

Summer 2018

Inference Framework for Model Update and Development

Xiao Lin

Follow this and additional works at: <https://scholarcommons.sc.edu/etd>



Part of the [Computer Engineering Commons](#)

Recommended Citation

Lin, X. (2018). *Inference Framework for Model Update and Development*. (Doctoral dissertation). Retrieved from <https://scholarcommons.sc.edu/etd/4921>

This Open Access Dissertation is brought to you by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact dillarda@mailbox.sc.edu.

INFERENCE FRAMEWORK FOR MODEL UPDATE AND DEVELOPMENT

by

Xiao Lin

Bachelor of Engineering
Jilin University 2009

Master of Science
Beihang University 2012

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in
Computer Science and Engineering
College of Engineering and Computing
University of South Carolina
2018

Accepted by:

Gabriel Terejanu, Major Professor

John Rose, Committee Member

Sourav Banerjee, Committee Member

Marco Valtorta, Committee Member

Jijun Tang, Committee Member

Anindya Chanda, Committee Member

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

© Copyright by Xiao Lin, 2018
All Rights Reserved.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Terejanu Gabriel, for his constant guidance and support. Gabriel has been a great mentor, friend and inspiration to me throughout my PhD study.

I would also thank Sourav Banerjee and Anindya Chanda. It is a great pleasure for me to work with you on exciting projects. I would like to thank all my committee members. Thanks for all the valuable suggestions and guidance in my research as well as in my career.

Last but not least, I would thank my family in China. I am so lucky to grow up in a happy family with constant support from all the family members.

ABSTRACT

Computational models play an important role in scientific discovery and engineering design. However, developing computational models is challenging, since the process always follows a path contaminated with errors and uncertainties. The uncertainties and errors inherent in computational models are the result of many factors, including experimental uncertainties, model structure inadequacies, uncertainties in model parameters and initial conditions, as well as errors due to numerical discretizations. To realize the full potential in applications it is critical to systematically and economically reduce the uncertainties inherent in all computational models.

The update and development of computational models is a recursive process between data assimilation and data selection. In data assimilation, measurements are incorporated into computational simulations to reduce the uncertainties of the model and in reverse, the simulations help determine where to acquire data such that most information can be provided.

Currently, data assimilation techniques are overwhelmed by data volume and velocity and increased complexity of computational models. In this work, we develop a novel data assimilation approach EnLLVM which is based on linear latent variable model. There are several advantages of this approach. First, it works well with high dimensional dynamic systems, but only requires a small number of samples. Second, it can absorb model structure error and reflect the error in the uncertainty of data assimilation results. In addition, data assimilation is performed without calculating likelihood of observation, thus it can be applied to data assimilation problems in which likelihood is intractable.

Obtaining informative data is also crucial, as data collection is an expensive endeavor for a number of science and engineering fields. Mutual information, which naturally measures information provided about one quantity by knowing the other quantity, has become a major design metric and has fueled a large body of work on experimental design. However, estimating mutual information is challenging and results are not reliable in high dimensions. In this work, we derive a lower bound of mutual information, which is computed in much lower dimensions. This lower bound can be applied to experimental design as well as other problems that require comparison of mutual information.

At last, we develop a general framework for building computational models. In this framework, hypotheses about unknown model structure are generated by using EnLLVM for data assimilation and lower bound of mutual information for finding relations between state variables and unknown structure function. Then, different hypotheses can be ranked with model selection technique. This framework not only provides a way to infer model discrepancy, but also could further contribute to scientific discoveries.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iii
ABSTRACT	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1 INTRODUCTION	1
1.1 Data assimilation	2
1.2 Data selection	4
CHAPTER 2 ENLLVM: FAST BAYESIAN INFERENCE FOR HIGH DIMEN- SIONAL PROBLEMS	6
2.1 Introduction	6
2.2 Methodology	9
2.3 Numerical experiments	26
2.4 Summary	30
CHAPTER 3 APPROXIMATE COMPUTATIONAL APPROACHES FOR BAYESIAN SENSOR PLACEMENT IN HIGH DIMENSIONS	34
3.1 Introduction	34
3.2 Bayesian Inference	37

3.3	Sensor Placement	41
3.4	Experiments	51
3.5	Conclusions	65
CHAPTER 4 BAYESIAN MODEL SELECTION FRAMEWORK FOR IDENTIFYING GROWTH PATTERNS IN FILAMENTOUS FUNGI		67
4.1	Introduction	67
4.2	Methods and Models	71
4.3	Numerical Results and Discussions	82
4.4	Conclusions	92
CHAPTER 5 HYPOTHESIS GENERATION FOR MODEL DISCREPANCY		93
5.1	Introduction	93
5.2	Methodology	95
5.3	Numerical experiments	96
5.4	Conclusion	100
BIBLIOGRAPHY		102

LIST OF TABLES

Table 2.1	EnKF vs EnPPCA: RMSE statistics of 100 trials	30
Table 3.1	Average and standard error over 100 trials for Eq. 3.34	54
Table 3.2	Average and standard error over 100 trials for Eq. 3.35	55
Table 3.3	Data set description	56
Table 3.4	Average and standard error over 40 times for real data sets	56
Table 4.1	Bayesian model comparison results for Dataset I ($\gamma_1 = 0.03$)	90
Table 4.2	Bayes factors for Dataset I ($\gamma_1 = 0.03$)	90
Table 4.3	Bayesian model comparison results or Dataset II ($\gamma_1 = 0.3$)	91
Table 4.4	Bayes factors for Dataset II ($\gamma_1 = 0.3$)	91
Table 5.1	Results for 100 trials	99
Table 5.2	Results for 100 trials	100

LIST OF FIGURES

Figure 1.1	Computational Framework	2
Figure 2.1	Fig.(a) shows Monte Carlo samples of $p(\mathbf{q}_{k+1 k}, \mathbf{d}_{k+1 k})$. The goal is to calculate conditional distribution given $\tilde{\mathbf{d}}_{k+1}$. Fig.(b) shows Gaussian approximation of $p(\mathbf{q}_{k+1 k}, \mathbf{d}_{k+1 k})$ and consequential conditional distribution.	12
Figure 2.2	Gaussian mixture approximation given by EnLLVM.	15
Figure 2.3	EnLLVM captures higher order statistics.	18
Figure 2.4	Workflow of EnLLVM	20
Figure 2.5	Posterior distribution of states in Lorenz 63 system after each update.	28
Figure 2.6	Results of data assimilation by EnFK: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete observation model.	31
Figure 2.7	Results of data assimilation by PPCA: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete observation model.	31
Figure 2.8	Results of data assimilation by EnKF: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete linear observation model with model error M4.	32
Figure 2.9	Results of data assimilation by EnPPCA: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete linear observation model with model error M4.	32

Figure 3.1	The dynamics of one puff. This figure shows the concentration surface of one puff at different time. The release occurs at $t = 0$ and the release location is $(0, -1163.5 \text{ km})$. The wind direction is 0.17 m/s and the wind speed is 4 m/s . p_y and q_y are 0.466 and 0.866 respectively. Note the color bar is different for each plot.	58
Figure 3.2	The grid and release location. The domain is grided by 11×21 . The pipeline is on the y axis, ranging from -3 km to 3 km .	59
Figure 3.3	Mutual information surface and sensor locations. Three sensor locations are selected in a greedy way. The heat map show mutual information $I(d_1, t; q)$, $I(d_1, d_2, t; q)$ and $I(d_1, d_2, d_3, t; q)$ at each grid point. Blue diamonds represent sensor locations selected over grid points, while green ones are selected by Bayesian optimization. Note the color bar for each plot is different.	61
Figure 3.4	Distributions of parameters. The histograms in Figure (a) are prior distribution and posterior distribution of release location at each time step. Figure (b) shows corresponding distribution of wind direction. The real release location is $(0, -1291.7 \text{ km})$ and the real wind direction is -0.026 rad . Both of them are represented by red vertical lines.	63
Figure 3.5	Results comparison. These two figures compare the performance of proposed strategy with 20 random sensor placements. Here, entropy is used to measure the uncertainty. Figure (a) shows entropy of release location after each update and Figure(b) shows joint entropy of release location and wind direction.	64
Figure 4.1	Branching types and corresponding mathematical model	81
Figure 4.2	Observation data. Ten observation points between 0 and 22.5 are evenly selected along x axis. Observation data at $t = 0.5$ is used to infer the initial condition for candidate models and data at $t = 5$ is used for model calibration.	83
Figure 4.3	Inferred initial condition. Initial condition for candidate models is estimated using observation data at $t = 0.5$. MSE is 0.0446 for hyphal density and 0.0018 for tip density.	85
Figure 4.4	Marginal posterior distribution of parameters for candidate models YH, YHD, YFH for Dataset I (a),(b),(c), and Dataset II (d),(e),(f).	87

Figure 4.5 Model predictions at $t = 5$ using calibrated models with Dataset
Scenario I (a) and Dataset II (b). 89

CHAPTER 1

INTRODUCTION

Computational science and engineering has now become the third pillar of scientific inquiry and engineering design along with theory and experimentation [18]. Computational simulations provide insights into the physical and chemical processes [77, 36] and are also used for solving engineering problems, such as condition-based maintenance of aircraft systems [80]. They are also instrumental in informing policy decisions in areas where the consequences of inaccurate predictions and poorly informed decisions could be catastrophic, such as disaster response due to contaminant release [71] and climate change [48]. These computational models are more complex than ever and increasingly operate at higher resolutions with more degrees of freedom.

The central challenge in developing computational models for scientific discovery, engineering design, or decision support is that the process follows a path contaminated with errors and uncertainties. The uncertainties and errors inherent in computational models are the result of many factors, including experimental uncertainties, model structure inadequacies, uncertainties in model parameters and initial conditions, as well as errors due to numerical discretizations. To realize the full potential in such applications it is critical to systematically and economically reduce the uncertainties inherent in all computational models.

A typical framework for model update and development integrates measurements with application simulations using a synergistic feedback and control-loop between the two, see Fig.1.1. Measurements are dynamically incorporated into computational simulations to reduce the uncertainties of the model and in reverse, the simulations

dynamically steer the measurement process [23]. This recursive process will lead to a well-developed computational model that can help accelerate scientific discoveries and decision-making under uncertainty.

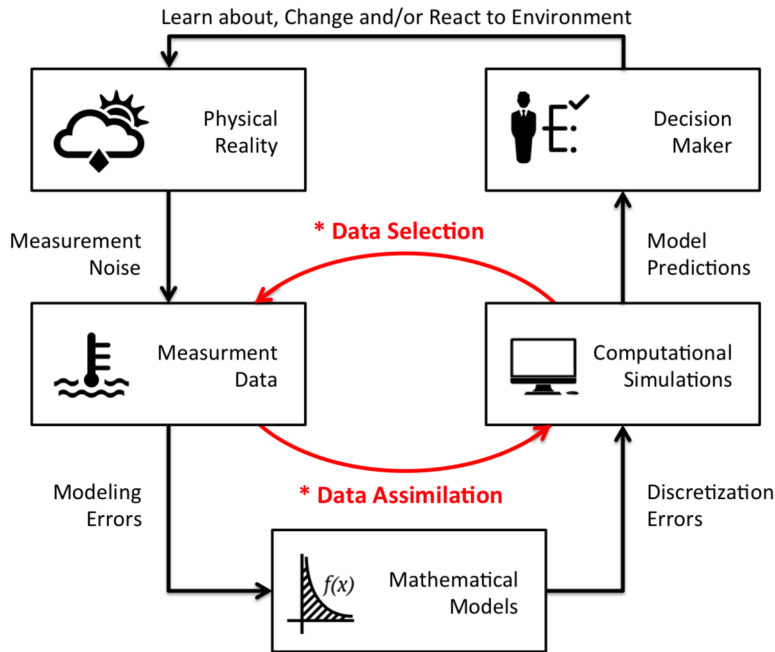


Figure 1.1: Computational Framework

1.1 DATA ASSIMILATION

Data assimilation is concerned with inferring certain quantity of interest (QoI). This QoI includes state variables of a dynamic system as well as unknown parameters in the computational model. The inference involves running computer simulations and solving inverse and forward problems. Generally, there are two categories of approaches, optimization methods and Bayesian inference. The optimization methods provide a single point estimate of unknown parameters by minimizing the discrepancy between model predictions and observational data. While Bayesian inference formulate the problem in a Bayesian framework where quantities have probabilistic description and posterior distribution of QoI can be obtained via Bayes rule. The ad-

vantage of Bayesian inference is the incorporation of prior information and the access to the full posterior distribution from which estimates with quantified uncertainties can be extracted. In this work, we employ probability to represent uncertainty and Bayesian inference to update the uncertainty of QoI in light of measurement data.

If the dynamic system is linear and the uncertainty involved can be described with Gaussian distribution, there is a closed-form solution for posterior distribution. It can be obtained by Kalman filter [47]. However, in most cases, the precondition for applying Kalman filter cannot be satisfied. Different assumptions and approximations have been made for quasi-optimal solutions maintaining both accuracy and tractability. Among the best understood and most frequently cited nonlinear filters are the Extended Kalman Filter (EKF), the Ensemble Kalman Filter (EnKF), the Unscented Kalman Filter (UKF), and the Particle Filter (PF). The EKF is based on the assumption that over small time increments, nonlinear system dynamics can be accurately modeled by a first-order Taylor series expansion [22]. However, when the process model and observation model are highly non-linear, EKF can give particularly poor performance [45]. The UKF, which is a derivative-free alternative to EKF, bypasses differentiation by using a deterministic sampling approach [45, 84]. The EnKF is a reduced rank sampling filter which propagates the states through the system nonlinearities and updates a relatively small ensemble of samples from which an assumed Gaussian distribution which captures the main characteristics in the uncertainty is estimated [33]. Due its simplicity in both theory and implementation, EnKF has been widely used, especially in meteorology where high dimensional data assimilation is performed. Unlike EnKF, PF achieves a complete Bayesian inference. It uses a sampling approach, estimating the posterior probability distribution, including its higher order moments, by propagating and updating a number of particles, without the assumption of Gaussian statistics [70].

Another commonly used approach is Markov Chain Monte Carlo (MCMC). MCMC

is also a sample-based method and has become an important computational workhorse in scientific computing for sampling from a large class of distributions. The most basic form of MCMC is Metropolis-Hastings (MH) algorithm [62, 4], which generates a sequence of correlated samples that form a Markov chain. A good estimate usually requires a large number of samples and takes much longer time than other approaches.

For high dimensional data assimilation, sampling-based methods usually suffer from *curse of dimensionality*. In this work, a novel filtering method is proposed to deal with this problem. The details of this method will be given in Chapter. 2. And in Chapter. 5, the proposed method will be applied to develop predictive models.

1.2 DATA SELECTION

Measurement data play a crucial role in developing models of physical systems. But not all measurements contain equal amount of information. Some data are more valuable, while others may help little in developing models. To select data strategically is of significance as data can be difficult and expensive to acquire. For example, experimental data collection in biological systems is usually a time consuming and costly operation due to its high reliance on human resources and specialized equipment. As a result, strategies based on design of experiments are needed to judiciously perform data collections that meet various constraints of biological experiments [56].

Data selection is usually decided by running simulations and maximizing certain target function. There are various choices towards the target function. It can be Mean Square Error (MSE), mutual information, entropy or measurements about inverse moment matrix in A-, D-, E-optimal design. Among all these criteria, mutual information is most common used, since it is a natural measure of dependence between the two quantities and others can be derived from it either directly or indirectly.

Estimating mutual information from samples is challenging. Some commonly used estimators include histogram based estimator, kernel density estimator, and k -nearest

neighbor estimator (kNN). In their survey, Walters-Williams and Li [83] show that parametric estimation usually outperform non-parametric estimation when data is drawn from a known family of distributions. However, this is not the case in most practical problems. Khan et al. [51] compare different estimators that quantify the dependence between random variables, and show that the kNN estimator of mutual information captures better the nonlinear dependence than other commonly used estimators.

Although kNN estimator provide good estimate in low dimensions, it works poorly in high dimensions due to the scarcity of samples. This issue is not directly addressed in the past literature. In Chapter. 3, we propose an approximate approach to select data. Instead of maximizing mutual information directly, we derive a lower bound of mutual information, which can be computed in a much lower dimension. This lower bound of mutual information is also applied in Chapter. 5 to measure relation of functions.

CHAPTER 2

ENLLVM: FAST BAYESIAN INFERENCE FOR HIGH DIMENSIONAL PROBLEMS

2.1 INTRODUCTION

Bayesian filtering is concerned with inferring certain quantity of interest (QoI). This QoI includes state variables of a dynamic system as well as unknown parameters in the computational models. The inference involves running computer simulations and solving forward and inverse problems. For a linear system, under the assumption of Gaussian probability distributions, the problem of estimating the states of the system has an exact closed-form solution given by the Kalman filter [47]. If the probability distributions are non-Gaussian or the system is nonlinear, in general no closed-form solutions are available. Different assumptions and approximations have been made for quasi-optimal solutions maintaining both accuracy and tractability. Among the best understood and most frequently cited nonlinear filters are the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the ensemble Kalman filter (EnKF), and the particle filter (PF).

The EKF is historically the first, and still the most widely adopted approach to nonlinear state estimation problems. It is based on the assumption that over small time increments, nonlinear system dynamics can be accurately modeled by a first-order Taylor series expansion [22]. However, when the process model and observation model are highly non-linear, EKF may give extremely poor performance [45]. The UKF, which is a derivative-free alternative to EKF, bypasses differentiation by us-

ing unscented transformation (UT) [45, 84, 74]. In UT, a relatively small number of representative points are selected to approximate the distribution of a Gaussian variable after a nonlinear transformation and the accuracy of the UT approximation decreases as the dimensionality increases [74].

The EnKF is a reduced rank sampling filter which propagates the states through the system nonlinearities and updates a relatively small ensemble of samples from which an assumed Gaussian distribution which captures the main characteristics in the uncertainty is estimated [33]. Due to its simplicity in both theory and implementation, EnKF has been widely used, especially in meteorology [40, 41] where high dimensional data assimilation is performed.

The Kalman filter variations discussed above all approximate posterior densities as Gaussians. For nonlinear, non-Gaussian systems, particle filter has significant advantages. The particle filter uses a Monte Carlo sampling approach, estimating the posterior probability distribution, including its higher order moments, by propagating and updating a number of particles, without the assumption of Gaussian statistics [70]. Compared with Kalman filter variations, particle filter works with any arbitrary prior distributions and is capable to capture multimodality of the posterior distribution. However, particle filter has mostly been applied to low dimensional problems. Snyder et al. [73] points out that particle filter has the tendency to collapse and for a good estimation, it is required that the number of samples scales exponentially with the problem size.

To cope with high dimensional problems, variations of particle filter have been developed. One class is multiple particle filter [26, 14, 1]. The core idea of multiple particle filter is using standard particle filters to update state variables in subspaces. In this way high dimensional filtering is converted to a bunch of lower dimensional problems which is easy to deal with and requires a smaller number of particles. However, since most of the time different states are coupled in process model, updating

in each particle filter requires propagating particles from other states. This approach completely ignores joint distribution even though coupled states are dependent. Also problems will arise when states are coupled in the observation model or only partial states are observable. Nakano et al. [63] proposed a merging particle filter (MPF) to avoid the collapse problem in standard particle filter. The performance of MPF, PF and EnKF are also compared by applying them to a 40-dimensional nonlinear dynamic system. The results show that as the number of samples increases, MPF converges faster than PF and after a certain number of samples they will both outperform EnKF. However, for a relatively small number of samples, for example $N < 256$ which is tested in their experiments, EnKF works much better than MPF and PF.

Another class of filters that aim at capturing multimodality is Gaussian sum filter. In Gaussian sum filter, arbitrary prior distribution is approximated with Gaussian mixture. Then each component of the mixture is updated through Kalman filter or its variations [74, 78], and the consequential posterior distribution is also represented by Gaussian mixture. A different way to combine Gaussian mixture models with EnKF is proposed in [27], where each ensemble member is updated through Gaussian mixture models and the updated ensemble is proved to be capable of capturing multimodality. The drawback of Gaussian mixture models is obvious. First, the mixture models are estimated from samples, which usually requires a large number of samples. Moreover, the estimation includes specifying various parameters such as weight and statistics of each Gaussian component, which is nontrivial and can be computationally expensive.

Stordal et al. [75] bridged EnKF and PF through an adaptive Gaussian sum filter. Basically, EnKF is equivalent to the Gaussian sum filter which takes sample covariance as covariance for each Gaussian kernel and has uniform weights. When model error is added, particle filter can also be viewed as the Gaussian sum filter whose Gaussian component has zero covariance. Stordal et al. [75] also pointed out that increasing the variance of model noise in particle filter can force the weights to

be more uniform so as to avoid filter collapse.

Often, general formulations that exploit the structure of physics-based models by using internal discrepancies to capture structural errors yield intractable likelihood functions. In fact, in many cases, the distribution of observation noise is unknown, and the observation model is viewed as a black box which outputs observation data given the input states. These problems make particle filter, EnKF unsuitable as they rely on computation of likelihood functions.

In this paper, we propose a novel Bayesian filtering approach which only requires a small number of samples even in high dimensional systems. The proposed method is based on linear latent variable models. Samples are first mapped into latent space which has a much lower dimensionality. Then update is performed in the latent space and posterior samples are mapped back to the original space. The proposed approach doesn't require Gaussian distribution assumption and is able to capture multimodality. Moreover, the method is performed without evaluating likelihood of observations, thus it can be applied to data assimilation problems in which likelihood is intractable. Other filtering methods mentioned above require direct computation of likelihood functions. In this paper, the proposed method is assessed on two nonlinear dynamic systems and also compared with EnKF on various scenarios.

The rest of paper is organized as follows. In section 2, the proposed method is detailed. The method is not restricted to a particular type of linear latent models and several options are provided. In section 3, two numerical experiments are performed to assess its predictability, which includes comparisons with EnKF on several different scenarios.

2.2 METHODOLOGY

Consider the following general parameterized nonlinear dynamical system perturbed by process noise \mathbf{w}_k , measurement noise \mathbf{v}_k and uncertain initial conditions. Given

a set of measurements, the goal is to infer the state of the system and parameters to improve model predictions and as a result decision making under uncertainty.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\theta}, \mathbf{w}_k) \quad (2.1)$$

$$\mathbf{d}_k = \mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}, \mathbf{v}_k) \quad (2.2)$$

$$\mathbf{x}_0 \sim p(\mathbf{x}_0) \quad (2.3)$$

Given a set of observation $\mathbf{D}_k = \{\tilde{\mathbf{d}}_i | 1 \leq i \leq k\}$, Bayesian filtering is the problem of finding the joint probability density function (pdf) of the states \mathbf{x}_k and parameters $\boldsymbol{\theta}$ conditioned on all the observations up to and including current time t_k , $p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{D}_k)$.

The posterior pdf, $p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{D}_k)$, may be obtained recursively using Bayes' rule:

$$p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{D}_k) = \frac{p(\tilde{\mathbf{d}}_k | \mathbf{x}_k, \boldsymbol{\theta}, \mathbf{D}_{k-1})p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{D}_{k-1})}{p(\tilde{\mathbf{d}}_k | \mathbf{D}_{k-1})} \quad (2.4)$$

In Eq.(2.4), $p(\tilde{\mathbf{d}}_k | \mathbf{x}_k, \boldsymbol{\theta}, \mathbf{D}_{k-1})$ is the likelihood function, $p(\tilde{\mathbf{d}}_k | \mathbf{D}_{k-1})$ is the normalization constant known as the evidence and the prior pdf $p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{D}_{k-1})$ is obtained as follows:

$$p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{D}_{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \boldsymbol{\theta})p(\mathbf{x}_{k-1}, \boldsymbol{\theta} | \mathbf{D}_{k-1})d\mathbf{x}_{k-1} \quad (2.5)$$

where $p(\mathbf{x}_{k-1}, \boldsymbol{\theta} | \mathbf{D}_{k-1})$ is posterior pdf at time t_{k-1} and $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \boldsymbol{\theta})$ can be obtained through Eq. (2.1).

If $f(\bullet)$ and $h(\bullet)$ are both linear models and all the probability distribution involved are Gaussian, then Bayesian filtering can be easily solved with Kalman filter. Otherwise, methods based on Monte Carlo samples are commonly adopted. However, current data assimilation methods suffer from the *curse of dimensionality* and struggle to deal with intractable likelihood functions. In general nonlinear dynamical systems, the measurement noise might not be explicitly presented in Eq. (2.2), it can be embedded into the observation model with unknown distribution, which makes it impossible to create likelihood function for EnKF and particle filter.

In this paper, we tackle Bayesian filtering from a different angle. Instead of relying on Eq. (2.2) to calculate likelihood, we operate on the joint distribution of \mathbf{x}_k , $\boldsymbol{\theta}$ and \mathbf{d}_k directly. For simplicity, in the followings, all the quantities of interest (QoIs) at any time k will be denoted by \mathbf{q}_k .

$$\mathbf{q}_k = [\mathbf{x}_k, \boldsymbol{\theta}]^T \quad (2.6)$$

Furthermore, let $\mathbf{q}_{k+1|k}$ denote \mathbf{q} which is obtained through Eq. (2.1) after observing \mathbf{D}_k but before $\tilde{\mathbf{d}}_{k+1}$. Thus $p(\mathbf{x}_{k+1}, \boldsymbol{\theta} \mid \mathbf{D}_k)$ can be written as $p(\mathbf{q}_{k+1|k})$. Similarly, let $\mathbf{d}_{k+1|k}$ denote \mathbf{d} which is obtained by propagating $\mathbf{q}_{k+1|k}$ through Eq. (2.2).

Assume there is a joint distribution $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$, then posterior pdf of $\mathbf{q}_{k+1|k}$ given $\tilde{\mathbf{d}}_{k+1}$ is the conditional distribution $p(\mathbf{q}_{k+1|k} \mid \tilde{\mathbf{d}}_{k+1})$ which is denoted by $\mathbf{q}_{k+1|k+1}$. Since there are no analytical forms for these distributions, they have to be estimated with Monte Carlo samples. This means that at any time k the uncertainty in the QoIs given the observation set \mathbf{D}_k is described by a set of N samples.

$$\mathbf{q}_{k|k}^i \sim p(\mathbf{q}_{k|k}), \quad \text{for } i = 1 \dots N \quad (2.7)$$

These samples are propagated through dynamics in Eq. (2.1) to obtain predictive samples at time $k + 1$, $\{\mathbf{q}_{k+1|k}^i\}_{i=1\dots N}$. Then $\{\mathbf{q}_{k+1|k}^i\}_{i=1\dots N}$ are propagated through Eq. (2.2) to obtain $\{\mathbf{d}_{k+1|k}^i\}_{i=1\dots N}$. Given the measurement $\tilde{\mathbf{d}}_{k+1}$, the goal is to obtain N samples, $\{\mathbf{q}_{k+1|k+1}^i\}_{i=1\dots N}$, from the posterior pdf of the QoIs, $p(\mathbf{q}_{k+1|k+1})$. The whole flow is shown in Fig. ???. The input of our method are $\{\mathbf{q}_{k+1|k}^i\}_{i=1\dots N}$, $\{\mathbf{d}_{k+1|k}^i\}_{i=1\dots N}$ and $\tilde{\mathbf{d}}_{k+1}$, and the expected output is $\{\mathbf{q}_{k+1|k+1}^i\}_{i=1\dots N}$.

If $[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}]$ follows multivariate Gaussian distribution, $p(\mathbf{q}_{k+1} \mid \tilde{\mathbf{d}}_{k+1})$ can be easily obtained [28]. If it is not Gaussian, we can still approximate it with a Gaussian distribution, and obtain conditional distribution. But in this case, the conditional distribution is likely to be inaccurate. As is shown in Fig. 2.1, a single Gaussian distribution is unable to capture multimodal property of samples and may lead to unreliable estimation of posterior $p(\mathbf{q}_{k+1} \mid \tilde{\mathbf{d}}_{k+1})$.

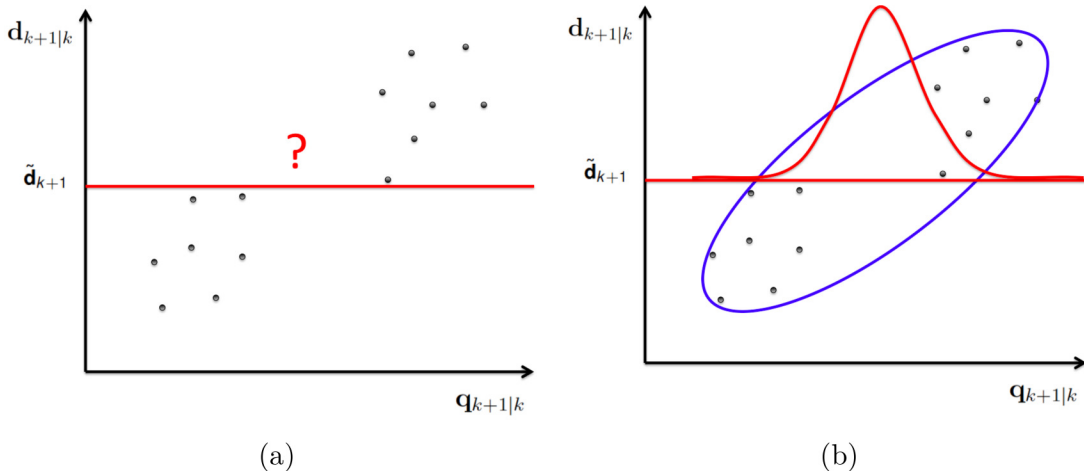


Figure 2.1: Fig.(a) shows Monte Carlo samples of $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$. The goal is to calculate conditional distribution given $\tilde{\mathbf{d}}_{k+1}$. Fig.(b) shows Gaussian approximation of $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$ and consequential conditional distribution.

We can also estimate the probability density $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$ from Monte Carlo samples. However, if only a small number of samples are available and $[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}]$ is in high dimensional space, then probability density estimation (PDE) will suffer badly from the *curse of dimensionality*. In this paper, we construct $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$ based on linear latent variable models and perform Bayesian inference in latent space.

2.2.1 LINEAR LATENT VARIABLE MODEL

The linear latent variable model (LLVM) is shown in Eq. (2.8). \mathbf{W} is a coefficient matrix, \mathbf{z} is the latent variable, $\boldsymbol{\mu}$ is bias and $\boldsymbol{\eta}$ is the noise. Assume $\mathbf{q}_{k+1|k}$ and $\mathbf{d}_{k+1|k}$ have a dimensionality of H_q and H_d respectively. Also denote the dimensionality of the joint space as H , then $H = H_q + H_d$. Let M be the dimensionality of latent variable \mathbf{z} . M is supposed to be much smaller than H . The grounds for latent model is that data points usually lie close to a manifold of much lower dimensionality than that of the original data space [5].

$$[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}]^T = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\eta} \quad (2.8)$$

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_{M \times M})$$

$$\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Psi})$$

Parameters \mathbf{W} , $\boldsymbol{\mu}$ and $\boldsymbol{\Psi}$ can be obtained with samples $\{\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i\}_{i=1 \dots N}$ via maximum likelihood estimation.

$$\mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Psi} = \arg \max \hat{p}(\{\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i\}_{i=1 \dots N} | \mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Psi})$$

We may restrict the structure of the parameters, such that the latent space will preserve certain property from the original data set. This will lead to different linear latent variable models, which will be discussed later in the paper.

From Eq. (2.8), a density estimator for $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$ can be easily obtained as is shown in Eq. (2.9).

$$\hat{p}(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (2.9)$$

$$\boldsymbol{\Sigma} = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi}$$

However, the pdf $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$ in Eq. (2.9) is still a single Gaussian distribution which might not be sufficient to describe original samples.

2.2.2 ENSEMBLE LINEAR LATENT VARIABLE MODEL

Here, we introduce an ensemble linear latent variable model (EnLLVM) to replace the single Gaussian approximation. EnLLVM will lead to a Gaussian mixture which is supposed to give a better description of original samples. EnLLVM is built in the following way. For each sample $\{\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i\}$ of $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$, we project it into the latent space, and there is a corresponding latent variable \mathbf{z}_i which follows Gaussian distribution:

$$\mathbf{z}_i \sim \mathcal{N}(\mathbb{E}[\mathbf{z} | \mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i], \text{Cov}[\mathbf{z} | \mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i])$$

This process is illustrated in Fig. 2.2. The black dots denote original samples and are projected into the latent space. For each sample, a corresponding latent variable is updated, which is denoted by a normal distribution in red. Back to original space, we will get a Gaussian representation for each sample. This is denoted by a blue circle. Then $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$ can be reconstructed by combining all the Gaussian representations. This will give a Gaussian mixture shown in Eq. (2.10). Since the model is linear Gaussian, $p(\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i)$ can be obtained according to the property of conditional Gaussian distribution [5].

$$\hat{p}_{en}(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}) = \frac{1}{N} \sum_{i=1}^N \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (2.10)$$

$$\boldsymbol{\mu}_i = \mathbf{W}\mathbf{E}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] + \boldsymbol{\mu} \quad (2.11)$$

$$\boldsymbol{\Sigma}_i = \mathbf{W}\text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]\mathbf{W}^T + \boldsymbol{\Psi} \quad (2.12)$$

where the mean and covariance of the latent variable are given by

$$\mathbf{E}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] = \text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]\mathbf{W}^T([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^T - \boldsymbol{\mu}) \quad (2.13)$$

$$\text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] = (\mathbf{I}_{M \times M} + \mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W})^{-1}. \quad (2.14)$$

We can also obtain $\mathbf{E}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]$ and $\text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]$ according to the prediction step of Kalman filter, which will give the following result

$$\mathbf{E}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] = \mathbf{W}^T\boldsymbol{\Sigma}^{-1}([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^T - \boldsymbol{\mu}) \quad (2.15)$$

$$\text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] = \mathbf{I}_{M \times M} - \mathbf{W}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}. \quad (2.16)$$

Here, $\boldsymbol{\Sigma}$ is the covariance of prior distribution $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$, which is given in Eq. (2.10). These two presentations are equivalent. However, the computational complexity can be different. As we can see, both representations involve inversion of a H by H matrix which will cost $O(H^3)$. The H by H matrix is $\boldsymbol{\Psi}$ in the first representation, and $\boldsymbol{\Sigma} = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi}$ in the second one. If $\boldsymbol{\Psi}$ is a diagonal matrix, this inversion will only cost $O(H)$. Although a M by M matrix also needs to be inverted,

M is supposed to be much smaller than H , thus $O(M^3)$ compared with $O(H^3)$ is a big computational saving. In fact, by restricting the structure of Ψ , we can have different latent linear models. This will be detailed in Sec. 2.2.4

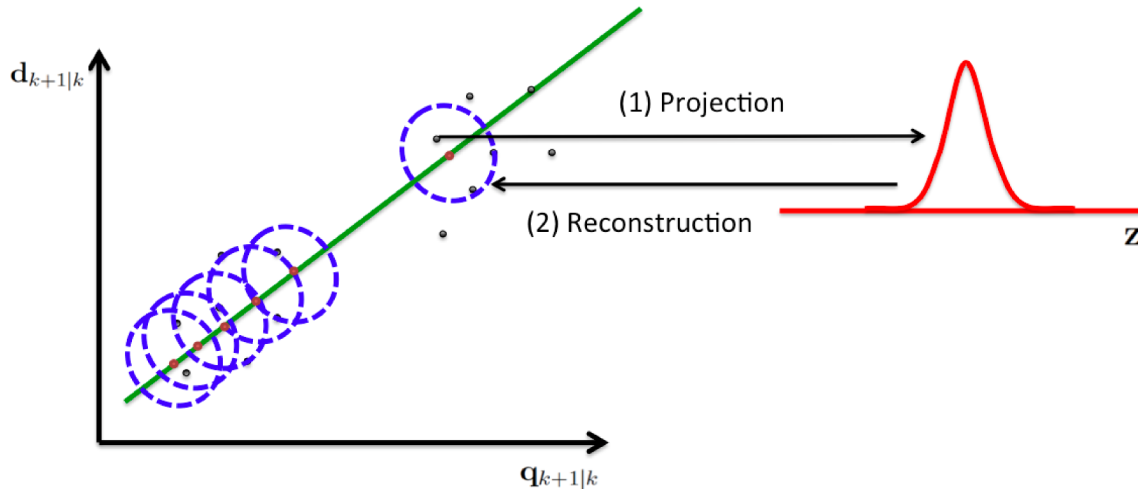


Figure 2.2: Gaussian mixture approximation given by EnLLVM.

2.2.3 PROPERTY OF ENLLVM

Although LLVM is naturally obtained from the latent model, it has a single Gaussian distribution which is unable to capture multimodality. To solve this problem, EnLLVM is developed. In this section, we will investigate the property of EnLLVM and compare it with LLVM.

Before making any statement, let's first examine what μ is in the latent model. All three parameters \mathbf{W} , μ , and Ψ can be obtained by maximizing the likelihood function shown in Eq. (2.9). The corresponding log likelihood function is given in Eq. (2.17).

$$\ln \hat{p}(\{\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i\}_{i=1\dots N} \mid \mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (2.17)$$

$$= \sum_{i=1}^N \ln p(\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i \mid \mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (2.18)$$

$$= -\frac{NM}{2} \ln(2\pi) - \frac{N}{2} \ln|\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{i=1}^N ([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} ([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] - \boldsymbol{\mu}) \quad (2.19)$$

Take the derivative of Eq. (2.19) with respect to $\boldsymbol{\mu}$, we have

$$\frac{d \ln \hat{p}(\{\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i\}_{i=1\dots N} \mid \mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Psi})}{d\boldsymbol{\mu}} = \frac{1}{2} \sum_{i=1}^N \boldsymbol{\Sigma}^{-1} ([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] - \boldsymbol{\mu}) \quad (2.20)$$

Set Eq. (2.20) to 0, we can obtain $\boldsymbol{\mu}$ which is the mean of $\{\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i\}_{i=1\dots N}$.

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N [\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i] \quad (2.21)$$

This means no matter what latent space is, $\boldsymbol{\mu}$ is always the the mean of original samples and thus different latent space is only determined by \mathbf{W} and $\boldsymbol{\Psi}$. With this fact, we can get the lemma below.

Lemma 1. EnLLVM provides the same estimate for the mean and covariance of the samples as LLVM, while it captures higher order statistics as compared with LLVM.

As shown in Eq. (2.9), the mean and covariance of LLVM is $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively. So we need to prove the following equations are true.

$$\text{E}_{en}[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}] = \boldsymbol{\mu} \quad (2.22)$$

$$\text{Cov}_{en}[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}] = \boldsymbol{\Sigma} \quad (2.23)$$

Proof. The mean and covariance of the Gaussian mixture in Eq. 2.10 are calculated

using the following relations:

$$\mathbb{E}_{en}[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}] = \frac{1}{N} \sum_{i=1}^N \boldsymbol{\mu}_i \quad (2.24)$$

$$= \frac{1}{N} \sum_{i=1}^N \mathbf{W}\mathbf{W}^T \boldsymbol{\Sigma}^{-1} ([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^T - \boldsymbol{\mu}) + \boldsymbol{\mu} \quad (2.25)$$

$$= \mathbf{W}\mathbf{W}^T \boldsymbol{\Sigma}^{-1} \underbrace{\left(\frac{1}{N} \sum_{i=1}^N [\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^T - \boldsymbol{\mu} \right)}_{\boldsymbol{\mu}} + \boldsymbol{\mu} \quad (2.26)$$

$$= \boldsymbol{\mu} \quad (2.27)$$

Eq. (2.26) uses the fact that $\boldsymbol{\mu}$ equals the mean of original samples. The covariance of EnLLVM can be decomposed into two parts as is shown below.

$$\begin{aligned} \text{Cov}_{en}[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}] &= \frac{1}{N} \sum_{i=1}^N \left[\boldsymbol{\Sigma}_i + \left(\boldsymbol{\mu}_i - \boldsymbol{\mu} \right) \left(\boldsymbol{\mu}_i - \boldsymbol{\mu} \right)^T \right] \\ &= \underbrace{\frac{1}{N} \sum_{i=1}^N \boldsymbol{\Sigma}_i}_{T_1} + \underbrace{\frac{1}{N} \sum_{i=1}^N \left(\boldsymbol{\mu}_i - \boldsymbol{\mu} \right) \left(\boldsymbol{\mu}_i - \boldsymbol{\mu} \right)^T}_{T_2} \end{aligned} \quad (2.28)$$

where

$$\begin{aligned} T_1 &= \mathbf{W} \left(\mathbf{I}_{M \times M} - \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \right) \mathbf{W}^T + \boldsymbol{\Psi} \\ &= \boldsymbol{\Sigma} - \mathbf{W}\mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{W}^T \end{aligned} \quad (2.29)$$

and

$$\begin{aligned} T_2 &= \mathbf{W}\mathbf{W}^T \boldsymbol{\Sigma}^{-1} \underbrace{\left[\frac{1}{N} \sum_{i=1}^N \left([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^T - \boldsymbol{\mu} \right) \left([\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^T - \boldsymbol{\mu} \right)^T \right]}_{\approx \boldsymbol{\Sigma}} \\ &\quad \times \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{W}^T \\ &= \mathbf{W}\mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{W}^T \end{aligned} \quad (2.30)$$

Substituting Eq. 2.29 and Eq. 2.30 back into Eq. 2.28 we obtain that the covariance of the EnLLVM approximation coincides with LLVM covariance.

$$\text{Cov}_{en}[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}] = \boldsymbol{\Sigma} . \quad (2.31)$$

So far, we have shown that EnLLVM provides the same estimate for the mean and covariance of samples as LLVM. As we know, LLVM cannot capture bimodality, since it gives a single Gaussian approximation. While EnLLVM is a mixture of Gaussian models, it is supposed to have better performance in capturing higher order statistics. This is shown in Fig. 2.3. The solid blue circle indicates bimodality of original samples. The dashed blue circles are components of the Gaussian mixture. Once observation data $\tilde{\mathbf{d}}_{k+1}$ is given, The conditional pdf $p(\mathbf{q}_{k+1|k}|\tilde{\mathbf{d}}_{k+1})$ can be obtained by updating each Gaussian components with $\tilde{\mathbf{d}}_{k+1}$ and the bimodality can be preserved. The details of updating process will be discussed in Sec. 2.2.5.

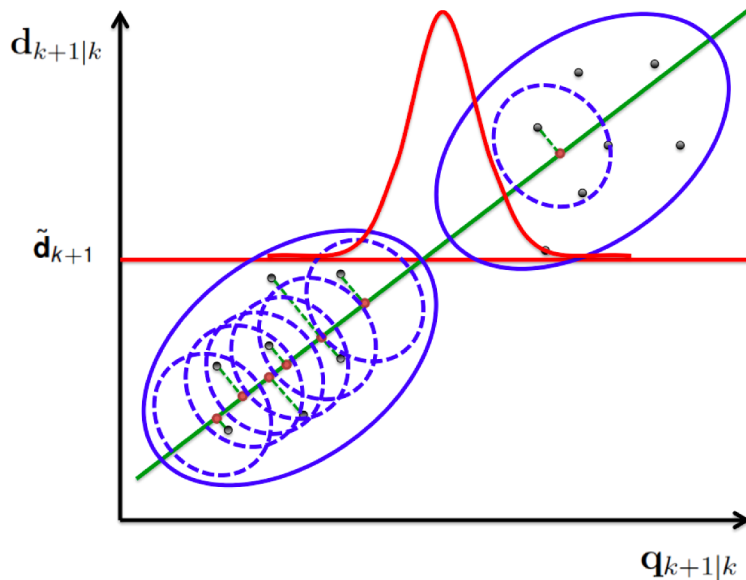


Figure 2.3: EnLLVM captures higher order statistics.

2.2.4 OPTIONS OF LLVM

As is discussed in Sec. 2.2.3, $\boldsymbol{\mu}$ is the mean of samples for all possible latent spaces. A particular latent model only depends on \mathbf{W} and $\boldsymbol{\Psi}$. By restricting the structure of \mathbf{W} and $\boldsymbol{\Psi}$, the latent space can preserve certain property from original samples. To set notation, we decompose Eq. (2.8) into two separate latent models as is shown in

Eq. (2.32) and (2.33).

$$\mathbf{q}_{k+1|k} = \mathbf{W}_q \mathbf{z} + \boldsymbol{\mu}_q + \boldsymbol{\eta}_q \quad (2.32)$$

$$\mathbf{d}_{k+1|k} = \mathbf{W}_d \mathbf{z} + \boldsymbol{\mu}_d + \boldsymbol{\eta}_d \quad (2.33)$$

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_{M \times M})$$

$$\boldsymbol{\eta}_q \sim \mathcal{N}(0, \boldsymbol{\Psi}_q)$$

$$\boldsymbol{\eta}_d \sim \mathcal{N}(0, \boldsymbol{\Psi}_d)$$

Now we have two separate latent models for $\mathbf{q}_{k+1|k}$ and $\mathbf{d}_{k+1|k}$ with a common latent variable \mathbf{z} . Parameters \mathbf{W}_q , \mathbf{W}_d , $\boldsymbol{\mu}_q$, $\boldsymbol{\mu}_d$, $\boldsymbol{\eta}_q$ and $\boldsymbol{\eta}_d$ can be obtained by comparing with Eq. (2.8). By doing this, we have made the assumption that the noise $\boldsymbol{\eta}_q$ and $\boldsymbol{\eta}_d$ are uncorrelated. This is a restriction on the structure of noise $\boldsymbol{\eta}$ and it is the case for commonly used linear latent models which we will discuss in this paper. Fig. 2.4 gives another interpretation of LLVM in our problem. Instead of taking it as the projection from $[\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k}]$ to z , we can view it from a generative viewpoint in which a sample of $\mathbf{q}_{k+1|k}$ is obtained by sampling latent variable z and noise η , then sampling the Gaussian representation conditioned on z and η . A sample of $\mathbf{d}_{k+1|k}$ is obtained in a similar way. Note that $\mathbf{q}_{k+1|k}$ and $\mathbf{d}_{k+1|k}$ share the same z . This process is shown with red color. Here we introduce three commonly used linear latent models.

Probabilistic Principal component analysis (PPCA) PCA is concerned with finding orthogonal directions in lower dimensional space, on which the projections have the largest variance. It is assumed that these projections in lower dimensional space can reserve most information from the original data set, thus can be used for dimensionality reduction and feature extraction. Probabilistic principal component analysis (PPCA) is the probabilistic interpretation of PCA. It can be viewed as a maximum likelihood solution of the probabilistic latent variable model given in

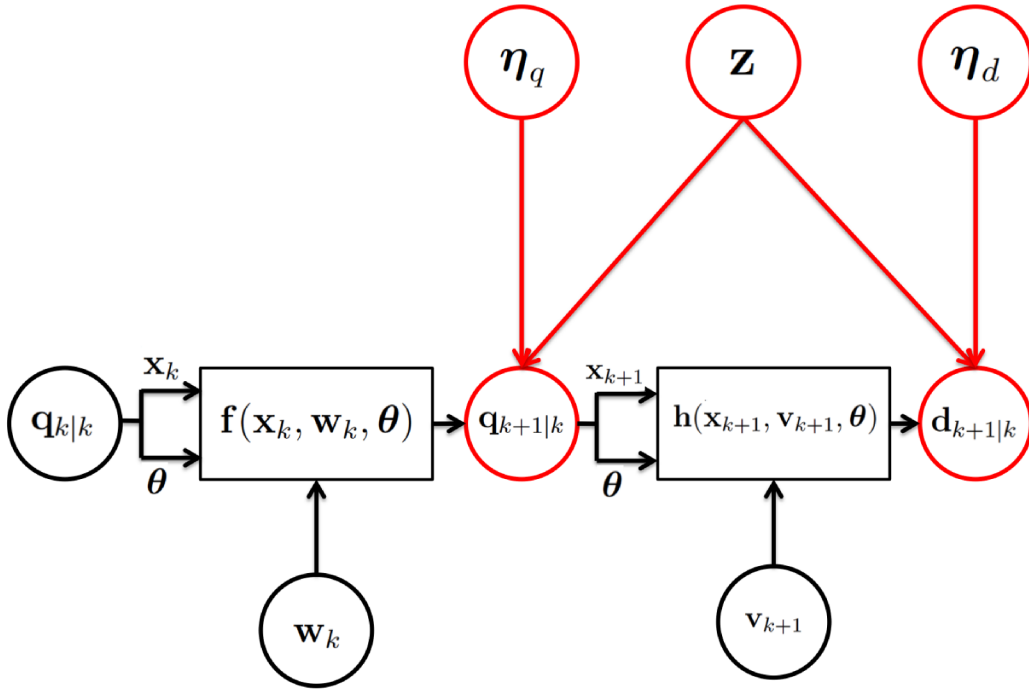


Figure 2.4: Workflow of EnLLVM

Eq. (2.8). We have mentioned that by restricting the structure of the covariance matrix Ψ , we can obtain different latent space. For PPCA, all off-diagonal values in Ψ are zero and values on the diagonal are the same.

$$\text{blkdiag}(\Psi_q, \Psi_d) = \sigma^2 \mathbf{I}_{D \times D}$$

As is discussed before, $\boldsymbol{\mu}$ is the mean of samples no matter what latent space is. The other two parameters \mathbf{W} and σ^2 can be also be obtained through maximum likelihood method. Tipping et al. [79] gave the exact closed-form solution. However, if the dimensionality of the original space is high, which is one of the main focus of this paper, an iterative expectation maximization (EM) procedure can be performed to lower the computational cost. The EM algorithm can also be implemented incrementally. Samples are processed in an incremental and asynchronous way, which can be advantageous if both N and H are large. For the implementation of EM procedure, one can refer to [5].

Factor analysis Factor analysis is closely related to PPCA. The only difference in the latent model is that the diagonal of Ψ can have different values, while for PPCA, all the diagonal values are the same.

$$\text{blkdiag}(\Psi_q, \Psi_d) = \text{diag}(\sigma_1^2 \dots \sigma_D^2)$$

Unlike PPCA, there is no closed-form maximum likelihood solution for \mathbf{W} . However, it can be done through EM method [5].

Probabilistic Canonical Correlation Analysis (PCCA) Given two random vectors \mathbf{q} and \mathbf{d} , CCA is concerned with finding two directions on which the projections of \mathbf{q} and \mathbf{d} have maximum correlation. Assuming \mathbf{q} has dimension H_q and \mathbf{d} has dimension H_d , the total number of canonical correlations is $\min\{H_q, H_d\}$. For PCCA, parameters \mathbf{W}_q , \mathbf{W}_d , $\boldsymbol{\mu}_q$, $\boldsymbol{\mu}_d$, $\boldsymbol{\eta}_q$ and $\boldsymbol{\eta}_d$ can also be obtained by maximizing the likelihood function. Bach et al. [2] gives the closed-form solution as well as EM algorithm. Note, unlike PPCA and factor analysis, $\boldsymbol{\eta}_q$ and $\boldsymbol{\eta}_d$ have off-diagonal values.

2.2.5 BAYESIAN UPDATE OF ENLLVM

Once observation data $\tilde{\mathbf{d}}_{k+1}$ is obtained, the latent variable \mathbf{z} for each component of the Gaussian mixture can be updated through Eq. (2.33). Then posterior $p(\hat{\mathbf{q}}_{k+1|k+1})$ can be estimated by sampling from the updated Gaussian mixture $\hat{p}_{en}(\mathbf{q}_{k+1|k+1})$. The Gaussian mixture is updated in following way. First, for each Gaussian component, latent variable \mathbf{z}_i is updated through Eq. (2.33). This update is performed in the same way as Gaussian component is initialized by each sample. The mean and covariance

matrix of \mathbf{z}_i given $\tilde{\mathbf{d}}_{k+1}$ are as follows

$$\begin{aligned} \mathbb{E}[\mathbf{z}_i|\tilde{\mathbf{d}}_{k+1}] &= \text{Cov}[\mathbf{z}_i|\tilde{\mathbf{d}}_{k+1}](\mathbf{W}^T\boldsymbol{\Psi}_d^{-1}(\tilde{\mathbf{d}}_{k+1} - \boldsymbol{\mu}_d) \\ &\quad + \text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^{-1}\mathbb{E}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]) \end{aligned} \quad (2.34)$$

$$\text{Cov}[\mathbf{z}_i|\tilde{\mathbf{d}}_{k+1}] = (\text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]^{-1} + \mathbf{W}_d^T\boldsymbol{\Psi}_d^{-1}\mathbf{W}_d)^{-1} \quad (2.35)$$

where $\mathbb{E}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]$ and $\text{Cov}[\mathbf{z}|\mathbf{q}_{k+1|k}^i, \mathbf{d}_{k+1|k}^i]$ are given in Eq. (2.13) and Eq. (2.14) respectively. Second, the weight of each Gaussian component is calculated, as $\tilde{\mathbf{d}}_{k+1}$ has different likelihood for each component. This update procedure shares the idea as particle filter in which each particle is assigned with a weight according to its likelihood. However, the likelihoods in EnLLVM and particle filter are different in essence. In particle filter, the likelihood of observation data is calculated through observation model that is Eq. (2.2). While in EnLLVM, the likelihood isn't directly related to the observation model. It is calculated through Eq. (2.33) which is a latent model that we created. Thus EnLLVM can be applied to the data assimilation problem in which typical likelihood is intractable. After \mathbf{z} and weight are updated for each Gaussian component, $\hat{p}_{en}(\mathbf{q}_{k+1|k+1})$ can be obtained as is shown in Eq. (2.36) from which posterior samples can be generated.

$$\hat{p}_{en}(\mathbf{q}_{k+1|k+1}) = \sum_{i=1}^N w_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (2.36)$$

$$\boldsymbol{\mu}_i = \mathbf{W}_q \mathbb{E}[\mathbf{z}_i|\tilde{\mathbf{d}}_{k+1}] + \boldsymbol{\mu}_q \quad (2.37)$$

$$\boldsymbol{\Sigma}_i = \mathbf{W}_q \text{Cov}[\mathbf{z}_i|\tilde{\mathbf{d}}_{k+1}] \mathbf{W}_q^T + \boldsymbol{\Psi}_q . \quad (2.38)$$

One may notice that, since we already have Gaussian representation for each sample $[\mathbf{q}_{k|k+1}^i, \mathbf{d}_{k|k+1}^i]$, given observation data $\tilde{\mathbf{d}}_{k+1}$, $p(\mathbf{q}_{k+1|k+1}^i)$ can be obtained directly by conditioning in the original joint space. But here we perform the update through latent model. This is due to the potential computational saving.

Lemma 2. Consider latent models shown in Eq. (2.32) and Eq. (2.33) with $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Psi}_z)$. Direct conditioning in the original space and update through latent models give equivalent posterior distribution $p(\mathbf{q}_{k+1|k+1})$.

Proof. The joint distribution of $\mathbf{q}_{k|k+1}$ and $\mathbf{d}_{k|k+1}$ is shown in Eq. (2.39).

$$[\mathbf{q}_{k|k+1}, \mathbf{d}_{k|k+1}]^T \sim \mathcal{N}(\mathbf{W}\boldsymbol{\mu}_z + \boldsymbol{\mu}, \mathbf{W}\boldsymbol{\Psi}_z\mathbf{W}^T + \boldsymbol{\Psi})$$

where the mean and covariance matrix can be decomposed as follows

$$\mathbf{W}\boldsymbol{\mu}_z + \boldsymbol{\mu} = \begin{bmatrix} \mathbf{W}_q\boldsymbol{\mu}_z + \boldsymbol{\mu}_q \\ \mathbf{W}_d\boldsymbol{\mu}_z + \boldsymbol{\mu}_d \end{bmatrix} \quad (2.39)$$

$$\mathbf{W}\boldsymbol{\Psi}_z\mathbf{W}^T + \boldsymbol{\Psi} = \begin{bmatrix} \mathbf{W}_q\boldsymbol{\Psi}_z\mathbf{W}_q^T + \boldsymbol{\Psi}_q & \mathbf{W}_q\boldsymbol{\Psi}_z\mathbf{W}_d^T \\ \mathbf{W}_d\boldsymbol{\Psi}_z\mathbf{W}_q^T & \mathbf{W}_d\boldsymbol{\Psi}_z\mathbf{W}_d^T + \boldsymbol{\Psi}_d \end{bmatrix}. \quad (2.40)$$

Given $\tilde{\mathbf{d}}_{k+1}$, the conditional distribution of $\mathbf{q}_{k+1|k+1}$ is also a Gaussian distribution of which the mean and covariance matrix are given below

$$\begin{aligned} \mathbb{E}[\mathbf{q}_{k+1|k+1}] &= \mathbf{W}_q\boldsymbol{\mu}_z + \boldsymbol{\mu}_q \\ &+ \mathbf{W}_q\boldsymbol{\Psi}_z\mathbf{W}_d^T(\mathbf{W}_d\boldsymbol{\Psi}_z\mathbf{W}_d^T + \boldsymbol{\Psi}_d)^{-1}(\tilde{\mathbf{d}}_{k+1} - \mathbf{W}_d\boldsymbol{\mu}_z + \boldsymbol{\mu}_d) \end{aligned} \quad (2.41)$$

$$\begin{aligned} \text{Cov}[\mathbf{q}_{k+1|k+1}] &= \mathbf{W}_q\boldsymbol{\Psi}_z\mathbf{W}_q^T + \boldsymbol{\Psi}_q \\ &- \mathbf{W}_q\boldsymbol{\Psi}_z\mathbf{W}_d^T(\mathbf{W}_d\boldsymbol{\Psi}_z\mathbf{W}_d^T + \boldsymbol{\Psi}_d)^{-1}\mathbf{W}_d\boldsymbol{\Psi}_z\mathbf{W}_q^T. \end{aligned} \quad (2.42)$$

If we update $\mathbf{q}_{k+1|k+1}$ through linear latent model, the mean and covariance matrix of latent variable \mathbf{z} are already given in Eq. (2.34) and Eq. (2.35). Since the goal here is to show the equivalence with Eq. (2.41) and Eq. (2.41), we will use the other representation given by Kalman filter, which is shown below

$$\mathbb{E}[\mathbf{z}|\tilde{\mathbf{d}}_{k+1}] = \boldsymbol{\Psi}_z + \boldsymbol{\Psi}_z\mathbf{W}_d^T(\mathbf{W}_d\boldsymbol{\Psi}_z\mathbf{W}_d^T + \boldsymbol{\Psi}_d)^{-1}(\tilde{\mathbf{d}}_{k+1} - \mathbf{W}_d\boldsymbol{\mu}_z + \boldsymbol{\mu}_d) \quad (2.43)$$

$$\text{Cov}[\mathbf{z}|\tilde{\mathbf{d}}_{k+1}] = (\mathbf{I}_{M \times M} - \boldsymbol{\Psi}_z\mathbf{W}_d^T(\mathbf{W}_d\boldsymbol{\Psi}_z\mathbf{W}_d^T + \boldsymbol{\Psi}_d)^{-1}\mathbf{W}_d)\boldsymbol{\Psi}_z. \quad (2.44)$$

Then

$$\mathbf{E}[\mathbf{q}_{k+1|k+1}] = \mathbf{W}_q \mathbf{E}[\mathbf{z}|\tilde{\mathbf{d}}_{k+1}] + \boldsymbol{\mu}_q \quad (2.45)$$

$$\begin{aligned} &= \mathbf{W}_q \boldsymbol{\mu}_z + \boldsymbol{\mu}_q \\ &\quad + \mathbf{W}_q \boldsymbol{\Psi}_z \mathbf{W}_d^T (\mathbf{W}_d \boldsymbol{\Psi}_z \mathbf{W}_d^T + \boldsymbol{\Psi}_d)^{-1} (\tilde{\mathbf{d}}_{k+1} - \mathbf{W}_d \boldsymbol{\mu}_z + \boldsymbol{\mu}_d) \end{aligned} \quad (2.46)$$

$$\text{Cov}[\mathbf{q}_{k+1|k+1}] = \mathbf{W}_q \text{Cov}[\mathbf{z}|\tilde{\mathbf{d}}_{k+1}] \mathbf{W}_q^T + \boldsymbol{\Psi}_d \quad (2.47)$$

$$\begin{aligned} &= \mathbf{W}_q \boldsymbol{\Psi}_z \mathbf{W}_q^T + \boldsymbol{\Psi}_q \\ &\quad - \mathbf{W}_q \boldsymbol{\Psi}_z \mathbf{W}_d^T (\mathbf{W}_d \boldsymbol{\Psi}_z \mathbf{W}_d^T + \boldsymbol{\Psi}_d)^{-1} \mathbf{W}_d \boldsymbol{\Psi}_z \mathbf{W}_q^T. \end{aligned} \quad (2.48)$$

As we can see, both ways give the equivalent update. However, conditioning in the data space requires the inversion of H_d by H_d matrix which is inevitable As is discussed before, if $\boldsymbol{\Psi}_d$ is a diagonal matrix, inversion of $\boldsymbol{\Psi}_d$ in Eq. (2.34) and Eq. (2.35) only costs $O(H_d)$. Although an inversion of M by M matrix is added and can not be avoided, the dimensionality of latent space is usually much less than the observation space. Thus less computational cost is expected by updating through latent model.

2.2.6 NOISE INFLATION

After Bayesian update, each Gaussian component is assigned with a weight which is calculated according to the corresponding likelihood. This may arise a degeneracy problem, that is, the weight is concentrated on very few Gaussian components. In this case, posterior samples are likely to be generated from the same component many times, which will fail to describe the real posterior distribution. Once the generated samples are propagated to the next time points and same operation is performed, less and less diversity will be kept. Another issue is to deal with model error. The existence of model error indicates that the observation data $\tilde{\mathbf{d}}_{k+1}$ may not be generated from Eq. (2.1) and Eq. (2.2), which further leads to the possibility that $[\tilde{\mathbf{d}}_{k+1}, \mathbf{q}_{k+1|k+1}]$ may be far away from the joint distribution $p(\mathbf{q}_{k+1|k}, \mathbf{d}_{k+1|k})$ described by the current latent model.

To avoid component degeneracy and accommodate for the model error, noise inflation is performed. We introduce a new hyperparameter α in the distribution of the noise $\boldsymbol{\eta}$

$$\boldsymbol{\eta}_d \sim \mathcal{N}(0, \alpha \boldsymbol{\Psi}_d) \quad (2.49)$$

This hyperparameter is the key in performing data assimilation in the presence of model error. Initially, after solving the LLVM, the Gaussian representation is sufficient to capture the predictive samples. However, since our models always depart from reality due to modeling approximations, there is a discrepancy between our model predictions $\mathbf{d}_{k+1|k}$ and the actual measurements $\tilde{\mathbf{d}}_{k+1}$. The hyperparameter α will be automatically tuned to accommodate for model discrepancy, it is determined in the same way as getting the parameters of the LLVM. Namely, α is given by the maximizing the likelihood function provided by Eq. (2.33).

$$\alpha = \arg \max \mathcal{N}(\tilde{\mathbf{d}}_{k+1}; \boldsymbol{\mu}_d, \mathbf{W}_d \mathbf{W}_d^T + \alpha \boldsymbol{\Psi}_d)$$

There is a closed form solution to Eq. (2.33). To make it more general, let's introduce a lemma first.

Lemma 3. For fixed observation data $\tilde{\mathbf{d}}$ and mean $\boldsymbol{\mu}$, the maximum value of $\mathcal{N}(\tilde{\mathbf{d}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is achieved when $\boldsymbol{\Sigma}^* = (\tilde{\mathbf{d}} - \boldsymbol{\mu})(\tilde{\mathbf{d}} - \boldsymbol{\mu})^T$

Proof. The log likelihood is given in Eq. (2.50).

$$\ln \mathcal{N}(\tilde{\mathbf{d}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{k}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} (\tilde{\mathbf{d}} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\tilde{\mathbf{d}} - \boldsymbol{\mu}) \quad (2.50)$$

where k is dimensionality of the Gaussian distribution. Take the derivative of $\ln \mathcal{N}(\tilde{\mathbf{d}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\Sigma}$, we have

$$\frac{\partial \ln \mathcal{N}(\tilde{\mathbf{d}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}} = -\frac{1}{2} \left(\frac{\partial \ln |\boldsymbol{\Sigma}|}{\partial \boldsymbol{\Sigma}} + \frac{\partial (\tilde{\mathbf{d}} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\tilde{\mathbf{d}} - \boldsymbol{\mu})}{\boldsymbol{\Sigma}} \right) \quad (2.51)$$

$$= -\frac{1}{2} (\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1} (\tilde{\mathbf{d}} - \boldsymbol{\mu})(\tilde{\mathbf{d}} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}) \quad (2.52)$$

Set Eq. (2.52) to 0, we obtain $\boldsymbol{\Sigma}^*$ that maximize the log likelihood.

$$\boldsymbol{\Sigma}^* = (\tilde{\mathbf{d}} - \boldsymbol{\mu})(\tilde{\mathbf{d}} - \boldsymbol{\mu})^T \quad (2.53)$$

In our case, the covariance matrix is $\mathbf{W}_d \mathbf{W}_d^T + \alpha \mathbf{\Psi}_d$, thus α^* can be obtained by solving the following equation:

$$\alpha^* \mathbf{I} = \mathbf{\Psi}_d^E \mathbf{\Psi}_d^{-1} \quad (2.54)$$

$$\mathbf{\Psi}_d^E = (\tilde{\mathbf{d}}_{k+1} - \boldsymbol{\mu}_d)(\tilde{\mathbf{d}}_{k+1} - \boldsymbol{\mu}_d)^T - \mathbf{W}_d \mathbf{W}_d^T \quad (2.55)$$

Usually, the exact solution of Eq. (2.54) does not exist, as $\mathbf{\Psi}_d^E \mathbf{\Psi}_d^{-1}$ may have nonzero off-diagonal values and different diagonal values. However, we can always find approximate solution by minimizing the Frobenius norm difference between $\alpha^* \mathbf{I}_{k \times k}$ and $\mathbf{\Psi}_d^E \mathbf{\Psi}_d^{-1}$, which leads to the following solution:

$$\alpha^* = \frac{1}{k} \text{tr}(\mathbf{\Psi}_d^E \mathbf{\Psi}_d^{-1}) \quad (2.56)$$

If $\alpha^* > 1$, the noise in Eq. (2.33) will be inflated, which will lead to larger uncertainty in the latent variable. This uncertainty will eventually be reflected in the uncertainty of QoIs through Eq. (2.32). This means the model error will be absorbed in the uncertainty of QoIs, which provides a way to investigate model error and develop models.

2.3 NUMERICAL EXPERIMENTS

In this section, the predictability of EnLLVM will be assessed by working on two chaotic systems. The first example will show the capability of capturing bimodal distribution. In the second example, EnLLVM and EnKF will be applied to a high dimensional system, and their performance will be compared on several different scenarios.

2.3.1 EXAMPLE 1 - LORENZ 63

The Lorenz 63 is a three dimensional system and was used to study multimodality in [78, 27].

$$\frac{dx_1}{dt} = -cx_1 + cx_2 \quad (2.57)$$

$$\frac{dx_2}{dt} = -x_1x_3 + rx_1 - y_2 \quad (2.58)$$

$$\frac{dx_3}{dt} = x_1x_2 - bx_3 \quad (2.59)$$

Here $c = 10$, $b = \frac{8}{3}$ and $r = 28$. The discrete measurement model is given in Eq. (2.60)

$$d_k = \sqrt{x_1(t_k)^2 + x_2(t_k)^2 + x_3(t_k)^2} + v_k, \quad v_k \sim \mathcal{N}(0, 1) \quad (2.60)$$

The initial state pdf is set to

$$p(x(t_0)) \sim 0.5\mathcal{N}([-0.2, -0.2, 8]^T, \sqrt{0.35}\mathbf{I}_3) + 0.5\mathcal{N}([0.2, 0.2, 8]^T, \sqrt{0.35}\mathbf{I}_3)$$

The total simulation time is 4 and the system is discretized with a time step $\Delta t = 0.1$. The measurement is generated by randomly selected initial states from Eq. (2.61) and propagated through Eq. (2.60). States are updated every 4 time steps.

The joint space of states and observation have a dimensionality of four, and in this simulation, we use two components for PPCA and 30 samples for Bayesian update. Fig. 2.5 shows the posterior distribution of states after each update. The horizontal lines indicate the truth. As we can see, EnPPCA is doing well in capturing the truth. Besides, it can also capture the bimodality of the states.

2.3.2 EXAMPLE 2 - LORENZ 96

In this example, EnPPCA and EnKF will be applied to track the states of Lorenz 96. Lorenz 96 is a 40 dimensional nonlinear system and commonly used for studying data assimilation. The system is given in Eq. (2.61).

$$\frac{dx_j(t)}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + 8, \quad \text{for } j = 1 \dots 40, \quad (2.61)$$

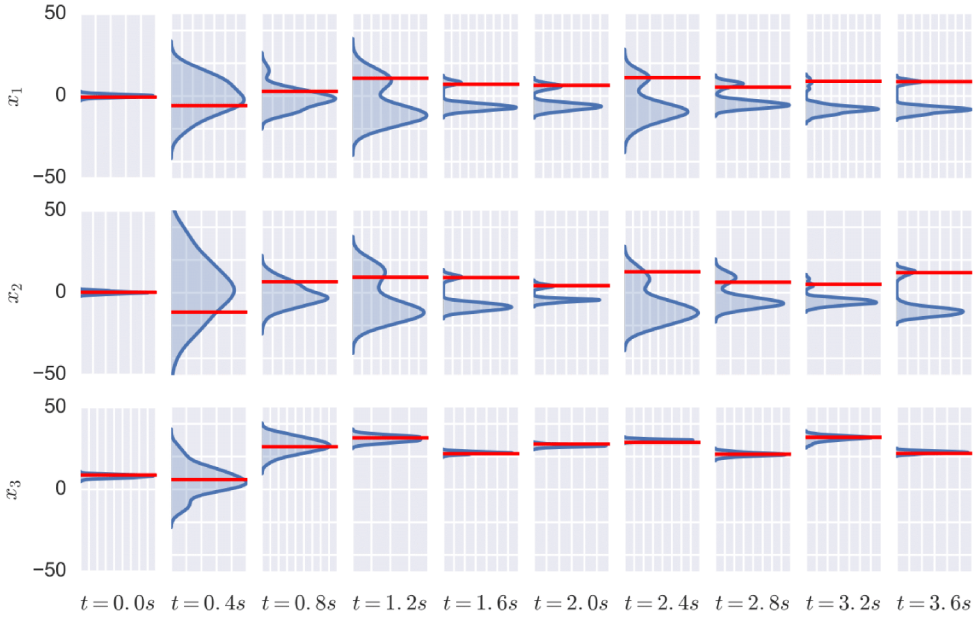


Figure 2.5: Posterior distribution of states in Lorenz 63 system after each update.

where $x_0 = x_{40}$, $x_{-1} = x_{39}$, and $x_{41} = x_1$. The system is discretized with a time step of $\Delta t = 0.001$. The prior is given by $\mathcal{N}(0, 1)$ and the observation model provides an incomplete observation of the state of the system at every $\Delta t = 0.1$. EnKF and EnPPCA will be compared in different scenarios. Besides the linear measurement model, two different nonlinear measurement models are also used. Another important scenario is the presence of model error. As is discussed earlier, EnPPCA has the property to absorb potential model error, and reflects it in the posterior distribution of QoI. Here, four different models are used to generate samples and they only differ in the constant term.

$$\left\{ \begin{array}{l}
\text{Data generation process:} \\
M1 : \frac{dx_j(t)}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + 9 \\
M2 : \frac{dx_j(t)}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + 10 \\
M3 : \frac{dx_j(t)}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + 11 \\
M4 : \frac{dx_j(t)}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + 12 \\
\text{True model:} \\
\frac{dx_j(t)}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + 8 \\
\text{Measurement models:} \\
d_j(t) = x_{2j-1}(t) + v_j(t), \quad v_j(t) \sim \mathcal{N}(0, I_{20}) \\
\text{Nonlinear measurement model:} \\
(I) : d_j(t) = x_{2j-1}(t)x_{2j}(t) + v_j(t), \quad v_j(t) \sim \mathcal{N}(0, I_{20}) \\
(II) : d_j(t) = x_{2j-1}(t)^2 + v_j(t), \quad v_j(t) \sim \mathcal{N}(0, I_{20})
\end{array} \right.$$

The joint space between the QoIs (states of the system) and the observable has dimensionality of 60. For this study, 5 components are used for EnPPCA, which means the dimensionality of the latent space is 5. Here, we use 30 samples to track the states of the system. Root mean square error (RMSE) is used as the metric to measure the predictability of EnPPCA and EnKF. RMSE is calculated between the mean of posterior samples and the true state value.

Table. 2.1 shows the RMSE statistics of 100 trials. As we can see, if no model error exists, the performance of EnKF is comparable with or better than EnPPCA in average RMSE for all three measurement models. However, if we look at predictive distribution given by EnKF, which is shown in Fig. 2.6, the 95% credible interval of posterior samples often fails to capture the true state. This is due to the limited number of samples which underestimate the covariance matrix of the states. Fig. 2.7

Table 2.1: EnKF vs EnPPCA: RMSE statistics of 100 trials

Model Error	EnKF		EnPPCA	
	Mean	STD	Mean	STD
Linear Measurement Model				
No err.	2.71	0.61	2.92	0.23
M1: 9	3.71	0.63	2.99	0.23
M2: 10	4.63	0.71	3.07	0.22
M3: 11	5.31	0.81	3.18	0.21
M4: 12	5.9	0.84	3.27	0.20
Nonlinear Measurement Model (I)				
No err.	1.87	0.53	2.85	0.26
M1: 9	3.3	0.63	2.91	0.22
M2: 10	4.37	0.47	3.02	0.21
M3: 11	5.01	0.44	3.17	0.17
M4: 12	5.51	0.49	3.32	0.17
Nonlinear Measurement Model (II)				
No err.	2.87	0.76	2.84	0.24
M1: 9	3.78	0.75	2.94	0.22
M2: 10	4.78	0.56	3.03	0.20
M3: 11	5.43	0.53	3.16	0.19
M4: 12	6.1	0.55	3.37	0.17

shows predictive trajectories provided by EnPPCA. As we can see, the truth nearly always fall in the 95% credible interval. Once model error is introduced, the RMSE increases for both EnKF and EnPPCA. However, EnPPCA has a smaller average RMSE than EnKF. And RMSE of EnKF increases significantly with the increasing model error, since there is no procedure to deal with model error. Fig. 2.8, Fig. 2.9 show the predictive trajectories of EnKF and EnPPCA when model M4 is used to generate samples. In this case, the model error is large and we can see that, 95% credible interval provided by EnKF can hardly capture the true state value. While for EnPPCA, it is still able to capture the truth.

2.4 SUMMARY

In this paper, a novel data assimilation method EnLLVM is proposed. EnLLVM is an ensemble method which is based on linear latent variable models. In EnLLVM,

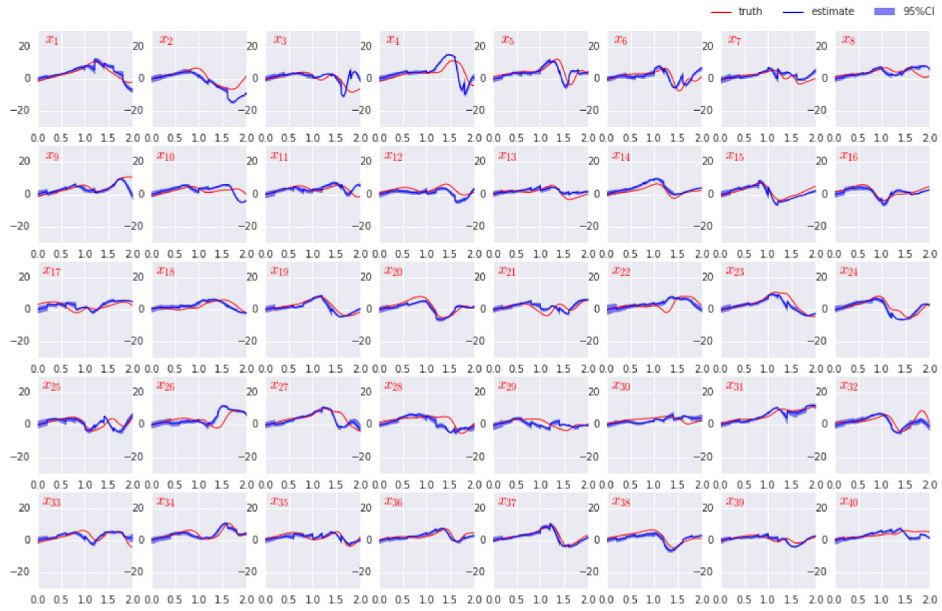


Figure 2.6: Results of data assimilation by EnFK: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete observation model.

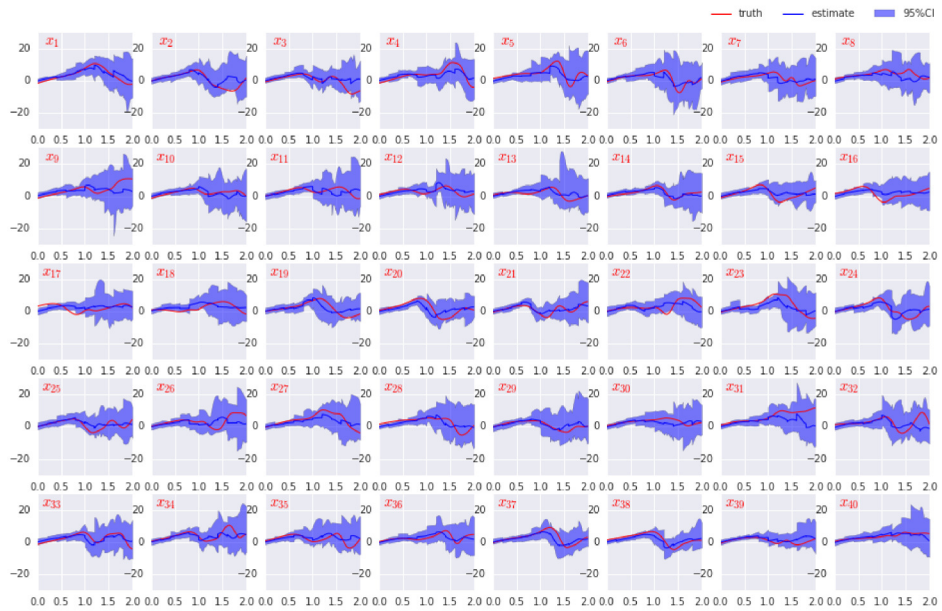


Figure 2.7: Results of data assimilation by PPCA: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete observation model.

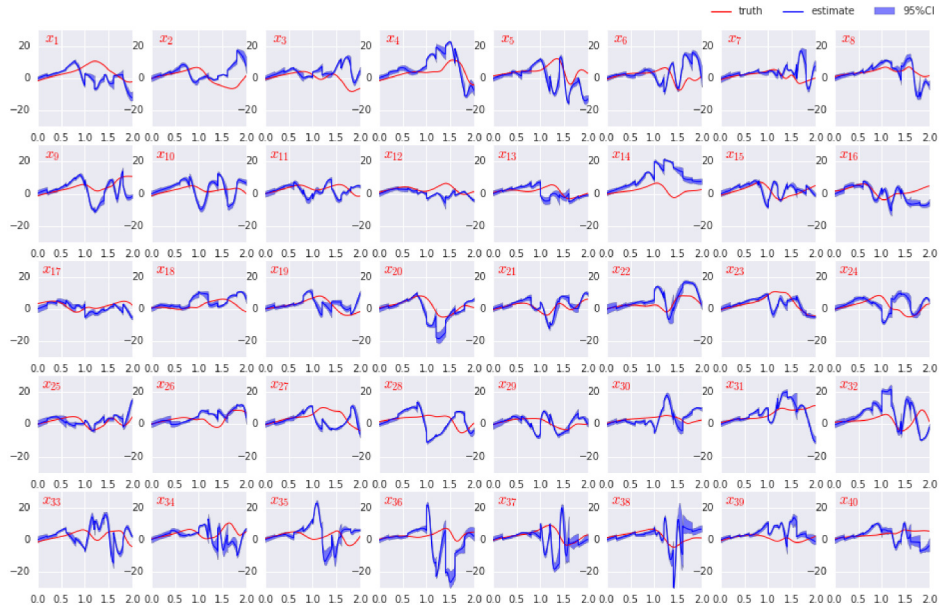


Figure 2.8: Results of data assimilation by EnKF: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete linear observation model with model error M4.

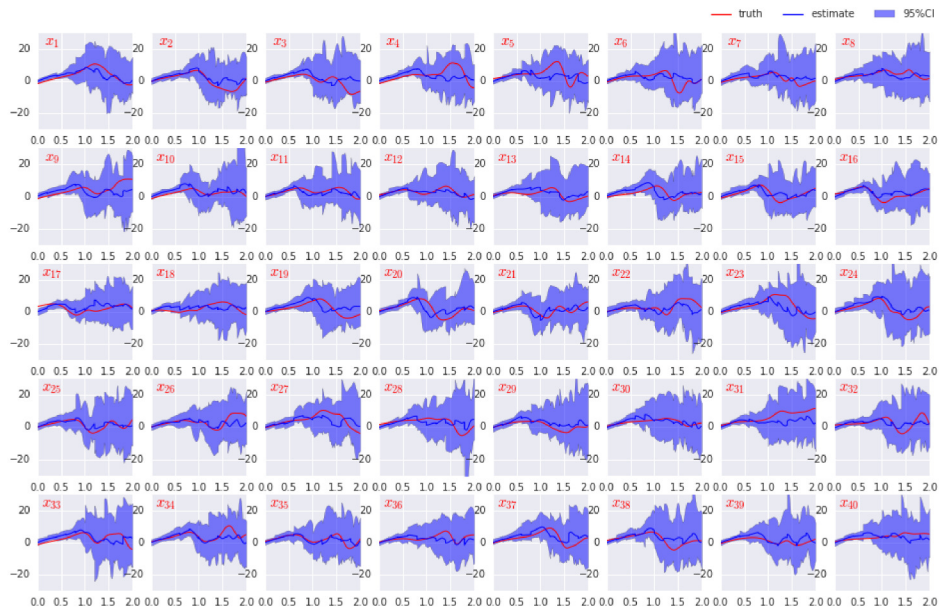


Figure 2.9: Results of data assimilation by EnPPCA: tracking the state of a 40 dimensional Lorenz system using only 30 samples and sparse incomplete linear observation model with model error M4.

the joint distribution of QoIs and observation is approximated by a Gaussian mixture which is constructed by projecting each sample into the latent space. Then Bayesian update of QoIs is performed by first updating the latent variable of each Gaussian component and then map it to the original space. By choosing proper linear latent models, e.g., PPCA or FA, one can avoid inverting big matrix in original space. Instead, only inversion of matrix which has the same dimensionality as the latent space is required. This can be a big saving in computations as the dimensionality of the latent space is supposed to be much lower than that of the original space. EnLLVM only requires a small number of samples.

Compared with EnKF, it doesn't require Gaussian distribution assumption and is able to capture multimodality. Since Bayesian update is performed in the latent space without calculating likelihood through observation model, EnLLVM can be applied to data assimilation problems where likelihood function is intractable.

However, EnLLVM is still an approximate method. Although linear latent model facilitates mapping between original space and latent space, the Gaussian presentation it provides is too simple to give an exact description of the probability distribution, even if ensemble method is used. Since samples are projected a much lower latent space, the loss of information is unavoidable.

Since EnLLVM is a general framework without specifying the latent model, We can try different latent models which preserve different kinds of properties from the origin samples. In this paper, the performance of EnPPCA is assessed by applying to two nonlinear dynamic systems Lorenz63 and Lorenz96. The first example shows the capability of EnPPCA to capture bimodal distribution. In the second example, we compared EnPPCA with EnKF on the same system in a bunch of different scenarios including different nonlinear observation models and different model error. Results show that EnPPCA gives consistent better performance.

CHAPTER 3

APPROXIMATE COMPUTATIONAL APPROACHES FOR BAYESIAN SENSOR PLACEMENT IN HIGH DIMENSIONS

3.1 INTRODUCTION

Sensor placement plays an important role in a range of engineering problems, such as grid coverage [25], tracking target [16, 85], and monitoring spatial phenomena [61]. These problems may range across various subjects, however, many of them are similar in essence. In this paper, we restrict ourselves to the problem where quantities of interest (QoIs) need to be inferred given sensor measurements. One example in this category is monitoring atmosphere dispersion event. In a chemical release accident, it is desirable to know the release parameters such as location, strength and time, so as to aid emergency response. These release parameters are QoIs and can be inferred from sensor measurements. Other QoIs include concentration of chemicals which can be inferred in unobserved regions.

These sensor placement problems always involve mathematical models which describe the spatial and/or temporal process in concern. Usually, QoIs are unknown model parameters and/or state variables. The inference involves running computer simulations and solving inverse and forward problems. Generally, there are two categories of approaches, optimization methods and Bayesian inference. The optimization methods provide a single point estimate of unknown parameters by minimizing the discrepancy between model predictions and observational data. While Bayesian inference formulate the problem in a Bayesian framework where quantities have proba-

bilistic description and posterior distribution of unknown parameters can be obtained via Bayes rule. The advantage of Bayesian inference is the incorporation of prior information and the access to the full posterior distribution from which estimates with quantified uncertainties can be extracted. Usually in Bayesian inference, Monte Carlo sampling are employed and inference problems are solved by Markov Chain Monte Carlo, particle filter or Ensemble Kalman filter for computation preference. For a thorough review of both optimization methods and Bayesian inference, one can refer to [42]. In this paper, we focus on Bayesian inference.

Once we have measurement data, QoI can be inferred. However, data collected from different locations may provide different amount of information towards QoI. Moreover, it is usually impractical to place sensors exhaustively due to high cost of installing and maintaining sensors. Thus sensors should be placed judiciously so as to maximize the information content. Sensor locations are commonly decided by running simulation and maximizing certain criterion. There are various choices towards the criterion. It can be Mean Square Error (MSE), mutual information, entropy or measurements about inverse moment matrix in A-, D-, E-optimal design. Among all these criteria, mutual information is most common used, since it is a natural measure of dependence between the two variables and others can be derived from it either directly or indirectly. Ertin (2003) [31] discussed maximum mutual information approach for dynamic sensor query problems. In his paper, mutual information between sensor data and target state was maximized at each step to decide which sensor was queried for tracking the target. And it was shown that this maximum mutual information approach was equivalent to minimizing expected posterior uncertainty in target state. This holds in the special case when the conditional distribution of the observable given the state is independent of the state as in the fixed additive Gaussian noise case. Andreas Krause (2008) [55] discussed sensor placements for prediction problems where mutual information between the observed locations and unobserved locations

is maximized so as to gain most information about observed locations. Xiaopei Wu (2012) [87] tackled a similar problem in soil moisture. Instead of applying maximum mutual information strategy globally, locations were first clustered according to soil moisture content, and then maximum mutual information was used in each cluster to select sensor locations.

Although mutual information is a perfect criterion theoretically, estimating mutual information is challenging. Some commonly used estimators include histogram based estimator, kernel density estimator, and k -nearest neighbor estimator (kNN). In their survey, Walters-Williams and Li [83] show that parametric estimation usually outperform non-parametric estimation when data is drawn from a known family of distributions. But this is not the case in most practical problems. Usually mutual information is estimated directly from Monte Carlo samples. Khan et al. [51] compare different estimators and show that the kNN estimator of mutual information captures better the nonlinear dependence than other commonly used estimators. The kNN estimator of mutual information, proposed by Kraskov et al. [54], is based on kNN estimator of entropy [53]. It is shown that by using matching distances in joint space and marginal spaces, the biases in entropy estimators could be canceled to some extent thus it is able to give a overall better performance.

However, a good estimation requires large sample size in small dimensions which is not always the case in real world applications. This issue is not directly addressed in the past literature. On many occasions, mutual information is adopted as a metric only for comparison purpose. For example, in sensor placement mutual information between QoIs and different locations are compared and there's no need to know the exact value of mutual information as long as the correct comparison results are provided. Thus, we can use alternatives that have consistent performance in such comparisons. In this paper, we propose a lower bound of mutual information that could be used as an alternative for comparisons purpose. The lower bound can be

computed in much lower dimensions than original space. We compare it with kNN estimator in full dimensions on both simulated and real data. The results demonstrate the consistent performance of the proposed lower bound.

Furthermore, Bayesian optimization [44] is introduced to facilitate maximizing the criterion over a continuous domain. This strategy has been used for experimental design in [86]. Bayesian optimization is much more efficient than the typical way which discretizes the domain into fine grid and make selection from all the grid points. Finally, a chemical release accident is simulated where sensors are placed to infer the release location. Mutual information is used as the criterion to place sensors. However, instead of comparing mutual information directly, we compare the proposed lower bound. And this criterion is maximized though Bayesian optimization. The proposed approach shows promising results.

The rest of the paper is organized as follows. In section 2, uncertainty modeling and Bayesian inference is introduced. The proposed sensor placement strategy is detailed in section 3. In section 4, a chemical release accident is simulated and the proposed approach is applied to infer release parameters. Conclusion is given in Section 5.

3.2 BAYESIAN INFERENCE

In this section, we introduce Bayesian inference for solving inverse problem. In the Bayesian framework, uncertainties in state variables and parameters are usually described with probability distributions. Measurement data is used to update the knowledge of these quantities. It is desirable for the posterior distribution to have a small uncertainty and at the same time to capture the true value. The connection between these quantities and observation data is embedded in the mathematical model which describes the process in concern. Since the proposed approach in this paper is universal, an abstract model will be introduced first.

3.2.1 UNCERTAINTY MODELING

To set notation, consider the following abstract model:

$$\mathcal{R}(u, \theta, x) = 0 \tag{3.1}$$

$$y = \mathcal{Y}(u, \theta, x) \tag{3.2}$$

$$q = \mathcal{Q}(u, \theta, x) . \tag{3.3}$$

Here, \mathcal{R} is some operator, u is the solution or the state variable and θ is a set of parameters, which usually have a physical interpretation. x denotes the scenario which defines the problem being considered. In sensor placement, x usually refers to sensor locations. \mathcal{Y} is a map from the solution to the prediction quantity y that can be compared with sensor measurements D . In addition, \mathcal{Q} defines QoI which is denoted by q . In our problem, q can be θ or u or other quantities inferred from θ and u . However, no matter what q is, model parameters and state variables need to be known first. Then other quantities can be obtained through Eq. (3.3). Usually, θ and u are unknown or partially unknown, and need to be inferred from data. Let τ denotes unknown parts of θ and u . In this paper, Bayesian inference is carried out to solve the problem.

Because the observation data is noisy due to sensor imprecision, the measurement noise ϵ usually follows a known pdf $p(\epsilon)$ that is defined by the specifications of the sensors. This results in the following relation between the observable d and model prediction y .

$$d = y + \epsilon . \tag{3.4}$$

Finally, the relation between the observable d and model parameters θ as well as state variable u is given by combining Eq. (4.2) and Eq. (4.5). This measurement model, Eq. (4.6) defines the likelihood function and Bayes rule can be used to update the knowledge of τ .

$$d = \mathcal{Y}(u, \theta, x) + \epsilon . \tag{3.5}$$

Since Bayes rule is used as the inference engine, then a prior probability distribution needs to be defined for τ , that is, $\tau \sim p(\tau)$. Note that the additive errors introduced in the previous equations are not a requirement; multiplicative errors or embedded errors are possible as well.

3.2.2 BAYESIAN INFERENCE

This paper employs probability to represent uncertainty and Bayesian inference to update the uncertainty of unknown quantity τ in light of observation data. In Bayesian inference, one seeks a complete probabilistic description of τ that make the model consistent with the observation data, D . The solution to this problem is the posterior probability density function of τ and it is defined by Bayes's Theorem,

$$p(\tau|D) = \frac{p(D|\tau)p(\tau)}{p(D)} . \quad (3.6)$$

Here, $p(D|\tau)$ is the likelihood function and it measures the agreement between the model output and the data for given values of the input τ . The denominator in Eq. (4.14) is called the marginal likelihood or evidence. Overall, this is just a normalization constant that ensures that the solution to the Bayes' inverse problem, $p(\tau|D)$ is indeed a proper pdf that integrates to one.

In a more general scenario, data is collected over a period of time, and the knowledge of τ is updated after each measurement. Let $D = \{d_1, d_2, \dots, d_M\}$ denote the measurement data collected at M time points. The final posterior distribution $p(\tau|D_M)$ can be obtained recursively as shown in Eq. 3.7

$$\begin{aligned} p(\tau|D_M) &= \frac{p(D_M|\tau)p(\tau)}{p(D_M)} \\ &= \frac{p(d_M, D_{M-1}|\tau)p(\tau)}{p(d_M, D_{M-1})} \\ &= \frac{p(d_M|D_{M-1}, \tau)p(D_{M-1}|\tau)p(\tau)}{p(d_M|D_{M-1})P(D_{M-1})} \\ &= \frac{p(d_M|D_{M-1}, \tau)p(\tau|D_{M-1})}{p(d_M|D_{M-1})} . \end{aligned} \quad (3.7)$$

As we can see, in the inference above, the posterior distribution obtained at current time will be the prior for the next time point. Note, the true value of model parameters remains unchanged, while other quantities like states usually change with time. Another thing worth mentioning is that during the observation time period, sensors can either remain fixed or be mobile. This raises two categories of sensor placement problems, static sensors and mobile sensors. If mobile sensors are used, sensor locations need to be selected at each time point given the current status of the system. In this case, the above observation data d_i are collected from different locations.

A vast number of computational approaches have been invented to solve Eq. 4.14. For linear models with Gaussian distribution and additive white noise, Kalman filter is the most accurate and efficient method to solve the inverse problem. However, this is not the case on most occasions, where models can be nonlinear and the distribution of states and/or parameters are non Gaussian. Then we need to use numerical sampling techniques, also known as Monte Carlo. Most commonly used methods include particle filter and Markov Chain Monte Carlo (MCMC) [50]. In particle filter, samples' weights are updated by calculating likelihood at each sample. For better estimation, more advanced procedures might be added, such as regularization and progressive correction [65]. On the other hand, MCMC has become a main computational workhorse in scientific computing from a large class of distributions. The most basic form of MCMC is Metropolis-Hasting (MH) algorithm [62, 4], which generates a sequence of correlated samples that form a Markov chain. Improved versions of MCMC such as delayed rejection adaptive metropolis (DRAM) and transitional MCMC (TMCMC) [38, 4, 17] are also used. Another important method is ensemble Kalman filter (EnKF). EnKF was first introduced by Evensen [32] and has been widely used in various applications due to its simplicity in both theory and implementation. It originates from Kalman filter but uses Monte Carlo approach to represent proba-

bility distributions. In EnKF, samples are called ensemble members. Each ensemble member is updated through similar formula as in Kalman filter. EnKF propagates a relatively small ensemble of samples through the system non-linearities and moves them such that their mean and covariance approximate the first two moments of the posterior distribution.

As we can see, once observation data are collected, various methods can be performed to implement Bayesian inference. However, sensors placed at different locations might provide different amount of information about τ . More specifically, the posterior distribution $p(\tau|D)$ in Eq. 4.14 is more likely to be different given different D . Some might have a larger uncertainty, while others have more accurate and confident estimation. Thus placing sensors judiciously is important and this will be detailed in the following section.

3.3 SENSOR PLACEMENT

The placing of sensors to infer the QoIs is formulated in Bayesian framework where each QoI has a probabilistic description. Here, we use q to denote all the QoI which might include model parameters, state variables or other quantities. Suppose we have prior information $p(\tau)$, when observation data d is collected, posterior $p(\tau|d)$ can be estimated via Bayes' rule which has been discussed in the previous section. Usually, the uncertainty of τ will be reduced after Bayesian inference. Meanwhile, any uncertainty of τ will be propagated to q through Eq. (3.3) and it is desirable for q to have small uncertainty. However, data collected at different locations will lead to different posterior distribution $p(q)$ which sometimes can be too wide to provide useful information. Thus sensors should be placed strategically. Here, we develop an approach which selects the most informative sensor locations. The approach is based on mutual information criterion which will be discussed first.

3.3.1 MUTUAL INFORMATION CRITERION

Mutual information criterion was first introduced by Lindley [59] who used this criterion to measure the expected amount of information provided about the unknown parameters θ by the measurement data d in an experiment. In this criterion, the information obtained from an experiment is quantified by the reduction in uncertainty of θ which is represented by Shannon entropy. Thus different experimental conditions can be compared and the one that leads to the most reduction in uncertainty of θ will be selected to perform the experiment.

Sensor placement shares the same idea. We use x to denote sensor locations, the amount of information provided by observing d at x will be

$$U(d, x) = H(q) - H(q|d, x) \quad (3.8)$$

$$= - \int_Q p(q) \log p(q) dq + \int_Q p(q|d, x) \log p(q|d, x) dq \quad (3.9)$$

where $H(q)$ is the entropy of prior distribution $p(q)$ and $H(q|d, x)$ represents the entropy of posterior distribution $p(q|d, x)$. The reduction in entropy quantifies how much information is gained by observing d at x . Since observation data can only be obtained after sensors are placed, the expected amount of information is calculated by marginalizing over all possible observations:

$$E_d[U(d, x)] = \int_{\mathcal{D}} U(d, x) p(d|x) dd . \quad (3.10)$$

Eq. (3.10) can be expanded as follows:

$$\begin{aligned} E_d[U(d, x)] &= - \int_{\mathcal{D}} \int_Q p(q) p(d|x) \log p(q) dq dd \\ &\quad + \int_{\mathcal{D}} \int_Q p(q|d, x) p(d|x) \log p(q|d, x) dq dd . \end{aligned} \quad (3.11)$$

In the first term on the right side of Eq. (3.11), d only appears in $p(d|x)$, so $p(d|x)$ can be integrated as 1 without affecting the result. Also q and x are independent.

Then we have

$$\int_{\mathcal{D}} p(q)p(d|x)d = p(q) \quad (3.12)$$

$$= \int_{\mathcal{D}} p(q, d|x)d . \quad (3.13)$$

The first term of Eq. (3.11) becomes $-\int_{\mathcal{D}} \int_Q p(q, d|x) \log p(q)dqdd$. Thus

$$\begin{aligned} E_d[U(d, x)] &= \int_{\mathcal{D}} \int_Q p(q|d, x)p(d|x) \log \frac{p(q|d, x)p(d|x)}{p(d|x)} dqdd \\ &\quad - \int_{\mathcal{D}} \int_Q p(q, d|x) \log p(q)dqdd \end{aligned} \quad (3.14)$$

$$\begin{aligned} &= \int_{\mathcal{D}} \int_Q p(q, d|x) \log \frac{p(q, d|x)}{p(d|x)} dqdd \\ &\quad - \int_{\mathcal{D}} \int_Q p(q, d|x) \log p(q)dqdd \end{aligned} \quad (3.15)$$

$$\begin{aligned} &= \int_{\mathcal{D}} \int_Q p(q, d|x) \log \frac{p(q, d|x)}{p(q)p(d|x)} dqdd \\ &= I(q; d|x) . \end{aligned} \quad (3.16)$$

We can see that the expected amount of information of q provided by sensors equals mutual information between q and sensor readings d . Therefore sensors should be placed where this mutual information is maximized. This is shown in Eq.(3.17)

$$x^* = \arg \max_x I(d; q|x) \quad (3.17)$$

In our paper, we adopt the kNN estimator of mutual information proposed by Kraskov et al. [54]. There are two kNN estimators:

$$I_1(q; d) \approx -\frac{1}{N} \sum_{i=1}^N (\psi(n_q(i) + 1) + \psi(n_d(i) + 1)) + \psi(k) + \psi(N) \quad (3.18)$$

$$I_2(q; d) \approx -\frac{1}{N} \sum_{i=1}^N (\psi(n_q(i)) + \psi(n_d(i))) + \psi(k) + \psi(N) - \frac{1}{k} . \quad (3.19)$$

Here, N represents the total number of samples and $\psi(k)$ is the digamma function. Let $l(i)$ denote the distance from the i th sample $[q(i), d(i)]$ to its k th nearest neighbor in the joint space, $l_q(i)$ and $l_d(i)$ denote the distances between the same points

projected into the q and d subspaces. For Eq. (3.18), $n_d(i)$ and $n_q(i)$ are the number of samples in the subspaces within the distance $l(i)$. For Eq. (3.19), $n_d(i)$ and $n_q(i)$ are the number of samples in the subspaces within the projected distances $l_q(i)$ and $l_d(i)$ respectively. Eq. (3.19) is expected to work better than of Eq. (3.18) in high dimensions. Both estimators are approximate. They are derived under the assumption that density is constant in small (hyper)cubes or (hyper)rectangles centered at each point with size specified by its distance to the k th nearest neighbor. So a small value of k will result in small system error but a large statistic error and vice-versa. It is recommended to use $k = 2 \sim 4$ [54].

Experiments in [54] show that the kNN estimator tends to underestimate mutual information when the N is not large and the proportion of the estimator on the true value scales as $1/\sqrt{N}$. Also the statistical errors scale roughly as m/\sqrt{N} for large dimension m . For $m = 6$ and $N = 50000$, kNN estimator starts to depart from the true value in some experiments. Thus in much higher dimensions, an enormous number of samples are required, which can be computationally too expensive, and impractical for cases when only a limited number of samples are available.

3.3.2 SENSOR PLACEMENT APPROACH

The main focus of this paper is to solve the problem of computing mutual information in high dimensions. Since it is usually impractical to get good enough samples in high dimensions, the kNN estimator becomes unreliable. To solve this problem, in this section we develop a novel approach which computes the lower bound of $I(d; q)$ in much lower dimensions. Also, in order to lower the computational cost, Bayesian optimization [44, 86] is introduced to generate a mutual information surface which greatly reduces the number of evaluations.

The derivation of lower bound is based on data processing inequality which is stated below [19].

Theorem 1. *If random variables J, V, Z form a Markov chain in the order denoted by $J \rightarrow V \rightarrow Z$, then $I(J; V) \geq I(J; Z)$.*

J, V, Z form a Markov chain means the conditional distribution of Z depends only on V and is conditionally independent of J , that is $p(Z|V) = p(Z|J, V)$. A special case of Theorem 1 is that if $Z = g(V)$, then $I(J; V) \geq I(J; Z)$ which means no function $g(\bullet)$ can increase the amount of information that V tells about J . We extend this idea by applying transformations on both J and V , which leads to Corollary 1.

Corollary 1. *For random variables J, V, Z and U , if $J \rightarrow V \rightarrow Z$ and $U = h(J)$, then $I(J; V) \geq I(U; Z)$.*

Proof. $J \rightarrow V \rightarrow Z$ implies $Z \rightarrow V \rightarrow J$ which further implies $Z \rightarrow J \rightarrow h(J)$. Thus $I(J; V) \geq I(J; Z) \geq I(U; Z)$. In particular, if $Z = g(V)$, then $I(J; V) \geq I(h(J), g(V))$. \square

In our case, for any transformation $g(\bullet)$ on d , and $h(\bullet)$ on q we have $h(q) \leftrightarrow q \leftrightarrow d \leftrightarrow g(d)$ which leads to

$$I(q; d) \geq I(q; g(d)) \geq I(h(q); g(d)) . \tag{3.20}$$

Thus, the problem is converted to looking for two proper transformations $g(\bullet)$ on d and $h(\bullet)$ on q . Since we want mutual information to be computed in lower dimension, the most favorable transformations are the ones that convert q and d to two one dimensional variables, at the same time keep the dependency between q and d . One such transformation is canonical correlation analysis (CCA) which will be discussed in the following section.

Given two random vectors J and V , canonical correlation analysis is concerned with finding two directions on which the projections of J and V have maximum correlation. Assuming J has dimension p_1 and V has dimension p_2 , there are p canonical correlations of J and V , where $p = \min\{p_1, p_2\}$. These p canonical correlations can be obtained by solving a generalized eigenvalue problem [3]. The solution will provide all p correlations which are ranked in decreasing order. Here, we only consider the first canonical correlation, that is the largest one among all the correlations, and use it to derive the lower bound of mutual information.

Let α and β denote two projecting directions, the first canonical correlation ρ_1 is defined as the maximum possible correlation between two projections $\alpha^T J$ and $\beta^T V$:

$$\rho_1 = \max_{\alpha, \beta} \text{corr}(\alpha^T J, \beta^T V) . \quad (3.21)$$

Here $\alpha^T J$ and $\beta^T V$ are both one dimensional random variables. Recall that our goal is to find two transformations $h(\bullet)$ and $g(\bullet)$ such that $I(h(J); g(V))$ can be computed in lower dimension. Thus we can choose $h(J) = \alpha^T J$ and $g(V) = \beta^T V$ where α and β are projecting directions of the first canonical correlation.

[3] discussed relation between CCA and mutual information. It is shown that if the joint distribution of J and V is Gaussian, the mutual information between J and V can be expressed as a function of canonical correlations ρ_i , $i = 1, 2, \dots, p$.

$$I(J, V) = -\frac{1}{2} \sum_{i=1}^p \log(1 - \rho_i^2) . \quad (3.22)$$

We can further denote $I(\alpha_i^T J, \beta_i^T V)$ as the mutual information between the i th canonical projections. Then

$$I(J, V) = \sum_{i=1}^p I(\alpha_i^T J, \beta_i^T V) \quad (3.23)$$

which means if $p = 1$, the mutual information of Gaussian variable J and V is equivalent to the mutual information of their first canonical projections, that is

$$I(J, V) = I(\alpha_1^T J, \beta_1^T V) . \quad (3.24)$$

In our case, $h(\bullet)$ and $g(\bullet)$ in Eq. (3.20) are selected to be the first canonical projections. In particular, if q contains only one quantity, then $h(q) = q$. Further more, if the joint distribution of q and d is Gaussian, then the equality will hold. As we can see, the mutual information between the first canonical projections is a lower bound of the original mutual information criterion. And we will use this lower bound as the criterion in the our sensor placement.

BAYESIAN OPTIMIZATION

By projecting the observable and QoI onto a lower dimensional space, the computation of mutual information becomes more reliable. However, to select the location with maximum mutual information over a continuous domain is still challenging, since mutual information is usually computed through Monte Carlo method and doesn't have a close form. A common way to solve this problem is to discretize the domain into grid then compare mutual information at each grid point. The drawback of this method is obvious. On one hand, if the grid is too fine, it will require vast computations which is not efficient. On the other hand, sparse grid may fail to capture the optimal point. In this paper, we apply Bayesian optimization [44] to facilitate the selection of sensor locations.

The basic idea of Bayesian optimization is to evaluate objective function $f(x)$ (which is mutual information in our case) at a small number of points where $f(x)$ is most likely to reach the maximum. Two concepts are needed to address, one is probabilistic prior on objective function $f(x)$, the other is acquisition function $a(x)$. Acquisition function takes in probabilistic distribution of $f(x)$ as argument and decides the point x at which the objective function $f(x)$ should be evaluated.

$$f(x) = I(q; d|x) \tag{3.25}$$

Gaussian process Like prior on variables, we use $p(f(x))$ to denote the prior distribution of $f(x)$. After calculating the estimates of the mutual information at various scenarios, $D_t = \{(x_i, f_i) | i = 1, 2, \dots, t\}$, the posterior distribution of $f(x)$ can be obtained through Bayes' rule

$$p(f(x)|D_t) = \frac{p(D_t|f(x))p(f(x))}{p(D_t)} . \quad (3.26)$$

The most common prior is Gaussian process which is denoted as

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) .$$

Here, $m(x)$ is the mean function and $k(x, x')$ is the covariance function. A Gaussian prior on $f(x)$ means that $f(x)$ at different points x follows a multivariate Gaussian distribution whose mean is $m(x)$ and covariance matrix is specified by $k(x, x')$.

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix} \right)$$

Covariance function $k(x, x')$ defines the correlation between two different points x and x' . A commonly used covariance function is squared exponential function,

$$k(x, x') = \exp(-(x - x')^T \Lambda^{-1} (x - x')) \quad (3.27)$$

where Λ is a diagonal matrix of which each entry λ on the diagonal specifies the correlation length of two points in the corresponding dimension. A large λ means $f(x)$ is smooth in that dimension. Usually, λ is obtained through maximum likelihood estimation.

Now let F_t denote observed outputs over $X_t = [x_1, x_2, \dots, x_t]$, F_n denote outputs over any $X_n = [x_{t+1}, x_{t+2}, \dots, x_n]$, M_t and M_n denote mean vectors $[m(x_1), m(x_2), \dots, m(x_t)]$ and $[m(x_{t+1}), m(x_{t+2}), \dots, m(x_n)]$ respectively. According to the Gaussian prior, we have

$$\begin{bmatrix} F_t \\ F_n \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} M_t \\ M_n \end{bmatrix}, \begin{bmatrix} K(X_t, X_t) & K(X_t, X_n) \\ K(X_n, X_t) & K(X_n, X_n) \end{bmatrix} \right) .$$

$K(X_t, X_n)$ is a covariance matrix of which the element at (i, j) is specified by $k(x_i, x_j)$. From this joint Gaussian distribution, we can obtain the conditional distribution of F_n given F_t

$$p(F_n|F_t) \sim \mathcal{N}(\mu(X_n), \Sigma(X_n)) \quad (3.28)$$

where

$$\mu(X_n) = M_n + K(X_n, X_t)K(X_t, X_t)^{-1}(F_t - M_t) \quad (3.29)$$

$$\Sigma(X_n) = K(X_n, X_n) - K(X_n, X_t)K(X_t, X_t)^{-1}K(X_t, X_n) . \quad (3.30)$$

The derivation of Eq. (3.29) and Eq. (3.30) is based on the fact

$p(F_n, F_t) = p(F_n|F_t)p(F_t)$. Also since if two sets of variables are joined Gaussian distribution then the conditional distribution of one set given the other is also Gaussian, we only need to work on the exponential terms of Gaussian densities. Thus we have

$$\begin{aligned} & -\frac{1}{2} \left(\begin{bmatrix} F_t \\ F_n \end{bmatrix} - \begin{bmatrix} M_t \\ M_n \end{bmatrix} \right)^T \begin{bmatrix} K(X_t, X_t) & K(X_t, X_n) \\ K(X_n, X_t) & K(X_n, X_n) \end{bmatrix}^{-1} \left(\begin{bmatrix} F_t \\ F_n \end{bmatrix} - \begin{bmatrix} M_t \\ M_n \end{bmatrix} \right) \\ &= -\frac{1}{2} (F_t - M_t)^T K(X_t, X_t)^{-1} (F_t - M_t) \\ & \quad - \frac{1}{2} (F_n - \mu(X_n))^T \Sigma^{-1}(X_n) (F_n - \mu(X_n)) . \end{aligned} \quad (3.31)$$

By comparing the related terms in Eq. (3.31), we can obtain $\mu(X_n)$ and $\Sigma(X_n)$. This also involves using partitioned matrix and the Schur complement to get inverse matrices. For details, one can refer to [5]. So we can get estimation of mean $\mu(f(x))$ and variance $\sigma^2(f(x))$ for each x . Thus a surface of $f(x)$ with uncertainty $\sigma^2(f(x))$ is generated. Each time after evaluating $f(x)$ at some x , the surface will be updated according to Eq. (3.29) and Eq. (3.30).

Acquisition function In the last section, we have discussed Gaussian process prior over objective function $f(x)$ and how to update distribution of $f(x)$ given evaluation at x . But where to collect real data D remains unsolved, since we want to place

the sensor where the maximum $f(x)$ is achieved. Here, acquisition function $a(x)$ will guide the searching process.

Acquisition function takes the mean $\mu(f(x))$ and the variance $\sigma^2(f(x))$ as two arguments, and has a property that large value of $a(x)$ is associated with potentially large value of $f(x)$. Thus we only need to evaluate $f(x)$ at the point where $a(x)$ reaches maximum. And $a(x)$ is usually much easier to evaluate.

One of the most popular acquisition function is expected improvement [11]. Let $f(x^+)$ denote the current maximum value of objective function. The improvement of the new evaluation will be

$$\delta(x) = \max\{0, f(x_{t+1}) - f(x^+)\} . \quad (3.32)$$

Since $f(x_{t+1})$ follows normal distribution with mean $\mu(f(x))$ and variance $\sigma(f(x))$ which can be obtained from Gaussian process, the expected improvement will be

$$\begin{aligned} EI(x) &= \int_{\delta=0}^{\delta=\infty} \delta \frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{(\mu(x) - (\delta + f(x^+)))^2}{2\sigma^2(x)}\right) d\delta \\ &= \sigma(x) \left[\frac{\mu(x) - f(x^+)}{\sigma(x)} \Phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right) + \phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right) \right] . \end{aligned} \quad (3.33)$$

Here Φ and ϕ represent cumulative density function and probability density function of the standard normal distribution respectively. As we can see the acquisition function achieves a trade-off between large values of $f(x)$ and large uncertainty in $f(x)$. Maximizing $EI(x)$ will give the next point to evaluate $f(x)$.

PLACEMENT OF MULTIPLE SENSORS

Bayesian optimization is powerful in looking for one single sensor location. However, in most cases, it is necessary to place multiple sensors so as to get enough information about QoI. Selecting multiple sensor locations is more complicated. When using sparse grids, the number of combinations of sensor locations where mutual information needs to be evaluated becomes prohibited computationally. Here, we adopt

a greedy approach that leverages the previously introduced lower bound. Suppose N sensors need to be placed. When we place the first sensor, the lower bound of $I(d_1(x); q)$ is maximized through Bayesian optimization and the point x with maximum mutual information is selected as the first sensor’s location. For the second sensor, the lower bound of $I(d_1, d_2(x); q)$ is maximized. This time d_1 is associated with the first sensor location which is fixed. The other sensors’ locations are selected in the same way, maximizing the lower bound of $I(d_1, d_2, \dots, d_i(x); q)$, until $i = N$. Here, we use $I(h_i(q), g_i(d_i^*(x)))$ to denote the lower bound and $d_i^*(x) = [d_1, d_2, \dots, d_i(x)]$. The whole process of sensor placement is shown in Algorithm. 1.

Algorithm 1 Sensor placement with Bayesian optimization

- 1: **for** $i = 1, 2, \dots, N$ **do**
 - 2: Evaluate $I(h_i(q), g_i(d_i^*(x)))$ at initial points $X_{init} = \{x_k | k = 1, 2, \dots, K\}$.
 - 3: Update Gaussian process on $I(h_i(q), g_i(d_i^*(x)))$ with $E_{init} = \{(I_k, x_k) | k = 1, 2, \dots, K\}$.
 - 4: **for** $j = 1, 2, \dots, M$ **do**
 - 5: Select next point x_{k+1} by maximizing acquisition function $a(x)$ associated with Gaussian process on $I(h_i(q), g_i(d_i^*(x)))$
 - 6: Evaluate $I(h_i(q), g_i(d_i^*(x)))$ at x_{k+1} .
 - 7: Update Gaussian process on $I(h_i(q), g_i(d_i^*(x)))$ with (I_{k+1}, x_{k+1}) .
 - 8: **end for**
 - 9: Select the point x with maximum value among $M + K$ evaluations as the i th sensor’s location.
 - 10: **end for**
-

Note that sensor locations are selected greedily by running simulations without any real data. Once observation data are collected, sensors can either stay still or move around. If mobile sensors are used, Algorithm. 1 need to be conducted after each Bayesian update.

3.4 EXPERIMENTS

In this section, we will conduct several experiments to assess the performance of the proposed methodology. In the first part, lower bound of mutual information is

compared with calculating mutual information directly. The idea behind this is that if a variable is more dependent on one variable than another, the mutual information and its lower bound should reflect the same trend. In the second part, a chemical release accident is simulated and sensors are placed to infer the release location. Sensor placement will be done by comparing lower bound of mutual information. And Bayesian optimization will be applied to facilitate the computation.

3.4.1 CONSISTENCY OF LOWER BOUND

SIMULATED DATA

Here we generate samples from two functions shown below.

$$y = x_1 + x_2 + \dots + x_m \quad (3.34)$$

$$y = x_1 + x_2 * x_1 + x_3 * x_2 + \dots + x_m * x_{m-1} \quad (3.35)$$

$x_i, i = 1, 2, \dots, m$ follows normal distribution with mean evenly distributed in $[0, 1]$ and standard deviation evenly distributed in $[0.1, 1]$,

$$x_i \sim \mathcal{N}\left(\frac{i-1}{m}, \left(\frac{9(i-1)}{10m} + 0.1\right)^2\right).$$

In order to compare mutual information between y and $X = [x_1, x_2, \dots, x_m]$ with different dependency, we first standardize X and then add noise ϵ_i on each x_i . Since mutual information is invariant to scaling, we can compute the scaled version directly without scaling back.

$$I(y; [x_1^* + \epsilon_1, x_2^* + \epsilon_2, \dots, x_m^* + \epsilon_m])$$

Here the superscript asterisk denotes standardized version. ϵ_i follows normal distribution $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and by increasing the value of σ^2 , mutual information between y and X is supposed to decrease.

We use both kNN methods to calculate mutual information and compare the proposed lower bound (CCA kNN) with mutual information in full dimensions in

different scenarios where different value of m and different number of samples are used. $k = 4$ is used for kNN method. We change the value of σ from 0.1 to 0.5 and run the experiments in 100 trials. The average value and standard error are shown in Table. 3.1 and Table. 3.2. Here *full kNN 1* and *full kNN 2* denote results given by Eq. (3.18) and Eq. (3.19) respectively.

For Eq. 3.34, analytic value of mutual information is accessible and is also shown in the table. As we can see, for a low dimensionality problem such as $m = 5$, both lower bound and two full kNN methods can give plausible results. They all decrease as σ increases and the value become significant smaller when σ grows bigger. However, when $m = 100$, results given by Eq. (3.18) become negative. Eq. (3.18) does provide better estimation in high dimensions. But there is no significant drop in the results when σ increases. On the other hand, the lower bound decreases significantly as dependency become weak. One may notice that, when $m = 100$ and sample size is 200 or 500, some lower bounds are slightly larger than the true values. This is due to small sample size and numerical errors in kNN. As we can see, when sample size comes to 1000, the lower bounds are all smaller than the true values.

As is shown in the table, calculating kNN directly in high dimensions always gives bad estimation. For Eq. 3.35, since it is nonlinear, we cannot get the analytic value of mutual information. However, the results are pretty similar to those of Eq. 3.34. The proposed lower bound gives more consistent performance than calculating kNN in high dimensions.

REAL DATA SETS

Here we work on 4 real data sets from UCI machine learning repository. The description of the data sets is shown in Table. 3.3. Two of the data sets are multivariate and the other two are univariate. Except *Combined cycle power plant*, the other three have small data size and *Residential building* has a large number of attributes. As

Table 3.1: Average and standard error over 100 trials for Eq. 3.34

m	No_smps	σ	full kNN 1	full kNN 2	CCA kNN	Truth
5	200	0.1	0.3016 \pm 0.0020	1.2059 \pm 0.0068	1.9123 \pm 0.0047	2.3076
		0.2	0.2738 \pm 0.0022	1.1696 \pm 0.0068	1.3929 \pm 0.0060	1.6290
		0.3	0.2353 \pm 0.0025	1.1301 \pm 0.0073	1.0687 \pm 0.0075	1.2470
		0.4	0.1971 \pm 0.0028	1.0672 \pm 0.0064	0.8389 \pm 0.0064	0.9905
		0.5	0.1558 \pm 0.0031	0.9983 \pm 0.0065	0.6765 \pm 0.0070	0.8047
	500	0.1	0.4841 \pm 0.0013	1.4488 \pm 0.0052	2.1138 \pm 0.0038	2.3076
		0.2	0.4441 \pm 0.0015	1.3926 \pm 0.0054	1.5088 \pm 0.0045	1.6290
		0.3	0.3913 \pm 0.0016	1.3227 \pm 0.0042	1.1573 \pm 0.0040	1.2471
		0.4	0.3308 \pm 0.0021	1.2331 \pm 0.0047	0.9088 \pm 0.0042	0.9905
		0.5	0.2698 \pm 0.0022	1.1507 \pm 0.0047	0.7238 \pm 0.0043	0.8047
	1000	0.1	0.6200 \pm 0.0009	1.6092 \pm 0.0034	2.1849 \pm 0.0028	2.3076
		0.2	0.5680 \pm 0.0010	1.5290 \pm 0.0033	1.5502 \pm 0.0031	1.6290
		0.3	0.5000 \pm 0.0012	1.4403 \pm 0.0036	1.1893 \pm 0.0031	1.2471
		0.4	0.4219 \pm 0.0013	1.3378 \pm 0.0034	0.9398 \pm 0.0027	0.9905
		0.5	0.3483 \pm 0.0017	1.2464 \pm 0.0032	0.7516 \pm 0.0030	0.8047
100	200	0.1	-0.2953 \pm 0.0004	0.5782 \pm 0.0071	2.1406 \pm 0.0061	2.3076
		0.2	-0.2952 \pm 0.0004	0.5618 \pm 0.0058	1.6708 \pm 0.0071	1.6290
		0.3	-0.2962 \pm 0.0004	0.5593 \pm 0.0074	1.3723 \pm 0.0076	1.2471
		0.4	-0.2961 \pm 0.0004	0.5514 \pm 0.0067	1.1412 \pm 0.0073	0.9905
		0.5	-0.2969 \pm 0.0004	0.5553 \pm 0.0061	0.9784 \pm 0.0080	0.8047
	500	0.1	-0.2864 \pm 0.0003	0.6127 \pm 0.0043	2.1966 \pm 0.0049	2.3076
		0.2	-0.2867 \pm 0.0003	0.6150 \pm 0.0046	1.5949 \pm 0.0048	1.6290
		0.3	-0.2861 \pm 0.0003	0.6153 \pm 0.0046	1.2533 \pm 0.0044	1.2471
		0.4	-0.2873 \pm 0.0003	0.6013 \pm 0.0045	1.0001 \pm 0.0044	0.9905
		0.5	-0.2886 \pm 0.0003	0.5992 \pm 0.0040	0.8326 \pm 0.0048	0.8047
	1000	0.1	-0.2802 \pm 0.0002	0.6373 \pm 0.0034	2.2334 \pm 0.0033	2.3076
		0.2	-0.2810 \pm 0.0002	0.6380 \pm 0.0029	1.6037 \pm 0.0035	1.6290
		0.3	-0.2813 \pm 0.0002	0.6281 \pm 0.0031	1.2280 \pm 0.0030	1.2471
		0.4	-0.2821 \pm 0.0002	0.6325 \pm 0.0035	0.9846 \pm 0.0028	0.9905
		0.5	-0.2833 \pm 0.0002	0.6230 \pm 0.0032	0.8019 \pm 0.0034	0.8047

is done in previous experiments, noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is added to each attribute after standardization and σ increases from 0 to 0.4. Here we start with 0 noise, because the real data set is already noisy. Each time 75% data are randomly selected, and it is repeated 40 times. The average value and standard error are shown in Table. 3.4.

As we can see, the lower bound works well in most cases except that in *CSM* data set, when $\sigma = 0.3$, the result is not significant different from that when $\sigma = 0.2$ or $\sigma = 0.4$. However, the results of $\sigma = 0$ and $\sigma = 0.2$ are significantly different which means for this data set, if the difference between dependencies is large, the lower bound can still be used for comparison purpose. On the other hand, full kNN only works well in data set *Combined cycle power plant* which has large data size and

Table 3.2: Average and standard error over 100 trials for Eq. 3.35

m	No_smps	σ	full kNN 1	full kNN 2	CCA kNN
5	200	0.1	0.1934 ± 0.0032	1.1435 ± 0.0057	0.8871 ± 0.0077
		0.2	0.1621 ± 0.0032	1.1180 ± 0.0068	0.7451 ± 0.0073
		0.3	0.1338 ± 0.0029	1.0650 ± 0.0072	0.6217 ± 0.0072
		0.4	0.1066 ± 0.0029	1.0199 ± 0.0073	0.5173 ± 0.0053
		0.5	0.0660 ± 0.0029	0.9735 ± 0.0069	0.4230 ± 0.0056
	500	0.1	0.3568 ± 0.0022	1.3747 ± 0.0043	0.9950 ± 0.0048
		0.2	0.3225 ± 0.0021	1.3301 ± 0.0050	0.8416 ± 0.0049
		0.3	0.2714 ± 0.0021	1.2489 ± 0.0051	0.6950 ± 0.0040
		0.4	0.2265 ± 0.0023	1.1876 ± 0.0048	0.5830 ± 0.0042
		0.5	0.1772 ± 0.0025	1.0947 ± 0.0050	0.4865 ± 0.0043
	1000	0.1	0.4800 ± 0.0014	1.5218 ± 0.0039	1.0327 ± 0.0042
		0.2	0.4348 ± 0.0015	1.4709 ± 0.0036	0.8710 ± 0.0032
		0.3	0.3752 ± 0.0015	1.3783 ± 0.0036	0.7250 ± 0.0031
		0.4	0.3108 ± 0.0017	1.2843 ± 0.0035	0.6035 ± 0.0031
		0.5	0.2507 ± 0.0017	1.1992 ± 0.0035	0.5044 ± 0.0028
100	200	0.1	-0.2953 ± 0.0004	0.5508 ± 0.0065	0.9422 ± 0.0080
		0.2	-0.2960 ± 0.0004	0.5528 ± 0.0064	0.8937 ± 0.0069
		0.3	-0.2963 ± 0.0005	0.5413 ± 0.0060	0.8365 ± 0.0073
		0.4	-0.2969 ± 0.0004	0.5608 ± 0.0080	0.7689 ± 0.0082
		0.5	-0.2973 ± 0.0004	0.5380 ± 0.0068	0.7045 ± 0.0080
	500	0.1	-0.2866 ± 0.0003	0.6070 ± 0.0042	0.8014 ± 0.0050
		0.2	-0.2873 ± 0.0003	0.6043 ± 0.0043	0.7496 ± 0.0041
		0.3	-0.2878 ± 0.0003	0.5939 ± 0.0042	0.6847 ± 0.0043
		0.4	-0.2892 ± 0.0003	0.6023 ± 0.0051	0.6074 ± 0.0039
		0.5	-0.2900 ± 0.0003	0.5994 ± 0.0047	0.5423 ± 0.0039
	1000	0.1	-0.2809 ± 0.0002	0.6312 ± 0.0029	0.7738 ± 0.0028
		0.2	-0.2817 ± 0.0003	0.6295 ± 0.0032	0.7245 ± 0.0030
		0.3	-0.2822 ± 0.0002	0.6271 ± 0.0031	0.6567 ± 0.0028
		0.4	-0.2832 ± 0.0002	0.6183 ± 0.0031	0.5790 ± 0.0031
		0.5	-0.2842 ± 0.0003	0.6163 ± 0.0031	0.5051 ± 0.0029

small dimensionality. For all the other three data sets, the results given by full kNN cannot be used for comparison.

From all these experiments, we can see that we can only use the estimation given by full kNN when the dimensionality is low and data size is not small. However, the proposed lower bound gives consistent performance and it works well for comparison purpose.

3.4.2 CHEMICAL RELEASE SIMULATION

In this section, a chemical release accident is simulated. The accident occurs in a pipeline and the plume advects and diffuses over the affected area. Sensors are placed

Table 3.3: Data set description

Data set	Size	No. of attributes	No. of outputs
Concrete slump test	103	7	3
CSM	217	11	1
Residential building	372	103	2
Combined cycle power plant	9568	4	1

to locate the release source. In this simulation, sensor locations are fixed during the experimental period and observation data are collected with a fixed time interval. At the same time, Bayesian inference is performed, where the estimation of release location is updated at each time point. The performance of the proposed method is assessed by looking at the uncertainty of the posterior distribution, which is quantified by entropy. We also compare sensor locations selected by Bayesian optimization and those selected on a prefixed grid.

Table 3.4: Average and standard error over 40 times for real data sets

Data set	σ	full kNN 1	full kNN 2	CCA kNN
Residential building	0.	-0.1322 ± 0.0037	1.2288 ± 0.0105	2.1211 ± 0.0094
	0.1	-0.0051 ± 0.0022	1.3410 ± 0.0079	1.8525 ± 0.0092
	0.2	-0.0156 ± 0.0023	1.3091 ± 0.0092	1.6276 ± 0.0073
	0.3	-0.0412 ± 0.0025	1.2828 ± 0.0097	1.4841 ± 0.0079
	0.4	-0.0670 ± 0.0021	1.2841 ± 0.0097	1.3865 ± 0.0066
Combined cycle power plant	0.	1.2957 ± 0.0009	1.2961 ± 0.0008	1.3002 ± 0.0010
	0.1	1.1976 ± 0.0009	1.1681 ± 0.0009	1.2233 ± 0.0013
	0.2	1.0236 ± 0.0008	0.9786 ± 0.0010	1.0993 ± 0.0012
	0.3	0.8900 ± 0.0010	0.8384 ± 0.0010	0.9807 ± 0.0015
	1.4	0.7816 ± 0.0011	0.7262 ± 0.0013	0.8762 ± 0.0015
Concrete slump test	0.	-0.0176 ± 0.0056	0.8124 ± 0.0130	0.8435 ± 0.0095
	0.1	0.0144 ± 0.0061	0.8482 ± 0.0129	0.7645 ± 0.0104
	0.2	0.0124 ± 0.0067	0.8284 ± 0.01247	0.6764 ± 0.0094
	0.3	0.0015 ± 0.0068	0.7940 ± 0.0112	0.5314 ± 0.0118
	0.4	0.0024 ± 0.0077	0.7860 ± 0.0182	0.4357 ± 0.0127
CSM	0.	-0.1903 ± 0.0039	0.7843 ± 0.0114	0.3611 ± 0.0069
	0.1	-0.1063 ± 0.0029	0.8462 ± 0.0113	0.3275 ± 0.0070
	0.2	-0.1041 ± 0.0032	0.8415 ± 0.0099	0.3086 ± 0.0063
	0.3	-0.1306 ± 0.0030	0.8634 ± 0.0092	0.3041 ± 0.0071
	0.4	-0.1287 ± 0.0036	0.8618 ± 0.0097	0.2854 ± 0.0068

DISPERSION MODEL

In this simulation, we adopt a 2D Gaussian puff based model from [68, 76] where the model is used for studying data assimilation in atmospheric dispersion. The chemical material sequentially released at the source is represented by a series of circular puffs. The advection and diffusion of the plume is decided by meteorological conditions. The concentration of each puff has a Gaussian-shape distribution and the concentration at each spacial point is the summation of contributions from all the puffs. For simplicity, deposition and puff splitting is ignored.

Each puff is characterized by three state variables: the center of the puff (X, Y) , the radius r and the mass Q . The advection is decided by the wind at the puff center and the radius of the puff is computed based on Pasuill parameterization [68, 76]. The dynamics of the k th puff at time t_i is as follows:

$$X_k(t_i) = X_k(t_{i-1}) + W_{spd} \cos(W_{dir}) \Delta T \quad (3.36)$$

$$Y_k(t_i) = Y_k(t_{i-1}) + W_{spd} \sin(W_{dir}) \Delta T \quad (3.37)$$

$$S_k(t_i) = S_k(t_{i-1}) + W_{spd} \Delta T \quad (3.38)$$

$$r_k(t_i) = p_y S_k(t_i)^{q_y} \quad (3.39)$$

$$Q_k(t_i) = Q_k(t_{i-1}) \quad (3.40)$$

where W_{spd} and W_{dir} denote wind speed and wind direction respectively. ΔT is the time interval from t_{i-1} to t_i and $S_k(t_i)$ is the distance the k th puff has transported in ΔT . p_y and q_y are Karlsruhe-Jülich diffusion coefficients [68, 76] which specifies meteorological conditions.

For the Gaussian shaped concentration, the mean is the puff center (X, Y) and the standard deviation is the radius r . At each spatial point (x, y) , the concentration is computed by summing up contributions of all the puffs

$$c_{(x,y)}(t_i) = \sum_{k=1}^K \frac{Q_k(t_i)}{2\pi(r_k(t_i))^2} \exp\left(-\frac{(X_k(t_i) - x)^2 + (Y_k(t_i) - y)^2}{2(r_k(t_i))^2}\right) \quad (3.41)$$

where K is the number of puffs released by the time point t_i . Fig. 3.1 shows the advection and diffusion of one Gaussian puff. Note that the color bar of each puff is different. In fact, due to the diffusing process, the average concentration is decreasing.

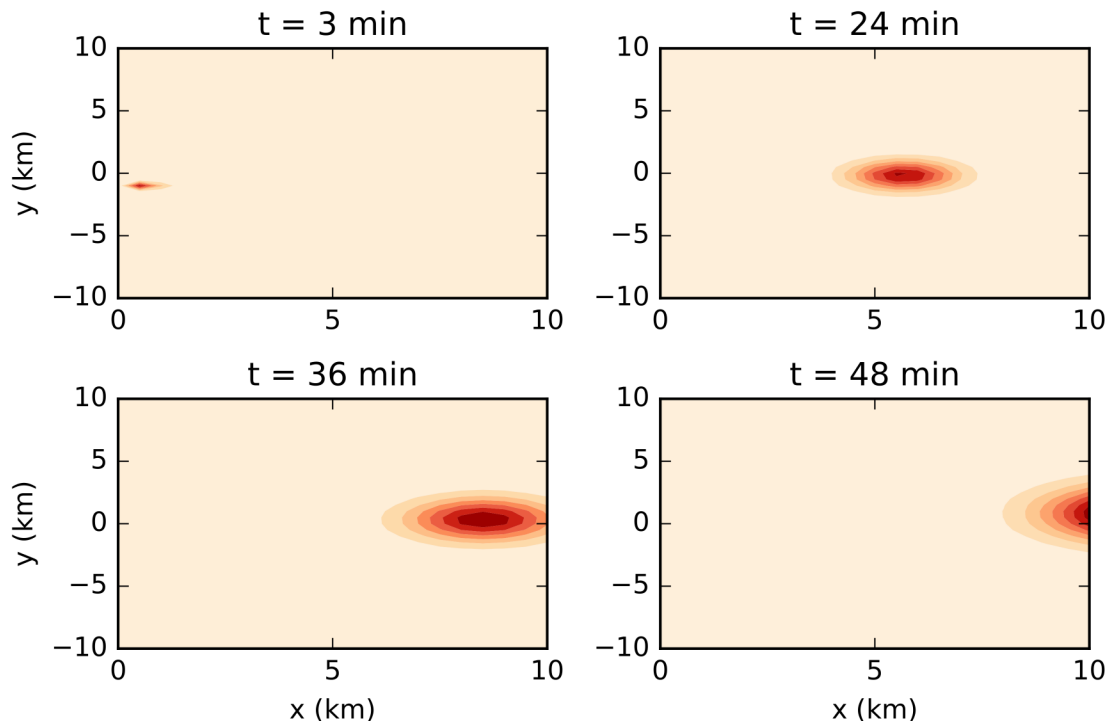


Figure 3.1: The dynamics of one puff. This figure shows the concentration surface of one puff at different time. The release occurs at $t = 0$ and the release location is $(0, -1163.5 \text{ km})$. The wind direction is 0.17 m/s and the wind speed is 4 m/s . p_y and q_y are 0.466 and 0.866 respectively. Note the color bar is different for each plot.

The following measurement model is used to relate model predictions with measurement data for each of the N sensors.

$$\ln(d_j) = \ln(c_j) + \epsilon_{meas_j} \quad j = 1, 2, \dots, N .$$

$$\epsilon_{meas_j} \sim \mathcal{N}(-0.005, 0.1^2)$$

Here c_j represents the concentration at the location of sensor j and d_j is the corresponding sensor reading.

SIMULATION SETTINGS

The affected area is a domain of $10 \times 20 \text{ km}^2$. The pipeline is from $(0, -3\text{km})$ to $(0, 3\text{km})$. The release accident can occur at any location along the pipeline. For simplicity, we assume that there is only one release source. For comparison purpose, we also grid the domain and select sensor locations on the grid points. The domain is shown in Fig. 3.2.

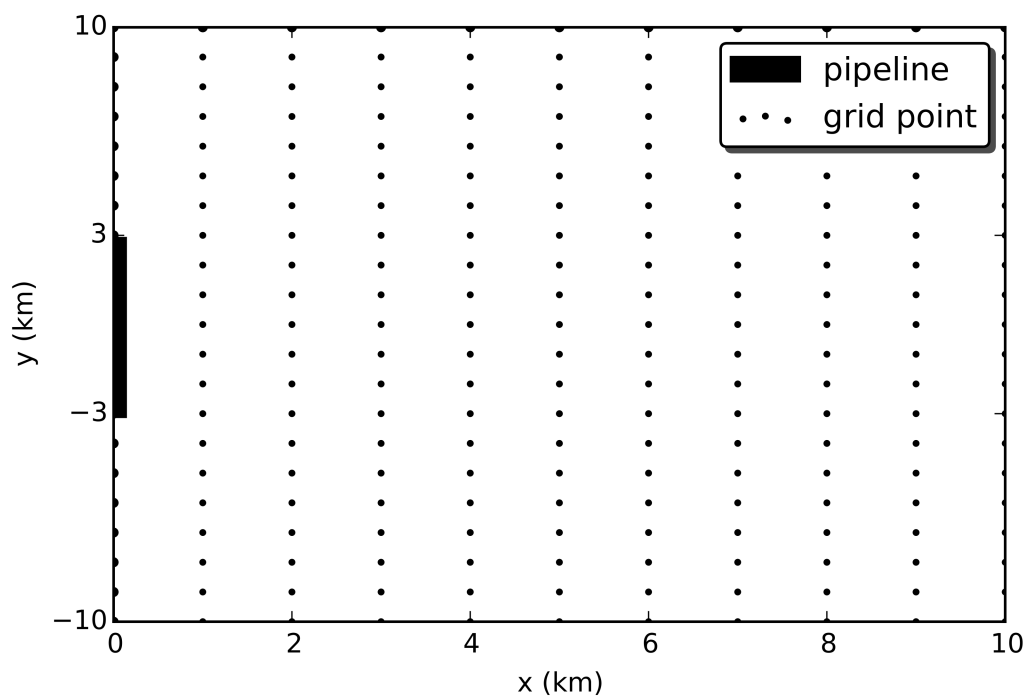


Figure 3.2: The grid and release location. The domain is grided by 11×21 . The pipeline is on the y axis, ranging from -3 km to 3 km.

The total simulation time is 30 *min* and the sampling interval is 1 *min*. Data collection starts from t_1 , since in the beginning there are no chemicals in the sensing area. Each time after the data is collected, Bayesian inference is performed. For the first 10 *min*, a series of Gaussian puffs are released every 1 *min* at the source from t_0 . The wind is from west with 10° standard deviation and the wind speed is 4 *m/sec*. In this simulation, there are two unknown model parameters, release location and wind

direction which both need to be inferred given measurement data. Although release location is the main concern, here we will include both parameters in q .

SELECTION OF SENSOR LOCATIONS

Suppose N sensors need to be placed to collect data, we use D to denote the whole vector of observation data, then $D = \{d_1, d_2, \dots, d_N\}$. As discussed before, optimal sensor locations can be selected by computing $I(D; q|x)$. However, the optimal sensor locations are very likely to change as the plume advects and diffuses over the district. Assume sensor data is collected at a series of discrete time t_1, t_2, \dots, t_M , then for each sensor $i, i = 1, 2, \dots, N$, $d_i = [d_i^{t_1}, d_i^{t_2}, \dots, d_i^{t_M}]$. Therefore, x which will provide the overall most information about q should be selected in the following way:

$$x^* = \arg \max_{x \in \mathcal{X}} I(d_1, d_2, \dots, d_N; q|x) . \quad (3.42)$$

There are several issues in computing Eq. (3.42). First, as we know, each d_i is an M dimensional vector, which means $I(d_1, d_2, \dots, d_N; q|x)$ needs to be computed in over NM dimensions. Second, since the domain is continuous and we don't have an analytical form of mutual information, it is difficult to select best sensor locations, not mentioning multiple sensors need to be placed at the same time. As discussed in the last section, these problems can be all addressed with our proposed method.

In this simulation, three sensors are placed. Sensor readings are collected during the simulation period to infer q . Since sensors are placed before any data is collected, which means the optimal sensor placement should provide most information about the release location in average sense for all possible initial conditions (release location and wind direction). The initial 1000 ensemble members are generated at random for the release location and the wind direction from a uniform distribution over the length of the pipeline and $\mathcal{N}(0, 10^\circ)$ respectively.

Three sensor locations are selected according to Algorithm 1. The three locations selected are (4.8 km, -2.8 km), (2.8 km, 2.6 km) and (3.4 km, 4.1 km), which are shown

in Fig. 3.3, represented by green diamonds. We also computed mutual information over the grid, the heat map in Fig. 3.3 shows the value of mutual information at each grid point. Three locations where mutual information has largest value are selected, represented by blue diamonds. We can see that sensor locations selected by both methods are spatially close.

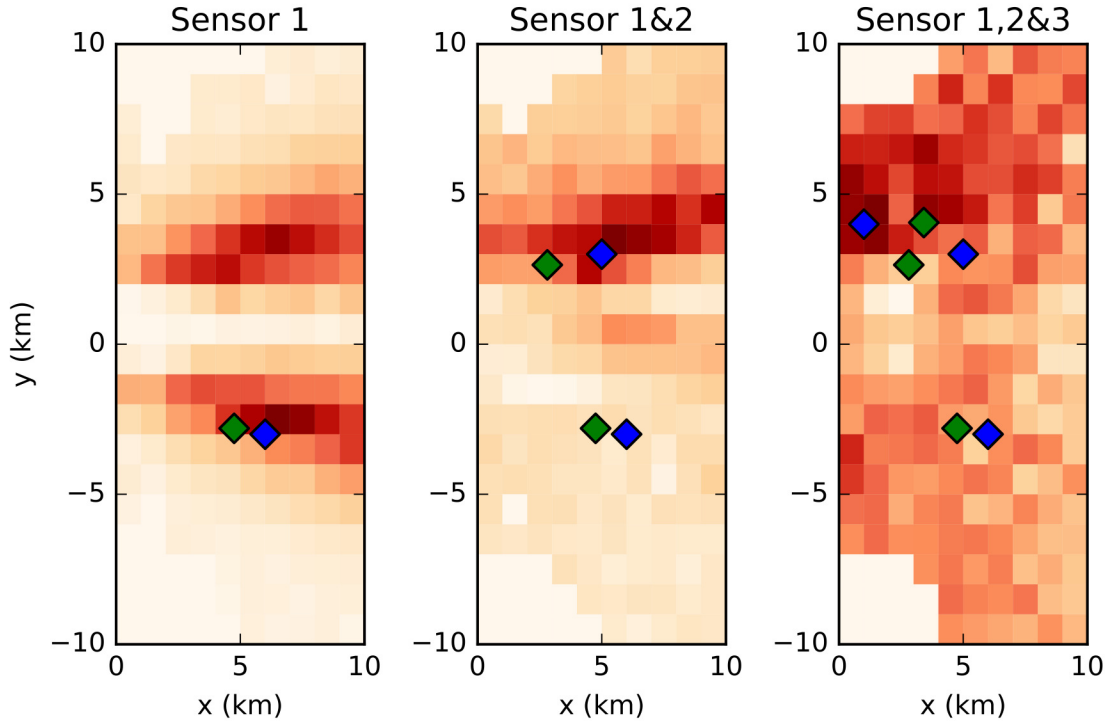


Figure 3.3: Mutual information surface and sensor locations. Three sensor locations are selected in a greedy way. The heat map show mutual information $I(d_1, t; q)$, $I(d_1, d_2, t; q)$ and $I(d_1, d_2, d_3, t; q)$ at each grid point. Blue diamonds represent sensor locations selected over grid points, while green ones are selected by Bayesian optimization. Note the color bar for each plot is different.

RESULT ANALYSIS

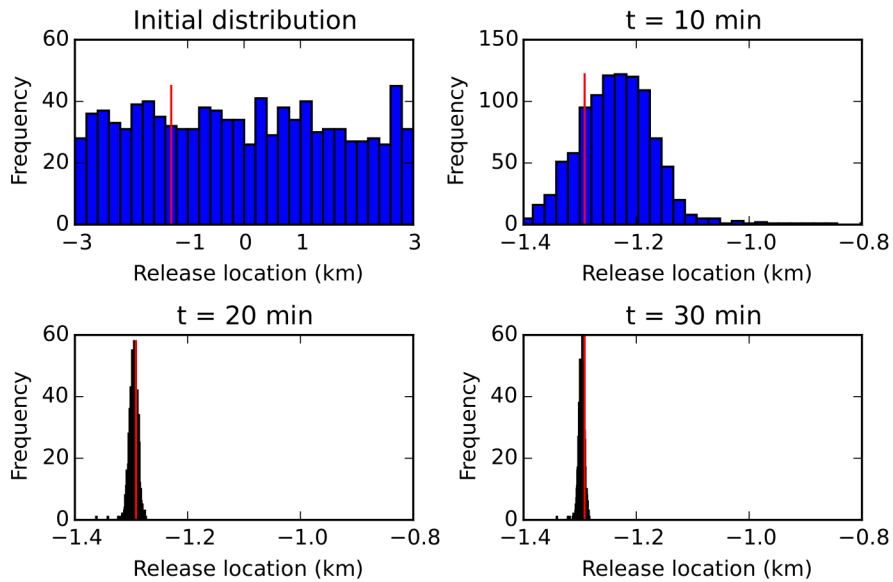
Estimation of Release Location After sensor data is collected and Bayesian inference is performed at each time point to estimate the release location. In this simulation, EnKF is used for Bayesian inference. How to use EnKF in our problem is detailed in Appendix. Here we simulate a release accident and see how the posterior

distribution of the release location changes. The real release location is (0, -1291.7 m), where the wind direction is -0.026 rad, both of which are randomly selected from their distributions. Observation data is collected and the distribution of the release location is updated at each time point from t_1 to t_{31} . Fig. 3.4 shows the initial prior distribution and posterior distributions after update at t_{11} , t_{21} and t_{31} . It is obvious that the real release location is captured by the posterior distribution and the uncertainty is decreasing as more data are collected. One may note that the change in uncertainty from t_0 to t_{11} is much bigger than that from t_{21} to t_{31} , which means most uncertainty is reduced in the first several updates. Due to the measurement noise and computational approximations, the uncertainty of release location can never be eliminated but might converge to a small level.

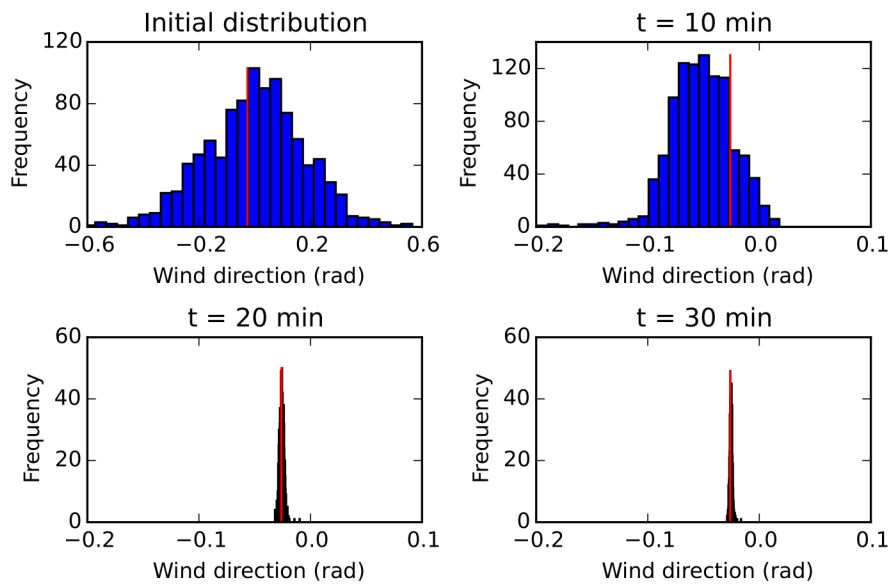
Comparison with Other Placements Here, we compare the selected sensor locations with 20 random placements under 50 different initial conditions. Each initial condition is a combination of release location and wind direction both of which are randomly sampled from their distributions. The performance is measured by conditional entropy. For each placement, the entropy of q is given by $H(q|\xi = \xi_i, x = x_j)$. Here, x represent the sensor location and the subscript j denotes a specific placement. ξ is initial condition and ξ_i implies a particular initial condition. To compare average performance under different initial conditions, conditional entropy $H(q|\xi, x = x_j)$ is computed by Eq. (3.43).

$$H(q|\xi, x = x_j) = \sum_{\xi_i} p(\xi = \xi_i)H(q|\xi = \xi_i, x = x_j) \quad (3.43)$$

Fig. 3.5 shows the average performance of each placement over 50 different initial conditions. We can see that mutual information via grids and mutual information via Bayesian optimization have similar performance, and they both outperforms most of random placements. They not only lead to smaller uncertainty in posterior distribution but also show a faster reduction in uncertainty.

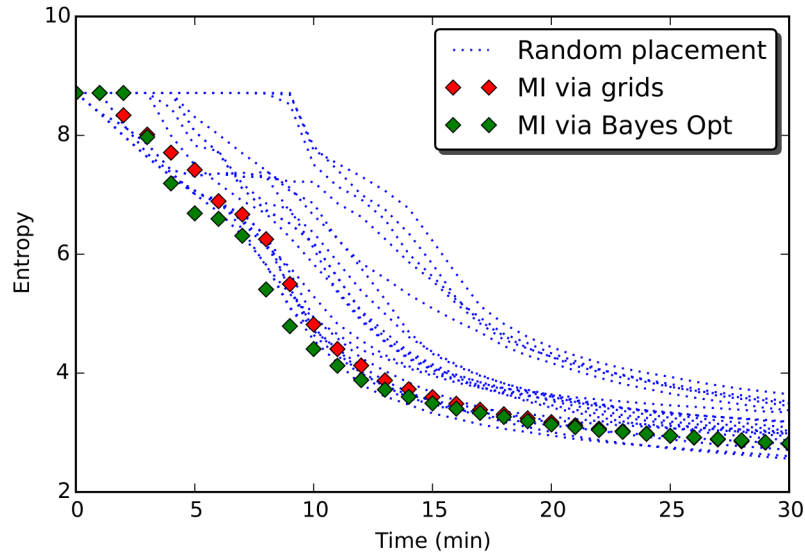


(a) Prior & posterior distribution of release location

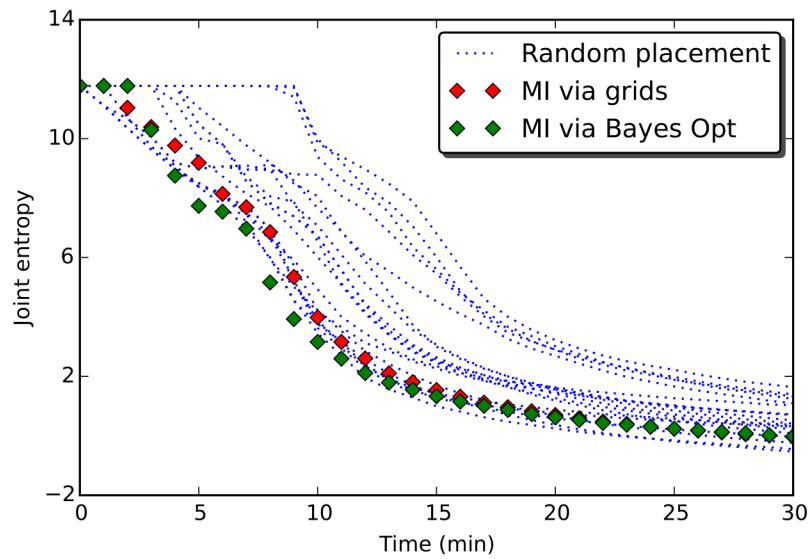


(b) Prior & posterior distribution of wind direction

Figure 3.4: Distributions of parameters. The histograms in Figure (a) are prior distribution and posterior distribution of release location at each time step. Figure (b) shows corresponding distribution of wind direction. The real release location is (0,-1291.7 km) and the real wind direction is -0.026 rad. Both of them are represented by red vertical lines.



(a) Entropy of release location



(b) Joint entropy

Figure 3.5: Results comparison. These two figures compare the performance of proposed strategy with 20 random sensor placements. Here, entropy is used to measure the uncertainty. Figure (a) shows entropy of release location after each update and Figure(b) shows joint entropy of release location and wind direction

In this paper, 3 sensors are placed to illustrate Algorithm. 1 and test the performance of the proposed methods. Since sensors are placed in a greedy manner before making any observations, It can be easily scaled up. Although more sensors means higher dimensionality in mutual information, we only need to compute its lower bound in 2d. Thus, the main cost is running simulations. Since Bayesian optimization has brought down the number of evaluations, only a small number simulations are needed in each run.

3.5 CONCLUSIONS

In this paper, we address the sensor placement problem where certain QoI need to be inferred from observation data. The QoI often include model parameters and/or state variables but can also be other quantities. Inferring QoI usually involves solving inverse problem which is formulated in Bayesian framework in this paper. On the other hand, since data collected at different locations are likely to provide different amount of information, sensors should be strategically placed.

Mutual information is one of the most common used criterion to guide the sensor placement. It naturally quantifies the dependence between two variables and has been widely discussed in the literature. However, computing mutual information is challenging and the estimation in high dimension is always unreliable. Thus we propose a novel approach, which compute the lower bound of mutual information in only two dimension. The approach is based on data inequality processing and canonical correlation analysis. It projects observation and QoI into two dimension where the projections have largest correlation. This lower bound of mutual information is shown to be effective as a metric to select sensor locations. In addition, we apply Bayesian optimization to deal with continuous domain. We place Gaussian prior on the metric, which generates a mutual information surface. Then evaluations are made according to acquisition function. In this way, the number of evaluations is greatly reduced.

A chemical dispersion accident is simulated and it shows that the proposed approach outperforms random sensor placements by offering an obviously faster reduction in uncertainty of QoI. Also the sensor locations selected by Bayesian optimization are close to those by discretizing the domain into a fine grid, but with a considerably less number of evaluations. The proposed approach is promising to address a vast range of sensor placement problems.

CHAPTER 4

BAYESIAN MODEL SELECTION FRAMEWORK FOR IDENTIFYING GROWTH PATTERNS IN FILAMENTOUS FUNGI

4.1 INTRODUCTION

Mycelial expansion of filamentous fungi has an enormous impact on agriculture, animal and human health and environmental sustainability [69, 39, 35, 6]. Hence, understanding how a mycelium expands in different environments is of great practical significance. The process is extremely complicated and is orchestrated by several basic physical and biochemical mechanisms [58]. A vast number of models have been developed in the past decades to describe mycelial growth. Generally, these models can be categorized into two groups: continuous and discrete. A continuous model, appearing in the form of differential equations, usually describes mycelial network by its average properties, such as biomass density, hyphal density and so on. The model proposed by Edelstein [29] for example, falls into this category and is the basis of many later models. This is a general model, which consists of various branching and anastomosis and could be calibrated for different species. However, the model is based on the assumption that unlimited nutrient is supplied and does not include the interaction between fungi and its environment. Later work [30, 24] improved this model by introducing processes such as uptake and translocation. Boswell [9] further incorporated the hyphal division into active and inactive according to whether they

are involved in the translocation of internal metabolites. This model was shown to be applicable to the nutritionally heterogeneous environments. Boswell and Davidson later developed several well-developed continuous models [7]. Unlike the continuous models that are based on averaging the fungal properties, discrete models consider the growth of each hypha and can produce a realistic visualization of fungal mycelium [8]. However, the construction of such a complex mycelial network is computationally expensive and it is much more difficult to include various processes into a discrete model than into a continuous model.

Mathematical structures that describe the complexity of these fungal growth patterns vary greatly. Thus, it is important to have access to a computational tool capable to rank alternative models in the light of new experimental data. This raises the issue of model selection, which is significant in several aspects. First, mycologists can identify the growth model that best describes the experimental data. This can determine the dominant growth patterns for various fungi species grown in diverse environments; these include the study of fungal growth patterns in presence of antifungal drugs. Second, as explained above, developing a new model is a nontrivial task. This consists of identifying dominant growth patterns, describing them mathematically, and validating whether the new model has improved descriptive power over its predecessors. Choosing from existing models on the other hand, can save time and energy in quickly identifying dominant growth patterns and guide the development of the new model. In addition, for a series of models derived from the same predecessor, the more complex ones may not always be the better choice than the simpler ones. According to Occam's razor, a more complex model should only be chosen if it offers a significant improvement [60]. This principle should direct the development of a new model, when additional processes are incorporated. Thus rigorous model selection can be an important tool in studying fungal growth.

The most straightforward way to evaluate the performance of a model is by look-

ing at how well it fits the observational data using measures such as mean square error. However, in the context of comparing different models, these type of measures are generally inappropriate. They favor heavily parameterized models able to “fit anything”, leading to over-fitting rather than an improved description of the true biology. This is in contrast with the aim of mathematical modeling, which is to simplify the complex process in the real world so as to study the key properties of the phenomenon of interest [7]. Complicated models capable to mirror the reality are meaningless to researchers as they describe spurious physical processes. Thus, when comparing different models there is more to consider. Namely, a good model selection approach should be a trade-off between data fitness and model complexity.

Two of the most commonly used approaches for model selection are Akaike information criterion (AIC) and Bayesian information criterion (BIC). Both are based on the likelihood function which reflects the how well the model fits the data and both have an additional penalty term for the number of parameters in the model. Penalty term in BIC is positively correlated with the number of observations and generally larger than the penalty in AIC [52, 46]. However, the number of parameters may not reflect the real complexity since different models could have different mathematical structures. Bayesian inference uses relative probabilities to compare the models under consideration. This general measure, which naturally embodies Occam’s razor, favors models that fit the data well while penalizing models that rely heavily on the data to adjust parameters. Smith and Spiegelhalter [72] pointed that under specific conditions, AIC and BIC are two particular cases of Bayesian model selection. In this work, we adopt a full Bayesian approach to the model selection problem.

Recent paper [52, 82] also talked about Bayesian model selection in biological applications. But a very important issue which is always ignored is the existence of model uncertainty. It should be noted that the model selected among a set of candidates may not be the best in describing the real process. It is just better than

the other candidates and it is quite possible that there still is a big discrepancy between its output and the real process. Unfortunately, this model discrepancy is always neglected in biological models, which causes the difficulty of describing real world data and restricts the use of advanced model based approaches. In fact, no model could fully describe the real process. The discrepancy which is due to the missing physics and numerical approximations is inevitable for mathematical models. Bryjarsdóttir [13] illustrates the importance of recognizing model discrepancy by giving a simple example where the model used for simulation is slightly different from the real model that generated observational data. It is shown that the prediction as well as the estimation of the parameters can be biased and over-confident. The inclusion of model discrepancy in the mathematical model is important not only for improving model predictions but also for understanding the deficiencies of the model, which can lead to further improvements. Thus, it is necessary to include model discrepancy in model based studies.

In this chapter, the discrepancy is modeled with parametric structure and calibrated with observation data. We will show that the introduced model discrepancy offers a way to evaluate the predictive capability of the best model in the context of model selection. If the model discrepancy is very large, the best model cannot be used even though it outperforms other candidates. On the other hand, the application of Bayesian framework always involves solving inverse problems, no matter in model calibration, experimental design or model selection. A variety of computational approaches have been developed to solve inverse problems. However, most approaches will fail, when model error is introduced. Thus the computational approaches need to be selected carefully in the presence of model error and this will be discussed later in the paper.

In this chapter, we develop a framework to perform Bayesian model selection as well as quantify the model discrepancy to study fungal growth. The models proposed

by Edelstein [29] are used to illustrate our approach, where different growth patterns are generated and compared, and the discrepancy of each model is quantified. Section 4.2 provides the necessary background and introduces our approach and models of interest. In Section 4.3, the simulation results are demonstrated to be consistent with the aim of the approach and shows promising results for studying fungal growth. In this paper, we only consider continuous models for the purpose of describing our methodology; discrete models can be accommodated in a similar manner.

4.2 METHODS AND MODELS

This section describes the Bayesian model selection framework in the presence of model uncertainty. The framework adopted here consists of the following three stages: (1) uncertainty modeling, (2) model calibration, and (3) model comparison. The methodology is first introduced from a general point of view followed by the computational approach and fungal growth models of interest.

To set notation, consider the following mathematical model for a biological system of interest:

$$\mathcal{R}(u, \theta_{model}) = 0 \tag{4.1}$$

$$y = f(u(\theta_{model})) . \tag{4.2}$$

Here, \mathcal{R} is some operator, u is the solution or the state variable and θ_{model} is a set of model parameters, which usually have a physical interpretation. For the purpose of this paper, Eq. 4.1 is a partial differential equation describing average properties of fungal mycelium in time and space such as hyphal density and tip density, whereas θ_{model} is the set of unobserved growth parameters. In addition, f is a map from the solution to the prediction quantity y that can be compared with experimental measurements D_{meas} . Note, that the model in Eq. 4.2 provides the avenue through which model parameters can be inferred from experimental data. However, there

are several issues that need to be address before carrying out model calibrations and model selections.

4.2.1 UNCERTAINTY MODELING

The main challenge in using computational models along with experimental data for scientific discoveries, such as determining fungal growth patterns, is that the process follows a path contaminated with errors and uncertainties. This includes measurement errors, structural uncertainties, errors due to numerical discretization, and uncertainty in model parameters. This work employs probability to represent uncertainty and Bayesian inference to update the uncertainty of model parameters in light of experimental data.

In general, among all these uncertainties, the most important ones are structural uncertainties that arise due to missing physics in Eq. 4.1. As a result, there is always some discrepancy between the output of the physical model and the values of the real process. A common approach for specifying the structural uncertainty model is that of Kennedy and O’Hagan [49]. In this approach, the true (but unknown) value of the observable, d_{true} , is assumed to be related to the model by,

$$d_{true} = f(u(\theta_{model})) + \epsilon_{struct} , \quad (4.3)$$

where ϵ_{struct} is a statistical model used to represent the structural uncertainty. This discrepancy model has its own parameters, θ_{struct} , called hyper-parameters, which define the probability density function (pdf) of the errors as follows:

$$\epsilon_{struct} \sim p_{struct}(\epsilon_{struct}|\theta_{struct}) . \quad (4.4)$$

Because, the experimental data is noisy due to sensor imprecision, the measurement error ϵ_{meas} usually follows a known pdf $p_{meas}(\epsilon_{meas})$ that is defined by the specifications of the experimental apparatus. This results in the following relation

between the observable d_{meas} and the true value of the observable d_{true} .

$$d_{meas} = d_{true} + \epsilon_{meas} . \quad (4.5)$$

Finally, the relation between the observable d_{meas} and model parameters is given by combining Eq. 4.3 and 4.5. This measurement model, Eq.4.6 defines the likelihood function and Bayes rule can be used to update the knowledge of both physical parameters and hyper-parameters, simply denoted by $\theta = [\theta_{model}, \theta_{struct}]^T$.

$$d_{meas} = f(u(\theta_{model})) + \epsilon_{struct} + \epsilon_{meas} . \quad (4.6)$$

Since Bayes rule is used as the inference engine, then a prior probability distribution needs to be defined for the parameters $\theta \sim p(\theta) = p(\theta_{model})p(\theta_{struct})$. Note that the additive errors introduced in the previous equations are not a requirement; multiplicative errors are possible as well.

To summarize, the statistical model for the biological system of interest is given by:

$$\mathcal{R}(u, \theta_{model}) = 0 \quad (4.7)$$

$$d_{meas} = f(u(\theta_{model})) + \epsilon_{struct} + \epsilon_{meas} \quad (4.8)$$

$$\epsilon_{struct} \sim p_{struct}(\epsilon_{struct}|\theta_{struct}) \quad (4.9)$$

$$\epsilon_{meas} \sim p_{meas}(\epsilon_{meas}) \quad (4.10)$$

$$\theta = [\theta_{model}, \theta_{struct}]^T \quad (4.11)$$

$$p(\theta) = p(\theta_{model})p(\theta_{struct}) . \quad (4.12)$$

So far, we have addressed the structural uncertainty in Eqs. 4.1 and 4.2 by introducing a discrepancy model according to Kennedy and O'Hagan formalism [49]. This arises due to the inadequate form of models in Eqs. 4.1 and 4.2. Without this discrepancy model the prediction as well as the estimation of the parameters can be biased and over-confident [13].

There is also a complementary approach to address structural uncertainty by specifying a set of candidate models, each described by different structural equations. This concept can be extended to build a set of K model classes $\mathcal{M} = \{M_1, M_2, \dots, M_K\}$, where each model class M_i can differ from the rest by at least one definition in the set of equations 4.7-4.12. For example, each model class M_i has its own structural equation \mathcal{R}_i and parameters θ_i and prior distribution over parameters $p(\theta_i|M_i)$. However, all candidate model classes should share the same measurement data, D_{meas} , which is the basis for comparing different models.

Each model class in the set \mathcal{M} is defined using the following mathematical structure, which is similar with the set of equations in Eqs. 4.7-4.12, but each has its own particularities.

$$M_i : \begin{cases} \mathcal{R}_i(u_i, \theta_{model}^i) = 0 \\ d_{meas} = f_i(u_i(\theta_{model}^i)) + \epsilon_{struct}^i + \epsilon_{meas} \\ \epsilon_{struct}^i \sim p_{struct}(\epsilon_{struct}^i | \theta_{struct}^i, M_i) \\ \epsilon_{meas} \sim p_{meas}(\epsilon_{meas}) \\ \theta_i = [\theta_{model}^i, \theta_{struct}^i]^T \\ p(\theta_i|M_i) = p(\theta_{model}^i|M_i)p(\theta_{struct}^i|M_i) \end{cases}, \text{ for } i = 1 \dots K \quad (4.13)$$

In this context, Bayesian analysis is also used to compute the relative model probabilities given experimental data, D_{meas} , which provide the basis for the proposed model comparison. Similar with model parameters, we also need to define the prior probability $p(M_i|\mathcal{M})$ for model class M_i . This is the probability assigned to M_i based on information that is independent of the data. In the presence of complete ignorance, a uniform prior is appropriate, i.e. $p(M_i|\mathcal{M}) = 1/K$. Finally, the set of Eqs. 4.13 and prior probabilities $p(M_i|\mathcal{M})$, provide the necessary abstract mathematical constructs to talk about Bayesian model calibration and Bayesian model selection.

4.2.2 BAYESIAN MODEL CALIBRATION

In Bayesian model calibration, one seeks a complete probabilistic description of model parameters that make the model consistent with the experimental data, D_{meas} . The solution to this problem is the posterior probability density function of model parameters and it is defined by the simple but powerful Bayes's Theorem,

$$p(\theta_i|M_i, D_{meas}) = \frac{p(D_{meas}|\theta_i, M_i)p(\theta_i|M_i)}{p(D_{meas}|M_i)} . \quad (4.14)$$

Here, $p(D_{meas}|\theta_i, M_i)$ is the likelihood function and it measures the agreement between the model output and the data for given values of the input parameters. In the context of the previous section, this is given by,

$$p(D_{meas}|\theta_i, M_i) = p_{conv} \left(D_{meas} - f_i(u_i(\theta_{model}^i)) \middle| \theta_{struct}^i, M_i \right) \quad (4.15)$$

where p_{conv} is given by the convolution of the two pdfs representing model discrepancies and measurement noise, $p_{struct}(\epsilon_{struct}^i|\theta_{struct}^i, M_i)$ and $p_{meas}(\epsilon_{meas})$ respectively.

$$p_{conv}(\epsilon_{conv}|\theta_{struct}^i, M_i) = \int p_{struct}(\epsilon_{struct}^i|\theta_{struct}^i, M_i)p_{meas}(\epsilon_{conv} - \epsilon_{struct}^i)d\epsilon_{struct} \quad (4.16)$$

A common approach is to choose the model for the two pdfs, $p_{struct}(\epsilon_{struct}^i|\theta_{struct}^i, M_i)$ and $p_{meas}(\epsilon_{meas})$ from the Gaussian family of pdfs, where the hyper-parameters θ_{struct}^i are normally used to parameterize the covariance matrix of ϵ_{struct}^i . In this case, the convolution between the two Gaussian pdfs in Eq. 4.16 is easily obtained analytically. This provides a tractable approach to evaluate the likelihood function $p(D_{meas}|\theta_i, M_i)$.

The denominator in Eq.4.14 is called the marginal likelihood or evidence. Overall, this is just a normalization constant that ensures that the solution to the Bayes' inverse problem, $p(\theta_i|M_i, D_{meas})$ is indeed a proper pdf that integrates to one. Nonetheless, this is a key quantity that will allow us to perform Bayesian model selection as described in the next section.

$$p(D_{meas}|M_i) = \int p(D_{meas}|\theta_i, M_i)p(\theta_i|M_i)d\theta_i . \quad (4.17)$$

In general, this integral cannot be computed analytically, which results in an intractable Bayesian inference problem. Thus one has to resort to some sort of approximation as it will be described in Section 4.2.4.

4.2.3 BAYESIAN MODEL COMPARISON AND OCCAM'S RAZOR INTERPRETATION

In the previous step, each model M_i is calibrated separately given the observation data D_{meas} . After model calibration, the uncertainties of both physical parameters and model discrepancy have been quantified for each individual model. It still remains to determine which model provides the best explanation for the observed data. Here, Bayesian inference can be used again to calculate posterior probability of models, which provides the basis for model comparison.

$$p(M_i|D_{meas}, \mathcal{M}) = \frac{p(D_{meas}|M_i)p(M_i|\mathcal{M})}{\sum_{k=1}^K p(D_{meas}|M_k)p(M_k|\mathcal{M})} \quad (4.18)$$

Note that in the presence of complete ignorance, $p(M_i|\mathcal{M}) = 1/K$, the relative posterior probability $p(M_i|D_{meas}, \mathcal{M})$ is determined entirely by the evidence, $p(D_{meas}|M_i)$, which is given by the normalization constant (the denominator) in the calibration phase introduced above. Thus, calculating model posterior probabilities can be easily performed in this step.

Models with corresponding high posterior probabilities are more likely to have generated the data, or in other words, they provide a better representation of the data than the rest of the models. This Bayesian model comparison can be viewed as a natural formalization of Occam's razor [60], which states that given multiple models that explain the data equally well, one should prefer the simplest. This can be easily seen by looking at the decomposition of the log-evidence by rearranging the terms in

Eq. 4.14 and taking the expectation with respect to the posterior distribution.

$$\begin{aligned} \ln p(D_{meas}|M_i) &= \ln p(D_{meas}|\theta_i, M_i) - \ln \frac{p(\theta_i|M_i, D_{meas})}{p(\theta_i|M_i)} \quad (4.19) \\ &= \int p(\theta_i|M_i, D_{meas}) \ln p(D_{meas}|\theta_i, M_i) d\theta_i \\ &\quad - \int p(\theta_i|M_i, D_{meas}) \ln \frac{p(\theta_i|M_i, D_{meas})}{p(\theta_i|M_i)} d\theta_i \end{aligned}$$

We can see that the log-evidence is the trade-off between the expected log-likelihood, which provides a measure of goodness-of-fit (how well the model fits the data), and Kullback-Leibler (KL) divergence [20] between posterior and prior pdfs. The latter is an information theoretic measure, which measures the departure from the prior pdf to the posterior pdf. The larger the divergence the more information the model has extracted from the data. Thus, the KL divergence can be interpreted as a measure of model complexity. Complex models have more degrees of freedom and thus are capable to extract more information from the data.

$$\ln p(D_{meas}|M_i) = \underbrace{E[\ln p(D_{meas}|\theta_i, M_i)]}_{\text{goodness-of-fit}} - \underbrace{D_{KL}(p(\theta_i|M_i, D_{meas})||p(\theta_i|M_i))}_{\text{model complexity}} \quad (4.20)$$

The log evidence embodies a trade-off between how well the model fits the data and how complex the model is. Thus, Bayesian model selection automatically enforces Occam's razor. This goes beyond looking at conventional measures such as root-mean-square-error (RMSE), which need to be used in conjunction with cross-validation methods to avoid over-fitting problems. Given the scarcity of experimental data in some situations, cross-validation is not as feasible as Bayesian framework, which allows models to be compared on the basis of a single data set. In practice, when comparing two different models M_i and M_j , Bayes factor B_{ij} is used to quantify how preferable M_i is against M_j . It is defined as a ratio of two evidences:

$$B_{ij} = \frac{p(D_{meas}|M_i)}{p(D_{meas}|M_j)} \quad (4.21)$$

Usually, Bayes factor of 100 is considered strong in model selection, while Bayes factor of 1-3.2 is “not worth more than a bare mention” [43, 82].

4.2.4 COMPUTATIONAL APPROACH

In this work we adopt numerical sampling techniques, also known as Monte Carlo. Markov Chain Monte Carlo (MCMC) sample-based methods have become the main computational workhorse in scientific computing for sampling from a large class of distributions. The most basic form of MCMC is Metropolis-Hastings (MH) algorithm [62, 4], which generates a sequence of correlated samples that form a Markov chain. The chain is started with an initial sample θ , and a candidate sample θ^* is sequentially generated from a symmetric proposal distribution $q(\theta^*|\theta)$, which usually has a Gaussian form. The candidate sample is more likely to be accepted if it causes an increase in the value of the posterior pdf. The probability of acceptance does not require the evaluation of the marginal likelihood, nonetheless this sampling technique does have its own challenges as described in the next paragraphs.

In MH algorithm, the variance of the proposal affects the distance from the current sample to the candidate sample. If the variance is large, then this will result in large steps from the current sample, which may miss the target distribution and yield low acceptance rates. On the other hand, small proposal variances yield highly correlated samples and the chain will move too slowly. Obviously, MH requires careful tuning of the proposal before using it to generate samples from the posterior distribution. In response, more efficient algorithms have been proposed to address this problem. Delayed Rejection Metropolis (DR) [62] employs a fixed number of proposals that depend on the rejected samples. Delaying rejection decreases the autocorrelation in the chain and forces it to move. On the other hand, Adaptive Metropolis (AM) [37] adapts the variance of the proposal based on the previous samples. Delayed Rejection Adaptive Metropolis (DRAM) algorithm [38] combines DR and AM approaches in a complimentary way that systematically remedies each other's shortcomings.

The most challenging cases for sampling algorithm involve big differences between the support of prior and posterior pdfs. These are due to the overlap between the

likelihood function and prior pdf that occurs in the regions of small value, far away from high density regions. On this occasion, the acceptance rate of DRAM may be low and it is quite likely to end up with highly correlated samples which are away from the high density region of the posterior pdf [17, 21]. Unfortunately, this happens commonly, when structural uncertainty is introduced. Since a small structural error is always assumed at the beginning, model output obtained by propagating prior samples are usually far away from the observation, which means likelihood and prior pdf can only be overlapped in low density regions. In this case, typical HM algorithms including DRAM can not guarantee the convergence of posterior samples, which is likely to cause a failure in model calibration stage.

Transitional MCMC (TMCMC) is an adaptive multilevel sampling method that aims to deal with this problem [4, 17]. Instead of directly sampling from the posterior distribution, it progressively samples intermediate distributions until the target distribution is sampled, see Eq. 4.22. The number of intermediate stages or levels, M , as well as their corresponding exponent, α_j , is automatically adapted based on the target probability of acceptance. The effect of the exponents α_j is to flatten the likelihood function to increase the overlap between the intermediate posterior distributions. Since the change between two adjacent intermediate distributions is small, sampling a new intermediate distribution based on the former one is more efficient. Multilevel sampling has been implemented in the C++ toolbox called QUESO [66], which can be downloaded online¹.

$$p(\theta_i|M_i, D_{meas}) = \frac{\prod_{j=1}^M p(D_{meas}|\theta_i, M_i)^{\alpha_j} p(\theta_i|M_i)}{p(D_{meas}|M_i)} \quad (4.22)$$

$$\sum_{j=1}^M \alpha_j = 1 \quad (4.23)$$

Note that all these numerical algorithms only provide samples from the posterior distribution, without requiring the evaluation of the evidence. However, calculating

¹<https://github.com/libqueso/queso>

the evidence is essential for Bayesian model selection. Given N samples from the prior distribution, a straightforward Monte Carlo estimates of $p(D_{meas}|M_i)$ such as

$$p(D_{meas}|M_i) \approx \frac{1}{N} \sum_{k=1}^N p(D_{meas}|\theta_i^k, M_i) ,$$

yields large variance as the main support of the prior pdf is usually very different from the high likelihood region. The multilevel sampling provides a natural way to estimate the evidence using thermodynamic integration. Namely, the variance of Monte Carlo estimators is significantly smaller at each level due to the flatness of the likelihood function, which provides a larger overlap between the likelihood and prior pdf. Thus, the overall natural logarithm of the evidence, $\ln p(D_{meas}|M_i)$ can be computed by summing over the log-evidences computed at all levels [17, 67].

4.2.5 FUNGAL GROWTH MODELS

Bayesian model selection can be a very useful tool in studying fungal growth. It not only helps researchers select the most suitable model for a particular fungus, but also provides a criterion for researchers to decide whether extra processes should be included when building a new model. For the rest of the paper, we will use an example to illustrate the method. The candidate models we use are generated from Edelstein's model [29], which consists of various types of branching, anastomosis and hyphal death and can be used to calibrate a variety of fungal models. In this model, hyphal growth is described by its average properties: hyphal density and tip density. Since a fungus grows radially and almost uniformly in all directions, only properties along the radial direction are considered.

$$\frac{\partial \rho}{\partial t} = nv - d \tag{4.24}$$

$$\frac{\partial n}{\partial t} = \frac{\partial(nv)}{\partial x} + \sigma . \tag{4.25}$$

Here, x is the distance from the spore center, ρ is hyphal density in units of filament length per unit area, n is tip density (number per unit area) and v is tip

extension rate. Both σ and d are uncertain terms which represent growth pattern as shown in Figure 4.1. Note that σ can be a combination of different branching types (e.g. YH, YFH, YHD etc.). Letters are used here to represent different growth patterns, e.g. YH describes a dichotomous branching and tip-hypha anastomosis. We will use this model to generate candidates which have different branching types, and then compare them via Bayesian model selection given observation data of hyphal density ρ and tip density n .




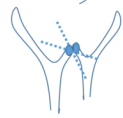

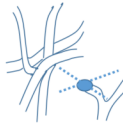

Branching	Biological type	Mathematical version	Symbol
	Dichotomous branching	$\sigma = \alpha n$	Y
	Lateral branching	$\sigma = \alpha_s \rho$	F
	Tip-hypha anastomosis	$\sigma = -\beta_2 n \rho$	H
	Tip-tip anastomosis	$\sigma = -\beta_1 n^2$	W
	Tip death	$\sigma = -\alpha_1 n$	T
	Tip death due to overcrowding	$\sigma = -\beta_3 \rho^2$	X
	Hyphal death	$d = \gamma_s \rho$	D

Figure 4.1: Branching types and corresponding mathematical model

4.3 NUMERICAL RESULTS AND DISCUSSIONS

The approach taken in this paper to demonstrate the effectiveness of the proposed Bayesian model comparison, is to generate synthetic data using the YHD fungal growth model and compare three candidate models YH, YHD and YFH in two particular scenarios. The goal is to showcase that for a particular scenario, the YHD candidate model is not the only model selected as the best representation of the data even though it has been used to generate the measurement data. This selection outcome is supported by the high model complexity level detected by the Bayesian model selection for this particular model and scenario.

4.3.1 SYNTHETIC DATA GENERATION

The data is generated by the following YHD model:

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= v n - \gamma_1 \rho \\ \frac{\partial n}{\partial t} &= -v \frac{\partial n}{\partial x} + \alpha_1 n - \beta_2 n \rho .\end{aligned}$$

The initial condition for this system is given by

$$\begin{aligned}\rho(x, t = 0) &= e^{-x^2/32} \\ n(x, t = 0) &= 0.5e^{(x-6.5)^2/8} ,\end{aligned}$$

and the boundary conditions are given by:

$$\begin{aligned}\rho(x = 0, t) &= 1, \quad \rho(x = 100, t) = 0 \\ n(x = 0, t) &= 0, \quad n(x = 100, t) = 0\end{aligned}$$

The evolution of the solution is computed within the $[0, 5]$ temporal domain and the $[0, 100]$ spatial domain. Two calibration datasets will be generated to showcase how Bayesian inference ranks various candidate models. The difference between the two dataset is given by the value of parameter γ_1 , which affects the ranking of the models.

Dataset (Scenario) I: $\gamma_1 = 0.03$ and Dataset (Scenario) II: $\gamma_1 = 0.3$.

The true values for the rest of the parameters are given by: $\nu = 0.8$, $\alpha_1 = 0.1$, and $\beta_2 = 0.3$. The datasets consists of 10 data points along x axis at $t = 0.5$ and $t = 5$. Figure 4.2 shows the synthetic data points for the first scenario. In the following section, the data at $t = 0.5$ is used to infer the initial condition for candidate models and data at $t = 5$ is used for model calibration.

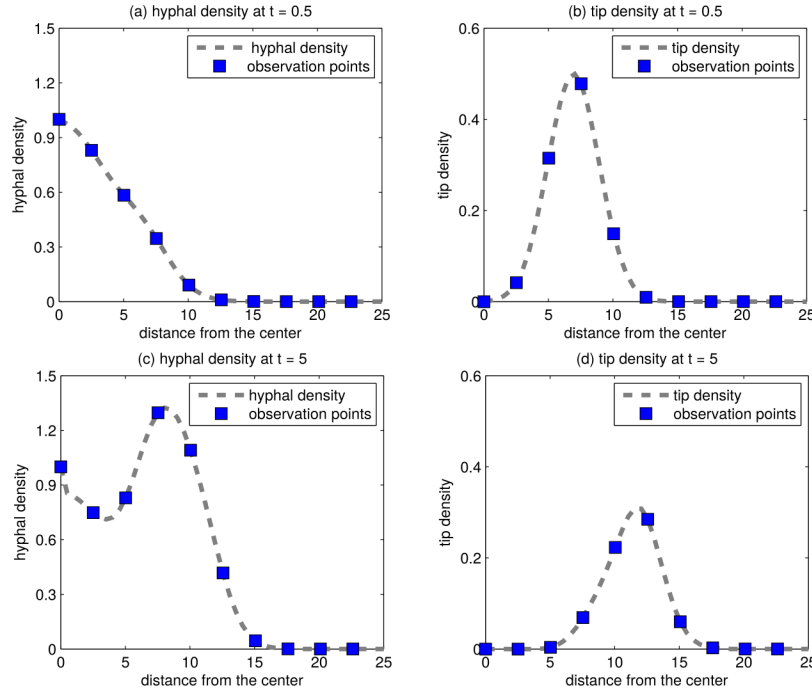


Figure 4.2: Observation data. Ten observation points between 0 and 22.5 are evenly selected along x axis. Observation data at $t = 0.5$ is used to infer the initial condition for candidate models and data at $t = 5$ is used for model calibration.

4.3.2 CANDIDATE MODELS AND UNCERTAINTY MODELING

Three candidate models are proposed here: YH, YHD, and YFH. In this section we provide the formulation of the corresponding three model classes. Thus, we include discrepancy models and prior probabilities for both model parameters and hyper-

parameters.

$$\mathcal{R}_{YH} : \begin{cases} \frac{\partial \rho}{\partial t} = nv \\ \frac{\partial n}{\partial t} = \frac{\partial(nv)}{\partial x} + \alpha_1 n - \beta_2 n \rho \end{cases} \quad (4.26)$$

$$\mathcal{R}_{YHD} : \begin{cases} \frac{\partial \rho}{\partial t} = nv - \gamma_1 \rho \\ \frac{\partial n}{\partial t} = \frac{\partial(nv)}{\partial x} + \alpha_1 n - \beta_2 n \rho \end{cases} \quad (4.27)$$

$$\mathcal{R}_{YFH} : \begin{cases} \frac{\partial \rho}{\partial t} = nv \\ \frac{\partial n}{\partial t} = \frac{\partial(nv)}{\partial x} + \alpha_1 n + \alpha_2 \rho - \beta_2 n \rho \end{cases} \quad (4.28)$$

The parameters in these three models are considered unknown and thus a prior distribution needs to be defined. Here, we use the same uniform prior distribution $\mathcal{U}(0, 1)$ for all the parameters of the three candidate models.

All candidate models have the same boundary conditions as the true model used to generate the synthetic data in the previous section. The models also share the same parameterized initial condition,

$$\rho(x, t = 0.5) = e^{-a(x-b)^2} \quad (4.29)$$

$$n(x, t = 0.5) = ce^{-d(x-f)^2} \quad (4.30)$$

Parameters a , b , c , d and f are estimated by minimizing the mean square error (MSE) between the observation data at $t = 0.5$ and the model shown above. This minimization is done using the function `fminunc` in Matlab. The results are shown in Fig. 4.3. The five optimized parameters are $a = 0.0357$, $b = 1.3116$, $c = 0.4761$, $d = 0.1223$, $f = 6.9407$ and the meas-square-error (MSE) is 0.0446 for hyphal density and 0.0018 for tip density.

Since we use synthetic data, no measurement noise is considered. Thus the observation model will only account for model discrepancy. Given that the two datasets provide measurements of both tip density and hyphal density, and both of them are

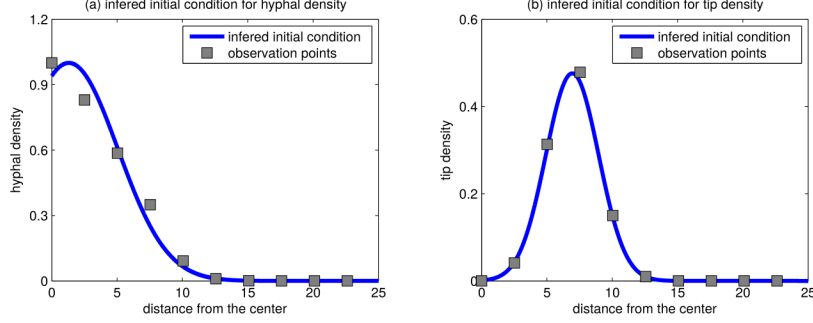


Figure 4.3: Inferred initial condition. Initial condition for candidate models is estimated using observation data at $t = 0.5$. MSE is 0.0446 for hyphal density and 0.0018 for tip density.

positive quantities, here, we use a multiplicative form for model discrepancies ϵ_ρ and ϵ_n .

$$d_\rho = \rho\epsilon_\rho, \quad \text{where} \quad \epsilon_\rho \sim \ln\mathcal{N}(0, \sigma_\rho^2) \quad (4.31)$$

$$d_n = n\epsilon_n, \quad \text{where} \quad \epsilon_n \sim \ln\mathcal{N}(0, \sigma_n^2) \quad (4.32)$$

The variance σ_ρ^2 and σ_n^2 are two hyper-parameters. They are adjusted during model calibration to give more flexibility in fitting the observation data. These two hyper-parameters are set to follow the inverse gamma distribution,

$$p(\sigma^2) = \frac{\beta^\alpha}{\Gamma(\alpha)} (\sigma^2)^{-\alpha-1} e^{-\frac{\beta}{\sigma^2}}. \quad (4.33)$$

There are two parameters needed to define this distribution: shape parameter α and scale parameter β . Since we have already determined the fitting of the initial condition, we have prior information regarding the magnitude of the model error. Namely, we can set $\beta = 0.0446$ for σ_ρ^2 and 0.0018 for σ_n^2 using the MSE obtained during the fitting of the initial condition. To indicate that this is an informative prior we set $\alpha = 10$.

Finally, the complete model class formulation for the corresponding YH growth model is shown in Eq. 4.34. The formulation for the other two models is easily obtained by replacing the fungal growth model and replacing the definition for the

parameter prior distribution in accordance with the model definition.

$$M_{YH} : \left\{ \begin{array}{l} \mathcal{R}_{YH} : \left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} = n\nu \\ \frac{\partial n}{\partial t} = \frac{\partial(n\nu)}{\partial x} + \alpha_1 n - \beta_2 n \rho \end{array} \right. \\ \\ \text{BC} : \left\{ \begin{array}{l} \rho(x=0, t) = 1, \rho(x=100, t) = 0 \\ n(x=0, t) = 0, n(x=100, t) = 0 \end{array} \right. \\ \\ \text{IC} : \left\{ \begin{array}{l} \rho(x, t=0.5) = e^{-a(x-b)^2}, \text{ with } a = 0.0357, b = 1.3116 \\ n(x, t=0.5) = ce^{-d(x-f)^2}, \text{ with } c = 0.4761, d = 0.1223 \\ \text{and } f = 6.9407 \end{array} \right. \\ \\ \text{Meas.Model} : \left\{ \begin{array}{l} d_\rho = \rho \epsilon_\rho, \quad \text{where } \epsilon_\rho \sim \ln \mathcal{N}(0, \sigma_\rho^2) \\ d_n = n \epsilon_n, \quad \text{where } \epsilon_n \sim \ln \mathcal{N}(0, \sigma_n^2) \end{array} \right. \\ \\ \text{Prior} : \left\{ \begin{array}{l} p(\nu | M_{YH}) = p(\alpha_1 | M_{YH}) = p(\beta_2 | M_{YH}) = \mathcal{U}(0, 1) \\ p(\sigma_\rho^2 | M_{YH}) = \frac{\beta_\rho^{\alpha_\rho}}{\Gamma(\alpha_\rho)} (\sigma_\rho^2)^{-\alpha_\rho - 1} e^{-\frac{\beta_\rho}{\sigma_\rho^2}}, \text{ with } \alpha_\rho = 10 \\ \text{and } \beta_\rho = 0.0446 \\ p(\sigma_n^2 | M_{YH}) = \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} (\sigma_n^2)^{-\alpha_n - 1} e^{-\frac{\beta_n}{\sigma_n^2}}, \text{ with } \alpha_n = 10 \\ \text{and } \beta_n = 0.0018 \end{array} \right. \end{array} \right. \quad (4.34)$$

In addition, since no additional information is considered to be known to discriminate the models in the absence of data, the prior model probability is given by $p(M_{YH} | \mathcal{M}) = p(M_{YHD} | \mathcal{M}) = p(M_{YFH} | \mathcal{M}) = 1/3$, where $\mathcal{M} = \{M_{YH}, M_{YHD}, M_{YFH}\}$.

4.3.3 BAYESIAN CALIBRATION RESULTS

With the final formulation described in the previous section, model calibration can be performed given observation data at $t = 5$. A number of 5000 samples are drawn

from the posterior distribution using TMCMC. Figs. 4.4(a)(b)(c) show the calibration results for Dataset I, and Figs. 4.4(d)(e)(f) provide the results for Dataset II.

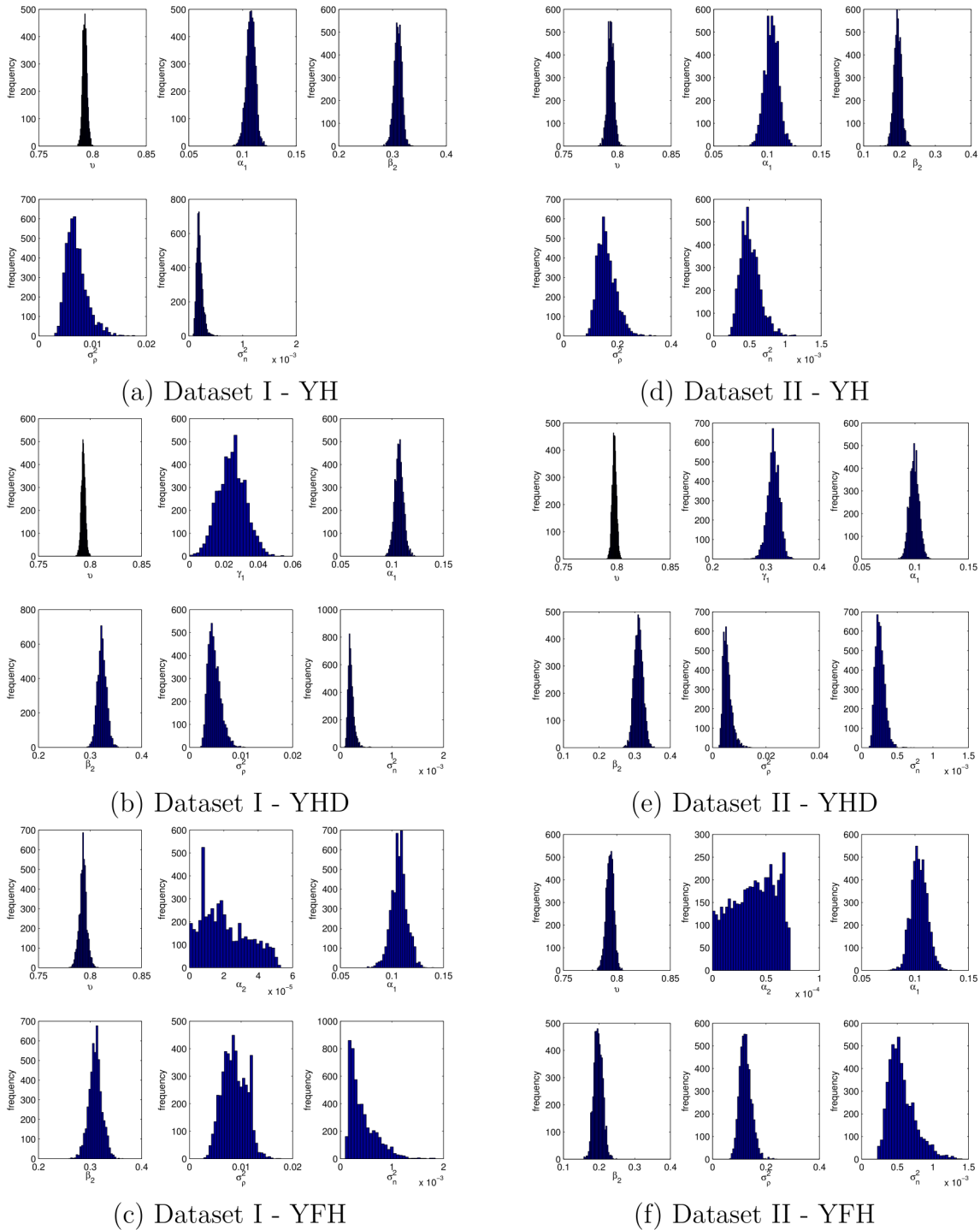


Figure 4.4: Marginal posterior distribution of parameters for candidate models YH, YHD, YFH for Dataset I (a),(b),(c), and Dataset II (d),(e),(f).

Due to model error, the outputs of a calibrated model may not be consistent with the data. So it is necessary to check the consistency of model outputs with the observation data. The discrepancy between the data and model outputs is quantified in this paper using posterior predictive checks [34, 10]. We check whether the data lies inside the 95% prediction interval. A prediction interval is a range in which a single new observation falls with a certain probability [15, 57]. The probability determines the width of the interval. Here, we use the 95% prediction interval. If most of the data is in the 95% prediction region, then the data would be considered a plausible outcome of the model.

In Fig. 4.5, hyphal density and tip density of the three candidate models are predicted at $t = 5$ with their associated uncertainties. Nearly all the observation data are within the 95% prediction interval. However, the prediction of different models have different interval ranges, which reflects the different amount of uncertainty within the calibrated models. In general, the more different is the candidate model from the true model, the bigger the model discrepancy during calibration. The discrepancy model provides increased flexibility to fit the data and consequently the prediction is more uncertain. From Fig. 4.5(a), we can see that YHD model fits the data best although the difference among three candidates is small when Dataset I is used. Thus in this case, there is no obvious way to discriminate just by looking at prediction of the models.

However, when Dataset II is used to calibrate the models, the prediction interval of YHD model is narrowest and captures all the data points. Just but looking at the prediction, YHD model is definitely the best choice. For YH model and YFH model there is a consistency problem between the model outputs and the observation data. Even though we account for model discrepancy in the observation model, the current statistical models used cannot compensate for the real discrepancy. Where to add the model discrepancy and how to model it is not trivial and beyond the scope of

this paper. New research in this direction can be found in Refs.[64, 13].

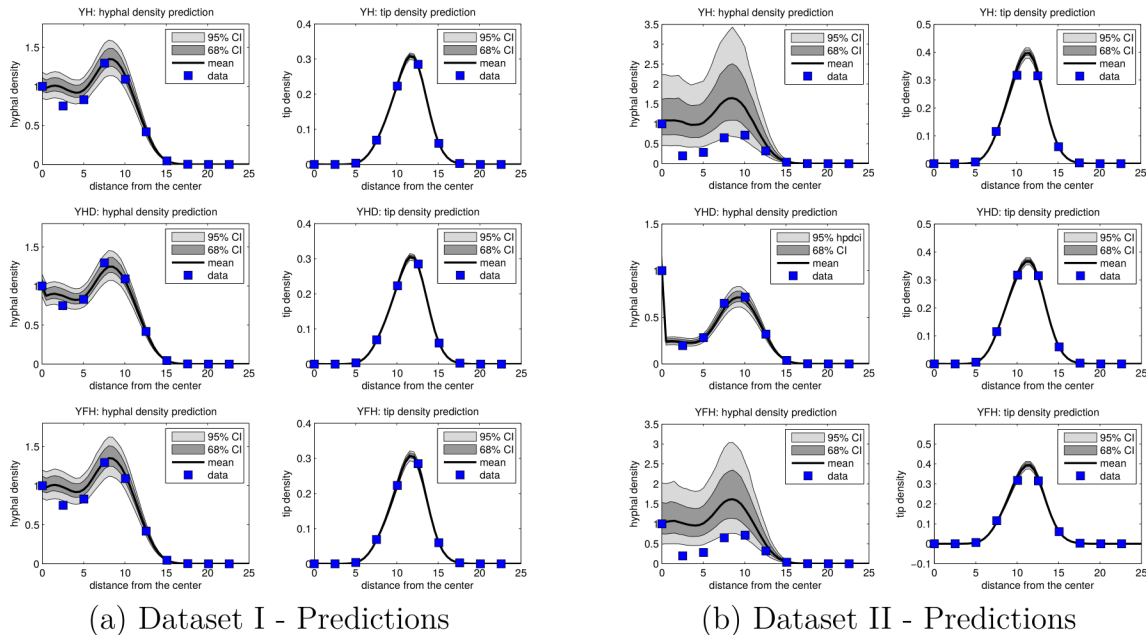


Figure 4.5: Model predictions at $t = 5$ using calibrated models with Dataset Scenario I (a) and Dataset II (b).

4.3.4 COMPARISON RESULTS

Note that for Dataset I, it is not clear how to discriminate the three models just based on how well they fit the data. As previously described, Bayesian model comparison offers a more comprehensive way to compare the models based on posterior model probability. Here, since a priori all models are equally likely, the posterior model probability is solely determined by the evidence. Table 4.1 shows the calculated log-evidence for the three models as well as the corresponding expected log-likelihood that measures goodness-of-fit and KL divergence that measures model complexity.

The results show that YH model has the largest evidence. The evidence of YHD model is slightly smaller than YH model but still much larger than YFH model. Since there are no prior preferences over three candidate models, then YH model might be selected as the most plausible one and the dominate branching types will be Dichotomous branching and tip-hyphae anastomosis. However, this is not necessarily

Table 4.1: Bayesian model comparison results for Dataset I ($\gamma_1 = 0.03$)

Model	Log evidence	Expectation of log likelihood	KL divergence
YH	20.844	35.807	14.964
YHD	21.384	39.403	18.020
YFH	-8.292	33.340	41.632

Table 4.2: Bayes factors for Dataset I ($\gamma_1 = 0.03$)

Bayes factor (/)	YH	YHD	YFH
YH	1	0.583	4.5e+12
YHD	1.716	1	7.7e+12
YFH	2.2e-13	1.3e-13	1

the best choice. Table 4.2 shows the Bayes factor of every two models. Considering a ratio of 100 as strong evidence for model selection, YFH model will be definitely cast aside. However, one can not select between YH model and YHD model confidently based on this criterion since the Bayes factor is 1.716 which is not worth more than a bare mention. One may notice that the real model that generates observation data is a YHD model, which means YHD model should be overwhelmingly supported by the data. However this is not the case here. The expectation of log likelihood represents how well the model fits the observation data and for this metric $YHD > YH > YFH$. This is consistent with the previous discussion about Fig. 4.5 which also shows that YHD model fits the data slightly better than the other two, although visually the difference between YHD and the other two models is not big. However, the evidence reflects a trade-off between the data fitness and model complexity. Although YHD model has the best data fitness, it is more complex than YH model, which can be seen from KL divergence. As a result the YHD model has a larger penalization than YH. This is consistent with the fact that in the true model $\gamma_1 = 0.03$, which is much smaller than those of other branching types. In this case, hyphal death can be taken as a non-dominant branching type which explains the above results. The posterior model probabilities are $p(M_{YH}|\mathcal{M}, D_{meas}) = 0.63$, $p(M_{YHD}|\mathcal{M}, D_{meas}) = 0.37$, and

$p(M_{YFH}|\mathcal{M}, D_{meas}) = 0.0$. Thus, there is not sufficient evidence to rule out model YH, or YHD for that matter. Both YH and YHD provide an adequate representation of the data, and the posterior model probability reflects this uncertainty.

When γ_1 is increased to 0.3, then the ranking of the models changes as shown in Table 4.3 and Table 4.4. For Dataset II, we can see that the evidence of YHD model is much larger than the other models and the dominant branching types will be Dichotomous branching, tip-hypha anastomosis and hyphal death. There is no doubt in this case that YHD is the best model to describe the data, which is in agreement with prediction shown in Fig. 4.5(b). The posterior model probability is one for YHD and zero for YH and YFH. In this case, YH and YFH have big deficiencies in fitting the data, thus neither of these two will be selected.

Table 4.3: Bayesian model comparison results for Dataset II ($\gamma_1 = 0.3$)

Model	Log evidence	Expectation of log likelihood	KL divergence
YH	-50.392	-0.787	49.605
YHD	17.021	35.253	18.232
YFH	-97.171	-4.432	92.739

Table 4.4: Bayes factors for Dataset II ($\gamma_1 = 0.3$)

Bayes factor (/)	YH	YHD	YFH
YH	1	5.3e-30	2.1e+20
YHD	1.9e+29	1	3.9e+49
YFH	4.8e-21	2.6e-50	1

In the case where only YH and YFH are considered, then YH again would be chosen over YFH according to the Bayes factor shown in Table 4.4. However, from Fig. 4.4(d), we can see that structural parameter σ_p^2 is around 0.2 which means structural error in hyphal growth model has a standard deviation of 0.45. This structural error is big considering hyphal density is smaller than 1 in our problem. This large uncertainty is also reflected in Fig. 4.5(b) where the prediction interval of hyphal density is quite wide for YH model. Thus, due to large discrepancy and wide

prediction intervals, one needs to reconsider whether this is a sufficient representation of the reality. The option in this case is to generate new alternative models and repeat the comparison process. The process also offers some insight into which branching types should further be considered by ignoring first the ones included in the models with low evidence values.

4.4 CONCLUSIONS

Mathematical models play an important role in studying fungi. A vast number of models have been developed to describe various processes involved in fungal growth. This raises the importance of model selection. This chapter develops a systematic framework based on Bayesian model selection for comparing models in the presence of model uncertainty. In contrast to more classical measures of data fitness, Bayesian model selection uses posterior model probabilities to rank candidate models. This measure naturally obeys Occam's razor since it uses the trade-off between data fitness and model complexity to select the best representation of the observed phenomenon. The application of this method is illustrated with an example where three fungal growth models are compared given observation data synthetically generated. The introduced model uncertainty not only helps in data fitting, but also provides a more accurate prediction as well as a perspective to evaluate the results of model selection. The promising results provide us with the hope that the proposed computational framework may become a widespread tool in studying fungal models.

CHAPTER 5

HYPOTHESIS GENERATION FOR MODEL DISCREPANCY

5.1 INTRODUCTION

In Chapter. 4, we introduced model discrepancy which offers a way to evaluate the predictive capability of the best model in the context of model selection. If the model discrepancy is very large, the best model cannot be used even though it outperforms other candidates. In this chapter, we will go deeper into the model discrepancy and try to explore the missing structure.

Simulation models are widely used to study complex physical process. These models are generally based on current scientific understanding. To improve the predictive performance, observation data are collected to infer parameters within the model. Sometimes these parameters themselves are of scientific interest as they have specific physical meaning. However, due to limitation of human knowledge, there is always a discrepancy between model output and the real process. Bryjarsdóttir [13] shows that if model discrepancy is ignored, predictions and inferences about parameters will be biased and overconfident.

A common way to include model discrepancy is proposed by Kennedy and O’Hagan [49], which is shown in Eq. 5.1.

$$d(x) = y(x) + \epsilon_{meas} = f(x, \theta) + \epsilon_{model} + \epsilon_{meas} . \quad (5.1)$$

where the real process output $y(x)$ is represented by physical model $f(x, \theta)$ plus model discrepancy ϵ_{model} . $d(x)$ is the observation data and ϵ_{meas} indicates measurement noise. Usually, we have prior knowledge about physical parameter θ and model

discrepancy ϵ_{model} . Once observation data are collected, knowledge about θ and ϵ_{model} will be updated through Bayesian inference.

In Chapter. 4, we model this discrepancy with a lognormal distribution $\ln\mathcal{N}(0, \sigma^2)$ where σ^2 is a hyperparameter and can be adjusted according to the observation data. This structure offers the flexibility to fit the data. Large discrepancy will lead to large value of σ^2 . This approach was used to show the necessity of including model discrepancy and it served its purpose. However, for general application, a single normal distribution is far from satisfactory as for different x the discrepancy ϵ_{model} are usually different.

A popular way to model the discrepancy is using Gaussian process. This method has been widely discussed in the literature [49, 13, 12]. Gaussian process is a distribution for a function. Each value of the function follows normal distribution and values together follows multivariate normal distribution. Once some function values are observed, the conditional distribution of the other can be easily calculated through Bayes' rule. The covariance between different values are usually quantified with distance-based kernels, which means if the function value is obtained at some point, the value of nearby points can also be inferred. This property makes Gaussian process work well for interpolation. In fact, Gaussian process alone often serves as an emulator [81]. However, in that case, a much larger number of observation data need to be collected. Bryjarsdóttir [13] conducted various experiments with Gaussian process over model discrepancy. The results show that for interpolation, Gaussian process remarkably improves the predictive performance of the original model. However, more information is required in order to make good inference of physical parameters. When comes to extrapolation, Gaussian process is unable to make good prediction even if more information about physical process is provided.

In fact, Eq. 5.1 has an obvious limitation, that is ϵ_{model} is not necessarily additive. ϵ_{model} can be embedded in $f(x, \theta)$ and this is common for systems described

by differential equations. A compensation term in observation model can not count for the real missing structure. In this chapter, instead of using surrogate for the discrepancy, we try to explore the behavior of the discrepancy, and uncover the missing structure. An improved model is not only helpful for system prediction and inference of parameters but also good for scientific discovery.

5.2 METHODOLOGY

To set notation, consider the following simulation model:

$$\mathcal{R}(\mathbf{x}, \boldsymbol{\theta}, \epsilon) = 0 \tag{5.2}$$

\mathcal{R} is some operator and $\boldsymbol{\theta}$ represents physical parameters. \mathbf{x} denotes state variable, which usually contains multiple states $\mathbf{x} = [x_1, x_2, \dots, x_M]$. ϵ represents the missing structure. Here we assume Eq. 5.2 is a set of differential equations. We will propose a method to generate hypothesis about the structure of ϵ .

The main idea of the method is that although ϵ is unknown, it is supposed to have connection with \mathbf{x} , and we can try to uncover this connection by measuring how strong ϵ is related to \mathbf{x} or transformations of \mathbf{x} such as x_1^2 , x_1x_2 , etc. However, we cannot treat ϵ as a single variable with some prior distribution. Since states usually change with time and space, ϵ should also have different value accordingly. In many cases, differential equations are solved with the process of discretization. This means states \mathbf{x} at different time points can be regarded as independent variables. Eq. 5.3 shows the discretized version of Eq. 5.2.

$$\mathcal{L}(\mathbf{x}(t), \boldsymbol{\theta}, \epsilon(t)) = 0, \quad t = 0, 1, 2, \dots T \tag{5.3}$$

As discussed above, $\epsilon(t)$ at each time point t can also be taken as an independent variable. In this way T new parameters are introduced and these parameters can be inferred along with physical parameters and states. So, although we don't know the

exact form of ϵ , we can still work on the values of ϵ at discretized time points. We can place a high-dimensional prior distribution over ϵ at all discretized time points and update it given observation data. Once we have posterior distribution of $\mathbf{x}(t)$ and $\epsilon(t)$, we can measure the dependence between them and generate hypothesis about the structure of ϵ .

This process includes two steps. The first one is to get posterior distribution of $\epsilon(t)$ and $\mathbf{x}(t)$. The second one is to measure their dependence. Since T affects the accuracy of the solution for differential equations, it cannot be too small. As a result, we have to solve the inverse problem in high dimensions. In Chapter. 2, we have proposed an Bayesian inference method which can be applied here. The second step is to measure the dependence between $\epsilon(t)$ and $\mathbf{x}(t)$ (or its transformations). Here, mutual information is adopted as the metric. For example, if we want to check which state is most related to ϵ , we can calculate $I([\epsilon(0), \epsilon(1), \dots, \epsilon(T)], [x_i(0), x_i(1), \dots, x_i(T)])$ for $i = 1, 2, \dots, M$ and select the state with highest mutual information. By doing this, we can claim certain state is contained in ϵ . Since mutual information need to be calculated in high dimensions we can first apply CCA to map both vectors to low dimensions as we did in Chapter. 3. This process is shown in Algorithm. 2. After selecting x_i , we can further complicate it by adding new ϵ , such as $\epsilon^* x_i$ and repeat the algorithm for inferring ϵ^* . All of these hypothesis can be used as candidate models, and in the end, model selection will be performed to select the best one. The process of conducting model selection has been discussed in Chapter. 4.

5.3 NUMERICAL EXPERIMENTS

In this section, we will first work on Lotka-Volterra equations and show how to use the proposed method to generate hypothesis for the model discrepancy. Then we will generate hypothesis for the fungal growth model which has been studied in Chapter. 4. Unlike Lotka-Volterra equations which only needs to be discretized on temporal axis,

Algorithm 2 Hypothesis generation about unknown structure ϵ

- 1: Discretize the model as well as ϵ .
 - 2: Place a high-dimensional Gaussian prior over ϵ at all discretized time points.
 - 3: Use EnLLVM to obtain the posterior distribution for ϵ as well as state variables.
 - 4: Use approximate MI in high dimensions proposed in Chapter. 3 to find which functions of \mathbf{x} are related to the unknown function ϵ .
 - 5: Based on the result of step 4, complicate it by introducing new ϵ and repeat step 1-4.
 - 6: Use this information to generate a number of hypothesis and perform Bayesian model selection to select the most plausible structure of ϵ .
-

the fungal growth model needs to be discretized on both temporal and spacial axis's and this will lead to much higher dimensionality.

5.3.1 LOTKA-VOLTERRA EQUATIONS

The Lotka-Volterra equation

$$\dot{x}_1(t) = -0.6x_1(t)x_2(t) + x_1(t) \quad (5.4)$$

$$\dot{x}_2(t) = 0.3x_1(t)x_2(t) - 0.4x_2(t) \quad (5.5)$$

is widely used to describe the dynamics of biological systems in which a prey and its predator interact. The states x_1 and x_2 are population of a prey and its predator, $\dot{x}_1(t)$ and $\dot{x}_2(t)$ represent the growth rates of the two populations over time. For the first experiment, we assume that $0.4x_2(t)$ in Eq. 5.5 is missing and we use ϵ to represent this missing structure. We will use the proposed method to generate hypothesis for ϵ .

First we discretize the equations and the discretized version is shown below:

$$x_1(t_{i+1}) = x_1(t_i) + \Delta t(-0.6x_1(t_i)x_2(t_i) + x_1(t_i)) \quad (5.6)$$

$$x_2(t_{i+1}) = x_2(t_i) + \Delta t(0.3x_1(t_i)x_2(t_i) - \epsilon(t_i)) \quad (5.7)$$

Δt is the time interval between t_{i+1} and t_i . Here $\Delta t = 0.05$ and the total time span is $[0, 5]$. So the dimensionality of ϵ is 100, $\epsilon = [\epsilon(0), \epsilon(1), \dots, \epsilon(199)]$. Observations

are made every 10 time steps. The initial conditions are $x_1(0) = 1$ and $x_2(0) = 2.5$. We place a uniform prior over $\epsilon(t_i)$, $\epsilon(t_i) \sim \mathcal{U}(0, 1)$ and perform Bayesian update using EnLLVM which we proposed in Chapter. 2. For EnLLVM, 200 samples are used and the dimensionality of latent space is 10. After getting posterior samples of ϵ , x_1 and x_2 , we calculate mutual information between $[\epsilon(0), \epsilon(1), \dots, \epsilon(99)]$ and $[x_1(1), x_1(2), \dots, x_1(100)]$, $[x_1(1), x_1(2), \dots, x_1(100)]$ respectively. Since the dimensionality is high, we use first map it to lower dimensions, as we did in Chapter. 3. The simulation runs 100 times, and the average of mutual information as well as the number of times that each state is selected over the other one are shown in Table. 5.1. As we can see, The average of mutual information between ϵ and x_2 is higher than that between ϵ and x_1 . And in 100 trials, there are 81 trials when x_2 has larger mutual information than x_1 . So we can come up with the hypothesis that x_2 is part of ϵ and a candidate model can be the one with ϵ replacing by ax_2 where a is some constant. We can also repeat the process by taking a as a variable and infer the structure of a . This might lead to new candidate model. Note that all the candidate models will be eventually ranked using Bayesian model selection, so false candidates never hurt.

We also test other scenarios and the results are also shown in Table. 5.1. As we can see, if the missing structure is $0.3x_1x_2$, then ϵ has nearly the same relation with x_1 and x_2 . In this case, we can choose both as candidates or complicate the structure by introducing a new ϵ . For example, assume we pick x_1 , we can further complicate it by multiplying a new ϵ , then the missing structure becomes ϵx_1 . Same procedure can be taken to infer the structure of new ϵ . As is shown in Table. 5.1, it is more related to x_2 than x_1 . Thus a plausible candidate model can be ax_1x_2 where a is some constant.

Table 5.1: Results for 100 trials

missing structure	avg. MI of x_1	avg. MI of x_2	no. of x_1	no. of x_2
x_1	1.3282	1.3014	62	38
$0.4x_2$	1.0196	1.1452	19	81
$0.3x_1x_2$	1.2303	1.2191	51	49
$0.3x_2$ (e.g. ϵx_1)	1.2301	1.2650	33	67

5.3.2 FUNGAL GROWTH MODELS

In Chapter. 4, Bayesian model comparison is conducted among different candidate models and the most plausible one is selected. In this section, we will show how to generate candidates for fungal growth models. For this experiment, the model we use is shown in Eq. (5.8).

$$\left\{ \begin{array}{l} \mathcal{R}_{Y?} : \\ \text{BC} : \\ \text{IC} : \end{array} \right. \left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} = 0.8n \\ \frac{\partial n}{\partial t} = \frac{0.8\partial(n)}{\partial x} + 0.03n + \epsilon \\ \rho(x=0, t) = 1, \rho(x=100, t) = 0 \\ n(x=0, t) = 0, n(x=100, t) = 0 \\ \rho(x, t=0.5) = e^{-a(x-b)^2}, \text{ with } a = 0.0357, b = 1.3116 \\ n(x, t=0.5) = ce^{-d(x-f)^2}, \text{ with } c = 0.4761, d = 0.1223, f = 6.9407 \end{array} \right. \quad (5.8)$$

x is the distance from the spore center, ρ is hyphal density in units of filament length per unit area and n is tip density (number per unit area). Here we take ϵ as the missing structure which can be any growth pattern shown in Figure 4.1. Note that ϵ might be also a combination of different branching types (e.g. YH, YFH, YHD etc.).

The discretized version is shown below.

$$\rho(x_i, t_{j+1}) = \rho(x_i, t_j) + 0.3\Delta t n(x_i, t_j) \quad (5.9)$$

$$\begin{aligned} n(x_i, t_{j+1}) = & -0.15 \frac{\Delta t}{\Delta x} (n(x_{i+1}, t_j) - n(x_{i-1}, t_j)) + (0.03\Delta t + 1)n(x_i, t_j) \\ & - \Delta t \epsilon(x_{i-1}, t_j) \end{aligned} \quad (5.10)$$

Here $\Delta t = 0.05$ and the total time span is $[0, 5]$. $\Delta x = 0.05$ and the observing radius is $[0, 5]$. Since it involves both time and space, the dimensionality is much higher. We use 20 steps on both t axis and x axis, so the dimensionality for ϵ is 400 and 1200 for combining ϵ , ρ and n . We place a uniform prior over $\epsilon(t_i)$, $\epsilon(t_i) \sim \mathcal{U}(0, 0.5)$ and perform Bayesian update using EnLLVM where 400 samples are used and the dimensionality of latent space is 20. Observations are made average 10 time steps on 5 positions. We test several different cases using the proposed method and the result is shown in Table. 5.2. As we can see, if the missing structure is 0.3ρ , ϵ will have larger mutual information with ρ than with n . And $a\rho$ is a plausible hypothesis for ϵ . However, since the method includes applying EnLLVM and computing low bound of mutual information both of which give approximate results, numerical errors are inevitable. Thus, we might not always get the expected results. As we can see, if we pick $0.3n$ as missing structure, although n is selected more times than ρ , the difference is not big. This also happens when we take $0.3n\rho$ as missing structure. First ρ will be selected, then ideally n should be selected. However, it is hard to decide whether ϵ is more related to n or ρ , as n has slightly larger average mutual information and ρ has slightly more times being picked over n . In this case, it is safe to include both of them as candidates and leave it to Bayesian model selection.

Table 5.2: Results for 100 trials

missing structure	avg. MI of ρ	avg. MI of n	no. of ρ	no. of n
0.3ρ	1.4165	1.3983	63	37
$0.3n$	1.4068	1.4137	42	58
$0.3n\rho$	1.4084	1.4041	56	44
$0.3n$ (e.g. $\epsilon\rho$)	1.4076	1.4067	46	54

5.4 CONCLUSION

In this chapter, we proposed a method to generate hypothesis for model discrepancy ϵ . Since model discrepancy is usually related to model states, we try to find which

state is more related to the discrepancy and then complicate it by introducing new *epsilon*. For differential equations, we work on the discretized version. The values of ϵ at different time point are taken as independent variables and updated together with states using EnLLVM. Then mutual information between discretized ϵ and discretized states is calculated.

The proposed method is tested on two models. The results of Lotka-Volterra equations are quite positive, as missing structures can be accurately identified. While for the fugal growth model, different structures can not always be clearly differentiated. The reason resides in the high dimensionality of the model. This method includes two steps, applying EnLLVM to get the posterior distribution and computing lower bound of mutual information, both of which require mapping from high dimensional space to lower dimensions. As we know, the loss of information during dimension reduction is inevitable, so both of these two steps can only give approximate results. Since the original space of fugal growth model is much higher, it is supposed to have larger approximation error.

In spite of its limitation, the proposed method provides a new perspective towards the model discrepancy. As more methods aiming at solving high dimensional inference are developed, we hope the proposed approach could help scientists to develop models and even make scientific discoveries.

BIBLIOGRAPHY

- [1] B. Ait-El-Fquih and I. Hoteit. “A Variational Bayesian Multiple Particle Filtering Scheme for Large-Dimensional Systems”. In: *IEEE Transactions on Signal Processing* 64.20 (Oct. 2016), pp. 5409–5422. ISSN: 1053-587X. DOI: 10.1109/TSP.2016.2580524.
- [2] F R Bach and M I Jordan. *A probabilistic interpretation of canonical correlation analysis*. Tech. rep. Technical Report 688. Department of Statistics, University of California, Berkeley, 2005.
- [3] Francis R. Bach and Michael I. Jordan. “Kernel Independent Component Analysis”. In: *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 1–48. ISSN: 1532-4435.
- [4] J. Beck and S. Au. “Bayesian Updating of Structural Models and Reliability using Markov Chain Monte Carlo Simulation”. In: *J. Eng. Mech.* 128.4 (2002), pp. 380–391.
- [5] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, Aug. 2006.
- [6] J. A. Blain. In: *The Filamentous Fungi: Industrial Mycology*. Edward Arnold Ltd, 1975, pp. 193–211.
- [7] G.P. Boswell and F.A. Davidson. “Mathematical modelling of mycelia: a question of scale”. In: *Fungal Biology Reviews* 21.1 (2007), pp. 30–41.
- [8] G.P. Boswell and F.A. Davidson. “Modelling hyphal networks”. In: *Fungal Biology Reviews* 26.1 (2012), pp. 30–38.
- [9] G.P. Boswell et al. “A mathematical approach to studying fungal mycelia”. In: *Mycologist* 17 (2003), pp. 165–171.
- [10] G.E.P. Box. In: *Bayesian Inference in Statistical Analysis*. New York, USA: Wiley Classics Library, 1973.
- [11] Eric Brochu, Vlad M Cora, and Nando de Freitas. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Mod-*

- eling and Hierarchical Reinforcement Learning*. Tech. rep. UBC TR-2009-023 and arXiv:1012.2599. University of British Columbia, Department of Computer Science, 2009. URL: <http://arxiv.org/abs/1012.2599>.
- [12] Corey Bryant and Gabriel Terejanu. “An information-theoretic approach to optimally calibrate approximate models”. In: (2012).
- [13] J. Brynjarsdóttir and A. O’Hagan. “Learning about physical parameters: the importance of model discrepancy ”. In: *Inverse Problems* 30.11 (2014), p. 114007.
- [14] M. F. Bugallo and P. M. Djuric. “Particle filtering in high-dimensional systems with Gaussian approximations”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2014, pp. 8013–8017.
- [15] C. Chatfield. “Calculating Interval Forecasts”. In: *Journal of Business and Economic Statistics* 11.2 (1993), pp. 121–135.
- [16] Yongqiang Cheng et al. “Information geometry of target tracking sensor networks”. In: *Information Fusion* 14.3 (2013), pp. 311–326.
- [17] J. Ching and Y. Chen. “Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging”. In: *J. Eng. Mech.* 133.7 (2007), pp. 816–832.
- [18] United States President’s Information Technology Advisory Committee. *Computational Science: Ensuring America’s Competitiveness*. 2005.
- [19] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN: 0471241954.
- [20] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.
- [21] K. Cowles and B.P. Carlin. “Markov chain monte carlo convergence diagnostics”. In: *JASA* (1996), pp. 883–904.
- [22] J. Crassidis and J. Junkins. *Optimal Estimation of Dynamic Systems*. CRC Press, 2004.
- [23] F. Darema. “Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements”. In: *Computational Science-ICCS 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 662–669.

- [24] F. A. Davidson and A. W. Park. “A mathematical model for fungal development in heterogeneous environments”. In: *Applied Mathematics Letters* 11.6 (1998), pp. 51–56.
- [25] Santpal Singh Dhillon and Krishnendu Chakrabarty. “Sensor placement for effective coverage and surveillance in distributed sensor networks”. In: *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*. Vol. 3. IEEE. 2003, pp. 1609–1614.
- [26] P. M. Djuric, T. Lu, and M. F. Bugallo. “Multiple Particle Filtering”. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. Vol. 3. Apr. 2007, DOI: 10.1109/ICASSP.2007.367053.
- [27] Laura Dovera and Ernesto Della Rossa. “Multimodal ensemble Kalman filtering using Gaussian mixture models”. In: *Computational Geosciences* 15.2 (2011), pp. 307–323. ISSN: 1573-1499. DOI: 10.1007/s10596-010-9205-3.
- [28] Morris L. Eaton. *Multivariate Statistics: a Vector Space Approach*. John Wiley and Sons., 1983, pp. 116–117.
- [29] L Edelstein. “The Propagation of Fungal Colonies: A Model for Tissue Growth”. In: *J. theor. Biol.* 98 (1982), pp. 679–701.
- [30] L. Edelstein and L. A. Segel. “Growth and metabolism in mycelial fungi”. In: *Journal of Theoretical Biology* 104 (1983), pp. 187–210.
- [31] E Ertin, JW Fisher, and LC Potter. “Maximum mutual information principle for dynamic sensor query problems”. In: *Information processing in sensor networks*. Springer. 2003, pp. 405–416.
- [32] G Evensen. *Data Assimilation. The Ensemble Kalman Filter*. Springer, Aug. 2009.
- [33] G. Evensen. “The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation”. In: *Ocean Dynamics* 53 (2003), pp. 343–367.
- [34] A. Gelman and C.R. Shalizi. “Philosophy and the practice of Bayesian statistics”. In: *British Journal of Mathematical and Statistical Psychology* 66 (2013), pp. 8–38.
- [35] N. A. R. Gow and G. M. Gadd. In: *The Growing Fungus*. Chapman and Hall, 1995.

- [36] Jeff Greeley, Peter Zapol, and Larry Curtiss. “Recent Trends in Computational CATALYSIS”. In: *SciDAC Review* (2008). URL: <http://www.scidacreview.org/0804/pdf/trends.pdf>.
- [37] Heikki Haario, Eero Saksman, and Johanna Tamminen. “An adaptive Metropolis algorithm”. In: *Bernoulli* 7.2 (Apr. 2001), pp. 223–242.
- [38] H. Haario et al. “DRAM: Efficient adaptive MCMC”. In: *Statistics and Computing* 16.4 (2006), pp. 339–354. ISSN: 0960-3174.
- [39] D.L. Hawksworth. “The magnitude of fungal diversity: the 1.5 million species estimate revisited”. In: *Mycological Research* 105.1 (2001), pp. 1422–1432.
- [40] P. L. Houtekamer and Herschel L. Mitchell. “A Sequential Ensemble Kalman Filter for Atmospheric Data Assimilation”. In: *Monthly Weather Review* 129.1 (2001), pp. 123–137. DOI: 10.1175/1520-0493(2001)129<0123:ASEKFF>2.0.CO;2.
- [41] P. L. Houtekamer and Fuqing Zhang. “Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation”. In: *Monthly Weather Review* 144.12 (2016), pp. 4489–4532. DOI: 10.1175/MWR-D-15-0440.1.
- [42] Michael Hutchinson, Hyondong Oh, and Wen-Hua Chen. “A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors”. In: *Information Fusion* 36 (2017), pp. 130–148.
- [43] H. Jeffreys. *The theory of probability*. OUP Oxford, 1998.
- [44] D. R. Jones, M. Schonlau, and W. J. Welch. “Efficient Global Optimization of Expensive Black-Box Functions”. In: *Journal of Global Optimization* 13 (1998), pp. 455–492.
- [45] S.J. Julier and J.K. Uhlmann. “Unscented Filtering and Nonlinear Estimation”. In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422.
- [46] J.B. Kadane and N.A. Lazar. “Methods and Criteria for Model Selection”. In: *Journal of the American Statistical Association* 99.465 (2004), pp. 279–290.
- [47] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Transactions of the ASME—Journal of Basic Engineering* 82 (1960). Series D, pp. 35–45.
- [48] Richard W. Katz et al. “Uncertainty analysis in climate change assessments”. In: *Nature Climate Change* 3 (2013), pp. 769–771.

- [49] M. C. Kennedy and A. O’Hagan. “Bayesian calibration of computer models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63 (2001), pp. 425–464.
- [50] Bahador Khaleghi et al. “Multisensor data fusion: A review of the state-of-the-art”. In: *Information Fusion* 14.1 (2013), pp. 28–44.
- [51] Shiraj Khan et al. “Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data”. In: *Physical Review E* 76.2 (2007), p. 026209.
- [52] P. Kirk, T. Thorne, and M.P. Stumpf. “Model selection in systems and synthetic biology”. In: *Current Opinion in Biotechnology* 24.4 (2013), pp. 767–774.
- [53] L F Kozachenko and N N Leonenko. “L. F. Kozachenko, N. N. Leonenko, “Sample Estimate of the Entropy of a Random Vector”, Probl. Peredachi Inf., 23:2 (1987), 9–16”. In: *Problemy Peredachi Informatsii* (1987).
- [54] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. “Estimating mutual information”. In: *Physical Review E* 69.6 (June 2004), p. 066138.
- [55] A Krause, A Singh, and C Guestrin. “Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies”. In: *J. Mach. Learn. Res.* 9 (June 2008), pp. 235–284. ISSN: 1532-4435.
- [56] Clemens Kreutz and Jens Timmer. “Systems biology: experimental design”. In: *FEBS Journal* 276.4 (2009), pp. 923–942.
- [57] J.F. Lawless and M. Fredette. “Frequentist prediction intervals and predictive distributions”. In: *Biometrika* 92.3 (2005), pp. 529–542.
- [58] Abigail C Leeder, Javier Palma-Guerrero, and N Louise Glass. “The social network: deciphering fungal language”. In: *Nature Reviews Microbiology* 9.6 (June 2011), pp. 440–451.
- [59] DV Lindley. “On a Measure of the Information Provided by an Experiment”. In: *Ann. Math. Statist.* 27(4) (1956), pp. 986–1005.
- [60] D.J.C. MacKay. “Model Comparison and Occam’s Razor”. In: *Information Theory, Inference, and Learning Algorithms*. Cambridge, England: Cambridge University Press, 2003, pp. 345–357.
- [61] Reza Madankan, Puneet Singla, and Tarunraj Singh. “Optimal information collection for source parameter estimation of atmospheric release phenomenon”.

- In: *American Control Conference, ACC 2014, Portland, OR, USA, June 4-6, 2014*. 2014, pp. 604–609.
- [62] A. Mira. “On Metropolis-Hastings algorithms with delayed rejection”. In: *Metron* 59 (2001), pp. 3–4.
- [63] Shinya Nakano, Genta Ueno, and Tomoyuki Higuchi. “Merging particle filter for sequential data assimilation”. In: *Nonlinear Processes in Geophysics* 14.4 (2007), pp. 395–408.
- [64] Todd A. Oliver et al. “Validating predictions of unobserved quantities”. In: *Computer Methods in Applied Mechanics and Engineering* 283 (2015), pp. 1310–1335. ISSN: 0045-7825.
- [65] Nadia Oudjane and Christian Musso. “Progressive correction for regularized particle filters”. In: *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*. Vol. 2. IEEE. 2000, THB2–10.
- [66] E. E. Prudencio and K. W. Schulz. “The parallel C++ statistical library ‘QUESO’: Quantification of Uncertainty for Estimation, Simulation and Optimization”. In: *Euro-Par 2011: Parallel Processing Workshops*. Springer, 2012, 398–407.
- [67] E. Prudencio and S.H. Cheung. “Parallel Adaptive Multilevel Sampling Algorithms for the Bayesian Analysis of Mathematical Models”. In: *International Journal for Uncertainty Quantification* 2.3 (2012), pp. 215–237.
- [68] KVU Reddy et al. “Data assimilation in variable dimension dispersion models using particle filters”. In: *Information Fusion, 2007 10th International Conference on*. IEEE. 2007, pp. 1–8.
- [69] D. Redecker, R. Kodner, and L.E. Graham. “Glomalean fungi from the Ordovician”. In: *Science* 289.1 (2000), pp. 1920–1921.
- [70] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter. Particle Filters for Tracking Applications*. Artech House, 2004.
- [71] Gary S Settles. “Fluid mechanics and homeland security”. In: *Annu. Rev. Fluid Mech.* 38 (2006), pp. 87–110.
- [72] A.F.M. Smith and D.J. Spiegelhalter. “Bayes Factors and Choice Criteria for Linear Models”. In: *Journal of the Royal Statistical Society, Ser.B* 42 (1980), pp. 213–220.

- [73] C. Snyder et al. “Obstacles to high-dimensional particle filtering”. In: *Monthly Weather Review* 136.12 (2008), pp. 4629–4640.
- [74] P. Stano et al. “Parametric Bayesian Filters for Nonlinear Stochastic Dynamical Systems: A Survey”. In: *IEEE Transactions on Cybernetics* 43.6 (2013), pp. 1607–1624. ISSN: 2168-2267. DOI: 10.1109/TSMCC.2012.2230254.
- [75] Andreas S. Stordal et al. “Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter”. In: *Computational Geosciences* 15.2 (Mar. 2011), pp. 293–305.
- [76] Gabriel Terejanu, Tarunraj Singh, and Peter D. Scott. “Unscented Kalman Filter/Smother for a CBRN Puff-Based Dispersion Model”. In: *11th International Conference on Information Fusion, Quebec City, Canada*. July 2007.
- [77] Gabriel Terejanu, Rochan R Upadhyay, and Kenji Miki. “Bayesian experimental design for the active nitridation of graphite by atomic nitrogen”. In: *Experimental Thermal and Fluid Science* 36 (2012), pp. 178–193.
- [78] Gabriel Terejanu et al. “Adaptive Gaussian sum filter for nonlinear Bayesian estimation”. In: *IEEE Transactions on Automatic Control* 56.9 (2011), pp. 2151–2156.
- [79] Michael E Tipping and Christopher M Bishop. “Probabilistic principal component analysis”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.
- [80] Eric J. Tuegel et al. “Reengineering Aircraft Structural Life Prediction Using a Digital Twin”. In: *International Journal of Aerospace Engineering* (2011).
- [81] Laura Uusitalo et al. “An overview of methods to evaluate uncertainty of deterministic models in decision support”. In: *Environmental Modelling & Software* 63 (2015), pp. 24–31.
- [82] J. Vanlier et al. “Optimal experiment design for model selection in biochemical networks”. In: *BMC Systems Biology* 8.20 (2014).
- [83] Janett Walters-Williams and Yan Li. “Estimation of Mutual Information: A Survey”. In: *Rough Sets and Knowledge Technology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 389–396.
- [84] E. Wan and R. van der Merwe. “The Unscented Kalman Filter”. In: ed. by S. Haykin. Wiley Publishing, 2001.

- [85] H Wang et al. “Entropy-based sensor selection heuristic for target localization”. In: *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*. Apr. 2004, pp. 36–45.
- [86] Brian P. Weaver et al. “Computational Enhancements to Bayesian Design of Experiments Using Gaussian Processes”. In: *Bayesian Analysis* 11.1 (2016), pp. 191–213.
- [87] X Wu, M Liu, and Y Wu. “In-situ Soil Moisture Sensing: Optimal Sensor Placement and Field Estimation”. In: *ACM Trans. Sen. Netw.* 8.4 (Sept. 2012), 33:1–33:30. ISSN: 1550-4859.