\phi = 1$ are shown in Figures 2.10 and 2.11. Figure 2.8 shows that the performance of Best-Fit and PALLAS increases with the time series length, while the performance of REVEAL and FBNNet are mostly stable. PALLAS perform better than the Best-Fit algorithm, especially when δ is smaller. This reflects the fact that ad-hoc binarization of the data becomes less accurate with a smaller difference between activation/inactivation levels in the observed data, which is determined by δ. Figure 2.9 shows that PALLAS beats GABNI in sensitivity throughout, as well

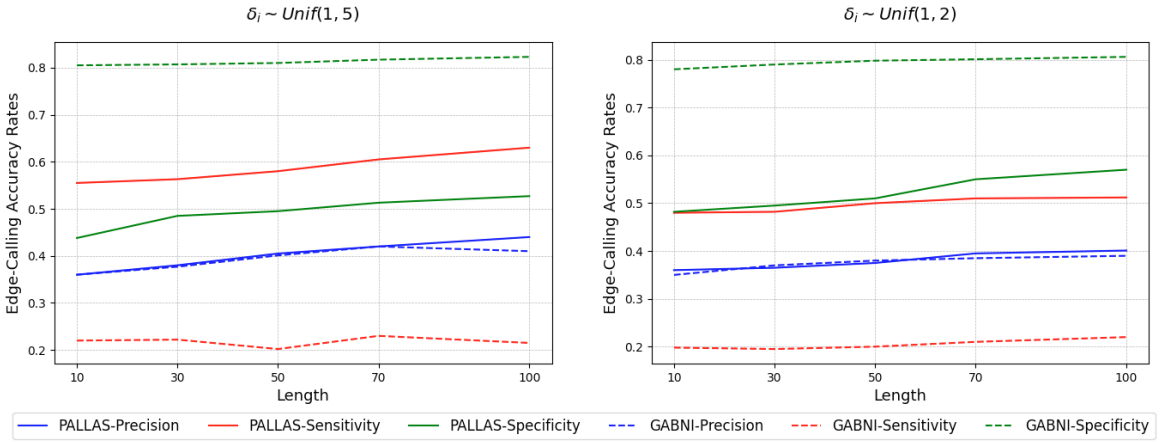

Figure 2.11: Comparison of edge-calling accuracy rates between the PALLAS and GABNI algorithms, under different δ ranges.

as in specificity under sufficient data. Indeed, it is shown in Figures 2.10 and 2.11 that GABNI detects very few edges under small sample size or high observation noise, which artificially inflates its specificity.

2.6.2.4 Experiment 4: Prototype Immunomic Protein-Protein Interaction Network

We investigate the PPI inference performance of the PALLAS algorithm using a prototype immunomic network during infection [78]. The model consists of three state variables, which represent immune activation of three distinct T-cell populations. We assume that the dynamic activity of the various T-cell populations on the model are measure through time series of LC-MS measurements of the corresponding cytokines (interferon-gamma specific to CD4+ T helper cells, interferon-gamma specific to CD8+ cytotoxic T cells, and interleukin-10 specific to the CD4+ regulator T cells).

Figure 2.12(a) depicts the model, which consists of a Boolean network with three nodes, labeled "A", "B", and "C". The interaction parameters $a_{i,j}$ can be read from the figure. For example, node "C" is activated by node "A" and inhibited by node "B". These interactions can be represented as $a_{31} = 1, a_{32} = -1, a_{33} = 0$.

Figure 2.12(b) depicts the resulting state-space. From the state space, we can see that these

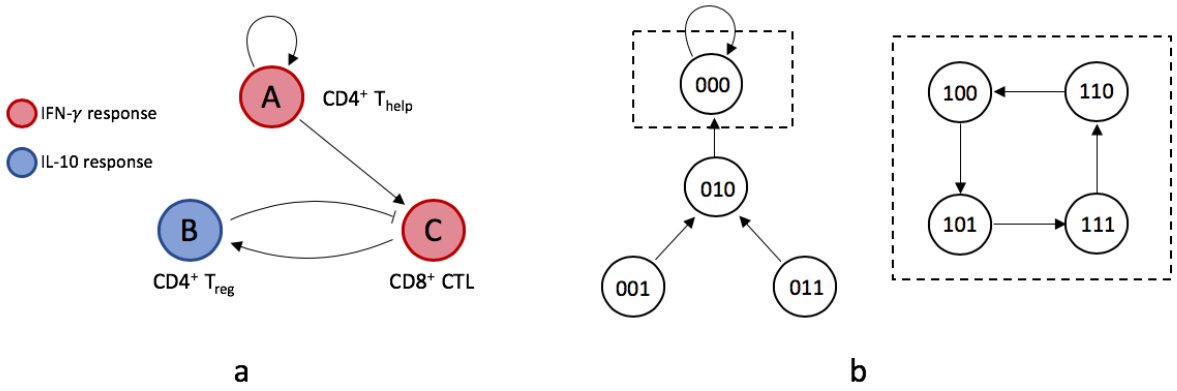

Figure 2.12: Example of a simple Boolean network model of immunomic interactions during response to infection, consisting of three nodes A, B, and C; node A is a promoter, B is a suppressor, while node C produces the effector response, while also promoting suppression of B (negative feedback). a. Network wiring diagram and transition rules. b. Basins of attraction in state-space, with attractors indicated by dashed rectangles.

states are partitioned into two basins of attraction: the first one corresponds to a single attractor, whereas the second one consists of an attractor cycle. However, the two behaviors of the system is only depends on the state of node "A". If it is not expressed (there is no "helper T-cell" response), then the system will always tend to the resting single-state attractor 000. If "A" is expressed (there is help), then the activity of the system corresponds to that of a attractor cycle with the effector response being turned on and off cyclically.

In this experiment, We assume negative regulation biases, $b_i = -1/2$, for $i = 1, 2, 3$ and the synthetic data is generated based on LC-MS model with parameters $\kappa = 4, \vartheta = 100, \zeta = 3, and \varphi = 0.01$. Predefined interval ranges for estimation are $\kappa \in [2, 6], \vartheta \in [80, 120], \zeta \in [1, 5], \varphi \in [0.01, 0.2]$. Average edge-calling accuracy rates and network function distances obtained over 10 repetitions of the experiment are displayed in Figure 2.13 and 2.14. In Figure 2.13, we can see that as the time series length increases, the algorithm is both sensitive and specific, with high precision, capturing well the topology of the network. In figure 2.14 we can see that, in addition to capturing the network edges well, the proposed algorithm can correctly identify the regulatory functions, which controls the system dynamics, as time series length increases.

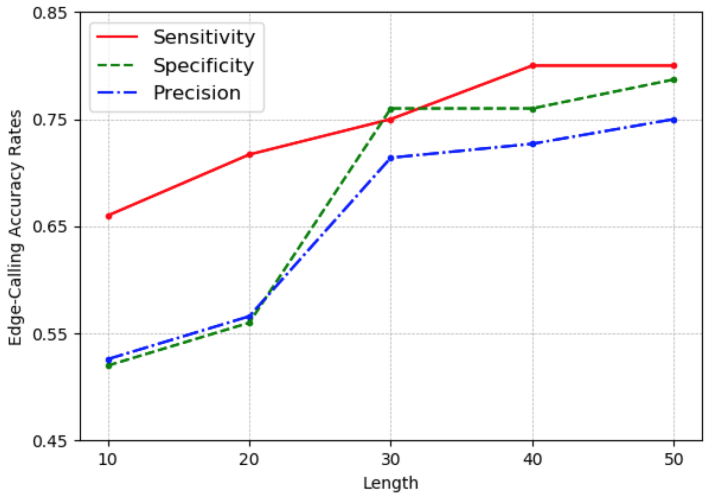

Figure 2.13: Average edge-calling accuracy rates.

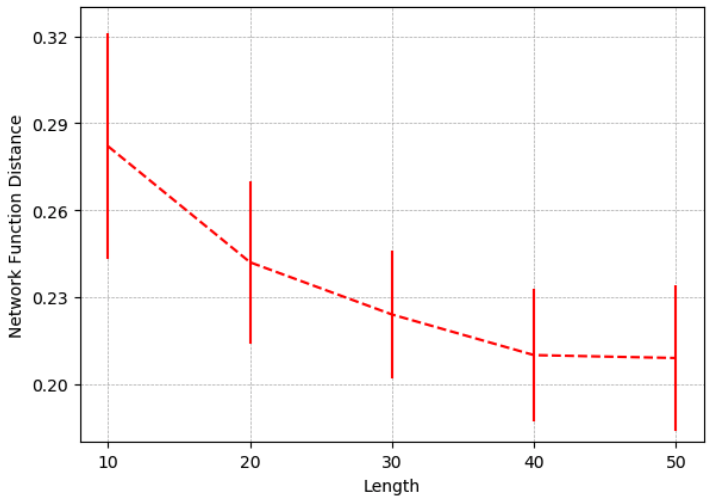

Figure 2.14: Average network function distance.

2.6.3 Experiments with Real Data

In this section, we demonstrate the application of PALLAS to real microarray data from well-known biological systems. The complete results, including both false positives and false negatives.

2.6.3.1 Experiment 1: E. Coli SOS DNA Repair System

First, we consider the SOS DNA repair system in E. Coli. In the normal state, the protein *LexA* is known to be a repressor to the SOS genes. When DNA is damaged, the protein *RecA* becomes activated and mediates *LexA* autocleavage, which causes activation of the SOS genes. After the activated SOS genes repair the damaged DNA, *RecA* stops mediating *LexA* autocleavage and *LexA* represses the SOS genes again. The full SOS DNA repair gene network is displayed in Figure 2.15 [79, 44]. We attempt to infer this network from gene expression time series datasets generated by [80] (`http://www.weizmann.ac.il/mcb/UriAlon/download/downloadable-data`). Each time series contains 50 measurements for every 6 minutes including the initial zero concentrations; we pick the third dataset in the database for this experiment, and compare the results against those found in [42, 43, 44].

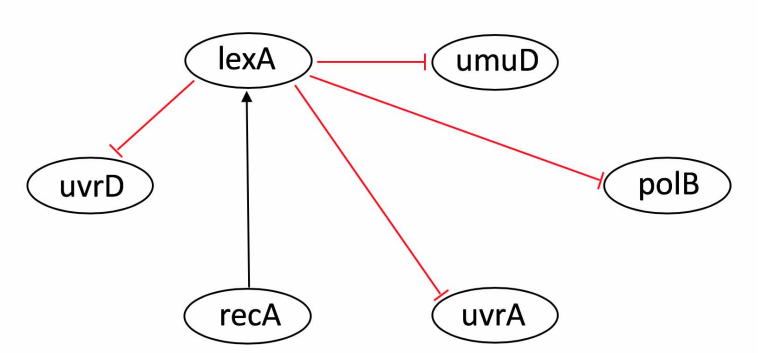

Figure 2.15: SOS DNA repair system in E.coli (the red edges are the ones successfully recovered by PALLAS).

The sparsity parameter λ in (2.26) is chosen to produce about half of the possible edges in the six-gene network. Figure 2.15 displays in red the edges of the original network that were suc-

cessfully recovered by a consensus of the top three networks found by PALLAS, according to the penalized likelihood score (the full network is displayed in Figure 2.16). We can see that all inhibitory edges from *lexA* were successfully detected. Although PALLAS infers the wrong direction between *recA* and *lexA*, the connection is detected. With a similar total number of inferred edges, [43] finds the opposite regulations, i.e., all the inhibitory edges are inferred as activating edges. While [42] finds most of the inhibitory edges, it misses the important edge from *lexA* to *uvrA*. Finally, [44] recovers only two of the edges.

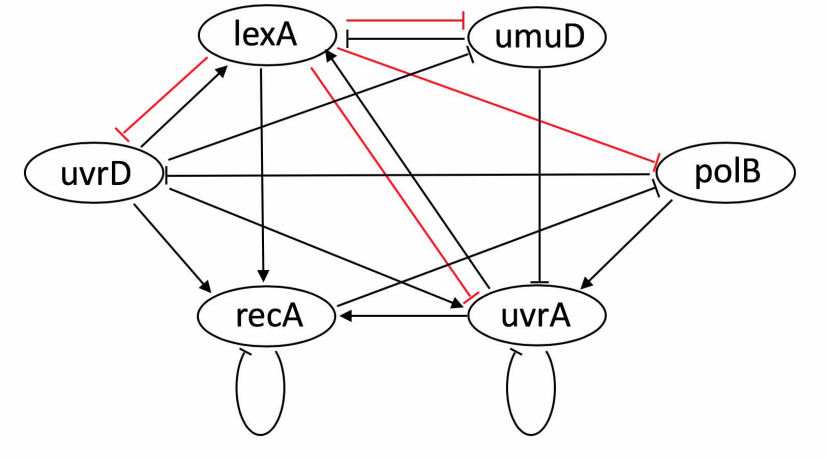

Figure 2.16: The SOS DNA repair system network inferred by PALLAS

2.6.3.2 Experiment 2: E. Coli Biofilm Formation Pathway

In this section, we demonstrate the performance of PALLAS on RNA-Seq time series expression data from a pathway involved in biofilm formation by E. Coli, namely, the Rpos(sigmaS)/MlrA/CsgD cascade, which involves eight genes: *Rpos, MlrA, CsgD, YciR, YoaD, BcsA, YaiC, YdaM*. Information on this pathway can be found in the KEGG database (`https://www.genome.jp/kegg/`) as well as in [81, 82, 83]. Figure 2.17 displays a consensus gene network derived from these sources. The gene expression data used is from the E. Coli Strain B/REL606 and is available at the Dryad Digital Repository (`https://datadryad.org/resource/doi:10.5061/`

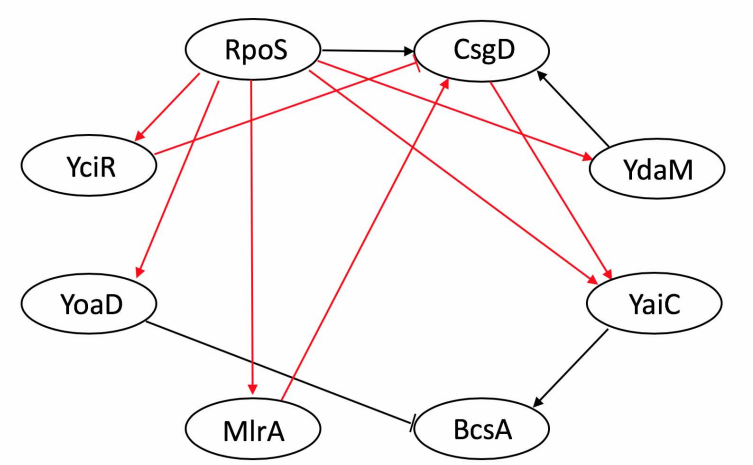

Figure 2.17: Biofilm architecture of Escherichia coli (the red edges are the ones successfully recovered by PALLAS).

`dryad.hj6mr`) [84]. This dataset consists of 3 bacterial samples and 9 time points evenly spaced for each sample. The genes in this pathway display similar values at low expression levels, but vary considerably at high expression levels. Accordingly, we assume a single baseline parameter $\mu_i \equiv \mu$ for all genes, but the parameters δ_i and ϕ_i are allowed to differ from gene to gene, for $i = 1, \ldots, 8$. The sequencing depth is set at $s = 1.02$ (1k-50k reads) reflecting the low read counts in the data set.

As in the previous experiment, the sparsity parameter λ in (2.26) is chosen to produce about half of the possible edges in the eight-gene network. Figure 2.17 displays in red the edges of the original network that were successfully recovered by a consensus of the top three networks found by PALLAS, according to penalized likelihood score (the full network is displayed in Figure 2.18). PALLAS successfully infers five out of the six important activating interactions from *RpoS*. Most of the other connections in the original network were correctly detected.

2.7 Conclusion

We presented PALLAS, a new framework for inference of Boolean gene regulatory networks from gene expression time series data and protein-protein interaction networks from proteomics expression time series data. The algorithm avoids ad-hoc binarization of the gene/proteomics ex-

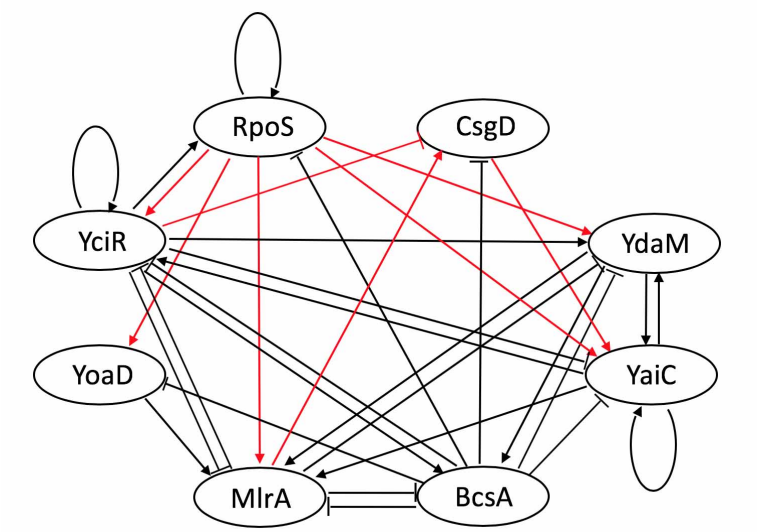

Figure 2.18: The biofilm system network inferred by PALLAS

pression data and allows inference of large networks by employing penalized maximum likelihood as a regularization method, applying auxiliary particle filter implementation of the Boolean Kalman filter and ABC-SMC algorithm for the computation of the likelihood, and using a novel version of the fish school search particle swarm algorithm to search the parameter space. Numerical experiments using synthetic time series data show that PALLAS outperforms other well-known inference methods. The performance of PALLAS was also demonstrated on real gene expression time series data from the SOS DNA repair and Biofilm formation pathways in E. Coli. As a sophisticated state-space method for Boolean GRN/PPI inference directly from noisy gene expression data, without the need of ad-hoc binarization, PALLAS is computationally expensive. Results indicate that execution time increases linearly with the number of fish used in the MFSS algorithm. The user can adjust the running time by changing the number of fish, at a cost to performance. Future work will include the implementation of PALLAS on high-performance parallel architectures, which will enable the inference of larger networks.

3. A NONLINEAR STOCHASTIC STATE SPACE APPROACH FOR EPIDMIC MODEL

3.1 Overview

Infectious disease outbreaks remain a major threat to global health. This is especially the case for highly pathogenic and transmissible diseases with pandemic potential. These global threats were recently exemplified by the 2009 swine flu outbreak and the ongoing COVID-19 pandemic caused by the novel Severe Acute Respiratory Syndrome coronavirus 2 (SARS-CoV-2). To effectively mitigate and control the spread of a disease epidemic, it is paramount for public health decision-making to be informed by an accurate understanding of the dynamics of the epidemic and the potential impact of intervention measures. To this end, epidemic models have become an important tool to help people better understand the epidemic and predict its future trends, explore intervention scenarios and ultimately control the epidemic, such as lock-down or vaccination.

Mathematical epidemiology models can be roughly divided into two main types: compartmental models and agent-based models. The agent-based model [85, 86, 87, 88, 89, 90, 91, 92] is a very detailed stochastic model where the agent represent single individuals and thus generally more complex and computationally expensive. On the other hand, population is assigned to compartments with labels in the compartmental model [93, 13, 94, 95, 96, 97, 98, 99], thus simplify the mathematical model and are fairly scalable. In this work, we propose a nonlinear state-space model motivated by the compartmental model.

The history of the compartment model can be traced back to the beginning of twentieth century, the most famous work by [12] whose susceptible - infected - recovered (SIR) model was used for modeling the cholera (London 1865) and plague (London 1665-1666, Bombay 1906) epidemics [13], which describes the number of people transmitted among the compartments. More specifically, it is a kind of weighted directed graph representation of a dynamic system. The susceptible refers to those healthy people who are susceptible to the disease and may get infected; the infected refers to those under infection; the recovered refers to those who recovered from infection and will

temporarily or permanently immune to the disease. However, there are several drawbacks in the original SIR model:

1) It is a deterministic model, meaning the model always performs the same for a given initial conditions, which cannot explain the nonignorable randomness of the observations [95, 100, 99];

2) It is only a temporal model which not consider the spread in geographical regions, e.g. human interaction induced by modern transportation [14];

3) It assumes all the parameters are known which is not the case in real world. Parameter estimation is needed for better understanding and forecasting epidemics, by using the noisy observations [100].

Many variants SIR models become available to solve those issues but not all of them, especially the parameter estimation since the dynamics of the epidemic are partially observed and the observations are noisy. To handle these issues, we propose a new framework which embed the classical compartmental model within the nonlinear state-space model. The state model is a spatial-temporal stochastic dynamic model, considering not only allow hidden states in one place change through time, but allow the states affect its neighbours' as well because of the human interaction induced by modern transportation. It is referred to as multinomial model based on a variant of SIR model - SEIRD model [101, 102, 103]. The observation model is designed by considering several prior knowledge including the each day testing rate, positivity rate, specificity and sensitivity of the tests. In addition, we also consider the difference of the test rate among symptomatic, asymptomatic patients, similar symptomatic of people without epidemic disease and healthy people.

In our model, one objective will be to estimate the continuous hidden state vector $X^t = (S^t, E^t, I^t, R^t, D^t)$, based on the noisy incomplete time series epidemic data. In addition, we will also estimate the parameters in the state model and observation model, including infection rate, inverse of the average latent time (the rate brings people from E to I) and so on, which can obtain the guidance of how to prevent and control the epidemic in reality.

3.2 Mathematical Models for Epidemics

3.2.1 Generalized SEIRD Model

The SEIRD model simulates the time dependent of an epidemic phenomenon. It models the dynamic interaction of people between five different compartments, namely, the susceptible (S), the exposed (E), the infected (I), the recovered (R) and the deceased (D). The classic SEIRD model can be described by the following equations:

$$\begin{aligned}
\frac{dS^t}{dt} &= -\lambda_S I^t \frac{S^t}{N} \\
\frac{dE^t}{dt} &= \lambda_S I^t \frac{S^t}{N} - \lambda_E E^t \\
\frac{dI^t}{dt} &= \lambda_E E^t - (\lambda_R + \lambda_D) I^t \\
\frac{dR^t}{dt} &= \lambda_R I^t \\
\frac{dD^t}{dt} &= \lambda_D I^t
\end{aligned} \tag{3.1}$$

with $S^t + E^t + I^t + R^t + D^t = N$.

The susceptible (S^t) is the part of the population not yet infected with the disease at time t, it will be initialed as the whole population without the infected information. The exposed (E^t) is the population that has been exposed to virus but does not show symptoms yet at time t. It is a latency period or incubation period for the coronavirus. The infected (I^t) is used to represent the infective population after the latent period at time t. The recovered (R^t) denotes the number of individuals who have been infected and then recovered at time t, and will not be reintroduced to the susceptible if the epidemic is assumed to be immunized after infected. The deceased (D^t) is the compartment which is removed from the total population because of the epidemic at time t. N is the total number of the population.

As we can see, the model is governed by the following parameters:

- λ_S is the infection rate, which is the probability that brings people from S to E category. It is multiplied with (S^t/N) to prevent from counting contacts between two individuals who

cannot infect each other. Also, the nonlinear term is involved here meaning that the infected speed not only depends on the infection rate, but also the percentage of infected people or susceptible people over the whole population;

- λ_E is the probability that transports people from E to I category or can be understood as the inverse of the average latent time of the epidemic;
- λ_R is the recovery rate which can also be explained as the inverse of the average recovery time;
- λ_D is the death rate;

3.2.2 Proposed Nonlinear State-Space Model

Given the complexity and reality of the epidemic, many different implementations of the classical SEIRD model become available [104, 103, 105, 101]. In order to handle all the drawbacks in the classic model and represent the reality as close as possible, we propose a nonlinear state-space framework based on SEIRD model. We extended the state model as a spatial-temporal stochastic model and came up with a new observation model which considers the accuracy of the tests and different testing rate of symptomatic and asymptomatic people.

3.2.2.1 State Model

Consider a state process $\{\mathbf{X}_i^t; i = 0, 1, \dots; t = 0, 1, \dots\}$, where $\mathbf{X}_i^t = [S_i^t, E_i^t, I_i^t, R_i^t, D_i^t]$ is a continuous state vector, which evolves according to

$$\mathbf{X}_i^t = \mathbf{f}(\mathbf{X}_i^{t-1}) + \mathbf{n}_i^{t-1} \tag{3.2}$$

for $t = 0, 1, \dots$ represents the time, and $i = 0, 1, \dots$ denotes the geographical region, where $\mathbf{f}$ is the nonlinear state dynamics function and $\mathbf{n}$ is the process noise which is dependent on the state in our model.

To be more precise, the state model can be represented as follows by using the multinomial model:

$$
\begin{aligned}
S_i^t &= S_i^{t-1} - N_{S_i}^{t-1} - \Sigma_j NB_{S_j}^{t-1} \\
E_i^t &= E_i^{t-1} + N_{S_i}^{t-1} - N_{E_i}^{t-1} + \Sigma_j NB_{S_j}^{t-1} \\
I_i^t &= I_i^{t-1} + N_{E_i}^{t-1} - N_{R_i}^{t-1} - N_{D_i}^{t-1} \\
R_i^t &= R_i^{t-1} + N_{R_i}^{t-1} \\
D_i^t &= D_i^{t-1} + N_{D_i}^{t-1}
\end{aligned}
\tag{3.3}
$$

where

$$
\begin{aligned}
N_{S_i}^t \mid X_i^t &\sim binomial(S_i^t; \frac{\lambda_{Si} I_i^t}{N_i^t}) \\
NB_{S_j}^t \mid X_i^t, X_j^t &\sim multinomial(S_i^t; \frac{c_{i0}\lambda_{Si} I_0^t}{N_0^t}, \frac{c_{i1}\lambda_{Si} I_1^t}{N_1^t}, \ldots, \frac{c_{ij}\lambda_{Si} I_j^t}{N_j^t}, j \neq i) \\
N_{E_i}^t \mid X_i^t &\sim binomial(E_i^t; \lambda_E) \\
(N_{R_i}^t, N_{D_i}^t) \mid X_i^t &\sim multinomial(I_i^t; \lambda_R, \lambda_D)
\end{aligned}
\tag{3.4}
$$

In our proposed model, most of the variables have the same meaning with the classic SEIRD model described above. However, since it is a stochastic event for the spread of the disease. The infection from one individual to another belongs to a stochastic manner. Thus, the stochastic property is involved here.

We assume all the transitions belong to binomial or multinomial distribution. $N_{S_i}^t$ is the number of people transport from S to I at time t in the region i effected by internal. It means that the probability of being infected not only depend on the infection rate, but the percentage of infected people in area i as well; $NB_{S_j}^t$ is the number of people bring from S to I at time t in the region i but effected by external region j. Similar with $N_{S_i}^t$, but it belongs to multinomial distribution and depends on the percentage of infected people in area j and calculated by the summation of all the other regions j with weight c_{ij}. This is the key part to influence the spatial domain in the state model which connect all geographical regions. Also, it will help us better understand the importance of

locking down some critical area in the early stage to control the epidemic; $N^t_{E_i}$ is the number of people transit from E to I at time t in region i; $N^t_{R_i}$ is the number of people move from I to R at time t in region i; and $N^t_{D_i}$ is the number of people removed at time t in region i.

Not only the states, but all the parameters which govern the model are designed with stochastic properties. We assume all the parameters belongs to the Beta distribution since it is a continuous distribution designed on the interval [0, 1]. Other than that, there are two differences in the parameters compare with the classic model:

- c_{ij} is a correlation factor, reflects the external interaction between region i and region j. It is the factor to connect all the geographical regions. c_{ij} is designed based on gravity models [106, 107, 108];

- λ_{S_i} is the infection rate but we assume it will different among all the regions because of the density of each region is different. Generally speaking, the disease in the area of larger density will spread much faster than sparsely populated places.

According to the standard state model representation, we can rewrite our model as follows to make the noise terms more clearly:

$$\begin{aligned}
S^t_i &= (1 - \frac{\lambda_S I^{t-1}_i}{N^t_i} - \Sigma_j \frac{c_{ij}\lambda_S I^{t-1}_j}{N^t_j}) \times S^{t-1}_i + n^{t-1}_{S_i} \\
E^t_i &= (1 - \lambda_E) \times E^{t-1}_i + (\frac{\lambda_S I^{t-1}_i}{N^t_i} + \Sigma_j \frac{c_{ij}\lambda_S I^{t-1}_j}{N^t_j}) \times S^{t-1}_i + n^{t-1}_{E_i} \\
I^t_i &= (1 - \lambda_R - \lambda_D) \times I^{t-1}_i + \lambda_E \times E^{t-1}_i + n^{t-1}_{I_i} \\
R^t_i &= R^{t-1}_i + \lambda_R \times I^{t-1}_i + n^{t-1}_{R_i} \\
D^t_i &= D^{t-1}_i + \lambda_D \times I^{t-1}_i + n^{t-1}_{D_i}
\end{aligned} \tag{3.5}$$

where the noise terms can be represented as:

$$
\begin{aligned}
n^t_{S_i} &= -(N^t_{S_i} - \frac{\lambda_S I^t_i}{N^t_i} S^t_i) - \Sigma_j (N B^t_{S_i} - \frac{c_{ij}\lambda_S I^t_j}{N^t_j} S^t_i) \\
n^t_{E_i} &= (N^t_{S_i} - \frac{\lambda_S I^t_i}{N^t_i} S^t_i) + \Sigma_j (N B^t_{S_i} - \frac{c_{ij}\lambda_S I^t_j}{N^t_j} S^t_i) - (N^t_{E_i} - \lambda_E E^t_i) \\
n^t_{I_i} &= (N^t_{E_i} - \lambda_E E^t_i) - (N^t_{R_i} - \lambda_R I^t_i) - (N^t_{D_i} - \lambda_D I^t_i) \\
n^t_{R_i} &= (N^t_{R_i} - \lambda_R I^t_i) \\
n^t_{D_i} &= (N^t_{D_i} - \lambda_D I^t_i)
\end{aligned}
\tag{3.6}
$$

3.2.2.2 Observation Model

Among the state variables we discussed in state model, only the number of deceased (D^t) can be directly observed but also come with the observation noise. Another measurement we can get will be the number of confirmed positivity cases every day, however, it can only be represented by the combination of the state variables and noise.

Assume the only two observations are $\{\mathbf{Y}^t_i; i = 0, 1, \dots; t = 0, 1, \dots\}$ where $\mathbf{Y}^t_i = [P^t_i, Q^t_i]$ (P^t_i is the number of confirmed cases and Q^t_i is the number of death at time t in the place i) evolves according to

$$
\mathbf{Y}^t_i = \mathbf{h}(\mathbf{X}^t_i) + \mathbf{v}^\mathbf{t}_\mathbf{i} \tag{3.7}
$$

for $t = 0, 1, \dots$ represents the time, and $i = 0, 1, \dots$ denotes the geographical region, where $\mathbf{h}$ is a general dynamics function which is a way to express the observations by the state variables and $\mathbf{v}$ is the observation noise which is dependent on the state in our model.

Specifically, the observation model can be represented in the following:

$$
\begin{aligned}
P_i^t &= NT_{FP_i}^t + NT_{TP_i}^t \\
Q_i^t &= D_{TP_i}^t
\end{aligned} \tag{3.8}
$$

where

$$
\begin{aligned}
N_{FP_i}^t | X_i^t &\sim binomial(\varepsilon_1 \times (1-\varepsilon_3) \times (S_i^t + R_i^t); \alpha) + binomial(\varepsilon_2 \times \varepsilon_3 \times (S_i^t + R_i^t); \alpha) \\
N_{TP_i}^t | X_i^t &\sim binomial(\varepsilon_2 \times (1-\varepsilon_4) \times I_i^t; 1-\beta) + binomial(\varepsilon_1 \times (\varepsilon_4 \times I_i^t + E_i^t), 1-\beta) \\
D_{TP_i}^t | X_i^t &\sim binomial(D_i^t; 1-\beta)
\end{aligned} \tag{3.9}
$$

- P_i^t is the number of confirmed positivity cases in region i at time t;
- Q_i^t is the number of death because of the epidemic in region i at time t;
- $NT_{FP_i}^t$ is the false positive of the number of test in the place i at time t;
- $NT_{TP_i}^t$ is the true positive of the number of test in the place i at time t;
- α and β are the false positive and false negative rate for testing a person respectively;
- ε_1 is the percentage of test rate of asymptomatic people;
- ε_2 is the percentage of test rate of symptomatic people;
- ε_3 is the percentage of people who are not infected by COVID19, but have a similar symptom, e.g., influenza;
- ε_4 is the percentage of infective people who are asymptomatic;
- $D_{TP_i}^t$ is the the true positive of the number of death in the place i at time t.

Comparably, we can rewrite the observation model as:

$$
\begin{aligned}
P_i^t &= \varepsilon_1(1-\varepsilon_3)(S_i^t + R_i^t)\alpha + \varepsilon_2\varepsilon_3(S_i^t + R_i^t)\alpha \\
&\quad + \varepsilon_2(1-\varepsilon_4)I_i^t(1-\beta) + \varepsilon_1(\varepsilon_4 I_i^t + E_i^t)(1-\beta) + {\upsilon_{p_i}}^t \\
Q_i^t &= D_i^t(1-\beta) + {\upsilon_{q_i}}^t
\end{aligned} \tag{3.10}
$$

where

$$
\begin{aligned}
{\upsilon_{p_i}}^t &= ({N_{FP_i}}^t - \varepsilon_1(1-\varepsilon_3)(S_i^t + R_i^t)\alpha - \varepsilon_2\varepsilon_3(S_i^t + R_i^t)\alpha) \\
&\quad + ({N_{TP_i}}^t - \varepsilon_2(1-\varepsilon_4)I_i^t(1-\beta) - \varepsilon_1(\varepsilon_4 I_i^t + E_i^t)(1-\beta)) \\
{\upsilon_{q_i}}^t &= {D_{TP_i}}^t - D_i^t(1-\beta)
\end{aligned} \tag{3.11}
$$

The observation model is one of the major contribution of this framework. We modeled the testing procedure which symptomatic and asymptomatic (the former is tested more), false positive and false negative rates of tests are taking into account. To be more precise, we split the total number of confirmed cases into two parts, one is the uninfected people but misdiagnosed as infected because of the false positive of the tests and the other one is the true infected people (the infected here are specifically refers to those suffering from COVID 19). Then in each part, symptomatic and asymptomatic aspects are considered. As we know, the uninfected people can also have a certain chance of having similar symptoms, e.g. influenza. Also, there are many infected people are asymptomatic. No matter infected or not, people with symptoms are more likely to be tested. In this procedure, ε_1 and ε_2 are correlated with ε_3 and ε_4 via two rates we can observe, namely, test positivity rate and testing rate.

Test positivity rate (R_P) means the percentage of positive tests over the total tests, then it can be represented as:

$$
{R_P}^t = \frac{P^t}{{N_{test}}^t} = \frac{\varepsilon_1 A\alpha + \varepsilon_2 B\alpha + \varepsilon_2 C(1-\beta) + \varepsilon_1 D(1-\beta)}{\varepsilon_1 A + \varepsilon_2 B + \varepsilon_2 C + \varepsilon_1 D} \tag{3.12}
$$

Similarly, testing rate (R_T) means the percentage of total tests over the total population, which is shown as following:

$$
{R_T}^t = \frac{{N_{test}}^t}{N^t} = \frac{\varepsilon_1 A + \varepsilon_2 B + \varepsilon_2 C + \varepsilon_1 D}{A + B + C + D} \tag{3.13}
$$

where $A = (1-\varepsilon_3)(S^t + R^t), B = \varepsilon_3(S^t + R^t), C = (1-\varepsilon_4)I^t, D = \varepsilon_4 I^t + E^t$

Then ε_1 and ε_2 can be represented by ε_3 and ε_4 in each iteration as:

$$
\begin{aligned}
{\varepsilon_1}^t &= \frac{(S^t + E^t + I^t + R^t){R_T}^t({R_P}^t(\varepsilon_3(S^t + R^t) + (1-\varepsilon_4)I^t) - \alpha\varepsilon_3(S^t + R^t) - (1-\beta)(1-\varepsilon_4)I^t)}{(1-\beta-\alpha)(\varepsilon_3(S^t + R^t)(\varepsilon_4 I^t + E^t) - (1-\varepsilon_3)(S^t + R^t)(1-\varepsilon_4)I^t)} \\
{\varepsilon_2}^t &= \frac{(S^t + E^t + I^t + R^t){R_T}^t}{1-\beta-\alpha} \times \\
&\frac{\alpha(1-\varepsilon_3)(S^t + R^t) + (1-\beta)(\varepsilon_4 I^t + E^t) - {R_P}^t((1-\varepsilon_3)(S^t + R^t) + (\varepsilon_4 I^t + E^t))}{\varepsilon_3(S^t + R^t)(\varepsilon_4 I^t + E^t) - (1-\varepsilon_3)(S^t + R^t)(1-\varepsilon_4)I^t}
\end{aligned}
\tag{3.14}
$$

Based on [109], which indicates the people with symptoms, e.g. fever, cough, loss of taste/smell, has higher possibility to be tested.

$$\varepsilon_1 = \zeta\varepsilon_2 \tag{3.15}$$

where $\zeta \sim [2, 4.3]$ according to different symptoms. Bring equation (3.15) into equation (3.14), then ε_3 can be represented as:

$${\varepsilon_3}^t = \frac{\zeta(1-\beta-{R_P}^t)(\varepsilon_4 I^t + E^t) + (1-\varepsilon_4)I^t(1-\beta-{R_P}^t) - \zeta(S^t + R^t)({R_P}^t - \alpha)}{(1-\zeta)(S^t + R^t)({R_P}^t - \alpha)} \tag{3.16}$$

In addition, according to this systematic review [110], the percentage of asymptomatic infected people ε_4 is around 0.2 and thus all the $\varepsilon_1, \varepsilon_2, \varepsilon_3$ can be calculated iteratively.

3.2.2.3 Model Assumptions and Limitations

As we know, any model cannot be suitable for all real scenarios and it must have its own conditions. I will summarize the assumptions and limitations of our proposed model:

- We have not considered the number of birth and natural deaths since we believe the population is balanced in the short period without the epidemic and also it is uncommon that the virus will transport from mothers to new birth.

- We assume the epidemic disease will be immunized after the infected which means the person in the recovery (R) category will not be reintroduced to susceptible.
- The infection rates among different ages are not considered, although we know that higher aged people are more vulnerable.
- The model is suitable to apply in some fixed period meaning that it is not accurate to fit the model to the epidemic from day one till the end because although we model the stochastic property of the parameters, we have not considered the trend of parameters over time, e.g. death rate will be high and recovery rate will be low at the beginning because of the lack of the effective medicine and experience. Also the different policy over time will also affect the parameters, e.g. ask people to keep the social distance and wear mask will greatly reduce the infection rate.

3.3 Epidemic Trend Estimation

Estimate the real trend of the epidemic is one of the main object of mathematical epidemiology model. To be specifically, it is important for the model to have an accurate estimation for the true values of state variables by using the noisy, incomplete, time series of reported epidemiological data.

As we know, Kalman filter first introduced by Rudolph E. Kalman [111] is an optimal filter for linear system which used for estimating the state of a time-varying system which is indirectly observed through noisy measurement. However, for nonlinear system, Kalman filter no longer applies, the solutions include Extended Kalman Filter (EKF) (local linearization of the equations)[112] or Unscented Kalman Filter (UKF) [113]. Since the EKF is based on linearizing the nonlinear system which may difficult to tune and gives unreliable estimates, we decide to use UKF to do the nonlinear state estimation to significant improve the accuracy.

3.3.1 Unscented Kalman Filter

The UKF is the filter that produce several sigma points around the current state with its covariance. Then, propagate these points by using the nonlinear map to get more accurate mean and

covariance. Based on [113, 114, 115], the procedure we used is shown below (every region is calculated parallel in the same procedure):

Recall our state space model eq (3.2) and eq (3.7), the state variables belong to:

$$\mathbf{X} \sim N(\mathbf{m}, \mathbf{P}) \tag{3.17}$$

and the process noise and observation noise belong to:

$$\begin{aligned} \mathbf{n} &\sim N(0, \mathbf{Q}) \\ \mathbf{v} &\sim N(0, \mathbf{R}) \end{aligned} \tag{3.18}$$

Prediction:

1) Generate sigma points:

$$\begin{aligned} \mathbf{X}^0_{t-1|t-1} &= \mathbf{m}_{t-1|t-1}, \\ \mathbf{X}^i_{t-1|t-1} &= \mathbf{m}_{t-1|t-1} + \left[\sqrt{n\mathbf{P}_{t-1|t-1}}\right]_i, \; i = 1, \ldots, n \\ \mathbf{X}^{i+n}_{t-1|t-1} &= \mathbf{m}_{t-1|t-1} - \left[\sqrt{n\mathbf{P}_{t-1|t-1}}\right]_{i-n}, \; i = n+1, \ldots, 2n \end{aligned} \tag{3.19}$$

2) Propagate the sigma point through the state model:

$$\hat{\mathbf{X}}^i_t = \mathbf{f}(\mathbf{X}^i_{t-1|t-1}), \; i = 0, \ldots, 2n \tag{3.20}$$

3) Compute predicted mean and predicted covariance:

$$\begin{aligned} \mathbf{m}_{t|t-1} &= \hat{\mathbf{X}}_{t|t-1} = \frac{1}{2n}\sum_{i=0}^{2n} \hat{\mathbf{X}}^i_t, \\ \mathbf{P}_{t|t-1} &= \frac{1}{2n}\sum_{i=0}^{2n} (\hat{\mathbf{X}}^i_t - \mathbf{m}_{t|t-1})(\hat{\mathbf{X}}^i_t + \mathbf{m}_{t|t-1})^T + \mathbf{Q}_{t-1}. \end{aligned} \tag{3.21}$$

(see Appendix A for the calculation of the $\mathbf{Q}_{t-1}$)

Update:

1) Update sigma points:

$$\begin{aligned}
\mathbf{X}^0_{t|t-1} &= \mathbf{m}_{t|t-1}, \\
\mathbf{X}^i_{t|t-1} &= \mathbf{m}_{t|t-1} + \left[\sqrt{n\mathbf{P}_{t|t-1}}\right]_i, \; i = 1, \ldots, n \\
\mathbf{X}^{i+n}_{t|t-1} &= \mathbf{m}_{t|t-1} - \left[\sqrt{n\mathbf{P}_{t|t-1}}\right]_{i-n}, \; i = n+1, \ldots, 2n
\end{aligned} \tag{3.22}$$

2) Propagate the sigma point through the observation model:

$$\hat{\mathbf{Y}}^i_t = \mathbf{h}(\mathbf{X}^i_{t|t-1}), \; i = 0, \ldots, n \tag{3.23}$$

3) Compute predicted mean, predicted covariance and the cross-covariance:

$$\begin{aligned}
\boldsymbol{\mu}_t &= \hat{\mathbf{Y}}_{t|t-1} = \frac{1}{2n}\sum_{i=0}^{2n} \hat{\mathbf{Y}}^i_t, \\
\mathbf{S}_t &= \mathbf{P}_{\hat{\mathbf{Y}}} = \frac{1}{2n}\sum_{i=0}^{2n} (\hat{\mathbf{Y}}^i_t - \boldsymbol{\mu}_t)(\hat{\mathbf{Y}}^i_t - \boldsymbol{\mu}_t)^T + \mathbf{R}_t, \\
\mathbf{C}_t &= \mathbf{P}_{\hat{\mathbf{X}}\hat{\mathbf{Y}}} = \frac{1}{2n}\sum_{i=0}^{2n} (\mathbf{X}^i_{t|t-1} - \mathbf{m}_{t|t-1})(\hat{\mathbf{Y}}^i_t - \boldsymbol{\mu}_t)^T,
\end{aligned} \tag{3.24}$$

(see Appendix A for the calculation of the $\mathbf{R}_t$)

4) Compute the filter gain, state mean and covariance:

$$\begin{aligned}
\mathbf{K}_t &= \mathbf{C}_t\mathbf{S}_t^{-1}, \\
\mathbf{m}_{t|t} &= \mathbf{m}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \boldsymbol{\mu}_t), \\
\mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T.
\end{aligned} \tag{3.25}$$

3.3.2 Maximum-Likelihood Adaptive Filtering Computation

In the previous sections, we have assumption that the parameters of the state model and observation model are known, however, it is not the case in real world. Thus, estimate the parameters based on the observation data is the key step to make our model useful in practice. In this case, we consider to use maximum-likelihood adaptive filtering to do the parameter estimation in nonlinear state-space models [116, 117, 118]. All of the variables following are only consider one region, so the corner mark of the region is ignored.

As we mentioned before, there are several parameters which govern the model eq(3.5) and eq(3.8), e.g. $\lambda_S, \lambda_E, \varepsilon_3, \varepsilon_4, \ldots$. Let call those paramters $\boldsymbol{\theta}$. $\boldsymbol{\theta}$ is a set of the continuous unknown model parameters. Then, suppose that the observation data we have $\mathbf{Y}_{1:t}$ (time series data up to time t). The log-likelihood of model $\boldsymbol{\theta}$ at time t is defined as

$$\begin{aligned} L_t(\boldsymbol{\theta}) &= \log\ p_{\boldsymbol{\theta}}(\mathbf{Y}_{1:t}) = \log\left[p_{\boldsymbol{\theta}}(\mathbf{Y}_t \mid \mathbf{Y}_{1:t-1}) p_{\boldsymbol{\theta}}(\mathbf{Y}_{t-1} \mid \mathbf{Y}_{1:t-2}) \cdots p_{\boldsymbol{\theta}}(\mathbf{Y}_2 \mid \mathbf{Y}_1) p_{\boldsymbol{\theta}}(\mathbf{Y}_1)\right] \\ &= L_{t-1}(\boldsymbol{\theta}) + \log\ p_{\boldsymbol{\theta}}(\mathbf{Y}_t \mid \mathbf{Y}_{1:t-1}), \end{aligned} \tag{3.26}$$

where

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{Y}_t \mid \mathbf{Y}_{1:t-1}) &= -\frac{1}{2} log\,|2\pi\,\mathbf{S}_t(\boldsymbol{\theta})| - \frac{1}{2}\,\mathbf{v}_t^T(\boldsymbol{\theta})\,\mathbf{S}_t^{-1}(\boldsymbol{\theta})\,\mathbf{v}_t(\boldsymbol{\theta}), \\ \mathbf{v}_t &= \mathbf{y}_t - \boldsymbol{\mu}_t. \end{aligned} \tag{3.27}$$

The quantities $\boldsymbol{\mu}_t$ and $\mathbf{S}_t$ are calculated in the UKF recursion. Then, the target is to maximize the log-likelihood $L_t(\theta)$:

$$\theta_{ML} = argmax_\theta\ \log\ p_\theta(\mathbf{Y}_{1:t}) \tag{3.28}$$

There are several ways for this optimization problem which only contain continuous parameters. For example, Expectation-Maximization (EM) algorithm, especially when it has a closed-form solution which avoids the recursive gradient calculation. However, there is no close-form solution when we maximizing Q function in the M-step since the correlation between the states and noise, and thus requires the numerical optimization, e.g. Broyden–Fletcher–Goldfarb–Shanno (BFGS)

algorithm [119], in each recursion. In this case, using numerical optimization insider EM is quite cumbersome [118] because we can directly calculate the gradients of the objective function (log-likelihood function) and do the optimization with the same effort.

The derivative of the log-likelihood function equation (3.26) is obtained by:

$$\begin{aligned}\frac{\partial L_t(\boldsymbol{\theta})}{\partial \theta_i} &= \frac{\partial L_{t-1}(\boldsymbol{\theta})}{\partial \theta_i} \\ &\quad - \frac{1}{2} tr(\mathbf{S}_t^{-1}(\boldsymbol{\theta}) \frac{\partial \mathbf{S}_t(\boldsymbol{\theta})}{\partial \theta_i}) - \frac{1}{2} \mathbf{v}_t^T(\boldsymbol{\theta}) \mathbf{S}_t^{-1}(\boldsymbol{\theta}) \frac{\partial \mathbf{S}_t(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{S}_t^{-1}(\boldsymbol{\theta}) \mathbf{v}_t(\boldsymbol{\theta}) + \mathbf{v}_t^T(\boldsymbol{\theta}) \mathbf{S}_t^{-1} \frac{\partial \mathbf{v}_t(\boldsymbol{\theta})}{\partial \theta_i}\end{aligned} \tag{3.29}$$

Then only $\frac{\partial \mathbf{S}_t(\theta)}{\partial \theta_i}$ and $\frac{\partial \mathbf{v}_t(\theta)}{\partial \theta_i}$ are need to computed recursively. (see the Appendix B for the details).

Metaheuristic algorithm can also be applied to this problem, e.g. the fish school search algorithm discussed in last chapter. In this case, there is no need for the gradient of the objective function anymore. (see the chapter 2 for the details).

3.4 Numerical Experiments

In this section, we present the result of a comprehensive set of numerical experiments, including simulations with different set of parameters and estimations of unknown parameters with synthetic data.

Here are the parameters selection based on the prior knowledge in the following simulation and will also be used in the synthetic generation: False positive rate $\alpha = 0.01$ and false negative rate $\beta = 0.15$ [120]; λ_S is different with different regions but the average will be 0.3; $\lambda_E = 0.1$ based on the CDC's website which the incubation period ranges from 2-14 days; $\lambda_R = 0.07$ based on CDC's website which the person will recovery in 2 weeks on average; $\lambda_D = 0.01$ based on the number of death reported every day.

3.4.1 Experiment 1: Simulation of Spatial and Temporal Dynamics of Epidemic

In the following cases of simulation, we would like to show our model's ability to demonstrate the spatial and temporal dynamics of epidemic. Fig 3.1, Fig 3.2, Fig 3.3, Fig 3.4 are the plots for

the results at day 1, day 60, day 90, and day 100, respectively.

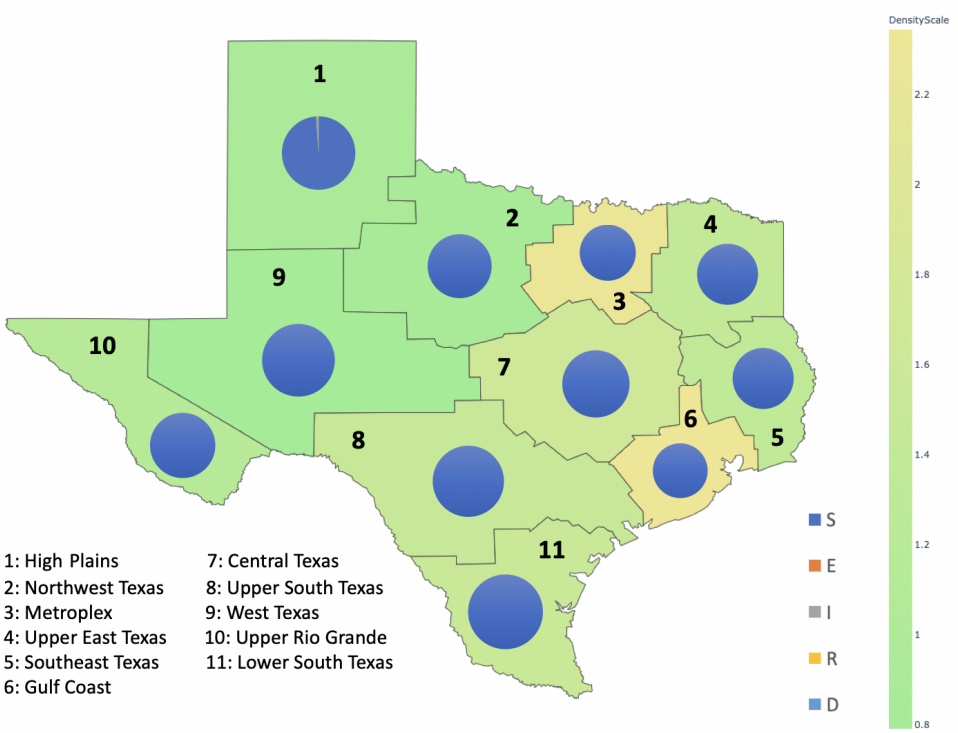

Figure 3.1: The simulation results at t = 1 day

From the Fig 3.1 we can see that the epidemic originated in region 1 with only a few people was infected. Then, in Fig 3.2, 60 day past, the epidemic spread to other regions, but still very limited.

However, in Fig 3.3, only another 30 days after, the epidemic spread much faster than before, especially in the originated region and the regions with large density, e.g. region 3 and region 6. Then, the last plot Fig 3.4, only another 10 days. The number of infected people almost doubled. From this simulation, we can clearly understand that the epidemic is easy to control at the very early stage, however, the epidemic will spread and develop at an exponential rate after a period of time. Meaning that introduce relevant policy to control the epidemic at the early stage is necessary and useful, e.g. keep social distance, wearing mask or even lockdown the limited critical area (region 1). In the next experiment, we will focus on one region to show the effect on the results with different

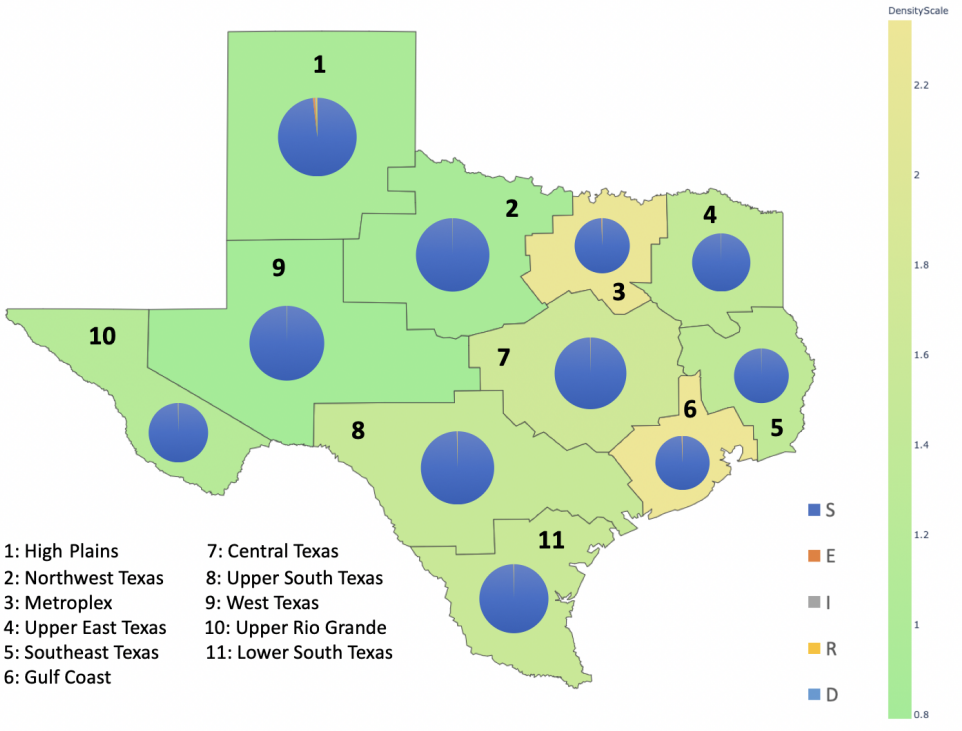

Figure 3.2: The simulation results at t = 60 days

policy.

3.4.2 Experiment 2: Simulation of Effect of Different Policy

In this experiment, we will compare the results of the same region (region 9) under different policy. Fig 3.5 is the result without any control. We can see that almost 80k people dead and the epidemic ended with herd immunity. Almost every has been infected which is the worst case.

Fig 3.6 presents the region 9 is under the policy of keeping social distance and wearing masks. From the figure we can see that the growth and decline have all eased a lot which because this policy will decrease the infection rate a lot in our model. However, it needs a bit long time to end the epidemic, although it will decrease the infected rate. In addition, many people still don't have the immunity which might cause the epidemic outbreak again.

Fig 3.7 depicts the scene that a certain number of vaccines will be delivered to people every day. In our model, it will transport a certain number of people from susceptible category to recovery

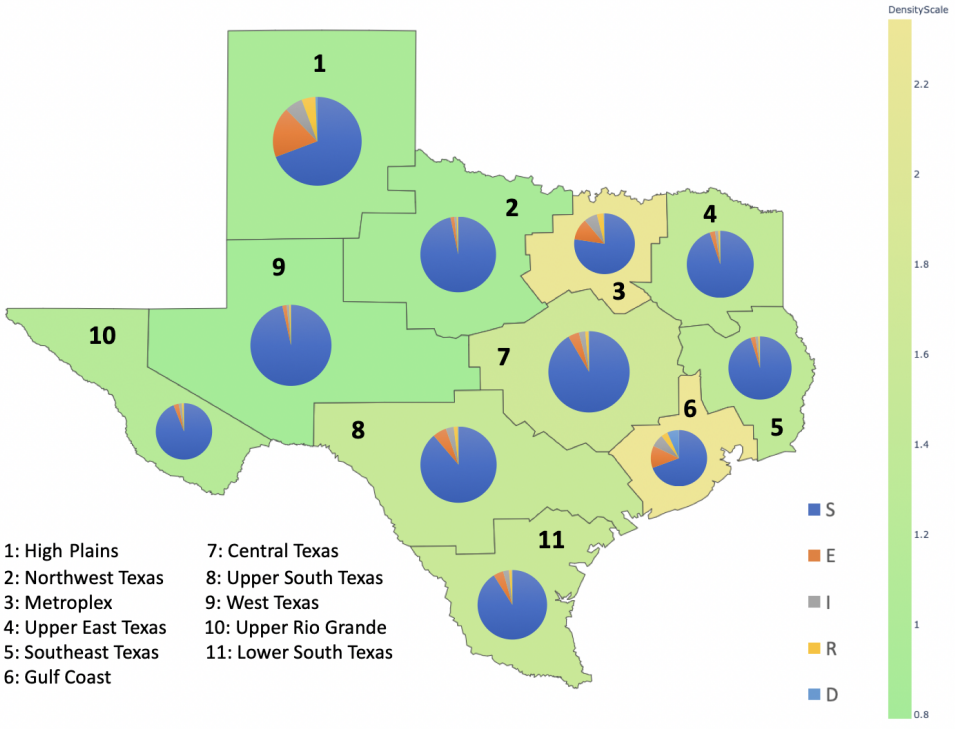

Figure 3.3: The simulation results at t = 90 days

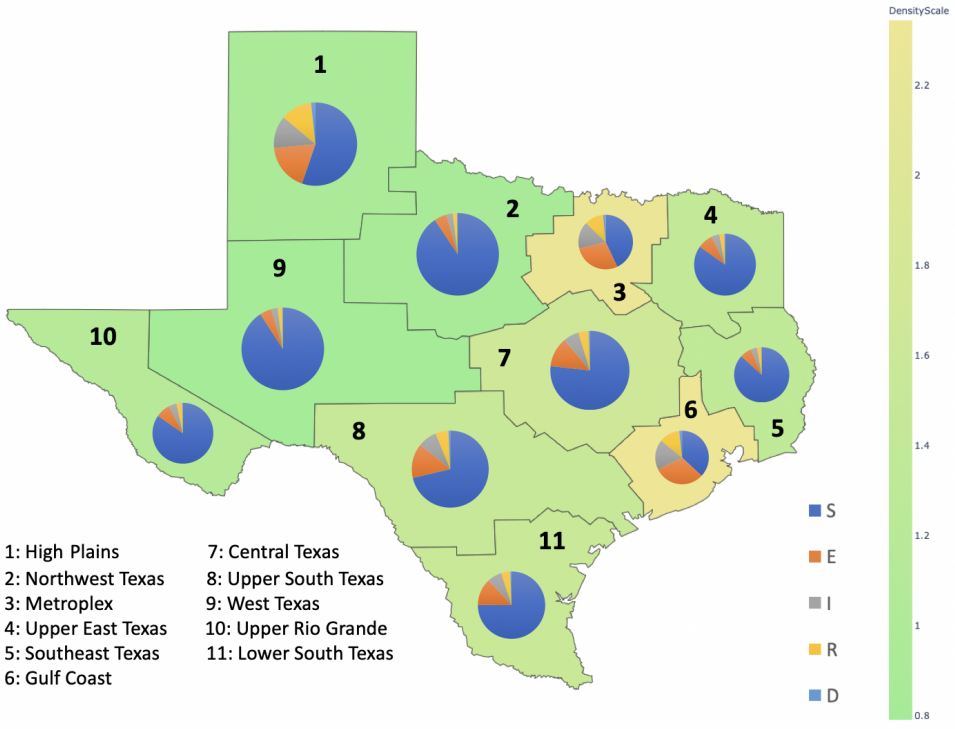

Figure 3.4: The simulation results at t = 100 days

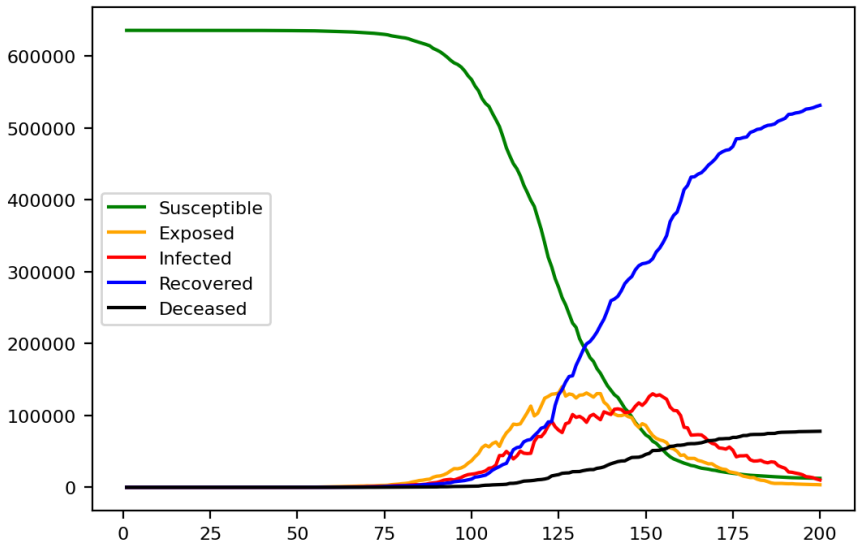

Figure 3.5: The results of region 9 without any control

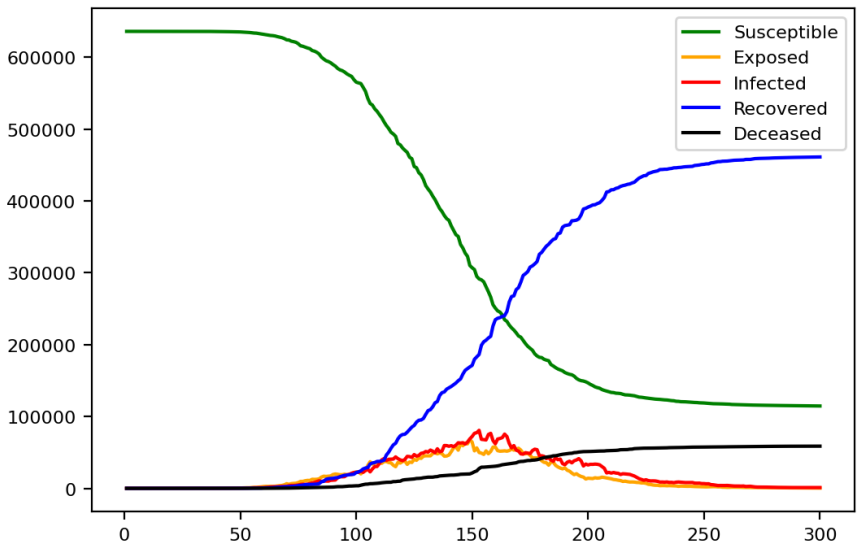

Figure 3.6: The results of region 9 with policy of keeping social distance or wearing mask

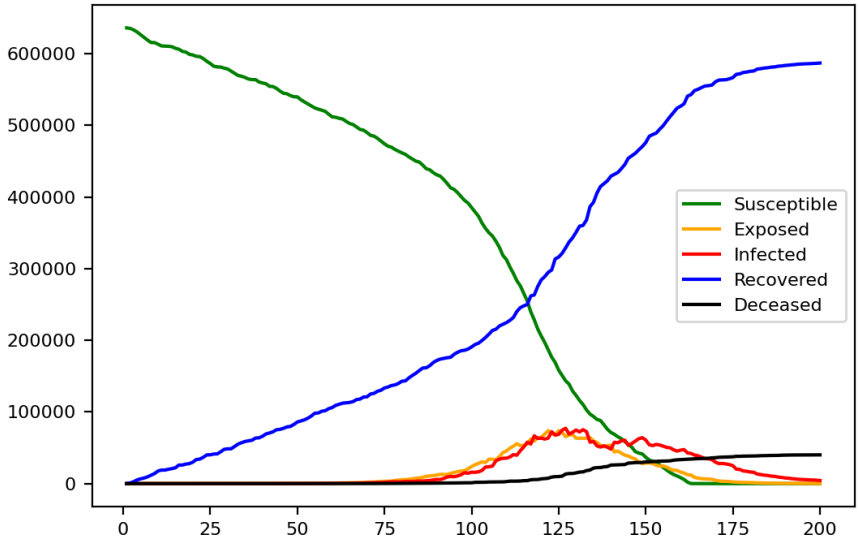

Figure 3.7: The results of region 9 with around 1500 vaccines per day

category, which means that the number of people in the recovery category will growth very fast and get to the herd immunity in a short period. At that time, almost all of the people have the immunity.

This simulation demonstrates that vaccination is the best way to stop the epidemic and if we can introduce some policy during this period will get even better results.

3.4.3 Experiment 3: Epidemic Trend Estimation

The ability to estimate the epidemic trend based on the reported noisy observations is important for a mathematical model. This experiment will demonstrate this ability with two scenarios. One is based on the assumption that we know all the system parameters. Another one is more close to reality that we need to estimate the unknown parameters first and check the performance.

Fig 3.9 is the figure that all of the parameters are known. Although it is not the case of real world, it presents that our model can estimate the epidemic trend well. Then Fig 3.8 depicts the situation that we need to estimate the unknown parameters by BFGS or metaheuristic algorithm and use those estimated parameters to predict the trend. In this experiment, we assume $\lambda_S, \lambda_R, \lambda_D$ are unknown. We can see that without the prior knowledge of the parameters, our model can still

estimate the epidemic trend well.

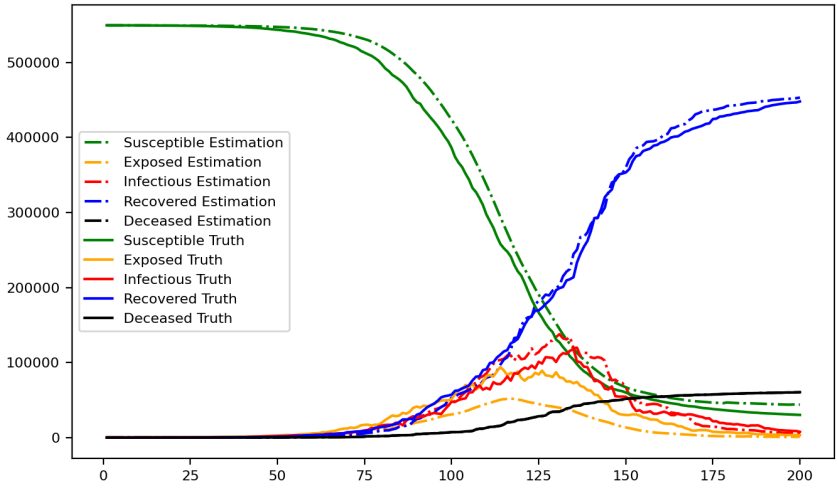

Figure 3.8: The estimation results of region 2 with all the parameters known

3.5 Conclusion

We developed a novel spatial-temporal nonlinear state space model for COVID-19 transmission, based on a discrete-time susceptible - exposed - infected - recovered - deceased (SEIRD) model, which can estimate the hidden states and parameters from a noisy, incomplete, time series of reported epidemiological data, by applying Unscented Kalman Filter (UKF), Maximum Likelihood (ML) adaptive filtering and Broyden–Fletcher–Goldfarb–Shanno (BFGS)/metaheuristic optimization. We used a comprehensive set of simulations and experiments, using synthetic data to demonstrate that our model can not only effectively simulate the different scenarios of epidemic, such as different lock-down patterns and vaccination scenarios, but also reliably estimate the unknown parameters which are important for predicting future trends of the epidemic and more accurately evaluating the effectiveness of public health policies.

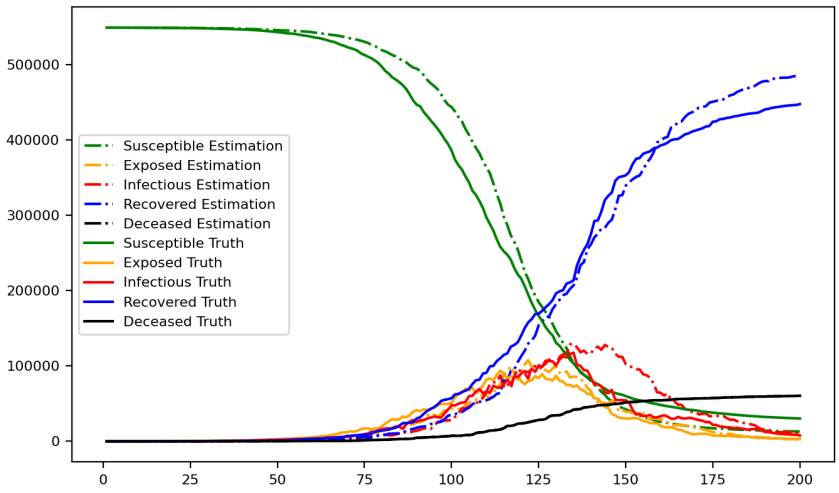

Figure 3.9: The estimation results of region 2 using the estimated unknown parameters

3.6 Appendix A

Given the process noise and observation noise in state model and observation model, the covariance matrix of those noises are calculated as:

1) Covariance of the process noise (**Q**):

$$\begin{aligned} Var(n^t_{S_i}) &= Var(N^t_{S_i}) + \sum_{j_1}\sum_{j_2} Cov(NB^t_{S_{j_1}}, NB^t_{S_{j_2}}), \quad j_1 \neq i, j_2 \neq i \\ &= S^t_i \frac{\lambda_S I^t_i}{N^t_i}(1 - \frac{\lambda_S I^t_i}{N^t_i}) + \sum_j S^t_i \frac{c_{ij}\lambda_S I^t_j}{N^t_j}(1 - \frac{c_{ij}\lambda_S I^t_j}{N^t_j}) \\ &\quad - \sum_{j_1}\sum_{j_2} S^t_i \frac{c_{ij_1}\lambda_S I^t_{j_1}}{N^t_{j_1}} \frac{c_{ij_2}\lambda_S I^t_{j_2}}{N^t_{j_2}}, \quad j_1 \neq j_2 \neq i, j \neq i \end{aligned} \tag{3.30}$$

$$
\begin{aligned}
Var(n^t_{E_i}) &= Var(N^t_{S_i}) + Var(N^t_{E_i}) + \sum_{j_1}\sum_{j_2} Cov(NB^t_{S_{j_1}}, NB^t_{S_{j_2}}), \quad j \neq i \\
&= S^t_i \frac{\lambda_S I^t_i}{N^t_i}(1 - \frac{\lambda_S I^t_i}{N^t_i}) + \sum_j S^t_i \frac{c_{ij}\lambda_S I^t_j}{N^t_j}(1 - \frac{c_{ij}\lambda_S I^t_j}{N^t_j}) \\
&\quad + E^t_i \lambda_E (1 - \lambda_E) - \sum_{j_1}\sum_{j_2} S^t_i \frac{c_{ij_1}\lambda_S I^t_{j_1}}{N^t_{j_1}} \frac{c_{ij_2}\lambda_S I^t_{j_2}}{N^t_{j_2}}, \quad j_1 \neq j_2 \neq i, j \neq i
\end{aligned} \tag{3.31}
$$

$$
\begin{aligned}
Var(n^t_{I_i}) &= Var(N^t_{E_i}) + Var(N^t_{R_i}) + Var(N^t_{D_i}) + 2Cov(N^t_{R_i}, N^t_{D_i}) \\
&= E^t_i \lambda_E (1 - \lambda_E) + I^t_i \lambda_R (1 - \lambda_R) + I^t_i \lambda_D (1 - \lambda_D) - 2 I^t_i \lambda_R \lambda_D \\
&= E^t_i \lambda_E (1 - \lambda_E) + I^t_i (\lambda_R + \lambda_D)(1 - (\lambda_R + \lambda_D))
\end{aligned} \tag{3.32}
$$

$$
Var(n^t_{R_i}) = Var(N^t_{R_i}) = I^t_i \lambda_R (1 - \lambda_R) \tag{3.33}
$$

$$
Var(n^t_{D_i}) = Var(N^t_{D_i}) = I^t_i \lambda_D (1 - \lambda_D) \tag{3.34}
$$

$$
Cov(n^t_{S_i}, n^t_{E_i}) = -Var(N^t_{S_i}) - \sum_{j_1}\sum_{j_2} Cov(NB^t_{S_{j_1}}, NB^t_{S_{j_2}}) = -Var(n^t_{S_i}) \tag{3.35}
$$

$$
Cov(n^t_{E_i}, n^t_{I_i}) = -Var(N^t_{E_i}) = -E^t_i \lambda_E (1 - \lambda_E) \tag{3.36}
$$

$$
\begin{aligned}
Cov(n^t_{I_i}, n^t_{R_i}) &= -Var(N^t_{R_i}) - Cov(N^t_{R_i}, N^t_{D_i}) \\
&= -I^t_i \lambda_R (1 - \lambda_R) + I^t_i \lambda_R \lambda_D \\
&= -I^t_i \lambda_R (1 - (\lambda_R + \lambda_D))
\end{aligned} \tag{3.37}
$$

$$
\begin{aligned}
Cov(n^t_{I_i}, n^t_{D_i}) &= -Var(N^t_{D_i}) - Cov(N^t_{R_i}, N^t_{D_i}) \\
&= -I^t_i \lambda_D (1-\lambda_D) + I^t_i \lambda_R \lambda_D \\
&= -I^t_i \lambda_D (1-(\lambda_R+\lambda_D)) \\
Cov(n^t_{R_i}, n^t_{D_i}) &= -I^t_i \lambda_R \lambda_D
\end{aligned}
\tag{3.38}
$$

The left will be all zero.

2) Covariance of the observation noise (**R**):

$$
\begin{aligned}
var(v_{p_i}{}^t) =& \varepsilon_1(1-\varepsilon_3)(S^t_i + R^t_i)\alpha(1-\alpha) + \varepsilon_2\varepsilon_3(S^t_i + R^t_i)\alpha(1-\alpha) \\
&+ \varepsilon_2(1-\varepsilon_4)I^t_i\beta(1-\beta) + \varepsilon_1(\varepsilon_4 I^t_i + E^t_i)\beta(1-\beta) + v_{p_i}{}^t \\
var(v_{q_i}{}^t) =& D^t_i \beta(1-\beta)
\end{aligned}
\tag{3.39}
$$

The left will be all zero.

3.7 Appendix B

Direct likelihood based optimization details [118].

$$
\begin{aligned}
\frac{\partial \mathbf{v}_t}{\partial \theta_i} &= -\frac{\partial \boldsymbol{\mu}_t}{\partial \theta_i} \\
\frac{\partial \boldsymbol{\mu}_t}{\partial \theta_i} &= \sum_j \{\frac{1}{2n}[\mathbf{H}_x(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) \times (\frac{\partial \mathbf{m}_{t-1|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{t-1|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j) + \frac{\partial h}{\partial \theta_i}(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta})]\} \\
\frac{\partial \mathbf{S}_t}{\partial \theta_i} &= \frac{1}{2n}\sum_j \{[\mathbf{H}_x(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta})(\frac{\partial \mathbf{m}_{t|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{t|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j) + \frac{\partial h}{\partial \theta_i}(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \frac{\partial \boldsymbol{\mu}_t}{\partial \theta_i}] \\
&\quad \times [h(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \boldsymbol{\mu}_t]^T + [h(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \boldsymbol{\mu}_t] \times [\mathbf{H}_x(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) \\
&\quad \times (\frac{\partial \mathbf{m}_{t-1|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{t-1|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j) + \frac{\partial h}{\partial \theta_i}(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \frac{\partial \boldsymbol{\mu}_t}{\partial \theta_i}]^T\} + \frac{\partial \mathbf{R}}{\partial \theta_i}
\end{aligned}
\tag{3.40}
$$

where $\mathbf{H_x}$ is the Jacobian of observation model $\mathbf{h}$, and $\mathbf{L}$ is the Cholesky factor ($\mathbf{P} = \mathbf{L}\mathbf{L}^T$). The partial derivative of the Cholesky factor $\frac{\partial \mathbf{L}}{\partial \theta_i}$ is calculated by

$$\frac{\partial \mathbf{L}}{\partial \theta_i} = \mathbf{L}\,\phi(\mathbf{L}^{-1}\frac{\partial \mathbf{P}}{\partial \theta_i}\mathbf{L}^{-T}) \tag{3.41}$$

where $\phi(\cdot)$ is a function returning the lower triangular part and half the diagonal of the argument as follows:

$$\phi_{ij}(\mathbf{M}) = \begin{cases} M_{ij}, & \text{if i} > \text{ j}, \\ \frac{1}{2}M_{ij}, & \text{if i} = \text{j}, \\ 0, & \text{if i} < \text{ j}. \end{cases}$$

$$\frac{\partial \mathbf{P}_{t|t}}{\partial \theta_i} = \frac{\partial \mathbf{P}_{t|t-1}}{\partial \theta_i} - \frac{\partial \mathbf{K}_t}{\partial \theta_i}\mathbf{S}_t\mathbf{K}_t^T - \mathbf{K}_t\frac{\partial \mathbf{S}_t}{\partial \theta_i}\mathbf{K}_t^T - \mathbf{K}_t\mathbf{S}_t\frac{\partial \mathbf{K}_t^T}{\partial \theta_i} \tag{3.42}$$

$$\begin{aligned} \frac{\partial \mathbf{P}_{t|t-1}}{\partial \theta_i} = \frac{1}{2n}\sum_j \{ & [\mathbf{Fx}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) \times (\frac{\partial \mathbf{m}_{t-1|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{t-1|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j) \\ & + \frac{\partial \mathbf{f}}{\partial \theta_i}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \frac{\partial \mathbf{m}_{t|t-1}}{\partial \theta_i}] \times [\mathbf{f}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \mathbf{m}_{t|t-1}]^T \\ & + [\mathbf{f}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \mathbf{m}_{t|t-1}] \times [\mathbf{Fx}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) \\ & \times (\frac{\partial \mathbf{m}_{t-1|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{t-1|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j) + \frac{\partial \mathbf{f}}{\partial \theta_i}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \frac{\partial \mathbf{m}_{t|t-1}}{\partial \theta_i}]^T \} + \frac{\partial \mathbf{Q}}{\partial \theta_i} \end{aligned} \tag{3.43}$$

where $\mathbf{F}_\mathbf{x}$ is the Jacobian of the state model $\mathbf{f}$.

$$\begin{aligned} \frac{\partial \mathbf{m}_{t|t}}{\partial \theta_i} &= \frac{\partial \mathbf{m}_{t|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{K}_t}{\partial \theta_i}\mathbf{v}_t + \mathbf{K}_t\frac{\partial \mathbf{v}_t}{\partial \theta_i} \\ \frac{\partial \mathbf{m}_{t|t-1}}{\partial \theta_i} &= \sum_j \{ \frac{1}{2n}[\mathbf{Fx}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) \times (\frac{\partial \mathbf{m}_{t-1|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{t-1|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j) \\ &\quad + \frac{\partial \mathbf{f}}{\partial \theta_i}(\mathbf{m}_{t-1|t-1} + \mathbf{L}_{t-1|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta})] \} \end{aligned} \tag{3.44}$$

$$
\begin{aligned}
\frac{\partial \mathbf{K}_t}{\partial \theta_i} &= \frac{\partial \mathbf{C}_t}{\partial \theta_i}\mathbf{S}_t^{-1} - \mathbf{C}_t\mathbf{S}_t^{-1}\frac{\partial \mathbf{S}_t}{\partial \theta_i}\mathbf{S}_t^{-1} \\
\frac{\partial \mathbf{C}_t}{\partial \theta_i} &= \sum_j \{\frac{1}{2n}[\frac{\partial \mathbf{L}_{t|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j(\mathbf{h}(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \boldsymbol{\mu}_t)^T + \mathbf{L}_{t|t-1} \\
&\quad \times \boldsymbol{\xi}_j[\mathbf{H}_x(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta})(\frac{\partial \mathbf{m}_{t|t-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{t|t-1}}{\partial \theta_i}\boldsymbol{\xi}_j) + \frac{\partial h}{\partial \theta_i}(\mathbf{m}_{t|t-1} + \mathbf{L}_{t|t-1}\boldsymbol{\xi}_j, \boldsymbol{\theta}) - \frac{\partial \boldsymbol{\mu}_t}{\partial \theta_i}]^T]\}
\end{aligned}
\tag{3.45}
$$

4. SUMMARY AND CONCLUSIONS

In this dissertation, we proposed several nonlinear state space models for efficiently state and parameter estimations in nonlinear dynamical systems with applications in biochemical regulatory networks and epidemic models, respectively.

In section 2, we presented PALLAS, a new framework for inference of Boolean gene regulatory networks (GRN) and protein-protein interaction network (PPI) from time series data. The algorithm avoids ad-hoc binarization of the expression data and allows inference of large networks by employing penalized maximum likelihood as a regularization method, applying particle filtering (in GRN framework) and ABC-SMC algorithm (in PPI framework) for the computation of the likelihood, and using a novel version of the fish school search particle swarm algorithm to search the parameter space. Numerical experiments using synthetic time series data show that PALLAS works well in PPI inference and outperforms other well-known GRN inference methods. The performance of PALLAS was also demonstrated on real gene expression time series data from the SOS DNA repair and Biofilm formation pathways in E. Coli. As a sophisticated state-space method for Boolean GRN inference directly from noisy gene expression data, without the need of ad-hoc binarization, PALLAS is computationally expensive. Results provided indicate that execution time increases linearly with the number of fish used in the MFSS algorithm. The user can adjust the running time by changing the number of fish, at a cost to performance. Future work will include the implementation of PALLAS on high-performance parallel architectures, which will enable the inference of larger networks.

In section 3, we proposed a nonlinear state-space framework to model the outbreak of the COVID 19. It is a stochastic and spatial-temporal model which satisfied the reality and can help people better understand the epidemic, predict its future trends, explore intervention scenarios and ultimately control the epidemics, such as lock-down or vaccination. By applying Unscented Kalman Filter (UKF), Maximum Likelihood (ML) adaptive filtering and Broyden–Fletcher–Goldfarb–Shanno (BFGS)/metaheuristic optimization algorithm, make our framework powerful than most of the ex-

isting methods since the ability of estimating the hidden states and parameters from a noisy, incomplete, time series of reported epidemiological data. A comprehensive set of simulations and experiments using synthetic data demonstrates that our model can not only effectively simulate the different scenarios of epidemic, but also reliably estimate the unknown parameters which are important for predicting future trends of the epidemic and more accurately evaluating the effectiveness of public health policies.

REFERENCES

[1] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*. Oxford university press, 2012.

[2] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[4] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. CRC press, 2011.

[5] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of object tracking*. Cambridge University Press, 2011.

[6] J. D. Murray, *Mathematical biology: I. An introduction*, vol. 17. Springer Science & Business Media, 2007.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[8] U. Braga-Neto, "Optimal state estimation for boolean dynamical systems," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 1050–1054, IEEE, 2011.

[9] M. Imani and U. M. Braga-Neto, "Maximum-likelihood adaptive filter for partially observed boolean dynamical systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 359–371, 2017.

[10] C. J. Bastos Filho, F. B. de Lima Neto, A. J. Lins, A. I. Nascimento, and M. P. Lima, "A novel search algorithm based on fish school behavior," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pp. 2646–2651, IEEE, 2008.

[11] C. Bastos-Filho and D. Nascimento, "An enhanced fish school search algorithm," in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on*, pp. 152–157, IEEE, 2013.

[12] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, pp. 700–721, 1927.

[13] V. Dukic, H. F. Lopes, and N. G. Polson, "Tracking epidemics with state-space seir and google flu trends," *Unpublished manuscript*, 2012.

[14] S. Zhong, Q. Huang, and D. Song, "Simulation of the spread of infectious diseases in a geographical environment," *Science in China Series D: Earth Sciences*, vol. 52, no. 4, pp. 550–561, 2009.

[15] R. De Smet and K. Marchal, "Advantages and limitations of current network inference methods," *Nature Reviews Microbiology*, vol. 8, no. 10, p. 717, 2010.

[16] E. P. van Someren, L. F. Wessels, and M. J. Reinders, "Linear modeling of genetic networks from experimental data.," in *Ismb*, pp. 355–366, 2000.

[17] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mrna expression levels during cns development and injury," in *Biocomputing'99*, pp. 41–52, World Scientific, 1999.

[18] K. Murphy, S. Mian, *et al.*, "Modelling gene expression data using dynamic bayesian networks," tech. rep., Technical report, Computer Science Division, University of California, Berkeley, CA, 1999.

[19] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Journal of computational biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[20] D. C. Weaver, C. T. Workman, and G. D. Stormo, "Modeling regulatory networks with weight matrices," in *Biocomputing'99*, pp. 112–123, World Scientific, 1999.

[21] T. Chen, H. L. He, and G. M. Church, "Modeling gene expression with differential equations," in *Biocomputing'99*, pp. 29–40, World Scientific, 1999.

[22] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863–14868, 1998.

[23] S. Barman and Y.-K. Kwon, "A novel mutual information-based boolean network inference method from time-series gene expression data," *PloS one*, vol. 12, no. 2, p. e0171097, 2017.

[24] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of theoretical biology*, vol. 22, no. 3, pp. 437–467, 1969.

[25] R. Albert and H. G. Othmer, "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster," *Journal of theoretical biology*, vol. 223, no. 1, pp. 1–18, 2003.

[26] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, "The yeast cell-cycle network is robustly designed," *Proceedings of the National Academy of Sciences*, vol. 101, no. 14, pp. 4781–4786, 2004.

[27] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle," *"Bionformatics"*, vol. 22, no. 14, pp. e124–e131, 2006.

[28] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery," *Journal of molecular medicine*, vol. 77, no. 6, pp. 469–480, 1999.

[29] I. Shmulevich, I. Gluhovsky, R. F. Hashimoto, E. R. Dougherty, and W. Zhang, "Steady-state analysis of genetic regulatory networks modelled by probabilistic boolean networks," *Comparative and functional genomics*, vol. 4, no. 6, pp. 601–608, 2003.

[30] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[31] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From boolean to probabilistic boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[32] D. Cheng and H. Qi, "A linear representation of dynamics of boolean networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2251–2258, 2010.

[33] H. Qi and D. Cheng, "Analysis and control of boolean networks: A semi-tensor product approach," in *2009 7th Asian Control Conference*, pp. 1352–1356, IEEE, 2009.

[34] A. Irrthum, L. Wehenkel, P. Geurts, *et al.*, "Inferring regulatory networks from expression data using tree-based methods," *PloS one*, vol. 5, no. 9, p. e12776, 2010.

[35] G. Sanguinetti *et al.*, "Gene regulatory network inference: an introductory survey," in *Gene Regulatory Networks*, pp. 1–23, Springer, 2019.

[36] A.-C. Haury, F. Mordelet, P. Vera-Licona, and J.-P. Vert, "Tigress: trustful inference of gene regulation using stability selection," *BMC systems biology*, vol. 6, no. 1, p. 145, 2012.

[37] V. A. Smith, J. Yu, T. V. Smulders, A. J. Hartemink, and E. D. Jarvis, "Computational inference of neural information flow networks," *PLoS computational biology*, vol. 2, no. 11, p. e161, 2006.

[38] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, "On learning gene regulatory networks under the boolean network model," *Machine learning*, vol. 52, no. 1-2, pp. 147–167, 2003.

[39] S. Liang, S. Fuhrman, and R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," 1998.

[40] S. Barman and Y.-K. Kwon, "A boolean network inference from time-series gene expression data using a genetic algorithm," *Bioinformatics*, vol. 34, no. 17, pp. i927–i933, 2018.

[41] L. Chen, D. Kulasiri, and S. Samarasinghe, "A novel data-driven boolean model for genetic regulatory networks," *Frontiers in physiology*, vol. 9, p. 1328, 2018.

[42] S. Kimura, K. Sonoda, S. Yamane, H. Maeda, K. Matsumura, and M. Hatakeyama, "Function approximation approach to the inference of reduced ngnet models of genetic networks," *BMC bioinformatics*, vol. 9, no. 1, p. 23, 2008.

[43] S. Kimura, S. Nakayama, and M. Hatakeyama, "Genetic network inference as a series of discrimination tasks," *Bioinformatics*, vol. 25, no. 7, pp. 918–925, 2009.

[44] M. Julfikar Islam, T. M.S.R., and A. M.A.H., "Gene regulatory network inference using prominent swarm intelligence methods," *Computational Biology and Bioinformatics*, vol. 4, no. 5, pp. 37–44, 2016.

[45] A. Jaimovich, G. Elidan, H. Margalit, and N. Friedman, "Towards an integrated protein–protein interaction network: A relational markov network approach," *Journal of Computational Biology*, vol. 13, no. 2, pp. 145–164, 2006.

[46] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences*, vol. 98, no. 8, pp. 4569–4574, 2001.

[47] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, *et al.*, "A comprehensive analysis of protein–protein interactions in saccharomyces cerevisiae," *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[48] K. Raman, "Construction and analysis of protein–protein interaction networks," *Automated experimentation*, vol. 2, no. 1, p. 2, 2010.

[49] S. Haider and R. Pal, "Boolean network inference from time series data incorporating prior biological knowledge," *BMC genomics*, vol. 13, no. S6, p. S9, 2012.

[50] Y. Hu, Y. Zhang, J. Ren, Y. Wang, Z. Wang, and J. Zhang, "Statistical approaches for the construction and interpretation of human protein-protein interaction network," *BioMed research international*, vol. 2016, 2016.

[51] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf, "Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of the Royal Society Interface*, vol. 6, no. 31, pp. 187–202, 2009.

[52] N. Ghaffari, M. R. Yousefi, C. D. Johnson, I. Ivanov, and E. R. Dougherty, "Modeling the next generation sequencing sample processing pipeline for the purposes of classification," *BMC bioinformatics*, vol. 14, no. 1, p. 307, 2013.

[53] T. J. Hardcastle and K. A. Kelly, "bayseq: empirical bayesian methods for identifying differential expression in sequence count data," *BMC bioinformatics*, vol. 11, no. 1, p. 422, 2010.

[54] S. Anders and W. Huber, "Differential expression analysis for sequence count data," *Genome biology*, vol. 11, no. 10, p. R106, 2010.

[55] Y. Sun, U. Braga-Neto, and E. R. Dougherty, "A systematic model of the lc-ms proteomics pipeline," *BMC genomics*, vol. 13, no. S6, p. S2, 2012.

[56] Y. Taniguchi, P. J. Choi, G.-W. Li, H. Chen, M. Babu, J. Hearn, A. Emili, and X. S. Xie, "Quantifying e. coli proteome and transcriptome with single-molecule sensitivity in single cells," *science*, vol. 329, no. 5991, pp. 533–538, 2010.

[57] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[58] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[59] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, pp. 107–113, IET, 1993.

[60] U. Braga-Neto, "Particle filtering approach to state estimation in boolean dynamical systems," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 81–84, IEEE, 2013.

[61] M. Imani and U. M. Braga-Neto, "Particle filters for partially-observed boolean dynamical systems," *Automatica*, vol. 87, pp. 238–250, 2018.

[62] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[63] M. K. Pitt, "Smooth particle filters for likelihood evaluation and maximisation," tech. rep., University of Warwick, Department of Economics, 2002.

[64] Y. Tan, F. B. L. Neto, and U. Braga-Neto, "Pallas: Penalized maximum likelihood and particle swarms for inference of gene regulatory networks from time series data," *bioRxiv*, 2020.

[65] A. Jasra, S. S. Singh, J. S. Martin, and E. McCoy, "Filtering via approximate bayesian computation," *Statistics and Computing*, vol. 22, no. 6, pp. 1223–1237, 2012.

[66] P. Del Moral, A. Doucet, and A. Jasra, "An adaptive sequential monte carlo method for approximate bayesian computation," *Statistics and Computing*, vol. 22, no. 5, pp. 1009–1020, 2012.

[67] T. A. Dean, S. S. Singh, A. Jasra, and G. W. Peters, "Parameter estimation for hidden markov models with intractable likelihoods," *Scandinavian Journal of Statistics*, vol. 41, no. 4, pp. 970–987, 2014.

[68] S. A. Sisson, Y. Fan, and M. M. Tanaka, "Sequential monte carlo without likelihoods," *Proceedings of the National Academy of Sciences*, vol. 104, no. 6, pp. 1760–1765, 2007.

[69] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, 1995.

[70] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[71] C. J. A. Bastos-Filho, F. B. D. Lima-Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima, "A novel search algorithm based on fish school behavior," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 2646–2651, 2008.

[72] Y. Tan, F. B. L. Neto, and U. B. Neto, "Inference of gene regulatory networks by maximum-likelihood adaptive filtering and discrete fish school search," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, IEEE, 2018.

[73] J. A. Sargo, S. M. Vieira, J. M. Sousa, and C. J. Bastos Filho, "Binary fish school search applied to feature selection: Application to icu readmissions," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pp. 1366–1373, IEEE, 2014.

[74] E. R. Dougherty, "Validation of inference procedures for gene regulatory networks," *Current genomics*, vol. 8, no. 6, pp. 351–359, 2007.

[75] E. Batchelor, A. Loewer, and G. Lahav, "The ups and downs of p53: understanding protein dynamics in single cells," *Nature Reviews Cancer*, vol. 9, no. 5, p. 371, 2009.

[76] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, pp. e124–e131, 2006.

[77] N. Berestovsky and L. Nakhleh, "An evaluation of methods for inferring boolean networks from time-series data," *PloS one*, vol. 8, no. 6, p. e66031, 2013.

[78] U. M. Braga-Neto and E. T. Marques Jr, "From functional genomics to functional immunomics: new challenges, old problems, big rewards," *PLoS computational biology*, vol. 2, no. 7, 2006.

[79] M. D. Sutton, B. T. Smith, V. G. Godoy, and G. C. Walker, "The sos response: recent insights into umudc-dependent mutagenesis and dna damage tolerance," *Annual review of genetics*, vol. 34, 2000.

[80] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics," *Proceedings of the national academy of sciences*, vol. 99, no. 16, pp. 10555–10560, 2002.

[81] R. Hengge, "Principles of c-di-gmp signalling in bacteria," *Nature Reviews Microbiology*, vol. 7, no. 4, p. 263, 2009.

[82] F. Mika and R. Hengge, "Small regulatory rnas in the control of motility and biofilm formation in e. coli and salmonella," *International journal of molecular sciences*, vol. 14, no. 3, pp. 4560–4579, 2013.

[83] F. Mika and R. Hengge, "Small rnas in the control of rpos, csgd, and biofilm architecture of escherichia coli," *RNA biology*, vol. 11, no. 5, pp. 494–507, 2014.

[84] J. R. Houser, C. Barnhart, D. R. Boutz, S. M. Carroll, A. Dasgupta, J. K. Michener, B. D. Needham, O. Papoulas, V. Sridhara, D. K. Sydykova, *et al.*, "Controlled measurement and comparative analysis of cellular components in e. coli reveals broad regulatory changes in response to glucose starvation," *PLoS computational biology*, vol. 11, no. 8, p. e1004400, 2015.

[85] M. L. C. Degli Atti, S. Merler, C. Rizzo, M. Ajelli, M. Massari, P. Manfredi, C. Furlanello, G. S. Tomba, and M. Iannelli, "Mitigation measures for pandemic influenza in italy: an individual based model considering different scenarios," *PloS one*, vol. 3, no. 3, p. e1790, 2008.

[86] L. Perez and S. Dragicevic, "An agent-based approach for modeling dynamics of contagious disease spread," *International journal of health geographics*, vol. 8, no. 1, pp. 1–17, 2009.

[87] E. Hunter, B. Mac Namee, and J. Kelleher, "An open-data-driven agent-based model to simulate infectious disease outbreaks," *PloS one*, vol. 13, no. 12, p. e0208775, 2018.

[88] S. L. Chang, N. Harding, C. Zachreson, O. M. Cliff, and M. Prokopenko, "Modelling transmission and control of the covid-19 pandemic in australia," *Nature communications*, vol. 11, no. 1, pp. 1–13, 2020.

[89] D. L. Chao, A. P. Oron, D. Srikrishna, and M. Famulare, "Modeling layered non-pharmaceutical interventions against sars-cov-2 in the united states with corvid," *medRxiv*, 2020.

[90] J. R. Koo, A. R. Cook, M. Park, Y. Sun, H. Sun, J. T. Lim, C. Tam, and B. L. Dickens, "Interventions to mitigate early spread of sars-cov-2 in singapore: a modelling study," *The Lancet Infectious Diseases*, vol. 20, no. 6, pp. 678–688, 2020.

[91] M. Kretzschmar, G. Rozhnova, and M. van Boven, "Isolation and contact tracing can tip the scale to containment of covid-19 in populations with social distancing," *Available at SSRN 3562458*, 2020.

[92] C. C. Kerr, R. M. Stuart, D. Mistry, R. G. Abeysuriya, G. Hart, K. Rosenfeld, P. Selvaraj, R. C. Nunez, B. Hagedorn, L. George, *et al.*, "Covasim: an agent-based model of covid-19 dynamics and interventions," *medRxiv*, 2020.

[93] D. Balcan, B. Goncontcalves, H. Hu, J. J. Ramasco, V. Colizza, and A. Vespignani, "Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model," *Journal of computational science*, vol. 1, no. 3, pp. 132–145, 2010.

[94] D. Osthus, K. S. Hickmann, P. C. Caragea, D. Higdon, and S. Y. Del Valle, "Forecasting seasonal influenza with a state-space sir model," *The annals of applied statistics*, vol. 11, no. 1, p. 202, 2017.

[95] E. Sebastian and P. Victor, "A state space approach for sir epidemic model," *International Journal of Difference Equations*, vol. 12, no. 1, pp. 79–87, 2017.

[96] M. J. Keeling, T. D. Hollingsworth, and J. M. Read, "Efficacy of contact tracing for the containment of the 2019 novel coronavirus (covid-19)," *J Epidemiol Community Health*, vol. 74, no. 10, pp. 861–866, 2020.

[97] R. Sameni, "Mathematical modeling of epidemic diseases; a case study of the covid-19 coronavirus," *arXiv preprint arXiv:2003.11371*, 2020.

[98] A. Godio, F. Pace, and A. Vergnano, "Seir modeling of the italian epidemic of sars-cov-2 using computational swarm intelligence," *International Journal of Environmental Research and Public Health*, vol. 17, no. 10, p. 3535, 2020.

[99] G. Kobayashi, S. Sugasawa, H. Tamae, and T. Ozu, "Predicting intervention effect for covid-19 in japan: state space modeling approach," *BioScience Trends*, 2020.

[100] G. Hooker, S. P. Ellner, L. D. V. Roditi, and D. J. Earn, "Parameterizing state–space models for infectious disease dynamics by generalized profiling: measles in ontario," *Journal of The Royal Society Interface*, vol. 8, no. 60, pp. 961–974, 2011.

[101] T. Rapolu, B. Nutakki, T. S. Rani, and S. D. Bhavani, "A time-dependent seird model for forecasting the covid-19 transmission dynamics," *medRxiv*, 2020.

[102] E. L. Piccolomiini and F. Zama, "Monitoring italian covid-19 spread by an adaptive seird model," *MedRxiv*, 2020.

[103] I. Korolev, "Identification and estimation of the seird epidemic model for covid-19," *Journal of econometrics*, vol. 220, no. 1, pp. 63–85, 2021.

[104] E. Loli Piccolomini and F. Zama, "Monitoring italian covid-19 spread by a forced seird model," *PloS one*, vol. 15, no. 8, p. e0237417, 2020.

[105] V. Tiwari, N. Bisht, and N. Deyal, "Mathematical modelling based study and prediction of covid-19 epidemic dissemination under the impact of lockdown in india," *medRxiv*, 2020.

[106] G. K. Zipf, "The p 1 p 2/d hypothesis: on the intercity movement of persons," *American sociological review*, vol. 11, no. 6, pp. 677–686, 1946.

[107] J. Truscott and N. M. Ferguson, "Evaluating the adequacy of gravity models as a description of human mobility for epidemic modelling," *PLoS Comput Biol*, vol. 8, no. 10, p. e1002699, 2012.

[108] Q. Chen, J. Yan, H. Huang, and X. Zhang, "Correlation of the epidemic spread of covid-19 and urban population migration in the major cities of hubei province, china," *Transportation Safety and Environment*, vol. 3, no. 1, pp. 21–35, 2021.

[109] W. E. Allen, H. Altae-Tran, J. Briggs, X. Jin, G. McGee, A. Shi, R. Raghavan, M. Kamariza, N. Nova, A. Pereta, *et al.*, "Population-scale longitudinal mapping of covid-19 symptoms, behaviour and testing," *Nature Human Behaviour*, vol. 4, no. 9, pp. 972–982, 2020.

[110] D. Buitrago-Garcia, D. Egli-Gany, M. J. Counotte, S. Hossmann, H. Imeri, A. M. Ipekci, G. Salanti, and N. Low, "Occurrence and transmission potential of asymptomatic and presymptomatic sars-cov-2 infections: A living systematic review and meta-analysis," *PLoS medicine*, vol. 17, no. 9, p. e1003346, 2020.

[111] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems [j]," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[112] G. Bishop, G. Welch, *et al.*, "An introduction to the kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.

[113] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pp. 153–158, Ieee, 2000.

[114] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[115] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[116] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE transactions on automatic control*, vol. 45, no. 5, pp. 910–927, 2000.

[117] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on gaussian filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2910–2921, 2006.

[118] J. Kokkala, A. Solin, and S. Särkkä, "Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems," *arXiv preprint arXiv:1504.06173*, 2015.

[119] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.

[120] T. Asai, "Covid-19: accurate interpretation of diagnostic tests—a statistical point of view," 2020.