ENVIRONMENTAL ADAPTIVE SAMPLING FOR MOBILE SENSOR NETWORKS USING GAUSSIAN PROCESSES

By

Yunfei Xu

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Mechanical Engineering

2011

ABSTRACT

ENVIRONMENTAL ADAPTIVE SAMPLING FOR MOBILE SENSOR NETWORKS USING GAUSSIAN PROCESSES

By

Yunfei Xu

In recent years, due to significant progress in sensing, communication, and embedded-system technologies, mobile sensor networks have been exploited in monitoring and predicting environmental fields (*e.g.*, temperature, salinity, pH, or biomass of harmful algal blooms). The conventional inverse problem approach based on physical transport models is computationally prohibitive for resource-constrained, multi-agent systems. In contrast, emphasizing practicality and usefulness, this work relies extensively on the phenomenological and statistical modeling techniques, in particular, Gaussian processes. However, such statistical models need to be carefully tailored such that they can be practical and usable for mobile sensor networks with limited resources. In this dissertation, we consider the problem of using mobile sensor networks to estimate and predict environmental fields modeled by spatio-temporal Gaussian processes.

In the first part of the dissertation, we first present robotic sensors that learn a spatio-temporal Gaussian process and move in order to improve the quality of the estimated covariance function. For a given covariance function, we then theoretically justify the usage of truncated observations for Gaussian process regression for mobile sensor networks with limited resources. We propose both centralized and distributed navigation strategies for resource-limited mobile sensing agents to move in order to reduce prediction error variances at points of interest. Next, we formulate a fully Bayesian approach for spatio-temporal Gaussian process regression such that multifactorial effects of observations, measurement

noise, and prior distributions of hyperparameters are all correctly incorporated in the posterior predictive distribution. To cope with computational complexity, we design sequential Bayesian prediction algorithms in which exact predictive distributions can be computed in constant time as the number of observations increases. Under this formulation, we provide an adaptive sampling strategy for mobile sensors, using the maximum *a posteriori* (MAP) estimation to minimize the prediction error variances.

In the second part of the dissertation, we address the issue of computational complexity by exploiting the sparsity of the precision matrix used in a Gaussian Markov random field (GMRF). The main advantages of using GMRFs are: (1) the computational efficiency due to the sparse structure of the precision matrix, and (2) the scalability as the number of measurements increases. We first propose a new class of Gaussian processes that builds on a GMRF with respect to a proximity graph over the surveillance region, and provide scalable inference algorithms to compute predictive statistics. We then consider a discretized spatial field that is modeled by a GMRF with unknown hyperparameters. From a Bayesian perspective, we design a sequential prediction algorithm to exactly compute the predictive inference of the random field. An adaptive sampling strategy is also designed for mobile sensing agents to find the most informative locations in taking future measurements in order to minimize the prediction error and the uncertainty in the estimated hyperparameters simultaneously.

ACKNOWLEDGMENT

I would like to thank everybody who helped to accomplish this dissertation. First of all, I would like to express my sincere appreciation and gratitude to my advisor, Dr. Jongeun Choi, who continuously encouraged, and supported me. This thesis is not possible without his invaluable advice and effort to improve the work. In addition, I would like to thank all my committee members, Dr. Ranjan Mukherjee, Dr. Xiaobo Tan, Dr. Guoming Zhu, and Dr. Tapabrata Maiti, for their contribution and suggestions.

I would also like to thank my colleagues, Mr. Mahdi Jadaliha and Mr. Justin Mrkva, for their help to obtain experimental data.

Last but not least, I cannot thank enough my parents and my wife for always being there for me. I would never have been able to accomplish what I have without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In recent years, due to drastic global climate changes, it is necessary to monitor the changing ecosystems over vast regions in lands, oceans, and lakes. For instance, for certain environmental conditions, rapidly reproducing harmful algal blooms in the lakes can cause the death of nearby fish and produce harmful conditions to aquatic life as well as human beings[1]. Besides natural disasters, there exist growing ubiquitous possibilities of the release of toxic chemicals and contaminants in the air, lakes, and public water systems. Hence, there are strong motivations to monitor and predict the environmental field undergoing often complex transport phenomena[2].

In this dissertation, we consider the problem of using mobile sensor networks to estimate and predict environmental fields that are modeled by spatio-temporal Gaussian processes.

[1] See http://www.glerl.noaa.gov/res/Centers/HABS/habs.html for more details.

[2] Common examples are diffusion, convection, and advection.

1.1 Background

Due to the recent advances of micro-electro-mechanical systems technology, wireless communications and digital electronics, the concept of sensor networks has been made viable [2]. A sensor network consists of a collection of low-cost, low-power, multifunctional sensing devices that are small in size and communicate in short distances. Endowing the nodes in a sensor network with mobility increases the network's capabilities drastically [7]. The sensor networks which consist of mobile sensing agents are more flexible than the ones with only static nodes. For example, the mobility allows the network to handle a large number of data sources with a much smaller number of moving sensors that visit the sources over time.

In a mobile sensor network, the resource limited sensing agents are required to collaborate in order to meet a specific objective. The cooperative control becomes essential. The most popular applications are in networks of autonomous ground vehicles, underwater vehicles, or aerial vehicles. Emerging technologies have been reported on the coordination of mobile sensing agents [28,37,46,52,68,69]. The mobility of mobile agents can be designed in order to perform the optimal sampling of the field of interest. Optimal sampling design is the process of choosing where to take samples in order to maximize the information gained. Recently in [35], Leonard *et al.* developed mobile sensor networks that optimize ocean sampling performance defined in terms of uncertainty in a model estimate of a sampled field. However, this approach optimized the collective patterns of mobile agents parameterized by a restricted number of parameters rather than optimizing individual trajectories. In [10], distributed learning and cooperative control were developed for multi-agent systems to discover peaks of the unknown field based on the recursive estimation of an unknown field. A typical sensor placement technique [16] that puts sensors at the locations where the entropy is high

tends to place sensors along the borders of the area of interest [32]. In [32], Krause *et al.* showed that seeking sensor placements that are most informative about unsensed locations is NP-hard, and they presented a polynomial time approximation algorithm by exploiting the submodularity of mutual information [14]. In a similar approach, in [58], Singh *et al.* presented an efficient planning of informative paths for multiple robots that maximizes the mutual information.

To find these locations that predict the phenomenon best, one needs a model of the spatio-temporal phenomenon. To this end, we use the Gaussian processes (Gaussian random fields) to model fields undergoing transport phenomena. Nonparametric Gaussian process regression (or Kriging in geostatistics) has been widely used as a nonlinear regression technique to estimate and predict geostatistical data [15,23,38,51]. A Gaussian process is a natural generalization of the Gaussian probability distribution. It generalizes a Gaussian distribution with a finite number of random variables to a Gaussian process with an infinite number of random variables in the surveillance region [51]. Gaussian process modeling enables us to predict physical values, such as temperature, salinity, pH, or biomass of harmful algal blooms, at any point with a predicted uncertainty level efficiently. For instance, near-optimal static sensor placements with a mutual information criterion in Gaussian processes were proposed in [31,32]. A distributed Kriged Kalman filter for spatial estimation based on mobile sensor networks is developed in [13]. Multi-agent systems that are versatile for various tasks by exploiting predictive posterior statistics of Gaussian processes were developed in [9] and [8]. However, Gaussian process regression, based on the standard mean and covariance functions, requires an inversion of a covariance matrix whose size grows as the number of observations increases.

The advantage of a fully Bayesian approach is that the uncertainty in the model parameters are incorporated in the prediction [5]. In [22], Gaudard *et al.* presented a Bayesian method that uses importance sampling for analyzing spatial data sampled from a Gaussian random field whose covariance function was unknown. However, the solution often requires Markov Chain Monte Carlo (MCMC) methods, which greatly increases the computational complexity. In [25], an iterative prediction algorithm without resorting to MCMC methods has been developed based on analytical closed-form solutions from results in [22], by assuming that the covariance function of the spatio-temporal Gaussian random field is known up to a constant.

Recently, there have been efforts to find a way to fit a computationally efficient Gaussian Markov random field (GMRF) on a discrete lattice to a Gaussian random field on a continuum space [17,27,56]. Such methods have been developed using a fitting with a weighted L_2-type distance [56], using a conditional-mean least-squares fitting [17], and for dealing with large data by fast Kriging [27]. It has been demonstrated that GMRFs with small neighborhoods can approximate Gaussian fields surprisingly well [56]. This approximated GMRF and its regression are very attractive for the resource-constrained mobile sensor networks due to its computational efficiency and scalability [34] as compared to the standard Gaussian process and its regression, which is not scalable as the number of observations increases.

1.2 Contribution

Here, we summarize the specific contributions of this dissertation in the order of chapters.

In Chapter 3, we develop covariance function learning algorithms for the sensing agents to perform nonparametric prediction based on a properly adapted Gaussian process for a

given spatio-temporal phenomenon. By introducing a generalized covariance function, we expand the class of Gaussian processes to include the anisotropic spatio-temporal phenomena. Maximum likelihood (ML) optimization is used to estimate hyperparameters for the associated covariance function. The proposed optimal navigation strategy for autonomous vehicles will maximize the Fisher information [30], improving the quality of the estimated covariance function.

In Chapter 4, we first present a theoretical foundation of Gaussian process regression with truncated observations. In particular, we show that the quality of prediction based on truncated observations does not deteriorate much as compared to that of prediction based on all cumulative data under certain conditions. The error bounds to use truncated observations are analyzed for prediction at a single point of interest. A way to select the temporal truncation size for spatio-temporal Gaussian processes is also introduced. Inspired by the analysis, we then propose both centralized and distributed navigation strategies for mobile sensor networks to move in order to reduce prediction error variances at points of interest. In particular, we demonstrate that the distributed navigation strategy produces an emergent, swarming-like, collective behavior to maintain communication connectivity among mobile sensing agents.

In Chapter 5, we formulate a fully Bayesian approach for spatio-temporal Gaussian process regression under practical conditions such as measurement noise and unknown hyperparmeters (particularly, the bandwidths). Thus, multifactorial effects of observations, measurement noise and prior distributions of hyperparameters are all correctly incorporated in the computed posterior predictive distribution. Using discrete prior probabilities and compactly supported kernels, we provide a way to design sequential Bayesian prediction al-

gorithms that can be computed (without using the Gibbs sampler) in constant time as the number of observations increases. An adaptive sampling strategy for mobile sensors, using the maximum *a posteriori* (MAP) estimation, has been proposed to minimize the prediction error variances.

In Chapter 6, we propose a new class of Gaussian processes for resource-constrained mobile sensor networks that builds on a Gaussian Markov random field (GMRF) with respect to a proximity graph over the surveillance region. The main advantages of using this class of Gaussian processes over standard Gaussian processes defined by mean and covariance functions are its numerical efficiency and scalability due to its built-in GMRF and its capability of representing a wide range of non-stationary physical processes. The formulas for predictive statistics are derived and a sequential field prediction algorithm is provided for sequentially sampled observations. For a special case using compactly supported weighting functions, we propose a distributed algorithm to implement field prediction by correctly fusing all observations.

In Chapter 7, We then consider a discretized spatial field that is modeled by a GMRF with unknown hyperparameters. From a Bayesian perspective, we design a sequential prediction algorithm to exactly compute the predictive inference of the random field. The main advantages of the proposed algorithm are: (1) the computational efficiency due to the sparse structure of the precision matrix, and (2) the scalability as the number of measurements increases. Thus, the prediction algorithm correctly takes into account the uncertainty in hyperparameters in a Bayesian way and also is scalable to be usable for the mobile sensor networks with limited resources. An adaptive sampling strategy is also designed for mobile sensing agents to find the most informative locations in taking future measurements in or-

der to minimize the prediction error and the uncertainty in the estimated hyperparameters simultaneously.

1.3 Organization

This dissertation is organized as follows. In Chapter 2, we first introduce the basic mathematical notations that will be used throughout the thesis. Then, we describe the general Gaussian processes and its usage in nonparametric regression problems. The notations for mobile sensor networks are also introduced in Chapter 2. In Chapter 3, we deal with the case where hyperparameters in the covariance function is deterministic but unknown. We design an optimal sampling strategy to improve the maximum likelihood estimation of these hyperparameters. In Chapter 4, we assume the hyperparameters in the covariance function are given which can be obtained using the approach proposed in Chapter 3. We then analyze the error bounds of prediction error using Gaussian process regression with truncated observations. Inspired by the analysis, we propose both centralized and distributed navigation strategies for mobile sensor networks to move in order to reduce prediction error variances at points of interest. In Chapter 5, we consider a fully Bayesian approach for Gaussian process regression in which the hyperparameters are treated as random variables. Using discrete prior probabilities and compactly supported kernels, we provide a way to design sequential Bayesian prediction algorithms that can be computed in constant time as the number of observations increases. To cope with the computational complexity brought by using standard Gaussian processes with covariance functions, in Chapter 6, we exploit the sparsity of the precision matrix by using Gaussian Markov random fields (GMRF). We first introduce a new class of Gaussian processes with built-in GMRF and show its capability of representing a

wide range of non-stationary physical processes. We then derive the formulas for predictive statistics and design sequential prediction algorithms with fixed complexity. In Chapter 7, we consider a discretized spatial field that is modeled by a GMRF with unknown hyperparameters. From a Bayesian perspective, we design a sequential prediction algorithm to exactly compute the predictive inference of the random field. An adaptive sampling strategy is also designed for mobile sensing agents to find the most informative locations in taking future measurements in order to minimize the prediction error and the uncertainty in the estimated hyperparameters simultaneously.

1.4 Publication

In this section, I list journal articles and conference proceedings that have been published (or will be published) related to the topic of this dissertation. Some of the work will be described in the following chapters.

1.4.1 Journal Articles

(J1) **Yunfei Xu**, Jongeun Choi, Sarat Dass, and Taps Maiti, "Bayesian prediction and adaptive sampling algorithms for mobile sensor networks," *IEEE Transactions on Automatic Control*, (to appear, 2012).

(J2) **Yunfei Xu**, Jongeun Choi, "Spatial prediction with mobile sensor networks using Gaussian Markov random fields," *Automatica*, (in review, 2011).

(J3) **Yunfei Xu**, Jongeun Choi, and Songhwai Oh, "Mobile sensor network navigation using Gaussian processes with truncated observations," *IEEE Transactions on Robotics*, vol.

27, no. 6, pp. 1118-1131, 2011.

(J4) **Yunfei Xu**, Jongeun Choi, "Stochastic adaptive sampling for mobile sensor networks using kernel regression," *International Journal of Control, Automation and Systems*, (conditionally accepted, 2011).

(J5) **Yunfei Xu**, Jongeun Choi, "Adaptive sampling for learning Gaussian processes using mobile sensor networks," *Sensors*, vol. 11, no. 3, pp. 3051-3066, 2011.

(J6) Mahdi Jadaliha, **Yunfei Xu**, and Jongeun Choi, "Gaussian process regression for sensor networks under localization uncertainty," *IEEE Transactions on Signal Processing*, (in review, 2011).

(J7) Jongeun Choi,**Yunfei Xu**, Justin Mrkva, Joonho Lee, and Songhwai Oh, "Navigation strategies for swarm intelligence using spatio-temproal Gaussian processes," *Robotics and Autonomous Systems*, (in review, 2010).

1.4.2 Conference Proceedings

(C1) **Yunfei Xu**, Jongeun Choi, Sarat Dass, and Taps Maiti, "Efficient Bayesian spatial prediction with mobile sensor networks using Gaussian Markov random fields," *in Proceedings of the 2012 American Control Conference (ACC)*, June 27-29, Montréal, Canada. (in review).

(C2) Mahdi Jadaliha, **Yunfei Xu**, and Jongeun Choi, "Gaussian process regression using Laplace approximations under localization uncertainty," *in Proceedings of the 2012 American Control Conference (ACC)*, June 27-29, Montréal, Canada. (in review).

(C3) **Yunfei Xu**, Jongeun Choi, "Spatial prediction with mobile sensor networks using Gaussian Markov random fields," *in Proceedings of 2011 ASME Dynamic Systems and Control Conference (DSCC)*, October 31-November 2, 2011, Arlington, VA, USA.

(C4) **Yunfei Xu**, Jongeun Choi, Sarat Dass, and Taps Maiti, "Bayesian prediction and adaptive sampling algorithms for mobile sensor networks," *in Proceedings of the 2011 American Control Conference (ACC)*, June 29-July 1, 2011, San Francisco, California, USA.

(C5) Songhwai Oh, **Yunfei Xu**, and Jongeun Choi, "Explorative navigation of mobile sensor networks using sparse Gaussian processes," *in Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, December 15-17, 2010, Atlanta, Georgia, USA.

(C6) **Yunfei Xu**, Jongeun Choi, "Stochastic adaptive sampling for mobile sensor networks using kernel regression," *in Proceedings of the 2010 American Control Conference (ACC)*, June 20-July 2, 2010, Baltimore, Maryland, USA.

(C7) **Yunfei Xu**, Jongeun Choi, "Optimal coordination of mobile sensor networks using Gaussian processes," *in Proceedings of 2009 ASME Dynamic Systems and Control Conference (DSCC)*, October 12-14, 2009, Hollywood, California, USA.

(C8) **Yunfei Xu**, Jongeun Choi, "Mobile sensor networks for learning anisotropic gaussian processes," *in Proceedings of the 2009 American Control Conference (ACC)*, June 10-12, 2009, St. Louis, Missouri, USA.

Chapter 2

Preliminaries

2.1 Mathematical Notation

Standard notation is used throughout this dissertation. Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{Z}$, $\mathbb{Z}_{\geq 0}$, $\mathbb{Z}_{>0}$ denote the sets of real numbers, non-negative real numbers, positive real numbers, integers, non-negative integers, and positive integers, respectively.

Let E, Var, Corr, Cov denote the expectation operator, the variance operator, the correlation operator, and the covariance operator, respectively.

Let $A^T \in \mathbb{R}^{m \times n}$ be the transpose of a matrix $A \in \mathbb{R}^{n \times m}$. Let $\mathrm{tr}(A)$ and $\det(A)$ denote the trace and the determinant of a matrix $A \in \mathbb{R}^{n \times n}$, respectively. Let $\mathrm{row}_i(A) \in \mathbb{R}^m$ and $\mathrm{col}_j(A) \in \mathbb{R}^n$ denote the i-th row and the j-th column of a matrix $A \in \mathbb{R}^{n \times m}$, respectively.

The positive definiteness and the positive semi-definiteness of a square matrix A are denoted by $A \succ 0$ and $A \succeq 0$, respectively.

Let $|x|$ denote the absolute value of a scalar x. Let $\|x\|$ denote the standard Euclidean norm (2-norm) of a vector x. The induced 2-norm of a matrix A is denoted by $\|A\|$. Let

$\|x\|_\infty$ denote the infinity norm of a vector x.

Let 1 denote the vector with all elements equal to one and I denote the identity matrix with an appropriate size. Let e_i be the standard basis vector of appropriate size with 1 on the i-th element and 0 on all other elements.

The symbol $\otimes$ denotes the Kronecker product. The symbol $\circ$ denotes the Hadamard product (also known as the entry-wise product and the Schur product).

A random variable x, which is distributed by a normal distribution of mean μ and covariance matrix Σ, is denoted by $x \sim \mathcal{N}(\mu, \Sigma)$. The corresponding probability density function is denoted by $\mathcal{N}(x; \mu, \Sigma)$.

The relative complement of a set $\mathcal{A}$ in a set $\mathcal{B}$ is denoted by $\mathcal{B} \setminus \mathcal{A} := \mathcal{B} \cap \mathcal{A}^c$, where $\mathcal{A}^c$ is the complement of $\mathcal{A}$. For a set $\mathcal{A} \in \mathcal{I}$, we define $z_\mathcal{A} = \{z_i \,|\, i \in \mathcal{A}\}$. Let $-\mathcal{A}$ denote the set $\mathcal{I} \setminus \mathcal{A}$.

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a tuple consisting of a set of vertices $\mathcal{V} := \{1, \cdots, n\}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The neighbors of $i \in \mathcal{V}$ in $\mathcal{G}$ are denoted by $\mathcal{N}_i := \{j \in \mathcal{V} \,|\, \{i, j\} \in \mathcal{E}\}$.

Other notation will be explained in due course.

2.2 Physical Process Model

In this section, we review important notions on the Gaussian process which will be used to model the physical phenomenon. In particular, we introduce a class of spatio-temporal Gaussian process model with anisotropic covariance functions. The properties of Gaussian Markov Random fields (GMRF) are also briefly reviewed.

2.2.1 Gaussian process

A Gaussian process can be thought of a generalization of a Gaussian distribution over a finite vector space to function space of infinite dimension. It is formally defined as follows [50, 51]:

Definition 2.2.1. *A Gaussian process (GP) is a collection of random variables, any finite number of which have a consistent*[1] *joint Gaussian distribution.*

A Gaussian process

$$z(x) \sim \mathcal{GP}\left(\mu(x), C(x, x'; \theta)\right) \tag{2.1}$$

is completely specified by its mean function $\mu(x)$ and covariance function $C(x, x'; \theta)$ which are defined as

$$\mu(x) = \mathrm{E}\left[z(x)\right],$$
$$C(x, x'; \theta) = \mathrm{E}\left[\left(z(x) - \mu(x)\right)\left(z(x') - \mu(x')\right)|\theta\right].$$

Although not needed to be done, we take the mean function to be zero for notational simplicity[2], *i.e.*, $\mu(x) = 0$. If the covariance function $C(x, x'; \theta)$ is invariant to translations in the input space, *i.e.*, $C(x, x'; \theta) = C(x - x'; \theta)$, we call it stationary. Furthermore, if the covariance function is a function of only the distance between the inputs, *i.e.*, $C(x, x'; \theta) = C(\left\|x - x'\right\|; \theta)$, then it is called isotropic.

In practice, a parametric family of functions is used instead of fixing the covariance

[1] It is also known as the marginalization property. It means simply that the random variables obey the usual rules of marginalization, etc.

[2] This is not a drastic limitation since the mean of the posterior process is not confined to zero [51].

function [5]. One common choice of a stationary covariance function is

$$C(x, x'; \theta) = \sigma_f^2 \exp\left\{ -\sum_{\ell=1}^{D} \frac{\left(x_\ell - x'_\ell\right)^2}{2\sigma_\ell^2} \right\}, \tag{2.2}$$

where x_ℓ is the ℓ-th element of $x \in \mathbb{R}^D$. From (2.2), it can be easily seen that the correlation between two inputs decreases as the distance between them increases. This decreasing rate depends on the choice of the length scales $\{\sigma_\ell\}$. A very large length scale means that the predictions would have little bearing on the corresponding input which is then said to be insignificant. σ_f^2 gives the overall vertical scale relative to the mean of the Gaussian process in the output space. These parameters play the role of hyperparameters since they correspond to the hyperparameters in neural networks and in the standard parametric model. Therefore, we define $\theta = (\sigma_f^2, \sigma_1, \cdots, \sigma_D)^T \in \mathbb{R}^{D+1}$ as the hyperparameter vector. A realization of a Gaussian process that is numerically generated is shown in Fig. 2.1.

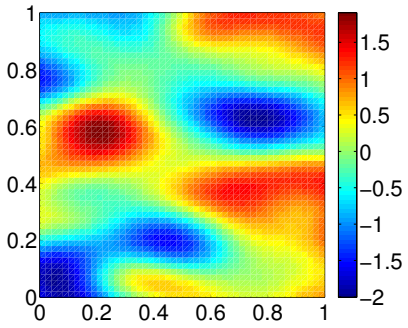


Figure 2.1: Realization of a Gaussian process. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

2.2.2 Spatio-temporal Gaussian process

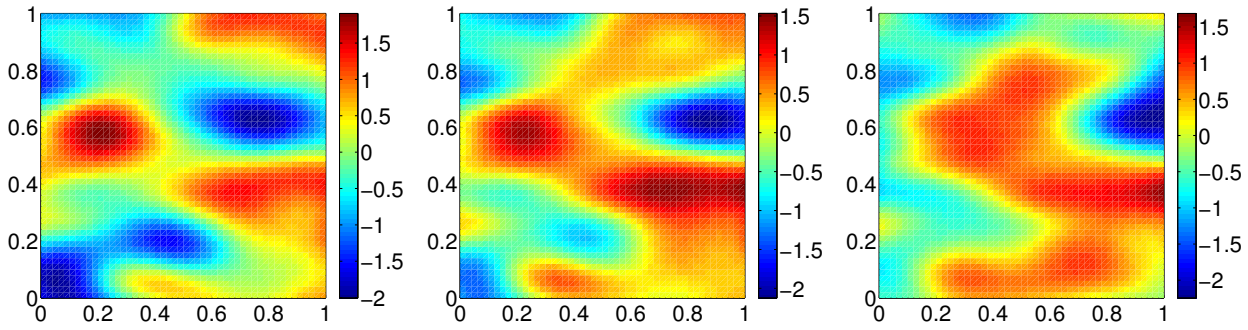
In the first part of this dissertation, spatio-temporal Gaussian processes are of particular interest. Consider a spatio-temporal Gaussian process

$$z(s,t) \sim \mathcal{GP}(\mu(s,t), C(s,t,s',t';\theta)),$$

which is a special case of the Gaussian process defined in (2.1), where $x = (s^T, t)^T \in \mathbb{R}^d \times \mathbb{R}_{\geq 0}$. We consider the following generalized anisotropic covariance function $C(x, x';\theta)$ with a hyperparameter vector $\theta := (\sigma_f^2, \sigma_1, \cdots, \sigma_d, \sigma_t)^T \in \mathbb{R}^{d+2}$:

$$C(x,x';\theta) = \sigma_f^2 \exp\left(-\sum_{\ell=1}^{d} \frac{(s_\ell - s'_\ell)^2}{2\sigma_\ell^2}\right) \exp\left(-\frac{(t-t')^2}{2\sigma_t^2}\right), \tag{2.3}$$

where $s, s' \in \mathcal{Q} \subset \mathbb{R}^d$, $t, t' \in \mathbb{R}_{\geq 0}$. $\{\sigma_1, \cdots, \sigma_d\}$ and σ_t are kernel bandwidths for space and time, respectively. (2.3) shows that points close in the measurement space and time indices are strongly correlated and produce similar values. In reality, the larger temporal distance two measurements are taken with, the less correlated they become, which strongly supports our generalized covariance function in (2.3). This may also justify the truncation (or windowing) of the observed time series data to limit the size of the covariance matrix for reducing the computational cost. A spatially isotropic version of the covariance function in (2.3) has been used in [35]. A realization of a spatio-temporal Gaussian process that is numerically generated is shown in Fig. 2.2.



(a) (b) (c)

Figure 2.2: Realization of Gaussian process at (a) $t = 1$, (b) $t = 5$, and (c) $t = 10$.

2.2.3 Gaussian Markov random field

The Gaussian Markov random field is formally defined as follows [54].

Definition 2.2.2. *(GMRF, [54, Definition 2.1]) A random vector $z = (z_1, \cdots, z_n)^T \in \mathbb{R}^n$ is called a GMRF with respect to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean μ and precision matrix $Q \succ 0$, if and only if its density has the form*

$$\pi(z) = \frac{|Q|^{1/2}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(z-\mu)^T Q(z-\mu)\right),$$

and $(Q)_{ij} \neq 0 \Leftrightarrow \{i, j\} \in \mathcal{E}$ for all $i \neq j$, where the precision matrix (or information matrix) $Q = C^{-1}$ is the inverse of the covariance matrix C, and $|Q|$ denotes the determinant of Q.

The Markov property of a GMRF can be shown by the following theorem.

Theorem 2.2.3. *([54, Theorem 2.4]) Let z be a GMRF with respect to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Then the followings are equivalent.*

1. *The pairwise Markov property:*

$$z_i \perp z_j \,|\, z_{-ij} \quad \text{if } \{i, j\} \notin \mathcal{E} \text{ and } i \neq j,$$

where $\perp$ denotes conditional independence and $z_{-ij} := z_{-\{i,j\}} = z_{\mathcal{I}\setminus\{i,j\}}$. This implies that z_i and z_j are conditionally independent given observations at all other vertices except $\{i, j\}$ if i and j are not neighbors.

2. *The local Markov property:*

$$z_i \perp z_{-\{i,\mathcal{N}_i\}} \,|\, z_{\mathcal{N}_i} \quad \text{for every } i \in \mathcal{I}.$$

3. *The global Markov property:*

$$z_{\mathcal{A}} \perp z_{\mathcal{B}} \,|\, z_{\mathcal{C}}$$

for disjoint sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ where $\mathcal{C}$ separates $\mathcal{A}$ and $\mathcal{B}$, and $\mathcal{A}$ and $\mathcal{B}$ are non-empty.

If a graph $\mathcal{G}$ has small cardinalities of the neighbor sets, its precision matrix Q becomes sparse with many zeros in its entries. This plays a key role in computation efficiency of a GMRF which can be greatly exploited by the resource-constrained mobile sensor network. For instance, some of the statistical inference can be obtained directly from the precision matrix Q with conditional interpretations.

Theorem 2.2.4. *([54, Theorem 2.3]) Let z be a GMRF with respect to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean μ and precision matrix $Q \succ 0$, then we have*

$$\begin{aligned}
\mathrm{E}(z_i \,|\, z_{-i}) &= \mu_i - \frac{1}{(Q)_{ii}} \sum_{j \in \mathcal{N}_i} (Q)_{ij}(z_j - \mu_j), \\
\mathrm{Var}(z_i \,|\, z_{-i}) &= \frac{1}{(Q)_{ii}}, \\
\mathrm{Corr}(z_i, z_j \,|\, z_{-ij}) &= -\frac{(Q)_{ij}}{\sqrt{(Q)_{ii}(Q)_{jj}}}, \quad \forall i \neq j.
\end{aligned}$$

2.3 Mobile Sensor Network

In this section, we explain the sensor network formed by multiple mobile sensing agents and present the measurement model used throughout the thesis.

Let N be the number of sensing agents distributed over the surveillance region $\mathcal{Q} \in \mathbb{R}^d$. The identity of each agent is indexed by $\mathcal{I} := \{1, 2, \cdots, N\}$. Assume that all agents are equipped with identical sensors and take noisy observations at time $t \in \mathbb{Z}_{>0}$. At time t, the sensing agent i takes a noise-corrupted measurement $y_i(t)$ at its current location $q_i(t) \in \mathcal{Q}$, *i.e.*,

$$y_i(t) = z(q_i(t), t) + \epsilon_i, \quad \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_w^2),$$

where the sensor noise ϵ_i is considered to be an independent and identically distributed Gaussian random variable. $\sigma_w^2 > 0$ is the noise level and we define the signal-to-noise ratio as

$$\gamma = \frac{\sigma_f^2}{\sigma_w^2}.$$

Notice that when a static field is considered, we have $z(s, t) = z(s)$.

For notational simplicity, we denote the collection of positions of all N agents at time t as $q(t)$, *i.e.*,

$$q(t) := \left(q_1(t)^T, \cdots, q_N(t)^T\right)^T \in \mathcal{Q}^N.$$

The collective measurements from all N mobile sensors at time t is denoted by

$$y_t := (y_1(t), \cdots, y_N(t))^T \in \mathbb{R}^N.$$

The cumulative measurements from time $t \in \mathbb{Z}_{>0}$ to time $t' \in \mathbb{Z}_{>0}$ is denoted by

$$y_{t:t'} := \left(y_t^T, \cdots, y_{t'}^T\right)^T \in \mathbb{R}^{N(t'-t+1)}.$$

The communication network of mobile agents can be represented by an undirected graph. Let $\mathcal{G}(t) := (\mathcal{I}, \mathcal{E}(t))$ be an undirected communication graph such that an edge $(i,j) \in \mathcal{E}(t)$ if and only if agent i can communicate with agent $j \neq i$ at time t. We define the neighborhood of agent i at time t by $\mathcal{N}_i(t) := \{j \in \mathcal{I} \,|\, (i,j) \in \mathcal{E}(t)\}$. Similarly, let $q^{[i]}(t)$ denote the vector form of the collection of positions in $\left\{q_j(t) \,|\, j \in \{i\} \cup \mathcal{N}_i(t)\right\}$. Let $y_t^{[i]}$ denote vector form of the collection of observations in $\left\{y(q_j(t),t) \,|\, j \in \{i\} \cup \mathcal{N}_i(t)\right\}$. The cumulative measurements of agent i from time t to time t' as $y_{t:t'}^{[i]}$.

2.4 Gaussian Processes for Regression

Suppose we have a data set $\mathcal{D} = \left\{(x^{(i)}, y^{(i)}) \,|\, i = 1, \cdots, n\right\}$ collected by mobile sensing agents where $x^{(i)}$ denotes an input vector of dimension D and $y^{(i)}$ denotes a scalar value of the noise corrupted output. The objective of probabilistic regression is to compute the predictive distribution of the function values $z_* := z(x_*)$ at some test input x_*.

For notational simplicity, we define the design matrix X of dimension $n \times D$ as the aggregation of n input vectors (*i.e.*, $\text{row}_i(X) := (x^{(i)})^T$), and the outputs are collected in a vector $y := (y^{(1)}, \cdots, y^{(n)})^T$. The corresponding vector of noise-free outputs is defined as $z := (z(x^{(1)}), \cdots, z(x^{(n)}))^T$.

The advantage of the Gaussian process formulation is that the combination of the prior and noise models can be carried out exactly via matrix operations [62]. The idea of Gaus-

sian process regression is to place a GP prior directly on the space of functions without parameterizing the function $z(\cdot)$, *i.e.*,

$$\pi(z|\theta) = \mathcal{N}(\mu, K),$$

where $\mu \in \mathbb{R}^n$ is the mean vector obtained by $(\mu)_i = \mu(x^{(i)})$, and $K := \mathrm{Cov}(z, z|\theta) \in \mathbb{R}^{n\times n}$ is the covariance matrix obtained by $(K)_{ij} = C(x^{(i)}, x^{(j)}; \theta)$. Notice that the GP model and all expressions are always conditional on the corresponding inputs. In the following, we will always neglect the explicit conditioning on the input matrix X.

The inference in the Gaussian process model is as follows. First, we assume a joint GP prior $\pi(z, z_*|\theta)$ over functions, *i.e.*,

$$\pi(z, z_*|\theta) = \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu(x_*) \end{bmatrix}, \begin{bmatrix} K & k \\ k^T & C(x_*, x_*; \theta) \end{bmatrix}\right), \tag{2.4}$$

where $k := \mathrm{Cov}(z, z_*|\theta) \in \mathbb{R}^n$ is the covariance between z and z_* obtained by $(k)_i = C(x^{(i)}, x_*; \theta)$. Then, the joint posterior is obtained using Bayes rule, *i.e.*,

$$\pi(z, z_*|\theta, y) = \frac{\pi(y|z)\pi(z, z_*|\theta)}{\pi(y|\theta)},$$

where we have used $\pi(y|z, z_*) = \pi(y|z)$. Finally, the desired predictive distribution $\pi(z_*|\theta, y)$ is obtained by marginalizing out the latent variables in z, *i.e.*,

$$\begin{aligned} \pi(z_*|\theta, y) &= \int \pi(z, z_*|\theta, y)dz \\ &= \frac{1}{\pi(y|\theta)} \int \pi(y|z)\pi(z, z_*|\theta, y)dz. \end{aligned} \tag{2.5}$$

Since we have the joint Gaussian prior given in (2.4) and

$$y|z \sim \mathcal{N}\left(z, \sigma_w^2 I\right),$$

the integral in (2.5) can be evaluated in closed-form and the predictive distribution turns out to be Gaussian, *i.e.*,

$$z_*|\theta, y \sim \mathcal{N}\left(\mu_{z_*|\theta,y}, \sigma^2_{z_*|\theta,y}\right), \tag{2.6}$$

where

$$\mu_{z_*|\theta,y} = \mu(x_*) + k^T(K + \sigma_w^2 I)^{-1}(y - \mu), \tag{2.7}$$

and

$$\sigma^2_{z_*|\theta,y} = C(x_*, x_*; \theta) - k^T(K + \sigma_w^2 I)^{-1}k. \tag{2.8}$$

For notational simplicity, we define the covariance matrix of the noisy observations as $C := \mathrm{Cov}(y, y|\theta) = K + \sigma_w^2 I$.

Chapter 3

Learning the Covariance Function

Even though, there have been efforts to utilize Gaussian processes to model and predict the spatio-temporal field of interest, most of recent papers assume that Gaussian processes are isotropic implying that the covariance function only depends on the distance between locations. Many studies also assume that the corresponding covariance functions are known *a priori* for simplicity. However, this is not the case in general as pointed out in literature [31, 32, 44], in which they treat the non-stationary process by fusing a collection of isotropic spatial Gaussian processes associated with a set of local regions. Hence, our objective in this Chapter is to develop theoretically-sound algorithms for mobile sensor networks to learn the anisotropic covariance function of a spatio-temporal Gaussian process. Mobile sensing agents can then predict the Gaussian process based on the estimated covariance function in a nonparametric manner.

In Section 3.1, we introduce a covariance function learning algorithm for an anisotropic, spatio-temporal Gaussian process. The covariance function is assumed to be deterministic but unknown *a priori* and it is estimated by the maximum likelihood (ML) estimator. In

Section 3.2, an optimal sampling strategy is proposed to minimize the Cramér-Rao lower bound (CRLB) of the estimation error covariance matrix. In Section 3.3, simulation results illustrate the usefulness of our proposed approach and its adaptability for unknown and/or time-varying covariance functions.

3.1 Learning the Hyperparameters

Without loss of generality, we consider a zero-mean spatio-temporal Gaussian process

$$z(s,t) \sim \mathcal{GP}\left(0, C(s,t,s',t';\theta)\right),$$

with the covariance function

$$C(s,t,s',t';\theta) = \sigma_f^2 \exp\left(-\sum_{\ell\in\{x,y\}} \frac{(s_\ell - s'_\ell)^2}{2\sigma_\ell^2}\right) \exp\left(-\frac{(t-t')^2}{2\sigma_t^2}\right),$$

where $s, s' \in \mathcal{Q} \subset \mathbb{R}^2$, $t, t' \in \mathbb{R}_{\geq 0}$, for modeling the field undergoing a physical transport phenomenon. $\theta = (\sigma_f, \sigma_x, \sigma_y, \sigma_t)^T \in \mathbb{R}^m$ is the hyperparameter vector, where $m = 4$. The assumption of zero-mean is not a strong limitation since the mean of the posterior process is not confined to zero [51].

If the covariance function $C(s,t,s',t';\theta)$ of a Gaussian process is not known *a priori*, mobile agents need to estimate parameters of the covariance function (*i.e.*, the hyperparameter vector $\theta \in \mathbb{R}^m$) based on the observed samples. In the case where measurement noise level σ_w is also unknown, it can be incorporated in the hyperparameter vector and be estimated. Thus, we have $\theta = (\sigma_f, \sigma_x, \sigma_y, \sigma_t, \sigma_w)^T \in \mathbb{R}^m$ where $m = 5$.

Existing techniques for learning the hyperparamters are based on the likelihood function. Given the observations $y = (y^{(1)}, \cdots, y^{(n)})^T \in \mathbb{R}^n$ collected by mobile sensing agents, the likelihood function is defined as

$$L(\theta|y) = \pi(y|\theta).$$

Notice that in this chapter, the hyperparameter vector θ is considered to be deterministic, and hence $\pi(y|\theta)$ should not be considered as conditional distribution.

At time t, a point estimate of the hyperparameter vector θ can be made by maximizing the log likelihood function. The maximum likelihood (ML) estimate $\hat{\theta} \in \mathbb{R}^m$ of the hyperparameter vector is obtained by

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \log L(\theta|y), \tag{3.1}$$

where Θ is the set of all possible choices of θ. The log likelihood function is given by

$$\log L(\theta|y) = -\frac{1}{2} y^T C^{-1} y - \frac{1}{2} \log\det(C) - \frac{n}{2} \ln 2\pi,$$

where $C := \mathrm{Cov}(y, y|\theta) \in \mathbb{R}^{n \times n}$ is the covariance matrix, and n is the total number of observations. Maximization of the log likelihood function can be done efficiently using gradient-based optimization techniques such as the conjugate gradient method [26, 43]. The partial derivative of the log likelihood function with respect to a hyperparameter $\theta_i \in \mathbb{R}$, *i.e.*, the i-th entry of the hyperparameter vector θ, is given by

$$\begin{aligned} \frac{\partial \ln L(\theta|y)}{\partial \theta_i} &= \frac{1}{2} y^T C^{-1} \frac{\partial C}{\partial \theta_i} C^{-1} y - \frac{1}{2} \operatorname{tr}\left(C^{-1} \frac{\partial C}{\partial \theta_i} \right) \\ &= \frac{1}{2} \operatorname{tr}\left((\alpha\alpha^T - C^{-1}) \frac{\partial C}{\partial \theta_i} \right), \end{aligned}$$

where $\alpha = C^{-1}y \in \mathbb{R}^n$. In general, the log likelihood function is a non-convex function and hence it can have multiple maxima.

As an alternative, when certain prior knowledge is available on the hyperparameters, a prior distribution can be imposed on the hyperparameter vector, *i.e.,* $\pi(\theta)$. Using Bayes' rule, the posterior distribution $\pi(\theta|y)$ is proportional to the likelihood $\pi(y|\theta)$ times the prior distribution $\pi(\theta)$, *i.e.,*

$$\pi(\theta|y) \propto \pi(y|\theta)\pi(\theta).$$

Then the maximum *a posteriori* (MAP) estimate $\hat{\theta} \in \mathbb{R}^m$ of the hyperparameter vector can be obtained similarly by

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \left(\log L(\theta|y) + \log \pi(\theta)\right). \tag{3.2}$$

Notice that when no prior information is available, the MAP estimate is equivalent to the ML estimate.

Once the estimate of the hyperparameter vector θ is obtained with confidence, it can be used as the true one for the mobile sensor network to predict the field of interest using Gaussian process regression in (2.6).

3.2 Optimal Sampling Strategy

Agents should find new sampling positions to improve the quality of the estimated covariance function in the next iteration at time $t+1$. For instance, to precisely estimate the anisotropic phenomenon, *i.e.*, processes with different covariances along x and y directions, sensing agents need to explore and sample measurements along different directions.

To this end, we consider a centralized scheme. Suppose that a central station (or a leader agent) has access to all measurements collected by agents. Assume that at time $t+1$, agent i moves to a new sampling position $\tilde{q}_i \in \mathcal{Q}$ and make an observation $y_i(t+1) \in \mathbb{R}$. The collection of the new sampling positions and new observations from all agents are denoted by $\tilde{q} \in \mathcal{Q}^N$ and $\tilde{y} \in \mathbb{R}^N$, respectively. The objective of the optimal sampling strategy is to find the best sampling positions $\tilde{q}$ such that the maximum likelihood (ML) estimate $\hat{\theta}_{t+1} \in \mathbb{R}^m$ at time $t+1$ is as close to the true hyperparameter vector $\theta^* \in \mathbb{R}^m$ as possible.

Consider the Fisher information matrix (FIM) that measures the information produced by $y_{1:t} \in \mathbb{R}^{Nt}$ and $\tilde{y} \in \mathbb{R}^N$ for estimating the true hyperparameter vector $\theta^* \in \mathbb{R}^m$ at time $t+1$. The Cramér-Rao lower bound (CRLB) theorem states that the inverse of the Fisher information matrix (denoted by $M \in \mathbb{R}^{m \times m}$) is a lower bound of the estimation error covariance matrix [30, 39]:

$$\mathrm{E}\left[(\hat{\theta}_{t+1} - \theta^*)(\hat{\theta}_{t+1} - \theta^*)^T\right] \succeq M^{-1},$$

where $\hat{\theta}_{t+1} \in \mathbb{R}^m$ represents the ML estimate of θ^* at time $t+1$. The Fisher information matrix (FIM) [30] is given by

$$(M)_{ij} = -\,\mathrm{E}\left[\frac{\partial^2 \ln L(\theta|\tilde{y}, y_{1:t})}{\partial\theta_i \partial\theta_j}\right],$$

where $L(\theta|\tilde{y}, y_{1:t})$ is the likelihood function at time $t+1$, and the expectation is taken with respect to $\pi(y_{1:t}, \tilde{y}|\theta)$. Notice that the likelihood is now a function of θ and $\tilde{y}$. The analytical

form of the FIM is given by

$$(M)_{ij} = \frac{1}{2} \operatorname{tr} \left(\tilde{C}^{-1} \frac{\partial \tilde{C}}{\partial \theta_i} \tilde{C}^{-1} \frac{\partial \tilde{C}}{\partial \theta_j} \right),$$

where $\tilde{C} \in \mathbb{R}^{N(t+1) \times N(t+1)}$ is defined as

$$\tilde{C} := \operatorname{Cov} \left(\begin{bmatrix} y_{1:t} \\ \tilde{y} \end{bmatrix}, \begin{bmatrix} y_{1:t} \\ \tilde{y} \end{bmatrix} \middle| \theta^* \right).$$

Since the true value θ^* is not available, we will evaluate the FIM at the currently best estimate $\hat{\theta}_t$.

We can expect that minimizing the Cramér-Rao lower bound results in a decrease of uncertainty in estimating θ [41]. The most common optimality criterion is D-optimality [21, 49]. It corresponds to minimizing the volume of the ellipsoid which represents the maximum confidence region for the maximum likelihood estimate of the unknown hyperparamters [21]. Using the D-optimality criterion [21, 49], the objective function $J(\cdot)$ is given by

$$J(\tilde{q}) := \det(M^{-1}).$$

However, if one hyperparamter has a very large variance compared to the others, the ellipsoid will be skinny and thus minimizing the volume may be misleading [21]. As an alternative, A-optimality which minimizes the sum of the variances is often used. The objective function $J(\cdot)$ based on A-optimality criterion is

$$J(\tilde{q}) := \operatorname{tr}(M^{-1}).$$

Hence, a control law for the mobile sensor network can be formulated as follows:

$$q(t+1) = \arg \min_{\tilde{q} \in \mathcal{Q}^N} J(\tilde{q}). \tag{3.3}$$

In (3.3), we only consider the constraint that robots should move within the region $\mathcal{Q}$. However, the mobility constraints, such as the maximum distance that a robot can move between two time indices, or the maximum speed with which a robot can travel, can be incorporated as additional constraints in the optimization problem [13].

The overall protocol for the sensor network is summarized as in Table 3.1.

Table 3.1: Centralized optimal sampling strategy at time t.

For $i \in \mathcal{I}$, agent i performs:

1: make an observation at current position $q_i(t)$, *i.e.*, $y_i(t)$
2: transmit the observation $y_i(t)$ to the central station

The central station performs:

1: collect the observations from all N agents, *i.e.*, y_t
2: obtain the cumulative measurements, *i.e.*, $y_{1:t}$
3: compute the maximum likelihood estimate $\hat{\theta}_t$ based on
$$\hat{\theta}_t = \arg \max_{\theta \in \Theta} \ln L(\theta | y_{1:t}),$$
starting with the initial point $\hat{\theta}_{t-1}$
4: compute the control in order to minimize the cost function $J(\tilde{q})$ via
$$q(t+1) = \arg \min_{\tilde{q} \in \mathcal{Q}^N} J(\tilde{q})$$
5: send the next sampling positions $\{q_i(t+1) \,|\, i \in \mathcal{I}\}$ to all N agents

For $i \in \mathcal{I}$, agent i performs:

1: receive the next sampling position $q_i(t+1)$ from the central station
2: move to $q_i(t+1)$ before time $t+1$

3.3 Simulation

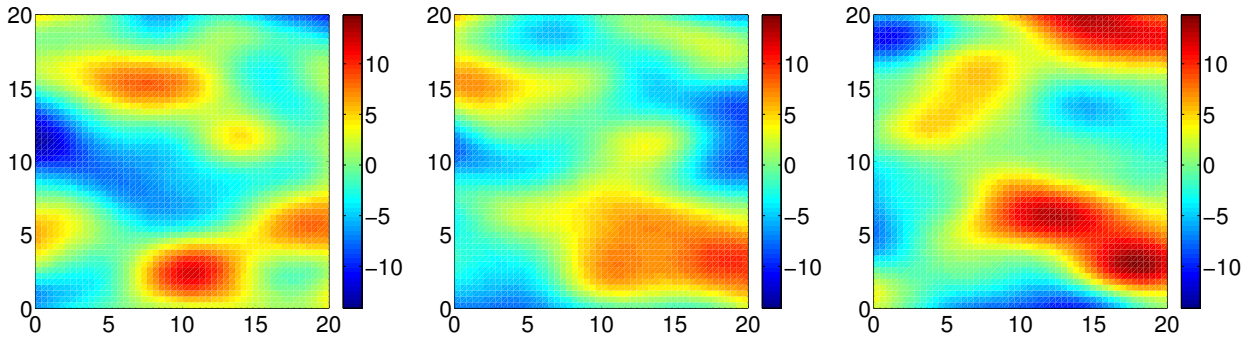
In this section, we evaluate the proposed approach for a spatio-temporal Gaussian process (Section 3.3.1) and an advection-diffusion process (Section 3.3.3). For both cases, we compare the simulation results using the proposed optimal sampling strategy with results using a benchmark random sampling strategy. In this random sampling strategy, each agent was initially randomly deployed in the surveillance region $\mathcal{Q}$. At time $t \in \mathbb{Z}_{>0}$, the next sampling position for agent i is generated randomly with the same mobility constraint, *viz.* a random position within a square region with length 2 centered at the current position $q_i(t)$. For fair comparison, the same values are used for all other conditions. In Section 3.3.2, our approach based on truncated observations is applied to a Gaussian process with a time-varying covariance function to demonstrate the adaptability of the proposed scheme.

3.3.1 Spatio-temporal Gaussian process

We apply our approach to a spatio-temporal Gaussian process. The Gaussian process was numerically generated for the simulation [51]. The hyperparameters used in the simulation were chosen such that $\theta = (\sigma_f, \sigma_x, \sigma_y, \sigma_t, \sigma_w)^T = (5, 4, 2, 8, 0.5)^T$. Snap shots of the realized Gaussian random field are shown in Fig. 3.1. In this case, $N = 5$ mobile sensing agents were initialized at random positions in a surveillance region $\mathcal{Q} = [0, 20] \times [0, 20]$. The initial values for the algorithm were given to be $\theta_0 = (1, 10, 10, 1, 0.1)^T$. A prior of the hyperparameter vector has been selected as

$$\pi(\theta) = \pi(\sigma_f)\pi(\sigma_x)\pi(\sigma_y)\pi(\sigma_t)\pi(\sigma_w),$$

where $\pi(\sigma_f) = \pi(\sigma_x) = \pi(\sigma_y) = \pi(\sigma_t) = \Gamma(5, 2)$, and $\pi(\sigma_w) = \Gamma(5, 0.2)$. $\Gamma(a, b)$ is a Gamma distribution with mean ab and variance ab^2 in which all possible values are positive. The gradient method was used to find the MAP estimate of the hyperparameter vector.



(a) (b) (c)

Figure 3.1: Snap shots of the realized Gaussian process at (a) $t = 1$, (b) $t = 10$, and (c) $t = 20$.

For simplicity, we assumed that the global basis is the same as the model basis. We considered a situation where at each time, measurements of agents are transmitted to a leader (or a central station) that uses our Gaussian learning algorithm and sends optimal control back to individual agents for next iteration to improve the quality of the estimated covariance function. The maximum distance for agents to move in one time step was chosen to be 1 for both x and y directions. The A-optimality criterion was used for optimal sampling.

For both proposed and random strategies, Monte Carlo simulations were run for 100 times and the statistical results are shown in Fig. 3.2. The estimates of the hyperparameters (shown in circles and error bars) tend to converge to the true values (shown in dotted lines) for both strategies. As can be seen, the proposed scheme (Fig. 3.2(a)) outperforms the random strategy (Fig. 3.2(b)) in terms of the A-optimality criterion.

Fig. 3.3 shows the predicted field along with agents' trajectories at time $t = 1$ and $t = 20$ for one trial. As shown in Fig. 3.1(a) and Fig. 3.3(a), at time $t = 1$, the predicted field is far

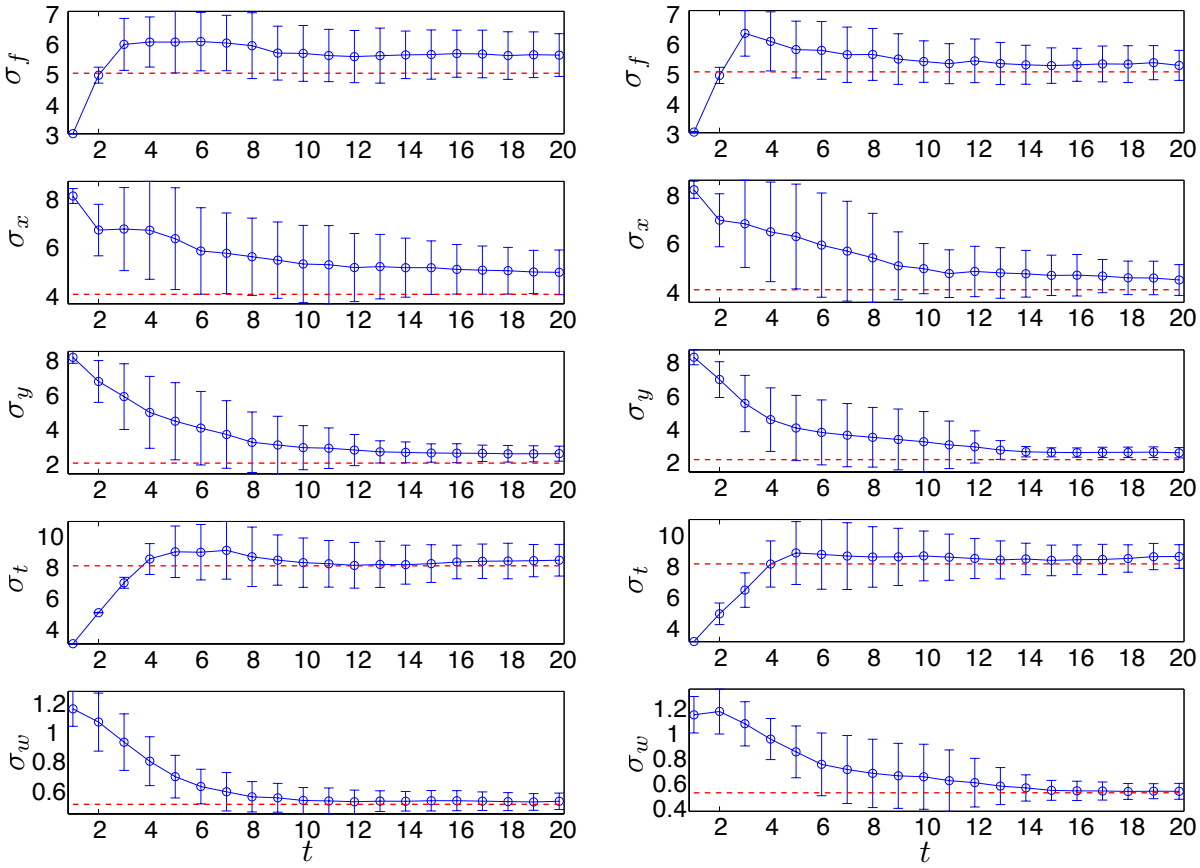


Figure 3.2: Monte Carlo simulation results (100 runs) for a spatio-temporal Gaussian process using (a) the random sampling strategy, and (b) the adaptive sampling strategy. The estimated hyperparameters are shown in blue circles with error-bars. The true hyperparameters that used for generating the process are shown in red dashed lines.

from the true field due to the inaccurate hyperparameters estimation and small number of observations. As time increases, the predicted field will be closer to the true field due to the improved quality of the estimated the covariance function and the cumulative observations. As expected, at time $t = 20$, the quality of the predicted field is very well near the sampled positions as shown in Fig. 3.3-(b). With 100 observations, the running time is around $30s$ using Matlab, R2008a (MathWorks) in a PC (2.4 GHz Dual-Core Processor). No attempt has been made to optimize the code. After converging to a good estimate of θ, agents can

switch to a decentralized configuration and collect samples for other goals such as peak tracking and prediction of the process [8–10].

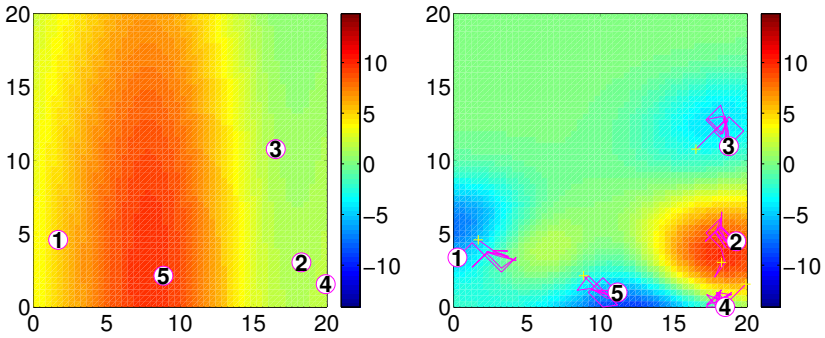


Figure 3.3: Predicted fields along with agents' trajectories at (a) $t = 1$ and (b) $t = 20$.

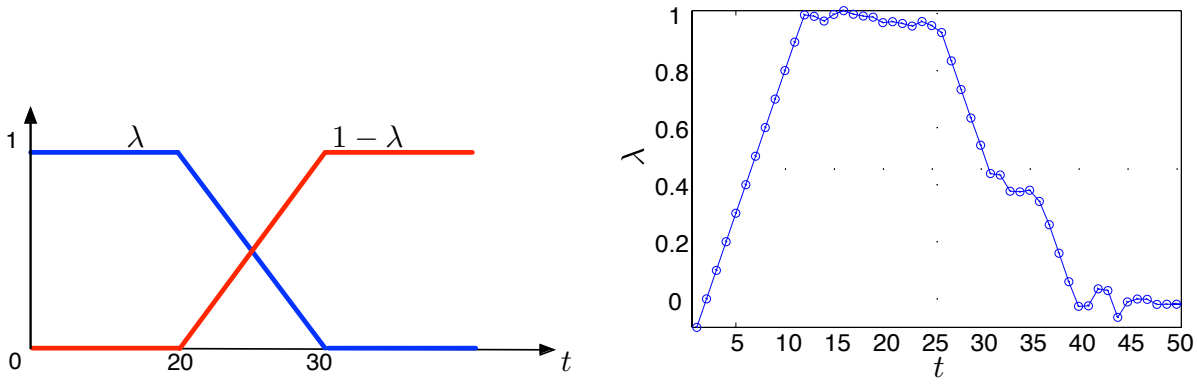
3.3.2 Time-varying covariance functions

To illustrate the adaptability of the proposed strategy to time-varying covariance functions, we introduce a Gaussian process defined by the following covariance function. The time-varying covariance function is modeled by a time-varying weighted sum of two known covariance functions $C_1(\cdot,\cdot)$ and $C_2(\cdot,\cdot)$ such as

$$C(\cdot,\cdot) = \lambda(t)C_1(\cdot,\cdot) + (1-\lambda(t))C_2(\cdot,\cdot), \tag{3.4}$$

where $\lambda(t) \in [0,1]$ is a time-varying weight factor that needs to be estimated. In the simulation study, $C_1(\cdot,\cdot)$ is constructed with $\sigma_f = 1$, $\sigma_x = 0.2$, $\sigma_y = 0.1$, $\sigma_t = 8$, and $\sigma_w = 0.1$; and $C_2(\cdot,\cdot)$ is with $\sigma_f = 1$, $\sigma_x = 0.1$, $\sigma_y = 0.2$, $\sigma_t = 8$, and $\sigma_w = 0.1$. This Gaussian process defined in (3.4) with theses particular C_1 and C_2 effectively models hyperparameter changes in x and y directions.

To improve the adaptability, the mobile sensor network uses only observations sampled during the last 20 iterations for estimating $\lambda(t)$ online. The true $\lambda(t)$ and the estimated $\lambda(t)$ are shown in Fig. 3.4(a), and (b), respectively. From Fig. 3.4, it is clear that the weighting factor $\lambda(t)$ can be estimated accurately after some delay about 5–8 iterations. The delay is due to using the truncated observations that contain past observations since the time-varying covariance function changes continuously in time.



(a) (b)

Figure 3.4: (a) Weighting factor $\lambda(t)$ and (b) the estimated $\lambda(t)$.

3.3.3 Advection-diffusion process

We apply our approach to a spatio-temporal process generated by physical phenomena (advection and diffusion). This work can be viewed as a statistical modeling of a physical process, *i.e.*, as an effort to fit a Gaussian process to a physical advection-diffusion process in practice. The advection-diffusion model developed in [29] was used to generate the experimental data numerically. An instantaneous release of Qkg of gas occurs at a location (x_0, y_0, z_0). This is then spread by the wind with mean velocity $u = (u_x, 0, 0)^T$ Assuming that all measurements are recorded at a level $z = 0$, and the release occurs at a ground level

Table 3.2: Parameters used in simulation.

Parameter	Notation	Unit	Value
Number of agents	N_s	-	5
Sampling time	t_s	min	5
Initial time	t_0	min	100
Gas release mass	Q	kg	10^6
Wind velocity in x axis	u_x	m/min	0.5
Eddy diffusivity in x axis	K_x	m^2/min	20
Eddy diffusivity in y axis	K_y	m^2/min	10
Eddy diffusivity in z axis	K_z	m^2/min	0.2
Location of explosion	x_0	m	2
Location of explosion	y_0	m	5
Location of explosion	z_0	m	0
Sensor noise level	σ_w	kg/m^3	0.1

(*i.e.*, $z_0 = 0$), the concentration C at an arbitrary location $(x, y, 0)$ and time t is described by the following analytical solution [11]:

$$C(x, y, 0, t) = \frac{Q \exp\left(-\frac{(\Delta x - u\Delta t)^2}{4K_x\Delta t} - \frac{\Delta y^2}{4K_y\Delta t}\right)}{4\pi^{\frac{3}{2}}(K_xK_yK_z)^{\frac{1}{2}}(\Delta t)^{\frac{3}{2}}} \tag{3.5}$$

where $\Delta x = x - x_0$, $\Delta y = y - y_0$, and $\Delta t = 5(t-1) + t_0$. The parameters used in the simulation study are shown in Table 3.2. Notice that this process generates an anisotropic concentration field with parameters $K_x = 20m^2/min$ and $K_y = 10m^2/min$ as in Table 3.2. The fields at time $t = 1$ and $t = 10$ are shown in Fig. 3.5. Notice the center of the concentration moved. In this case, $N = 5$ mobile sensing agents were initialized at random positions in a surveillance region $\mathcal{Q} = [-50, 150] \times [-100, 100]$.

The initial values for the algorithm was chosen to be $\theta_0 = (100, 100, 100)^T$ where we assumed $\sigma_f = 1$ and $\sigma_w = 0.1$. For this application, we did not assume any prior knowledge about the covariance function. Hence, the MAP estimator was the same as the ML estimator.

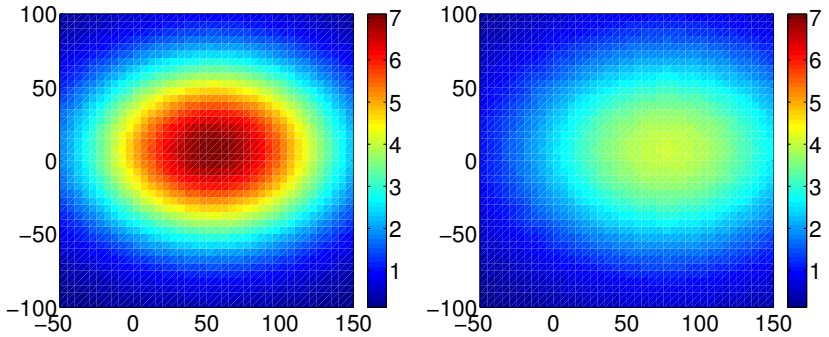


Figure 3.5: Snap shots of the advection-diffusion process at (a) $t = 1$ and (b) $t = 10$.

The gradient method was used to find the ML estimate.

We again assumed that the global basis is the same as the model basis and assumed all agents have the same level of measurement noises for simplicity. In our simulation study, agents start sampling at $t_0 = 100min$ and take measurements at time t with a sampling time of $t_s = 5min$ as in Table 3.2.

Monte Carlo simulations were run for 100 times, and Fig. 3.6 shows the estimated σ_x, σ_y, and σ_t with (a) the random sampling strategy and (b) the optimal sampling strategy, respectively. With 100 observations, the running time at each time step is around $20s$ using Matlab, R2008a (MathWorks) in a PC (2.4 GHz Dual-Core Processor). No attempt has been made to optimize the code. As can be seen in Fig. 3.6, the estimates of the hyperparameters tend to converge to similar values for both strategies. Clearly, the proposed strategy outperforms the random sampling strategy in terms of the estimation error variance.

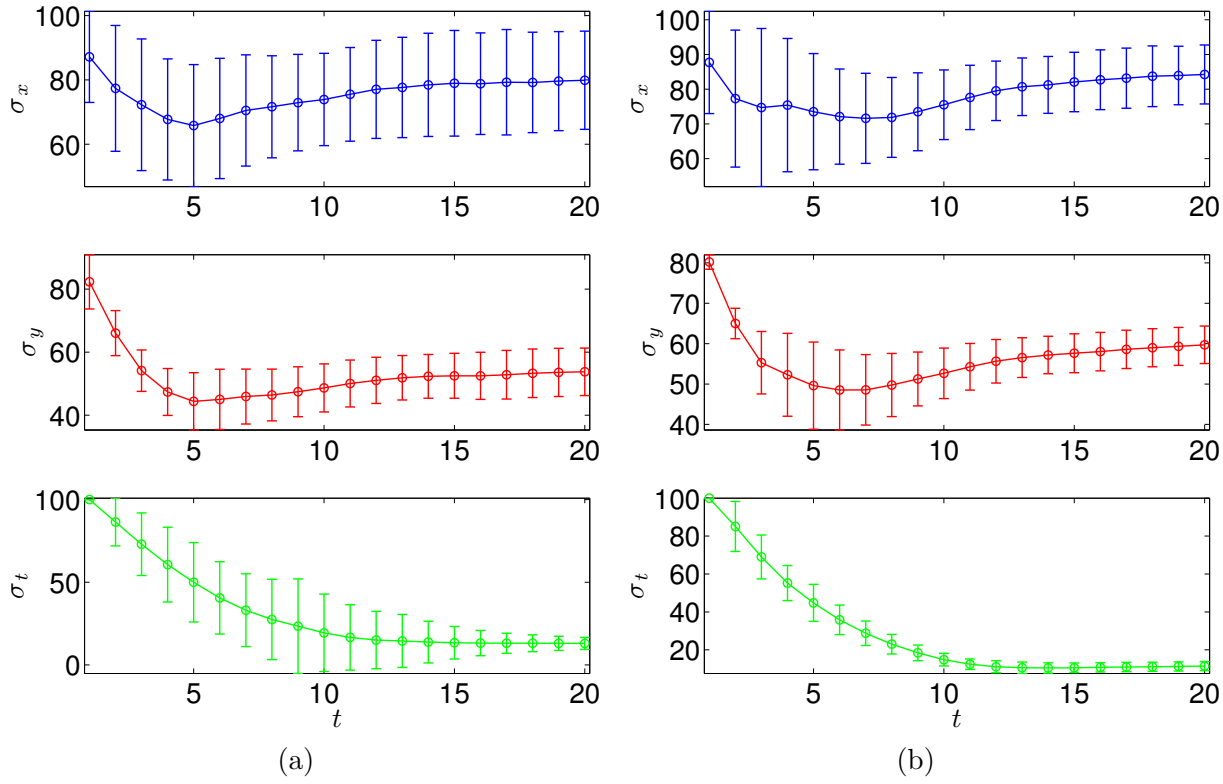


Figure 3.6: Simulation results (100 runs) for a advection-diffusion process. The estimated hyperparameters with (a) random sampling and (b) optimal sampling.

Chapter 4

Prediction with Known Covariance Function

The main reason why the nonparametric prediction using Gaussian processes is not popular for resource-constrained multi-agent systems is the fact that the optimal prediction must use all cumulatively measured values in a non-trivial way [23, 38]. In this case, a robot needs to compute the inverse of the covariance matrix whose size grows as it collects more measurements. With this operation, the robot will run out of memory quickly. Therefore, it is necessary to develop a class of prediction algorithms using spatio-temporal Gaussian processes under a fixed memory size.

The space-time Kalman filter model proposed in [18,40] and utilized in [9] partially solved this problem by modeling the spatio-temporal field as a sum of a zero-mean Gaussian process, which is uncorrelated in time, and a time-varying mean function (see (6) and (12) in [18]). The zero-mean Gaussian process represents a spatial structure that is independent from one time point to the next as described in [18] by assuming that the dynamical environmental

process is governed by a relatively large time scale. This formulation in turn provides the Markov property in time, which makes the optimal prediction recursive in time. However, the value of a temporal mean function at a point (realized by a stable linear system) consists of a linear sum of colored white noises, and transient responses that converge to zero values exponentially fast [9], which can not represent a wide range of spatio-temporal phenomena in a fully nonparametric manner [51].

A simple way to cope with this dilemma is to design a robot so that it predicts a spatio-temporal Gaussian process at the current (or future) time based on truncated observations, *e.g.*, the last m observations from a total of n of observations as shown in Fig. 4.1. This seems intuitive in the sense that the last m observations are more correlated with the point of interest than the other $r = n - m$ observations (Fig. 4.1) in order to predict values at current or future time. Therefore, it is very important to analyze the performance degradation and trade-off effects of prediction based on truncated observations compared to the one based on all cumulative observations.

The second motivation is to design and analyze distributed sampling strategies for resource-constrained mobile sensor networks. Developing distributed estimation and coordination algorithms for multi-agent systems using only local information from local neighboring agents has been one of the most fundamental problems in mobile sensor networks [10, 13, 28, 46, 52, 68, 69]. Emphasizing practicality and usefulness, it is critical to synthesize and analyze distributed sampling strategies under practical constraints such as measurement noise and a limited communication range.

In Section 4.1, we propose to use only truncated observations to bound the computational complexity. The error bounds in using truncated observations are analyzed for prediction at

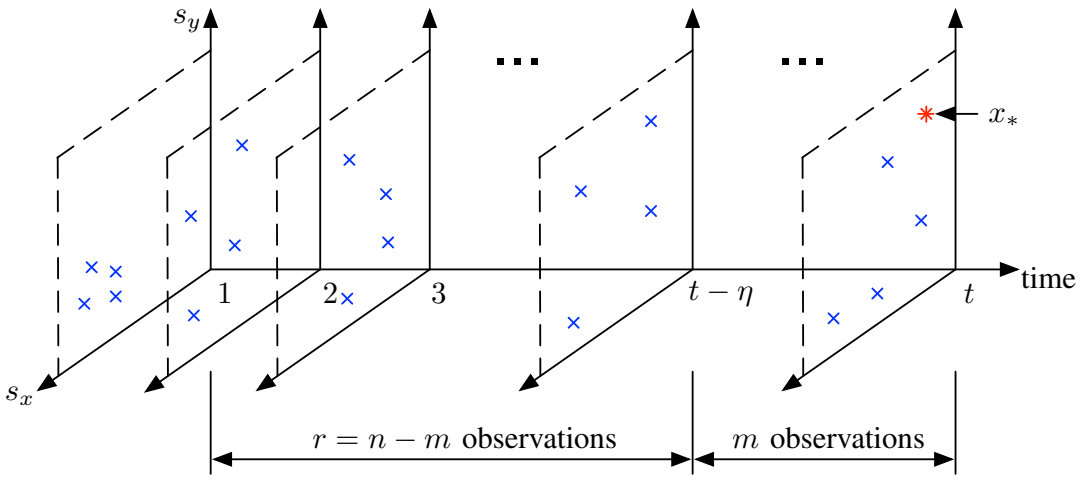


Figure 4.1: Robot predicts a scalar value at x_* (denoted by a red star) based on cumulative n spatio-temporal observations (denoted by blue crosses). Near-optimal prediction can be obtained using truncated observations, *e.g.*, the last m observations. In this case, $x = (s_x, s_y, t)^T$.

a single point in Section 4.1.1. A way of selecting a temporal truncation size is also discussed in Section 4.1.2. To improve the prediction quality, centralized and distributed navigation strategies for mobile sensor networks are proposed in Section 4.2. In Section 4.3, simulation results illustrate the usefulness of our schemes under different conditions and parameters.

4.1 GPR with Truncated Observations

As mentioned in above, one drawback of Gaussian process regression is that its computational complexity and memory space increase as more measurements are collected, making the method prohibitive for robots with limited memory and computing power. To overcome this increase in complexity, a number of approximation methods for Gaussian process regression have been proposed. In particular, the sparse greedy approximation method [59], the Nystrom method [63], the informative vector machine [33], the likelihood approxima-

tion [57], and the Bayesian committee machine [61] have been shown to be effective for many problems. However, these approximation methods have been proposed without theoretical justifications.

In general, if measurements are taken from nearby locations (or space-time locations), correlation between measurements is strong and correlation exponentially decays as the distance between locations increases. If the correlation function of a Gaussian process has this property, intuitively, we can make a good prediction at a point of interest using only measurements nearby. In the next subsection, we formalize this idea and provide a theoretical foundation for justifying Gaussian process regression with truncated observations proposed in this chapter.

4.1.1 Error bounds in using truncated observations

Consider a zero-mean Gaussian process

$$z(x) \sim \mathcal{GP}(0, \sigma_f^2 C(x, x')). \tag{4.1}$$

Notice that we denote the covariance function as $\sigma_f^2 C(x, x')$ in which $C(x, x') := \mathrm{Corr}(z(x), z(x'))$ is the correlation function. Recall that the predictive distribution of $z_* := z(x_*)$ at a point of interest x_* given observations $y = (y^{(1)}, \cdots, y^{(n)})^T$ is Gaussian, *i.e.*,

$$z_*|y \sim \mathcal{N}\left(\mu_{z_*|y}, \sigma^2_{z_*|y}\right), \tag{4.2}$$

where

$$\mu_{z_*|y} = k^T C^{-1} y, \tag{4.3a}$$

and

$$\sigma^2_{z_*|y} = \sigma^2_f(1 - k^T C^{-1} k). \tag{4.3b}$$

In (4.3a) and (4.3b), we have defined $C := \mathrm{Corr}(y, y) \in \mathbb{R}^{n\times n}$, and $k := \mathrm{Corr}(y, z_*) \in \mathbb{R}^n$. Notice that in this chapter, we assume the hyperparameter vector $\theta \in \mathbb{R}^m$ is given, and hence we neglect the explicit conditioning on θ.

Without loss of generality, we assume that the first m out of n observations are used to predict z_*. Let $r = n - m$, $y_m = (y^{(1)}, \cdots, y^{(m)})^T$, $y_r = (y^{(m+1)}, \cdots, y^{(n)})^T$. Then the covariance matrix $K \in \mathbb{R}^{n\times n}$ and $k \in \mathbb{R}^n$ can be represented as

$$K = \begin{bmatrix} K_m & K_{mr} \\ K_{mr}^T & K_r \end{bmatrix}, \quad k = \begin{bmatrix} k_m \\ k_r \end{bmatrix}.$$

Using truncated observations, we can predict the value z_* as

$$\mu_{z_*|y_m} = k_m^T C_m^{-1} y_m, \tag{4.4}$$

with a prediction error variance given by

$$\sigma^2_{z_*|y_m} = \sigma^2_f(1 - k_m^T C_m^{-1} k_m), \tag{4.5}$$

where $C_m = K_m + \sigma_w^2 I \in \mathbb{R}^{m\times m}$.

The following result shows the gap between predicted values using truncated measurements and all measurements.

Theorem 4.1.1. *Consider a Gaussian process $z(x) \sim \mathcal{GP}(0, \sigma_f^2 C(x, x'))$, we have*

$$\mu_{z_*|y} - \mu_{z_*|y_m} = (k_r - K_{mr}^T C_m^{-1} k_m)^T (C_r - K_{mr}^T C_m^{-1} K_{mr})^{-1} (y_r - K_{mr}^T C_m^{-1} y_m), \quad (4.6a)$$

and

$$\sigma^2_{z_*|y} - \sigma^2_{z_*|y_m} = -\sigma_f^2 (k_r - K_{mr}^T C_m^{-1} k_m)^T (C_r - K_{mr}^T C_m^{-1} K_{mr})^{-1} (k_r - K_{mr}^T C_m^{-1} k_m) < 0. \quad (4.6b)$$

Proof. We can rewrite (4.3a) as

$$\mu_{z_*|y} = \begin{bmatrix} k_m \\ k_r \end{bmatrix}^T \begin{bmatrix} C_m & K_{mr} \\ K_{mr}^T & C_r \end{bmatrix}^{-1} \begin{bmatrix} y_m \\ y_r \end{bmatrix}, \quad (4.7a)$$

and (4.3b) as

$$\sigma^2_{z_*|y} = \sigma_f^2 \left(1 - \begin{bmatrix} k_m \\ k_r \end{bmatrix}^T \begin{bmatrix} C_m & K_{mr} \\ K_{mr}^T & C_r \end{bmatrix}^{-1} \begin{bmatrix} k_m \\ k_r \end{bmatrix} \right). \quad (4.7b)$$

Using the identity based on matrix inversion lemma (see Appendix A.2), (4.7a) and (4.7b) become

$$\begin{aligned} \mu_{z_*|y} &= k_m^T C_m^{-1} y_m \\ &\quad + (k_r - K_{mr}^T C_m^{-1} k_m)^T (C_r - K_{mr}^T C_m^{-1} K_{mr})^{-1} (y_r - K_{mr}^T C_m^{-1} y_m), \end{aligned}$$

and

$$\sigma^2_{z_*|y} = \sigma_f^2 \left(1 - k_m^T C_m^{-1} k_m\right) \\ - \sigma_f^2 (k_r - K_{mr}^T C_m^{-1} k_m)^T (C_r - K_{mr}^T C_m^{-1} K_{mr})^{-1} (k_r - K_{mr}^T C_m^{-1} k_m).$$

Hence, by the use of (4.4) and (4.5), we obtain (4.6a) and 4.6b. □

Corollary 4.1.2. *The prediction error variance $\sigma^2_{z_*|y_m}$ is a non-increasing function of m.*

Proof. The proof is straightforward from Theorem 4.1.1 by letting $n = m + 1$. □

Considering an ideal case in which the measurements y_m are not correlated with the remaining measurements y_r, we have the following result.

Proposition 4.1.3. *Under the assumptions used in Theorem 4.1.1 and for given $y_r \sim \mathcal{N}(0, C_r)$, if $K_{mr} = 0$, then $\mu_{z_*|y} - \mu_{z_*|y_m} = k_r^T C_r^{-1} y_r$ and $\sigma^2_{z_*|y} - \sigma^2_{z_*|y_m} = -\sigma_f^2 k_r^T C_r^{-1} k_r$. In addition, we also have*

$$\left|\mu_{z_*|y} - \mu_{z_*|y_m}\right| \le \left\|k_r^T C_r^{-1}\right\| \sqrt{r}\bar{y}(p_1)$$

with a non-zero probability p_1. For a desired p_1, we can find $\bar{y}(p_1)$ by solving

$$p_1 = \prod_{1 \le i \le r} \left(1 - 2\Phi\left(-\frac{\bar{y}(p_1)}{\lambda_i^{1/2}}\right)\right), \tag{4.8}$$

where Φ is the cumulative normal distribution and $\{\lambda_i \,|\, i = 1, \cdots, r\}$ are the eigenvalues of $C_r = U \Lambda U^T$ with a unitary matrix U, i.e., $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_r)$.

Proof. The first statement is straightforward from Theorem 4.1.1.

For the second statement, we can represent y_r as $y_r = C_r^{1/2}u = U\Lambda^{1/2}u = U\tilde{y}$, where u is a vector of independent standard normals and $C_r = U\Lambda U^T$ and $C_r^{1/2} = U\Lambda^{1/2}$. By using the Cauchy-Schwarz inequality and norm inequalities, we have

$$\begin{aligned}\left|\mu_{z_*|y} - \mu_{z_*|y_m}\right| &= \left|k_r^T C_r^{-1} y_r\right| = \left|k_r^T C_r^{-1} U\tilde{y}\right| \\ &\leq \left\|k_r^T C_r^{-1}\right\| \|U\tilde{y}\| = \left\|k_r^T C_r^{-1}\right\| \|\tilde{y}\| \\ &\leq \left\|k_r^T C_r^{-1}\right\| \sqrt{r}\,\|\tilde{y}\|_\infty \leq \left\|k_r^T C_r^{-1}\right\| \sqrt{r}\bar{y}.\end{aligned}$$

Recall that we have $u \sim \mathcal{N}(0, I)$ and $\tilde{y} \sim \mathcal{N}(0, \Lambda)$, where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_r)$. Then we can compute the probability $p_1 = \Pr(\|\tilde{y}\|_\infty \leq \bar{y})$ as follows.

$$\begin{aligned}p_1 &= \Pr\left(\max_{1\leq i\leq r}\left|\tilde{y}^{(i)}\right| \leq \bar{y}\right) = \Pr\left(\max_{1\leq i\leq r}\left|\lambda_i^{1/2}u_i\right| \leq \bar{y}\right) \\ &= \prod_{1\leq i\leq r}\Pr\left(\lambda_i^{1/2}|u_i| \leq \bar{y}\right) = \prod_{1\leq i\leq r}\Pr\left(|u_i| \leq \frac{\bar{y}}{\lambda_i^{1/2}}\right) \\ &= \prod_{1\leq i\leq r}\left(1 - 2\Phi\left(-\frac{\bar{y}}{\lambda_i^{1/2}}\right)\right),\end{aligned}$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution. □

Hence, if the magnitude of K_{mr} is small, then the truncation error from using truncated measurements will be close to $k_r^T C_r^{-1} k_r$. Furthermore, if we want to reduce this error, we want k_r to be small, *i.e.*, when the covariance between z_* and the remaining measurements y_r is small. In summary, if the following two conditions are satisfied: (1) the correlation between measurements y_m and the remaining measurements y_r is small and (2) the correlation between z_* and the remaining measurements y_r is small, then the truncation error is small and $\mu_{z_*|y_m}$ can be a good approximation to $\mu_{z_*|y}$. This idea is formalized in a more general

setting in the following theorem.

Theorem 4.1.4. *Consider a zero-mean Gaussian process* $z(x) \sim \mathcal{N}(0, \sigma_f^2 C(x, x'))$ *with the correlation function*

$$C(x, x') = \exp\left\{-\frac{\left\|x - x'\right\|^2}{2\ell^2}\right\}, \tag{4.9}$$

and assume that we have collected n *observations,* $y^{(1)}, \cdots, y^{(n)}$*. Suppose that* K_{mr} *is small enough such that* $\left\|K_{mr}^T C_m^{-1} k_m\right\| \leq \|k_r\|$*, and* $\left\|K_{mr}^T C_m^{-1} y_m\right\| \leq \delta_2 \|y_r\|$ *and for some* $\delta_2 > 0$*. Given* $0 < p_2 < 1$*, choose* $\bar{y}(p_2)$ *such that* $\max_{i=m+1}^{n} \left|y^{(i)}\right| < \bar{y}(p_2)$ *with probability* p_2 *and* $\epsilon > 0$ *such that* $\epsilon < 2\gamma r(1 + \delta_2)\bar{y}(p_2)$ *where* γ *is the signal-to-noise ratio. For* x_**, if the last* $r = n - m$ *data points satisfy*

$$\left\|x^{(i)} - x_*\right\|^2 > 2\sigma_\ell^2 \log\left(2\gamma \frac{1}{\epsilon} r(1 + \delta_2)\bar{y}(p_2)\right),$$

then, with probability p_2*, we have*

$$\left|\mu_{z_*|y} - \mu_{z_*|y_m}\right| < \epsilon.$$

Proof. Let $A = C_m^{-1} K_{mr}$ and $B = K_{mr}^T C_m^{-1} K_{mr}$ for notational convenience. Then

$$\begin{aligned}
\left|\mu_{z_*|y} - \mu_{z_*|y_m}\right| &= \left\|(k_r^T - k_m^T A)(C_r - B)^{-1}(y_r - A^T y_m)\right\| \\
&\leq \left\|k_r^T - k_m^T A\right\| \left\|(C_r - B)^{-1}(y_r - A^T y_m)\right\| \\
&\leq \left\|k_r^T - k_m^T A\right\| \times (\left\|(C_r - B)^{-1} y_r\right\| + \left\|(C_r - B)^{-1} A^T y_m\right\|) \\
&\leq 2\|k_r\| \left(\left\|(C_r - B)^{-1} y_r\right\| + \left\|(C_r - B)^{-1} A^T y_m\right\|\right)
\end{aligned}$$

Since K_r is positive semi-definite, and C_m is positive definite, we have$K_r - B$ is positive

semi-definite. Then we have

$$(C_r - B)^{-1} = (K_r + 1/\gamma I - B)^{-1} \preceq \gamma I.$$

Combining this result, we get

$$\begin{aligned}\left|\mu_{z_*|y} - \mu_{z_*|y_m}\right| &\leq 2\gamma \left\|k_r\right\| \left(\left\|y_r\right\| + \left\|A^T y_m\right\|\right) \\ &\leq 2\gamma(1+\delta_2) \left\|k_r\right\| \left\|y_r\right\| \\ &\leq 2\gamma(1+\delta_2)\sqrt{r} C_{\max} \left\|y_r\right\|,\end{aligned}$$

where $C(x^{(i)}, x_*) \leq C_{\max}$ for $i \in \{m+1, \cdots, n\}$. Define $\bar{y}(p_2)$ such that $\max_{i=m+1}^{n} \left|y^{(i)}\right| \leq \bar{y}(p_2)$ with probability p_2. Then, with probability p_2, we have

$$\left|\mu_{z_*|y} - \mu_{z_*|y_m}\right| \leq 2\gamma r(1+\delta_2) C_{\max} \bar{y}(p_2).$$

Hence, for $\epsilon > 0$, if

$$C_{\max} < \frac{\epsilon}{2\gamma r(1+\delta_2)\bar{y}(p_2)} \tag{4.10}$$

with probability p_2, we have

$$\left|\mu_{z_*|y} - \mu_{z_*|y_m}\right| < \epsilon.$$

Let $l^2 = \min \left\|x^{(i)} - x_*\right\|^2$ for any $i \in \{m+1, \cdots, n\}$. Then (4.10) becomes, with probability

p_2,

$$\exp\left(-\frac{l^2}{2\sigma_\ell^2}\right) \leq C_{\max} < \frac{\epsilon}{2\gamma r(1+\delta_2)\bar{y}(p_2)}$$
$$l^2 > -2\sigma_\ell^2 \log\left(\frac{\epsilon}{2\gamma r(1+\delta_2)\bar{y}(p_2)}\right)$$

For $\epsilon < 2\gamma r(1+\delta_2)\bar{y}(p_2)$, we have

$$l^2 > 2\sigma_\ell^2 \log\left(2\gamma\frac{1}{\epsilon}r(1+\delta_2)\bar{y}(p_2)\right),$$

and this completes the proof. □

Remark 4.1.5. *The last part of Proposition 4.1.3 and Theorem 4.1.4 seek a bound for the difference between predicted values using all and truncated observations with a given probability since the difference is a random variable.*

Example 4.1.6. *We provide an illustrative example to show how to use the result of Theorem 4.1.4 as follows. Consider a Gaussian process defined in* (4.1) *and* (4.9) *with* $\sigma_f^2 = 1$, $\sigma_\ell = 0.2$, *and* $\gamma = 100$. *If we have any randomly chosen* 10 *samples* ($m = 10$) *within* $(0,1)^2$ *and we want to make prediction at* $x_* = (1,1)^T$. *We choose* $\bar{y}(p_2) = 2\sigma_f = 2$ *such that* $\max_{i=m+1}^{n} \left|y^{(i)}\right| < \bar{y}(p_2)$ *with probability* $p_2 = 0.95$. *According to Theorem 4.1.4, if we have an extra sample* $x^{(11)}$ ($r = 1$) *at* $(2.5, 2.5)^T$, *which satisfies the condition* $\left\|x^{(11)} - x_*\right\| > 0.92$, *then the difference in prediction using with and without the extra sample is less than* $\epsilon = 0.01$ *with probability* $p_2 = 0.95$.

Example 4.1.7. *Motivated by the results presented, we take a closer look at the usefulness of using a subset of observations from a sensor network for a particular realization of the Gaussian process. We consider a particular realization shown in Fig. 4.2, where crosses*

represent the sampling points of a Gaussian process defined in (4.1) *and* (4.9) *with* $\sigma_f^2 = 1$, $\sigma_\ell = 0.2$, *and* $\gamma = 100$ *over* $(0,1)^2$. *We have selected* y_m *as the collection of observations (blue crosses) within the red circle of a radius* $R = 2\sigma_\ell = 0.4$ *centered at a point (a red star) located at* $x_* = (0.6, 0.4)^T$. *If a measurement is taken outside the red circle, the correlation between this measurement and the value at* x_* *decreases to* 0.135. *The rest of observations (blue crosses outside of the red circle) are selected as* y_r. *The prediction results are shown in Table 4.1. In this particular realization, we have* $z_* = 1.0298$. *It can be seen that the prediction means and variances using only* y_m *are close to the one using all observations. We also compute the prediction at* x_* *with* y_r *which is far from the true value with a large variance.*

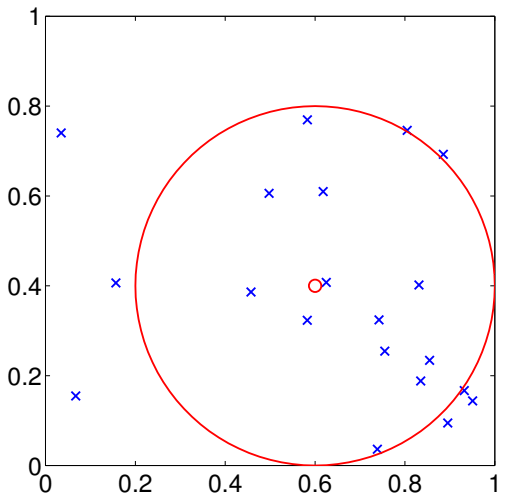


Figure 4.2: Example of the selection of truncated observations. The parameters used in the example are: $\sigma_f^2 = 1$, $\sigma_\ell = 0.2$, $\sigma_w = 0.1$.

Table 4.1: Prediction means and variances using y, y_m, and y_r.

	$n = 20$	$m = 12$	$r = 8$
$\mu_{z_*\|y}$	1.0515	1.0633	0.3491
$\sigma^2_{z_*\|y}$	0.0079	0.0080	0.9364

The result of Theorem 4.1.4 and Examples 4.1.6 and 4.1.7 all suggest the usage of observations that are highly correlated with the point of interest.

4.1.2 Selecting temporal truncation size

In previous subsection, we have obtained the error bounds for the prediction at a single point. In general, the observations made close to that point are more informative than the others.

Consider a zero-mean spatio-temporal Gaussian process

$$z(s,t) \sim \mathcal{GP}(0, \sigma_f^2 C(s,t,s,t')), \tag{4.11}$$

with covariance function

$$\begin{aligned} C(x,x') &= C_s(s,s')C_t(t,t') \\ &= \exp\left(-\sum_{\ell\in\{x,y\}} \frac{(s_\ell - s'_\ell)^2}{2\sigma_\ell^2}\right) \exp\left(-\frac{(t-t')^2}{2\sigma_t^2}\right). \end{aligned} \tag{4.12}$$

We define η as the truncation size, and our objective is to use only the observations made during the last η time steps, *i.e.*, from time $t-\eta+1$ to time t, to make prediction at time t. In general, a small η yields faster computation but lower accuracy and a large η yields slower computation but higher accuracy. Thus, the truncation size η should be selected according to a trade-off relationship between accuracy and efficiency.

Next, we show an approach to select the truncation size η in an averaged performance sense. Given the observations and associated sampling locations and times (denoted by $\mathcal{D}$ which depends on η), the generalization error $\epsilon_{x_*,\mathcal{D}}$ at a point $x_* = (s_*^T, t_*)^T$ is defined as

the prediction error variance $\sigma^2_{z_*|\mathcal{D}}$ [60, 64]. For a given t_* not knowing user specific s_* *a priori*, we seek to find η that guarantees a low prediction error variance uniformly over the entire space $\mathcal{Q}$, *i.e.*, we want $\epsilon_{\mathcal{D}} = \mathrm{E}_{s_*}[\sigma^2_{z_*|\mathcal{D}}]$ to be small [60, 64]. Here E_{s_*} denotes the expectation with respect to the uniform distribution of s_*.

According to Mercer's Theorem, we know that the kernel function C_s can be decomposed into

$$C_s(s, s') = \sum_{i=1}^{\infty} \lambda_i \phi_i(s) \phi_i(s'),$$

where $\{\lambda_i\}$ and $\{\phi_i(\cdot)\}$ are the eigenvalues and corresponding eigenfunctions, respectively [60]. In a similar way shown in [60], the input dependent generalization error $\epsilon_{\mathcal{D}}$ for our spatio-temporal Gaussian process can be obtained as

$$\begin{aligned} \epsilon_{\mathcal{D}} &= \mathrm{E}_{s*}\left[\sigma_f^2\left(1 - \mathrm{tr}\left(kk^T(K + 1/\gamma I)^{-1}\right)\right)\right] \\ &= \sigma_f^2\left(1 - \mathrm{tr}\left(\mathrm{E}_{s_*}[kk^T](K + 1/\gamma I)^{-1}\right)\right). \end{aligned} \tag{4.13}$$

We have

$$\mathrm{E}_{s_*}[kk^T] = \Psi \Lambda^2 \Psi^T \circ k_t k_t^T, \tag{4.14}$$

and

$$K = \Psi \Lambda \Psi^T \circ K_t K_t^T, \tag{4.15}$$

where $(\Psi)_{ij} = \phi_j(s_i)$, $(k_t)_j = C_t(t^{(j)}, t_*)$, $(K_t)_{ij} = C_t(t^{(i)}, t^{(j)})$, and $(\Lambda)_{ij} = \lambda_i \delta_{ij}$. δ_{ij} denotes the Dirac delta function. $\circ$ denotes the Hadamard (element-wise) product [60]. Hence, the input-dependent generalization error $\epsilon_{\mathcal{D}}$ can be computed analytically by plugging (4.14) and (4.15) into (4.13). Notice that $\epsilon_{\mathcal{D}}$ is a function of inputs (*i.e.*, the sampling locations and times). To obtain an averaged performance level without the knowledge of the

algorithmic sampling strategy *a priori*, we use an appropriate sampling distribution which models the stochastic behavior of the sampling strategy. Thus, further averaging over the observation set $\mathcal{D}$ with the samping distribution yields $\epsilon(\eta) = \mathrm{E}_{\mathcal{D}}[\epsilon_{\mathcal{D}}]$ which is a function of the truncation size η only. This averaging process can be done using Monte Carlo methods. Then η can be chosen based on the averaged performance measure $\epsilon(\eta)$ under the sampling distribution.

An alternative way, without using the eigenvalues and eigenfunctions, is to directly and numerically compute $\epsilon_{\mathcal{D}} = \mathrm{E}_{s*}[\sigma^2_{z*|\mathcal{D}}]$ uniformly over the entire space $\mathcal{Q}$ with random sampling positions at each time step. An averaged generalization error with respect to the temporal truncation size can be plotted by using such Monte Carlo methods. Then the temporal truncation size η can be chosen such that a given level of the averaged generalization error is achieved.

Example 4.1.8. *Consider a problem of selecting a temporal truncation size η for spatio-temporal Gaussian process regression using observations from* 9 *agents. The spatio-temporal Gaussian process is defined in* (4.1) *and* (4.9) *with* $\sigma_f^2 = 1$, $\sigma_x = \sigma_y = 0.2$, $\sigma_t = 5$, *and* $\gamma = 100$ *over* $(0, 1)^2$. *The Monte Carlo simulation result is shown in Fig. 4.3. The achieved generalization error $\epsilon_{\mathcal{D}}$ are plotted in blue circles with error-bars with respect to the temporal truncation size η. As can be seen, an averaged generalization error (in blue circles) under* 0.1 *can be achieved by using observations taken from last* 10 *time steps.*

Notice that the prediction error variances can be significantly minimized by optimally selecting the sampling positions. Hence, the selected η guarantees at least the averaged performance level of the sensor network when the optimal sampling strategy is used.

By using a fixed truncation size η, the computational complexity and memory space

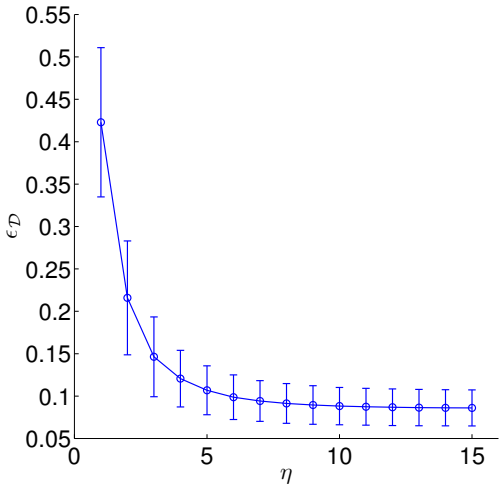


Figure 4.3: Example of selecting a temporal truncation size η. The parameters used in the example are: $\sigma_f^2 = 1$, $\sigma_x = \sigma_y = 0.2$, $\sigma_t = 5$, $\gamma = 100$.

required for making prediction (*i.e.*, evaluating (4.3a) and (4.3b)) do not increase as more measurements are collected. Our next objective is to improve the quality of the prediction by carefully selecting the future sampling positions for the mobile sensor network.

4.2 Optimal Sampling Strategies

At time t, the goal of the mobile sensor network is to make prediction at pre-specified points of interest $\{p_j = (v_j, \tau_j) \,|\, j \in \mathcal{J}\}$ indexed by $\mathcal{J} := \{1, \cdots, M\}$. From here on, points of interest will be referred to as *target points*. The introduction of target points is motivated by the fact that the potential environmental concerns should be frequently monitored. For instance, the target points can be assigned at the interface of a factory and a lake, sewage systems, or polluted beaches. Thus, the introduction of target points, which can be arbitrarily specified by a user, provides a flexible way to define a geometrical shape of a subregion of interest in a surveillance region. Notice that the target points can be changed by a user at any time. In particular, we allow that the number of target points M can be larger than that of agents

N, which is often the case in practice. The prediction of $z_j := z(p_j)$ of the Gaussian process at a target point p_j can be obtained as in (4.3a) and (4.3b).

4.2.1 Centralized navigation strategy

Consider the case in which a central station receives collective measurements from all N mobile sensors and performs the prediction. Let the central station discard the oldest set of measurements $y_{t-\eta+1}$ after making the prediction at time t. At the next time index $t+1$, using the remained observations $y_{t-\eta+2:t}$ in the memory along with new measurements y_{t+1} from all N agents at time $t+1$, the central station will predict $z(s_*, t_*)$ evaluated at target points $\{p_j \,|\, j \in \mathcal{J}\}$. Hence, agents should move to the most informative locations for taking measurements at time $t+1$ [32].

For notational simplicity, let $\bar{y} \in \mathbb{R}^{N(\eta-1)}$ be the remained observations, *i.e.*, $\bar{y} := y_{t-\eta+2:t}$, and $\tilde{y} \in \mathbb{R}^N$ be the measurements that will be taken at positions $\tilde{q} = (\tilde{q}_1^T, \cdots, \tilde{q}_N^T)^T \in \mathcal{Q}^N$ and time $t+1$. In contrast to the information-theoretic control strategies using the conditional entropy or the mutual information criterion [14, 32], in this chapter, the mobility of the robotic sensors will be designed such that they directly minimize the average of the prediction error variances over target points, *i.e.*,

$$J_c(\tilde{q}) = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \sigma^2_{z_j|\bar{y},\tilde{y}}(\tilde{q}), \tag{4.16}$$

where $|\mathcal{J}| = M$ is the cardinality of $\mathcal{J}$. The prediction error variance at each of M target points is given by

$$\sigma^2_{z_j|\bar{y},\tilde{y}}(\tilde{q}) = \sigma_f^2 \left(1 - k_j(\tilde{q})^T C(\tilde{q})^{-1} k_j(\tilde{q})\right), \quad \forall j \in \mathcal{J},$$

where $k_j(\tilde{q})$ and $C(\tilde{q})$ are defined as

$$k_j(\tilde{q}) = \begin{bmatrix} \mathrm{Corr}(\bar{y}, z_j) \\ \mathrm{Corr}(\tilde{y}, z_j) \end{bmatrix}, \quad C(\tilde{q}) = \begin{bmatrix} \mathrm{Corr}(\bar{y}, \bar{y}) & \mathrm{Corr}(\bar{y}, \tilde{y}) \\ \mathrm{Corr}(\tilde{y}, \bar{y}) & \mathrm{Corr}(\tilde{y}, \tilde{y}) \end{bmatrix}.$$

In order to reduce the average of prediction error variances over target points $\{p_j \,|\, j \in \mathcal{J}\}$, the central station solves the following optimization problem

$$q(t+1) = \arg \min_{\tilde{q} \in \mathcal{Q}^N} J_c(\tilde{q}). \tag{4.17}$$

Notice that in this problem set-up, we only consider the constraint that robots should move within the region $\mathcal{Q}$. However, the mobility constraints such as the maximum distance a robot can move between two time indices or the maximum speed a robot can travel, can be incorporated as additional constraints in the optimization problem [13].

The sensor network configuration $q(t)$ can be controlled by a gradient descent algorithm such that $q(t)$ can move to a local minimum of J_c for the prediction at time $t+1$. The gradient descent control algorithm is given by

$$\frac{dq(\tau)}{d\tau} = -\nabla_q J_c(q(\tau)), \tag{4.18}$$

where $\nabla_x J_c(x)$ denotes the gradient of $J_c(x)$ at x. A critical point of $J_c(q)$ obtained in (4.18) will be $q(t+1)$. The analytical form of $\partial \sigma^2_{z_j|\bar{y},\tilde{y}}(\tilde{q}) / \partial \tilde{q}_{i,\ell}$, where $\tilde{q}_{i,\ell}$ is the ℓ-th element in $\tilde{q}_i \in \mathcal{Q}$, can be obtained by

$$\frac{\partial \sigma^2_{z_j|\bar{y},\tilde{y}}(\tilde{q})}{\partial \tilde{q}_{i,\ell}} = k_j^T C^{-1} \left(\frac{\partial C}{\partial \tilde{q}_{i,\ell}} C^{-1} k_j - 2 \frac{\partial k_j}{\partial \tilde{q}_{i,\ell}} \right), \quad \forall i \in \mathcal{I}, \ell \in \{1, 2\}.$$

Other more advanced non-linear optimization techniques may be applied to solve the optimization problem in (4.17) [3].

The centralized sampling strategy for the mobile sensor network with the cost function J_c in (4.16) is summarized in Table 4.2. Notice that the prediction in the centralized sampling strategy uses temporally truncated observations. A decentralized version of the centralized sampling strategy in Table 4.2 may be developed using the approach proposed in [42] in which each robot incrementally refines its decision while intermittently communicating with the rest of the robots.

4.2.2 Distributed navigation strategy

Now, we consider a case in which each agent in the sensor network can only communicate with other agents within a limited communication range R. In addition, no central station exists. In this section, we present a distributed navigation strategy for mobile agents that uses only local information in order to minimize a collective network performance cost function.

The communication network of mobile agents can be represented by an undirected graph. Let $\mathcal{G}(t) := (\mathcal{I}, \mathcal{E}(t))$ be an undirected communication graph such that an edge $(i, j) \in \mathcal{E}(t)$ if and only if agent i can communicate with agent j at time t. We define the neighborhood of agent i at time t by $\mathcal{N}_i(t) := \{j \in \mathcal{I} \,|\, (i, j) \in \mathcal{E}(t)\}$. In particular, we have

$$\mathcal{N}_i(t) = \left\{ j \in \mathcal{I} \,|\, \left\| q_i(t) - q_j(t) \right\| < R, j \neq i \right\}.$$

Note that in our definition above, " $<$ " is used instead of " $\leq$ " in deciding the communication range.

At time $t \in \mathbb{Z}_{>0}$, agent i collects measurements $\left\{ y_j(t) \,|\, j \in \{i\} \cup \mathcal{N}_i(t) \right\}$ sampled at

Table 4.2: Centralized sampling strategy at time t.

Input:
(1) Number of agents N
(2) Positions of agents $\{q_i(t) \,|\, i \in \mathcal{I}\}$
(3) Hyperparameters of the Gaussian process $\theta = (\sigma_f^2, \sigma_x, \sigma_y, \sigma_t)^T$
(4) Target points $\{p_j \,|\, j \in \mathcal{J}\}$
(5) Truncation size η

Output:
(1) Prediction at target points $\left\{\mu_{z_j|y_{t-\eta+1:t}} \,|\, j \in \mathcal{J}\right\}$
(2) Prediction error variance at target points $\left\{\sigma^2_{z_j|y_{t-\eta+1:t}} \,|\, j \in \mathcal{J}\right\}$

For $i \in \mathcal{I}$, agent i performs:

1: make an observation at current position $q_i(t)$, *i.e.*, $y_i(t)$
2: transmit the observation $y_i(t)$ to the central station

The central station performs:

1: collect the observations from all N agents, *i.e.*, $y_t = (y_1(t), \cdots, y_N(t))^T$
2: obtain the cumulative measurements, *i.e.*, $y_{t-\eta+1:t} = (y_{t-\eta+1}^T, \cdots, y_t^T)^T$
3: **for** $j \in \mathcal{J}$ **do**
4: make prediction at a target point p_j

$$\mu_{z_j|y_{t-\eta+1:t}} = k^T C^{-1} y,$$

with a prediction error variance given by

$$\sigma^2_{z_j|y_{t-\eta+1:t}} = \sigma_f^2(1 - k^T C^{-1} k),$$

where $y = y_{t-\eta+1:t}$, $k = \mathrm{Corr}(y, z_j)$, and $C = \mathrm{Corr}(y, y)$
5: **end for**
6: **if** $t \geq \eta$ **then**
7: discard the oldest set of measurements taken at time $t - \eta + 1$, *i.e.*, $y_{t-\eta+1}$
8: **end if**
9: compute the control with the remained data $y_{t-\eta+2:t}$

$$q(t+1) = \arg\min_{\tilde{q} \in \mathcal{Q}^N} J_c(\tilde{q}),$$

via

$$\frac{dq(\tau)}{d\tau} = -\nabla_q J_c(q(\tau))$$

10: send the next sampling positions $\{q_i(t+1)\}_{i=1}^N$ (a critical point of $J_c(\tilde{q})$) to all N agents

For $i \in \mathcal{I}$, agent i performs:

1: receive the next sampling position $q_i(t+1)$ from the central station
2: move to $q_i(t+1)$ before time $t+1$

$\{q_j(t) \,|\, j \in \{i\} \cup \mathcal{N}_i(t)\}$ from its neighbors and itself. The collection of these observations and the associated sampling positions in vector forms are denoted by $y_t^{[i]}$ and $q_t^{[i]}$, respectively. Similarly, for notational simplicity, we also define the cumulative measurements that have been collected by agent i from time $t-\eta+1$ to t as

$$y_{t-\eta+1:t}^{[i]} = \left((y_{t-\eta+1}^{[i]})^T, \cdots, (y_t^{[i]})^T\right)^T.$$

In contrast to the centralized scheme, in the distributed scheme, each agent determines the sampling points based on the local information from neighbors. After making the prediction at time t, agent i discards the oldest set of measurements $y_{t-\eta+1}^{[i]}$. At time $t+1$, using the remained observations $y_{t-\eta+2:t}^{[i]}$ in the memory along with new measurements $y_{t+1}^{[i]}$ from its neighbors in $\mathcal{N}_i(t+1)$, agent i will predict $z(s_*, t_*)$ evaluated at target points $\{p_j \,|\, j \in \mathcal{J}\}$.

For notational simplicity, let $\bar{y}^{[i]}$ be the remained observations of agent i, *i.e.*, $\bar{y}^{[i]} := y_{t-\eta+2:t}^{[i]}$. Let $\tilde{y}^{[i]}$ be the new measurements that will be taken at positions of agent i and its neighbors $\tilde{q}^{[i]} \in \mathcal{Q}^{|\mathcal{N}_i(t+1)|+1}$, and at time $t+1$, where $|\mathcal{N}_i(t+1)|$ is the number of neighbors of agent i at time $t+1$. The prediction error variance obtained by agent i at each of M target points (indexed by $\mathcal{J}$) is given by

$$\sigma^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]}) = \sigma_f^2 \left(1 - k_j^{[i]}(\tilde{q}^{[i]})^T C^{[i]}(\tilde{q}^{[i]})^{-1} k_j^{[i]}(\tilde{q}^{[i]})\right), \quad \forall j \in \mathcal{J},$$

where $k_j^{[i]}(\tilde{q}^{[i]})$ and $C^{[i]}(\tilde{q}^{[i]})$ are defined as

$$k_j^{[i]}(\tilde{q}^{[i]}) = \begin{bmatrix} \mathrm{Corr}(\bar{y}^{[i]}, z_j) \\ \mathrm{Corr}(\tilde{y}^{[i]}, z_j) \end{bmatrix}, \quad C^{[i]}(\tilde{q}^{[i]}) = \begin{bmatrix} \mathrm{Corr}(\bar{y}^{[i]}, \bar{y}^{[i]}) & \mathrm{Corr}(\bar{y}^{[i]}, \tilde{y}^{[i]}) \\ \mathrm{Corr}(\tilde{y}^{[i]}, \bar{y}^{[i]}) & \mathrm{Corr}(\tilde{y}^{[i]}, \tilde{y}^{[i]}) \end{bmatrix}. \tag{4.19}$$

The performance of agent i can be evaluated by the average of the prediction error variances over target points, *i.e.*,

$$J^{[i]}(\tilde{q}^{[i]}) = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \sigma^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]}), \quad \forall i \in \mathcal{I}.$$

One criterion to evaluate the network performance is the average of individual performance, *i.e.*,

$$J(\tilde{q}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} J^{[i]}(\tilde{q}^{[i]}). \tag{4.20}$$

However, the discontinuity of the function J occurs at the moment of gaining or losing neighbors, *e.g.*, at the set

$$\left\{ \tilde{q} \mid \left\| \tilde{q}_i - \tilde{q}_j \right\| = R \right\}.$$

A gradient decent algorithm for mobile robots that minimizes such J may produce hybrid system dynamics and/or chattering behaviors when robots lose or gain neighbors.

Therefore, we seek to minimize an upper-bound of J that is continuously differentiable. Consider the following function

$$\bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]}) = \sigma_f^2 \left(1 - k_j^{[i]}(\tilde{q}^{[i]})^T \bar{C}^{[i]}(\tilde{q}^{[i]})^{-1} k_j^{[i]}(\tilde{q}^{[i]}) \right), \quad \forall j \in \mathcal{J}, \tag{4.21}$$

where $\bar{C}^{[i]}(\tilde{q}^{[i]})$ is defined as

$$\bar{C}^{[i]}(\tilde{q}^{[i]}) = \begin{bmatrix} \mathrm{Corr}(\bar{y}^{[i]}, \bar{y}^{[i]}) & \mathrm{Corr}(\bar{y}^{[i]}, \tilde{y}^{[i]}) \\ \mathrm{Corr}(\tilde{y}^{[i]}, \bar{y}^{[i]}) & \mathrm{Corr}(\tilde{y}^{[i]}, \tilde{y}^{[i]}) + \tilde{C}^{[i]}(\tilde{q}^{[i]}) \end{bmatrix}.$$

Notice that $\bar{C}^{[i]}(\tilde{q}^{[i]})$ is obtained by adding a positive semi-definite matrix $\tilde{C}^{[i]}(\tilde{q}^{[i]})$ to the

lower right block of $C^{[i]}(\tilde{q}^{[i]})$ in (4.19), where

$$\tilde{C}^{[i]}(\tilde{q}^{[i]}) = \text{diag}\left(\Phi(d_{i1})^{-1}, \cdots, \Phi(d_{i(|\mathcal{N}_i(t+1)|+1)})^{-1}\right) - \frac{1}{\gamma} I,$$

where $d_{ij} := \left\|\tilde{q}_i - \tilde{q}_j\right\|$ is the distance between agent i and agent $j, \forall j \in \{i\} \cup \mathcal{N}_i(t+1)$. $\Phi : [0, R) \mapsto (0, \gamma]$ is a continuously differentiable function defined as

$$\Phi(d) = \gamma\phi\left(\frac{d + d_0 - R}{d_0}\right), \tag{4.22}$$

where

$$\phi(h) = \begin{cases} 1, & h \leq 0, \\ \exp\left(\frac{-h^2}{1-h^2}\right), & 0 < h < 1. \end{cases}$$

An example of $\Phi(d)$ where $\gamma = 100$, $R = 0.4$, and $d_0 = 0.1$ is shown in the red dotted line in Fig. 4.4. Notice that if $\Phi(d) = \gamma$ is used (the blue solid line in Fig. 4.4), we have $\bar{C}^{[i]}(\tilde{q}^{[i]}) = C^{[i]}(\tilde{q}^{[i]})$. We then have the following result.

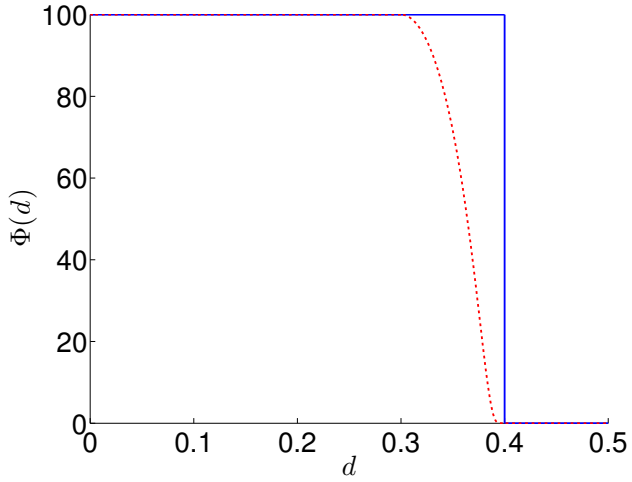


Figure 4.4: Function $\Phi(d)$ in (4.22) with $\gamma = 100$, $R = 0.4$, and $d_0 = 0.1$ is shown in a red dotted line. The function $\Phi(d) = \gamma$ is shown in a blue solid line.

Proposition 4.2.1. *$\bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]})$ is an upper-bound of $\sigma^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]})$, $\forall i \in \mathcal{I}$.*

Proof. Let $A := C^{[i]}(\tilde{q}^{[i]})$ and $B := \mathrm{diag}(0, \tilde{C}^{[i]}(\tilde{q}^{[i]}))$. The result follows immediately from the fact that $(A+B)^{-1} \preceq A^{-1}$ for any $A \succ 0$ and $B \succeq 0$. □

Hence, we construct a new cost function as

$$J_d(\tilde{q}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]}). \tag{4.23}$$

By Proposition 4.2.1, J_d in (4.23) is an upper-bound of J in (4.20).

Next, we show that J_d is continuously differentiable when agents gain or lose neighbors. In doing so, we compute the partial derivative of J_d with respect to $\tilde{q}_{i,\ell}$, where $\tilde{q}_{i,\ell}$ is the ℓ-th element in $\tilde{q}_i \in \mathcal{Q}$, as follows.

$$\begin{aligned} \frac{\partial J_d(\tilde{q})}{\partial \tilde{q}_{i,\ell}} &= \frac{1}{|\mathcal{I}|} \sum_{k \in \mathcal{I}} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[k]},\tilde{y}^{[k]}}(\tilde{q}^{[k]})}{\partial \tilde{q}_{i,\ell}} \\ &= \frac{1}{|\mathcal{I}|} \sum_{k \in \{i\} \cup \mathcal{N}_i} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[k]},\tilde{y}^{[k]}}(\tilde{q}^{[k]})}{\partial \tilde{q}_{i,\ell}}, \quad \forall i \in \mathcal{I}, \ell \in \{1,2\}. \end{aligned} \tag{4.24}$$

We then have the following.

Proposition 4.2.2. *The cost function J_d in* (4.23) *is of class $\mathcal{C}^1$,* i.e.*, it is continuously differentiable.*

Proof. We need to show that the partial derivatives of J_d with respect to $\tilde{q}_{i,\ell}, \forall i \in \mathcal{I}, \ell \in \{1,2\}$ exist and are continuous. Without loss of generality, we show that $\partial J_d / \partial \tilde{q}_{i,\ell}$, $\forall \ell \in$

$\{1, 2\}$ is continuous at any point $\tilde{q}^*$ in the following boundary set defined by

$$\mathcal{S}_{ik} := \{\tilde{q} \,|\, d_{ik} = \|\tilde{q}_i - \tilde{q}_k\| = R\}\,.$$

First, we consider a case in which $\tilde{q} \notin \mathcal{S}_{ik}$ and $d_{ik} < R$, *i.e.*, $k \in \mathcal{N}_i$ and $i \in \mathcal{N}_k$. By the construction of $\bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}$ in (4.21) using (4.22), when we take the limit of the partial derivative, as d_{ik} approaches R from below (as $\tilde{q}$ approaches $\tilde{q}^*$), we have that

$$\begin{aligned}
\lim_{d_{ik}\to R_-} \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]})}{\partial \tilde{q}_{i,\ell}} &= \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]}\backslash\tilde{q}_k)}{\partial \tilde{q}_{i,\ell}}, \\
\lim_{d_{ik}\to R_-} \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[k]},\tilde{y}^{[k]}}(\tilde{q}^{[k]})}{\partial \tilde{q}_{i,\ell}} &= \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[k]},\tilde{y}^{[k]}}(\tilde{q}^{[k]}\backslash\tilde{q}_i)}{\partial \tilde{q}_{i,\ell}} = 0,
\end{aligned}$$

where $\tilde{q}^{[a]}\backslash\tilde{q}_b$ denotes the collection of locations of agent a and its neighbors excluding $\tilde{q}_b$. Hence we have

$$\lim_{d_{ik}\to R_-} \frac{\partial J_d(\tilde{q})}{\partial \tilde{q}_{i,\ell}} = \frac{\partial J_d(\tilde{q}^*)}{\partial \tilde{q}_{i,\ell}}. \tag{4.25}$$

Consider the other case in which $\tilde{q} \notin \mathcal{S}_{ik}$ and $d_{ik} > R$, *i.e.*, $k \notin \mathcal{N}_i$ and $i \notin \mathcal{N}_k$. When d_{ik} approaches R from above (as $\tilde{q}$ approaches $\tilde{q}^*$), we have

$$\lim_{d_{ik}\to R_+} \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]})}{\partial \tilde{q}_{i,\ell}} = \frac{\partial \bar{\sigma}^2_{z_j|\bar{y}^{[i]},\tilde{y}^{[i]}}(\tilde{q}^{[i]})}{\partial \tilde{q}_{i,\ell}},$$

and hence

$$\lim_{d_{ik}\to R_+} \frac{\partial J_d(\tilde{q})}{\partial \tilde{q}_{i,\ell}} = \frac{\partial J_d(\tilde{q}^*)}{\partial \tilde{q}_{i,\ell}}. \tag{4.26}$$

Therefore, from (4.25) and (4.26), we have

$$\lim_{d_{ik}\to R_-} \frac{\partial J_d(\tilde{q})}{\partial \tilde{q}_{i,\ell}} = \lim_{d_{ik}\to R_+} \frac{\partial J_d(\tilde{q})}{\partial \tilde{q}_{i,\ell}} = \frac{\partial J_d(\tilde{q}^*)}{\partial \tilde{q}_{i,\ell}}.$$

This completes the proof due to Theorem 4.6 in [53]. □

By using J_d in (4.23), a gradient descent algorithm can be used to minimize the network performance cost function J_d in (4.23) for the prediction at $t+1$.

$$\frac{dq(\tau)}{d\tau} = -\nabla_q J_d(q(\tau)). \tag{4.27}$$

Note that the partial derivative in (4.24), which builds the gradient flow in (4.27), is a function of positions in $\cup_{j\in\mathcal{N}_i(t)}\mathcal{N}_j(t)$ only. This makes the algorithm distributed. A distributed sampling strategy for agent i with the network cost function J_d in (4.23) is summarized in Table 4.3. In this way, each agent with the distributed sampling strategy uses spatially and temporally truncated observations.

4.3 Simulation

In this section, we apply our approach to a spatio-temporal Gaussian process with a covariance function in (4.12). The Gaussian process was numerically generated through circulant

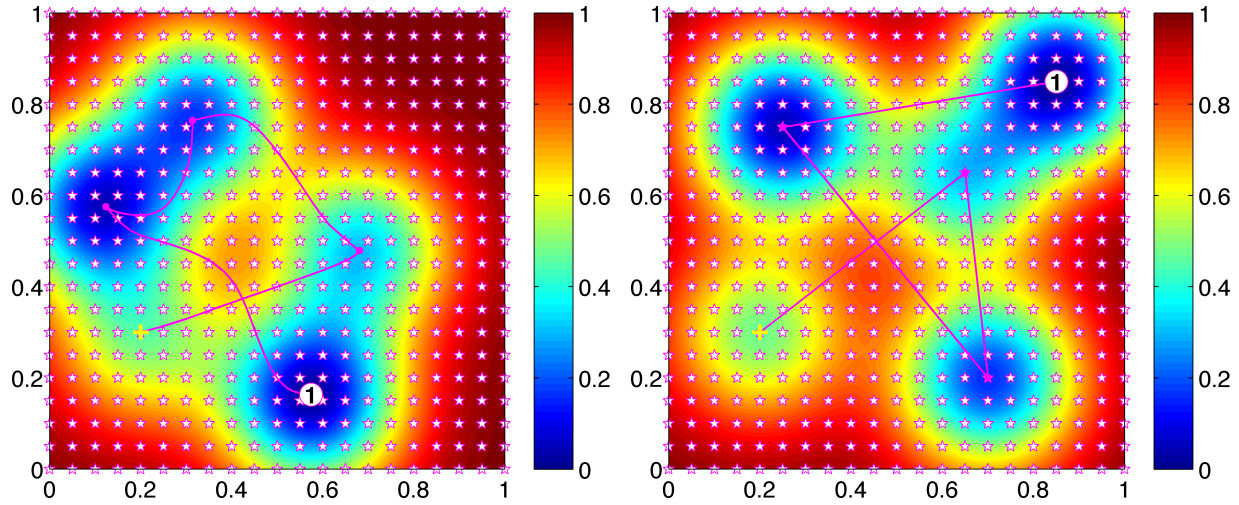
embedding of the covariance matrix for the simulation [20]. The hyperparameters used in the simulation were chosen to be $\theta = (\sigma_f^2, \sigma_x, \sigma_y, \sigma_t)^T = (1, 0.2, 0.2, 5)^T$. The surveillance region $\mathcal{Q}$ is given by $\mathcal{Q} = (0, 1)^2$. The signal-to-noise ratio $\gamma = 100$ is used throughout the simulation which is equivalent to a noise level of $\sigma_w = 0.1$. In our simulation, $N = 9$ agents sample at time $t \in \mathbb{Z}_{>0}$. The initial positions of the agents are randomly selected. The truncation size $\eta = 10$ is chosen using the approach introduced in Section 4.1.2 that guarantees the averaged performance level $\epsilon(\eta = 10) < 0.1$ under a uniform sampling distribution (see Example 4.1.8).

In the figures of simulation results, the target positions, the initial positions of agents, the past sampling positions of agents, and the current positions of agents are represented by white stars, yellow crosses, pink dots, and white circles with agent indices, respectively.

4.3.1 Gradient-based algorithm vs. exhaustive search algorithm

To evaluate the performance of the gradient-based algorithm presented in Section 4.2, we compare it with the exhaustive search algorithm over sufficiently many grid points, which guarantees the near-optimum. Due to the exponential complexity of the grid-based exhaustive search algorithm as the number of agents increases, its usage for multiple robots is prohibitive. Hence, we consider a simple case in which only one mobile agent samples and makes prediction on 21×21 target points over $\mathcal{Q}$. The grid points used in the exhaustive search are the same as the target points, *i.e.*, 21×21 grid points. The initial positions of the agents for both cases were set to $(0.2, 0.3)^T$. The prediction error variances at $t = 5$ for the proposed algorithm and the exhaustive search algorithm are shown in Figs. 4.5-(a) and (b), respectively. At time $t = 5$, the averaged prediction error variance over target points is

0.636 which is close to 0.613 achieved by the exhaustive search. Therefore, this simulation study shows that the performance of the gradient-based algorithm is comparable to that of the exhaustive search algorithm for the given problem.



(a) Gradient-based algorithm (b) Exhaustive search algorithm

Figure 4.5: Prediction error variances at $t = 5$ achieved by (a) using the gradient-based algorithm, and (b) using the exhaustive search algorithm. The trajectories of the agent are shown in solid lines.

4.3.2 Centralized sampling scheme

Consider a situation where a central station has access to all measurements collected by agents. At each time, measurements sampled by agents are transmitted to the central station that uses the centralized navigation strategy and sends control commands back to individual agents.

Case 1: First, we consider a set of fixed target points, *e.g.*, 6×6 grid points on $\mathcal{Q}$ at a fixed time $t = 10$. At each time step, the cost function J_c in (4.16), which is the average of prediction error variances at target points, is minimized due to the proposed centralized navigation strategy in Section 4.2.1. As a benchmark strategy, we consider

a random sampling scheme in which a group of 9 agents takes observations at randomly selected positions within the surveillance region $\mathcal{Q}$.

In Fig. 4.6-(a), the blue circles represent the average of prediction error variances over target points achieved by the centralized scheme, and the red squares indicate the average of prediction error variances over target points achieved by the benchmark strategy. Clearly, the proposed scheme produces lower averaged prediction error variances at target points as time increases, which demonstrates the usefulness of our scheme.

Case 2: Next, we consider the same 6×6 grid points on $\mathcal{Q}$ as in case 1. However, at time t, we are now interested in the prediction at the next sampling time $t+1$. At each time step, the cost function J_c is minimized. Fig. 4.6-(b) shows the average of prediction error variances over target points achieved by the centralized scheme with truncation (in red squares) and without truncation (in blue circles). With truncated observations, *i.e.*, with only observations obtained from latest $\eta = 10$ time steps, we are able to maintain the same level of the averaged prediction error variances (around 0.05 in Fig. 4.6-(b)).

Fig. 4.7-(a), (c), and (e) show the true field, the predicted field, and the prediction error variance at time $t = 1$, respectively. To see the improvement, the counterpart of the simulation results at time $t = 5$ are shown in Fig. 4.7-(b), (d), and (f). At time $t = 1$, agents have little information about the field and hence the prediction is far away from the true field, which produces a large prediction error variance. As time increases, the prediction becomes close to the true field and the prediction error variances are reduced due to the proposed navigation strategy.

Case 3: Now, we consider another case in which 36 target points (plotted in Fig. 4.8 as white stars) are evenly distributed on three concentric circles to form a ring shaped subregion

of interest. As in case 2, we are interested in the prediction at the next time iteration $t+1$. The average of prediction error variances over these target points at each time step achieved by the centralized scheme with truncation (in red squares) and without truncation (in blue circles) are shown in Fig. 4.6-(c). The prediction error variances at time $t = 1$ and $t = 5$ are shown in Fig. 4.8-(a) and (b), respectively. It is shown that agents are dynamically covering the ring shaped region to minimize the average of prediction error variances over the target points.

4.3.3 Distributed sampling scheme

Consider a situation in which the sensor network has a limited communication range R, *i.e.*, $\mathcal{N}_i(t) := \{j \in \mathcal{I} \mid \|q_i(t) - q_j(t)\| < R, j \neq i\}$. At each time $t \in \mathbb{Z}_{>0}$, agent i collects measurements from itself and its neighbors $\mathcal{N}_i(t)$ and makes prediction in a distributed fashion. The distributed strategy is used to navigate itself to move to the next sampling position. To be comparable with the centralized scheme, the same target points as in case 2 of Section 4.3.2 are considered.

Fig. 4.9 shows that the cost function, which is an upper-bound of the averaged prediction error variance over target points and agents, deceases smoothly from time $t = 1$ to $t = 2$ by the gradient descent algorithm with a communication range $R = 0.4$. Significant decreases occur whenever one of the agent gains a neighbor. Notice that the discontinuity of minimizing J in (4.20) caused by gaining or losing neighbors is eliminated due to the construction of J_d in (4.23). Hence, the proposed distributed algorithm is robust to gaining or losing neighbors.

The following study shows the effect of different communication range. Intuitively, the larger the communication range is, the more information can be obtained by the agent and

hence the better prediction can be made. Figs. 4.10-(a) and (b) show the average of prediction error variances over all target points and agents in blue circles with error-bars indicating the standard deviation among agents for the case $R = 0.3$ and $R = 0.4$, respectively. In both cases, $d_0 = 0.1$ in (4.22) was used. The average of prediction error variances is minimized quickly to a certain level. It can be seen that the level of achieved averaged prediction error variance with $R = 0.4$ is lower than the counterpart with $R = 0.3$.

Now, assume that each agent only predict the field at target points within radius R (local target points). The average of prediction error variances, over only local target points and agents, are also plotted in Fig. 4.10 in red squares with the standard deviation among agents. As can be seen, the prediction error variances at local target points (the red squares) are significantly lower than those for all target points (the blue circles).

Fig. 4.11 shows the prediction error variances obtained by agent 1 along with the edges of the communication network for different communication range R and different time step k. In Fig. 4.11, the target positions, the initial positions, and the current positions are represented by white stars, yellow crosses, and white circles, respectively. Surprisingly, the agents under the distributed navigation algorithm produce an emergent, swarm-like behavior to maintain communication connectivity among local neighbors. Notice that this collective behavior emerged naturally and was not generated by the flocking or swarming algorithm as in [10].

This interesting simulation study (Fig. 4.11) shows that agents won't get too close each other since the average of prediction error variances at target points can be reduced by spreading over and covering the target points that need to be sampled. However, agents won't move too far away each other since the average of prediction error variances can be

reduced by collecting measurements from a larger population of neighbors. This trade-off is controlled by the communication range. With the intertwined dynamics of agents over the proximity graph, as shown in Fig. 4.11, mobile sensing agents are coordinated in each time iteration in order to dynamically cover the target positions for better collective prediction capability.

Table 4.3: Distributed sampling strategy at time t.

Input:
(1) Number of agents N
(2) Positions of agents $\{q_i(t) \,|\, i \in \mathcal{I}\}$
(3) Hyperparameters of the Gaussian process $\theta = (\sigma_f^2, \sigma_x, \sigma_y, \sigma_t)^T$
(4) Target points $\{p_j \,|\, j \in \mathcal{J}\}$
(5) Truncation size η
Output:
(1) Prediction at target points $\left\{ \mu_{z_j | y_{t-\eta+1:t}^{[i]}} \,|\, i \in \mathcal{I}, j \in \mathcal{J} \right\}$
(2) Prediction error variances at target points $\left\{ \sigma^2_{z_j | y_{t-\eta+1:t}^{[i]}} \,|\, i \in \mathcal{I}, j \in \mathcal{J} \right\}$

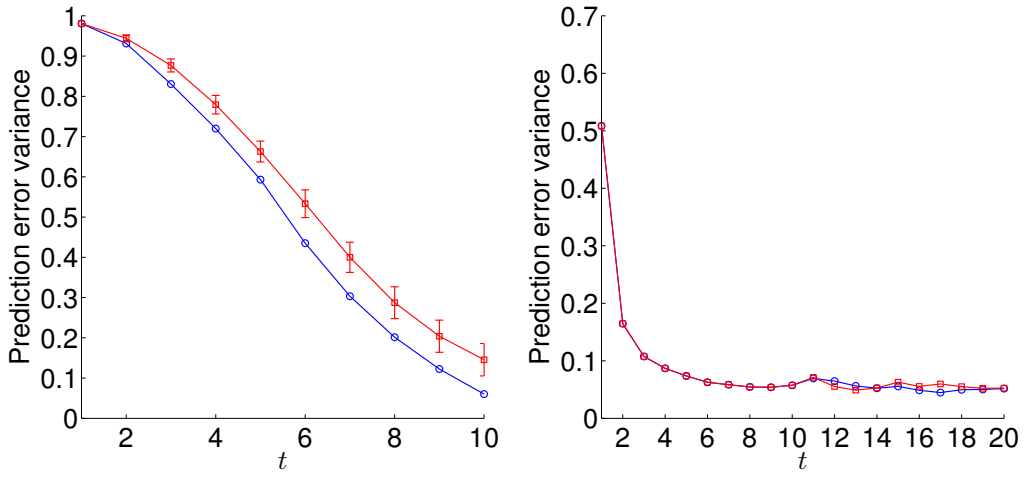
For $i \in \mathcal{I}$, agent i performs:
1: make an observation at $q_i(t)$, *i.e.*, $y_i(t)$
2: transmit the observation to the neighbors in $\mathcal{N}_i(t)$
3: collect the observations from neighbors in $\mathcal{N}_i(t)$, *i.e.*, $y^{[i]}(t)$
4: obtain the cumulative measurements, *i.e.*, $y_{t-\eta+1:t}^{[i]} = \left((y_{t-\eta+1}^{[i]})^T, \cdots, (y_t^{[i]})^T\right)^T$
5: **for** $j \in \mathcal{J}$ **do**
6: make prediction at a target point p_j

$$\mu_{z_j | y_{t-\eta+1:t}^{[i]}} = k^T C^{-1} y,$$

with a prediction error variance given by

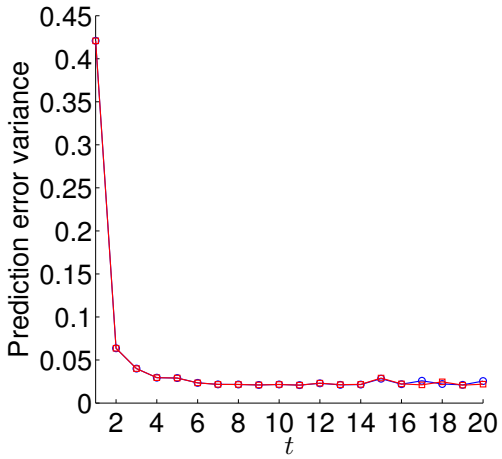
$$\sigma^2_{z_j | y_{t-\eta+1:t}^{[i]}} = \sigma_f^2 (1 - k^T C^{-1} k),$$

where $y = y_{t-\eta+1:t}^{[i]}$, $k = \mathrm{Corr}(y, z_j)$, and $C = \mathrm{Corr}(y, y)$
7: **end for**
8: **if** $t \geq \eta$ **then**
9: discard the oldest set of measurements taken at time $t - \eta + 1$, *i.e.*, $y_{t-\eta+1}^{[i]}$
10: **end if**
11: **while** $t \leq \tau \leq t + 1$ **do**
12: compute $\nabla_{q_\ell} J^{[i]}$ with the remained data $y_{t-\eta+2}^{[i]}$
13: send $\nabla_{q_\ell} J^{[i]}$ to agent ℓ in $\mathcal{N}_i(\tau)$
14: receive $\nabla_{q_i} J^{[\ell]}$ from all neighbors in $\mathcal{N}_i(\tau)$
15: compute the gradient $\nabla_{q_i} J_d = \sum_{\ell \in \mathcal{N}_i(\tau)} \nabla_{q_i} J^{[\ell]} / |\mathcal{I}|$
16: update position according to $q_i(\tau + \delta t) = q_i(\tau) - \alpha \nabla_{q_i} J_d$ for a small step size α
17: **end while**



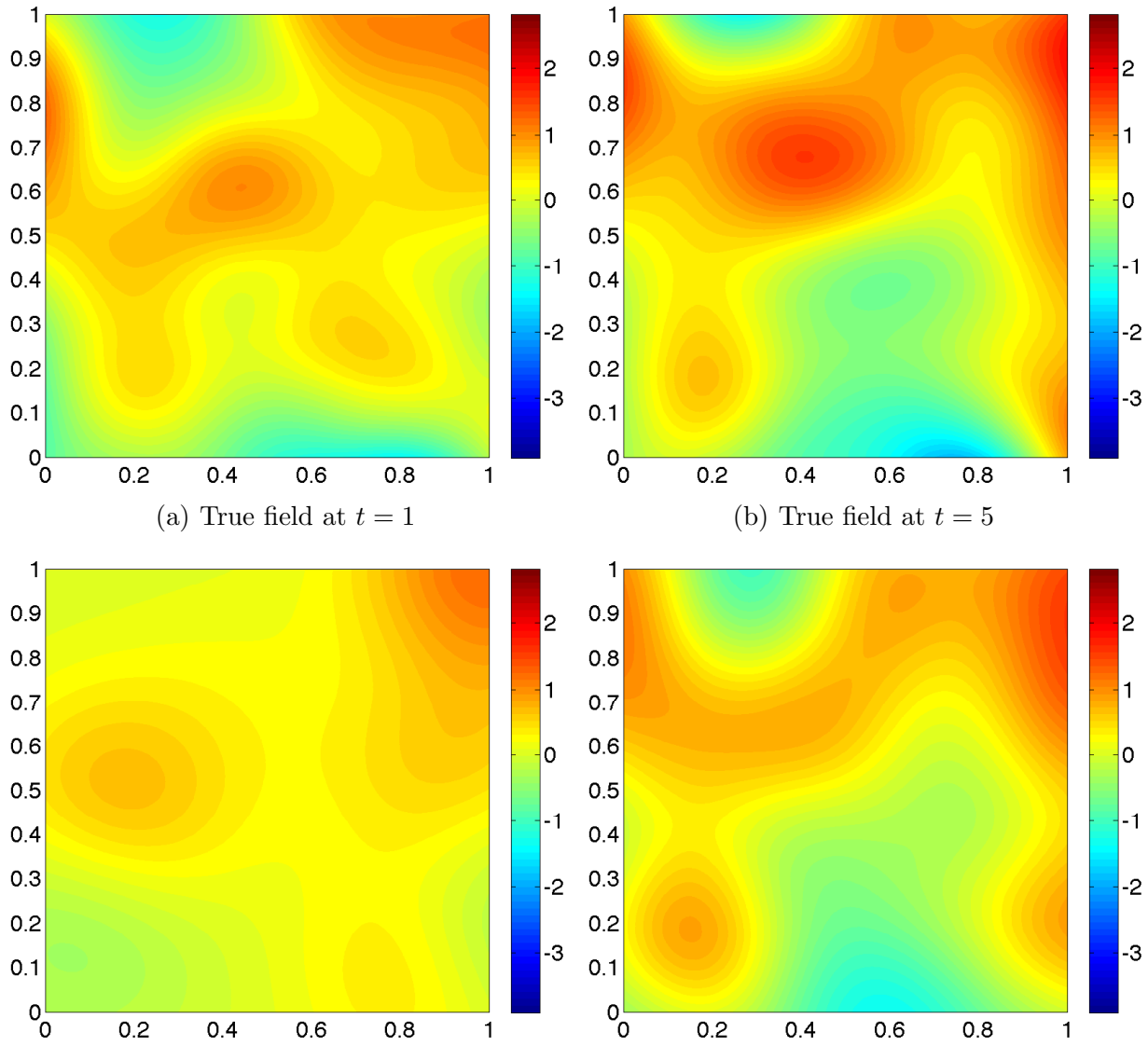
(a) Case 1

(b) Case 2



(c) Case 3

Figure 4.6: Average of prediction error variances over target points (in blue circles) achieved by the centralized sampling scheme using all collective observations for (a) case 1, (b) case 2, and (c) case 3. In (a), the target points are fixed at time $t = 10$, and the counterpart achieved by the benchmark random sampling strategy is shown in red squares with error-bars. In (b) and (c), the target points are at $t + 1$ and change over time. The counterpart achieved by using truncated observations are shown in red squares.



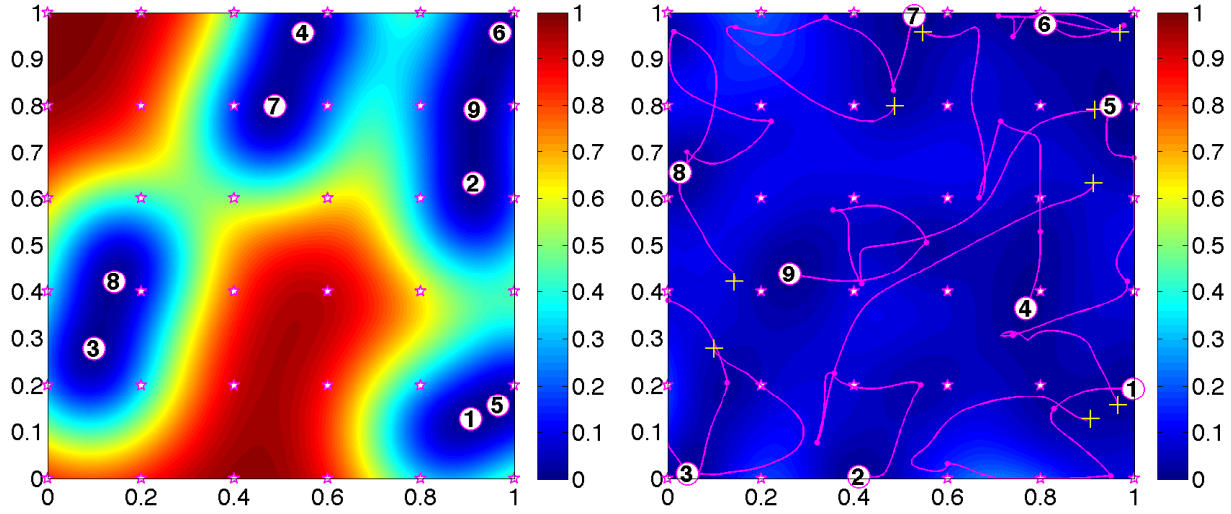
(a) True field at $t = 1$

(b) True field at $t = 5$

(c) Predicted field at $t = 1$

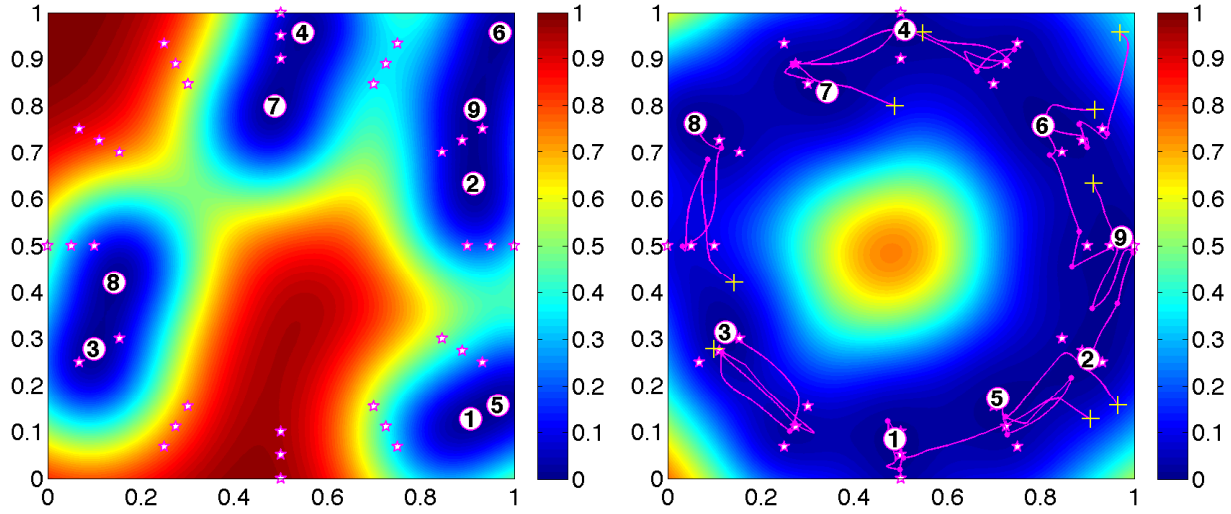
(d) Predicted field at $t = 5$

Figure 4.7: Simulation results at $t = 1$ and $t = 5$ obtained by the centralized sampling scheme for case 2.



(e) Prediction error variance at $t = 1$ (f) Prediction error variance at $t = 5$

Figure 4.7: Simulation results at $t = 1$ and $t = 5$ obtained by the centralized sampling scheme for case 2 (cont'd).



(a) Prediction error variance at $t = 1$ (b) Prediction error variance at $t = 5$

Figure 4.8: Simulation results obtained by the centralized sampling scheme for case 3. The trajectories of agents are shown in solid lines.

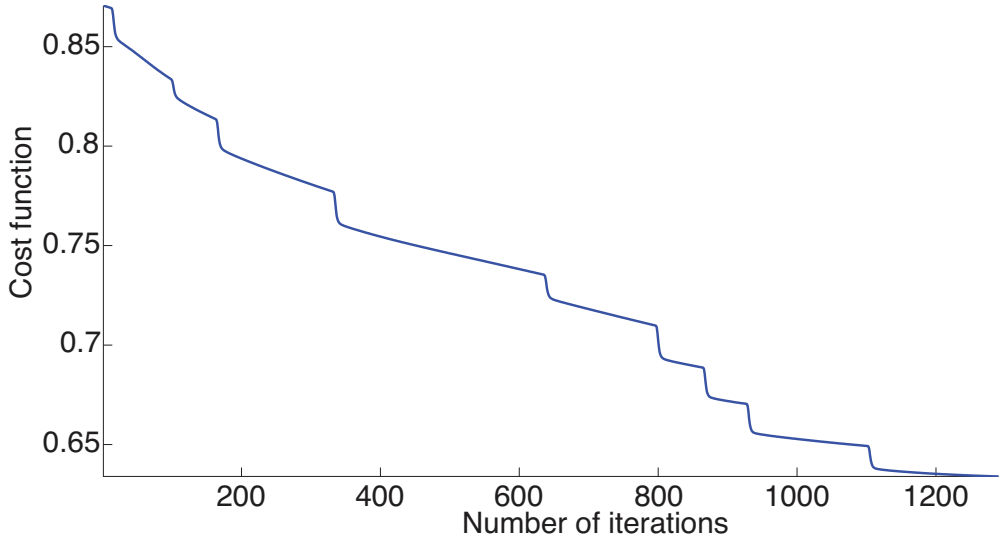
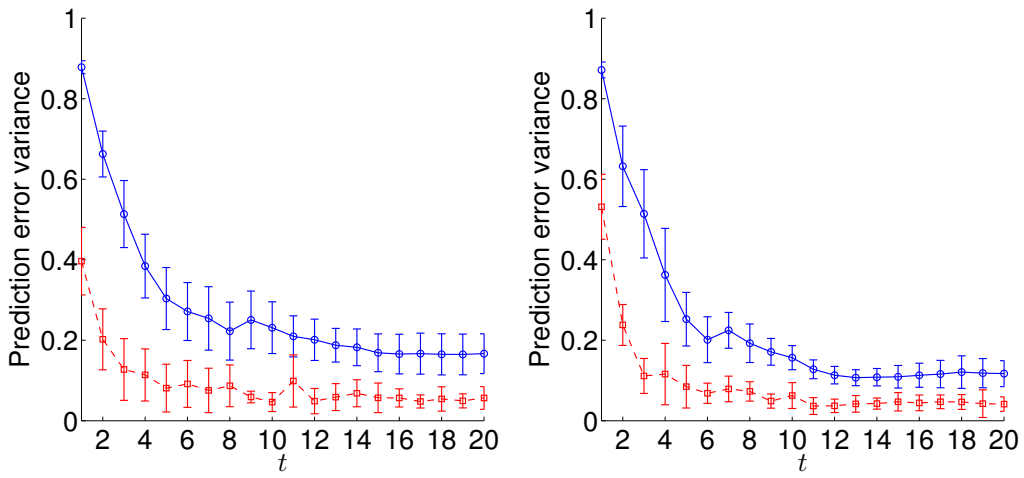
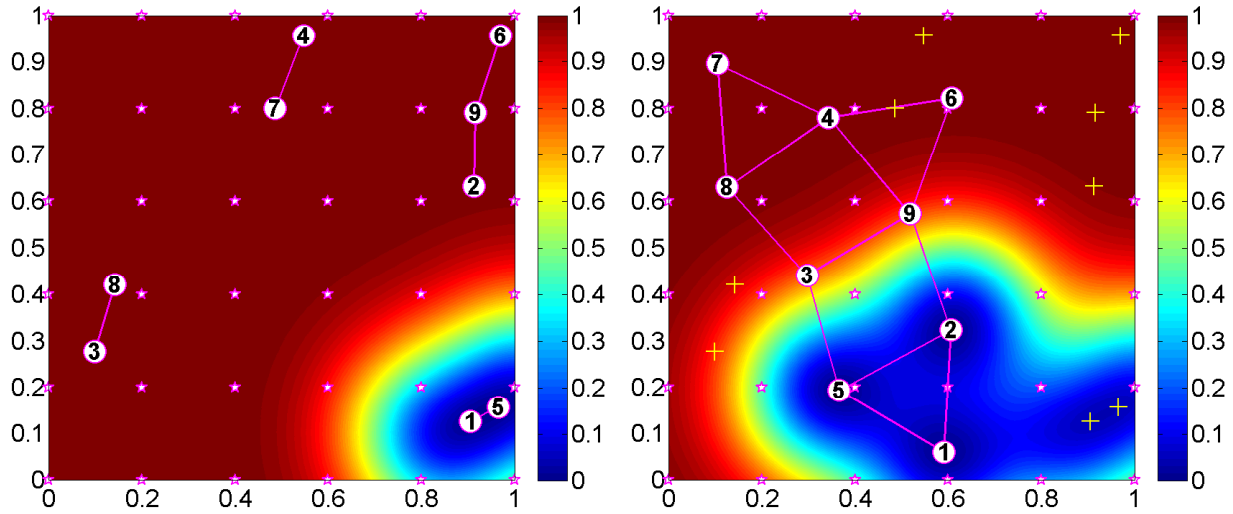


Figure 4.9: Cost function $J_d(\tilde{q})$ from $t = 1$ to $t = 2$ with a communication range $R = 0.4$.



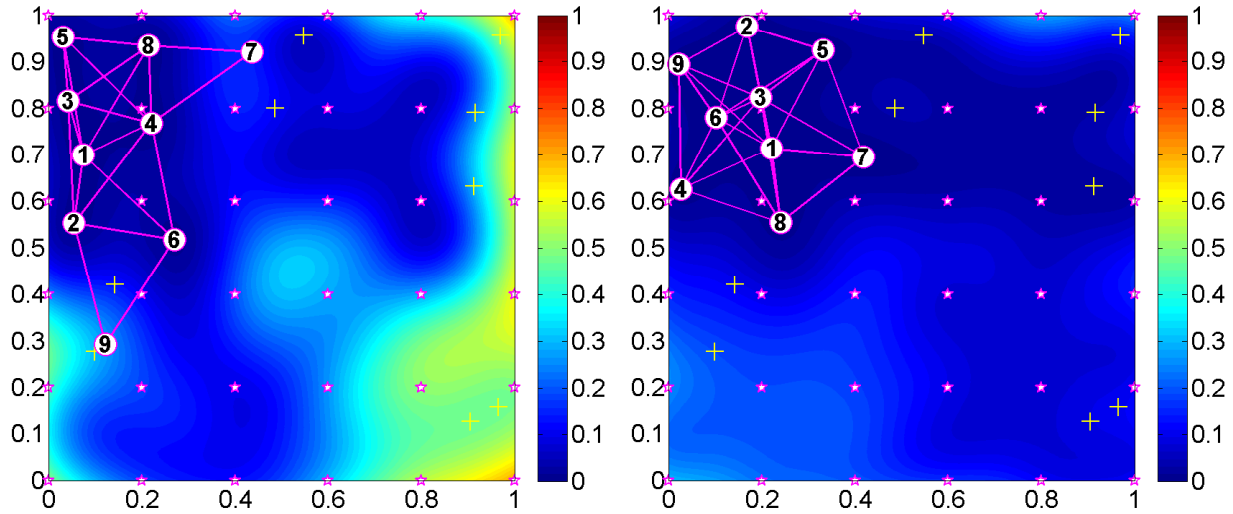
(a) $R = 0.3$ (b) $R = 0.4$

Figure 4.10: Average of prediction error variances over all target points and agents achieved by the distributed sampling scheme with a communication range (a) $R = 0.3$, and (b) $R = 0.4$. The average of prediction error variances over all target points and agents are shown in blue circles. The average of prediction error variance over local target points and agents are shown in red squares. The error-bars indicate the standard deviation among agents.



(a) $R = 0.3$, $t = 1$

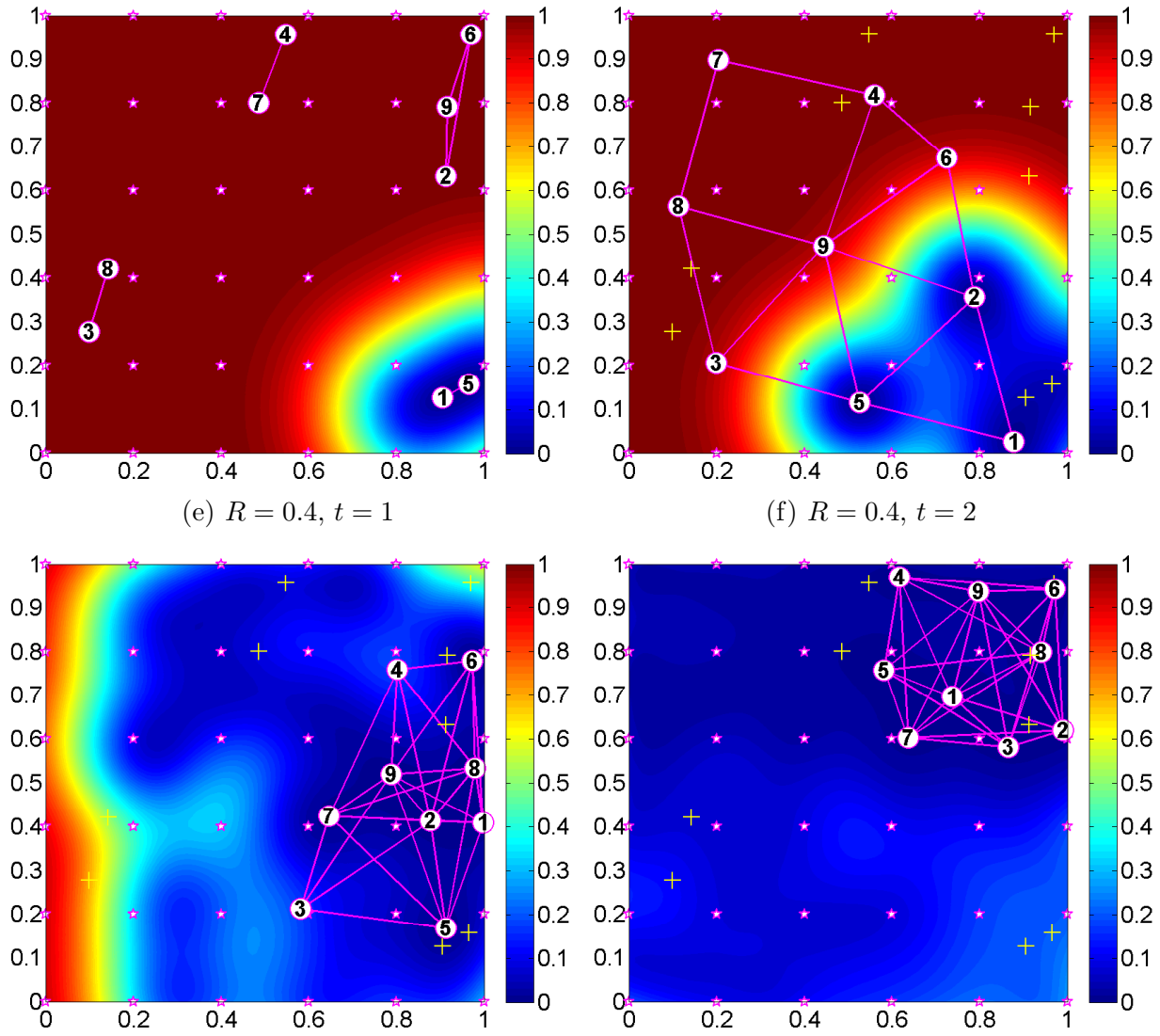
(b) $R = 0.3$, $t = 2$



(c) $R = 0.3$, $t = 5$

(d) $R = 0.3$, $t = 20$

Figure 4.11: Simulation results obtained by the distributed sampling scheme with different communication ranges. The edges of the graph are shown in solid lines.



(e) $R = 0.4$, $t = 1$

(f) $R = 0.4$, $t = 2$

(g) $R = 0.4$, $t = 5$

(h) $R = 0.4$, $t = 20$

Figure 4.11: Simulation results obtained by the distributed sampling scheme with different communication ranges (cont'd). The edges of the graph are shown in solid lines.

Chapter 5

Fully Bayesian Approach

Recently, there has been an increasing exploitation of mobile sensor networks in environmental monitoring [10, 13, 35, 37]. Gaussian process regression (or kriging in geostatistics) has been widely used to draw statistical inference from geostatistical and environmental data [15, 51]. For example, near-optimal static sensor placements with a mutual information criterion in Gaussian processes were proposed in [32]. A distributed kriged Kalman filter for spatial estimation based on mobile sensor networks was developed in [13]. Multi-agent systems that are versatile for various tasks by exploiting predictive posterior statistics of Gaussian processes were developed in [8, 9].

The significant computational complexity in Gaussian process regression due to the growing number of observations (and hence the size of covariance matrix) has been tackled in different ways. In [67], the authors analyzed the conditions under which near-optimal prediction can be achieved using only truncated observations. This motivates the usage of sparse Gaussian process proposed in [45]. However, they both assumed the covariance function is known a priori, which is unrealistic in practice. On the other hand, unknown parameters in

the covariance function can be estimated by the maximum likelihood (ML) estimator. Such ML estimates may be regarded as the true parameters and then used in the prediction [65]. However, the point estimate itself needs to be identified using sufficient amount of measurements. Instead, a maximum *a posterior* (MAP) estimate can use the prior to provide the point estimate with a small number of measurements. However, it fails to incorporate the uncertainty in the estimate into the prediction.

The advantage of a fully Bayesian approach, which will be adopted in this work, is that the uncertainty in the model parameters are incorporated in the prediction [5]. In [22], Gaudard *et al.* presented a Bayesian method that uses importance sampling for analyzing spatial data sampled from a Gaussian random field whose covariance function was unknown. However, the assumptions made in [22], such as noiseless observations and time-invariance of the field, limit the applicability of the approach on mobile sensors in practice. The computational complexity of a fully Bayesian prediction algorithm has been the main hurdle for applications in resource-constrained robots. In [25], an iterative prediction algorithm without resorting to Markov Chain Monte Carlo (MCMC) methods has been developed based on analytical closed-form solutions from results in [22], by assuming that the covariance function of the spatio-temporal Gaussian random field is known up to a constant. Our work builds on such Bayesian approaches used in [22,25] and explores new ways to synthesize practical algorithms for mobile sensor networks under more relaxed conditions.

In Section Section 5.1, we provide fully Bayesian approaches for spatio-temporal Gaussian process regression under more practical conditions such as measurement noise and the unknown covariance function . In Section 5.2, using discrete prior probabilities and compactly supported kernels, we provide a way to design sequential Bayesian prediction algorithms in

which the exact predictive distributions can be computed in constant time as the number of observations increases. In particular, a centralized sequential Bayesian prediction algorithm is developed in Section 5.2.1, and its distributed implementation among sensor groups is provided for a special case in Section 5.2.2. An adaptive sampling strategy for mobile sensors, utilizing the maximum *a posteriori* (MAP) estimation of the parameters, is proposed to minimize the prediction error variances in Section 5.2.3. In Section 5.3, the proposed sequential Bayesian prediction algorithms and the adaptive sampling strategy are tested under practical conditions for spatio-temporal Gaussian processes.

5.1 Fully Bayesian Prediction Approach

In this chapter, we consider a spatio-temporal Gaussian process denoted by

$$z(x) \sim \mathcal{GP}\left(\mu(x), \sigma_f^2 C(x, x'; \theta)\right),$$

where $z(x) \in \mathbb{R}$ and $x := (s^T, t)^T \in \mathcal{Q} \times \mathbb{Z}_{>0}$ contains the sampling location $s \in \mathcal{Q} \subset \mathbb{R}^D$ and the sampling time $t \in \mathbb{Z}_{>0}$. The mean function is assumed to be

$$\mu(x) = f(x)^T \beta,$$

where $f(x) := (f_1(x), \cdots, f_p(x))^T \in \mathbb{R}^p$ is a known regression function, and $\beta \in \mathbb{R}^p$ is an unknown vector of regression coefficients. The correlation between $z(x)$ and $z(x')$ is taken as

$$C(x, x'; \theta) = C_s\left(\frac{\|s - s'\|}{\sigma_s}\right) C_t\left(\frac{|t - t'|}{\sigma_t}\right), \tag{5.1}$$

which is governed by spatial and temporal distance functions $C_s(\cdot)$ and $C_t(\cdot)$. We assume that $C_s(\cdot)$ and $C_t(\cdot)$ are decreasing kernel functions over space and time, respectively, so that the correlation between two inputs decreases as the distance between spatial locations (respectively, time indices) increases. The decreasing rate depends on the spatial bandwidth σ_s (respectively, the time bandwidth σ_t) for given fixed time indices (respectively, spatial locations). The signal variance σ_f^2 gives the overall vertical scale relative to the mean of the Gaussian process in the output space. We define $\theta := (\sigma_s, \sigma_t)^T \in \mathbb{R}^2$ for notational simplicity.

Given the collection of noise corrupted observations from mobile sensing agents up to time t, we want to predict $z(s_*, t_*)$ at a prespecified location $s_* \in \mathcal{S} \subset \mathcal{Q}$ and current (or future) time t_*. To do this, suppose we have a collection of n observations $\mathcal{D} = \left\{(x^{(i)}, y^{(i)}) \,|\, i = 1, \ldots, n\right\}$ from N mobile sensing agents up to time t. Here $x^{(i)}$ denotes the i-th input vector of dimension $D + 1$ (*i.e.*, the sampling position and time of the i-th observation) and $y^{(i)}$ denotes the i-th noise corrupted measurement. If all observations are considered, we have $n = Nt$. Notice that the number of observations n grows with the time t. For notational simplicity, let $y := (y^{(1)}, \cdots, y^{(n)})^T \in \mathbb{R}^n$ denote the collection of noise corrupted observations. Based on the spatio-temporal Gaussian process, the distribution of the observations given the parameters β, σ_f^2, and θ is Gaussian, *i.e.*,

$$y|\beta, \sigma_f^2, \theta \sim \mathcal{N}(F\beta, \sigma_f^2 C)$$

with F and C defined as

$$\begin{aligned} F &:= \Big(f(x^{(1)}), \cdots, f(x^{(n)})\Big)^T \in \mathbb{R}^{n\times p}, \\ C &:= \mathrm{Corr}(y, y|\theta) = \left[C(x^{(i)}, x^{(j)}; \theta) + \frac{1}{\gamma}\delta_{ij}\right] \in \mathbb{R}^{n\times n}, \end{aligned} \tag{5.2}$$

where δ_{ij} is the Kronecker delta which equals to one when $i = j$, and zero, otherwise.

5.1.1 Prior selection

To infer the unknown parameters β, σ_f^2, and θ in a Bayesian framework, the collection of them is considered to be a random vector with a prior distribution reflecting the *a priori* belief of uncertainty for them. In this chapter, we use the prior distribution given by

$$\pi(\beta, \sigma_f^2, \theta) = \pi(\beta|\sigma_f^2)\pi(\sigma_f^2)\pi(\theta), \tag{5.3}$$

where

$$\beta|\sigma_f^2 \sim \mathcal{N}(\beta_0, \sigma_f^2 T).$$

The prior for $\pi(\sigma_f^2)$ is taken to be the inverse gamma distribution, chosen to guarantee positiveness of σ_f^2 and a closed-form expression for the posterior distribution of σ_f^2 for computational ease of the proposed algorithms. To cope with the case where no prior knowledge on β is available, which is often the case in practice, we propose to use a noninformative prior. In particular, we take $\beta_0 = 0$, $T = \alpha I$, and subsequently, let $\alpha \to \infty$. Any proper prior $\pi(\theta)$ that correctly reflects the priori knowledge of θ can be used.

5.1.2 MCMC-based approach

According to the Bayes rule, the posterior distribution of β, σ_f^2, and θ is given by

$$\pi(\beta, \sigma_f^2, \theta|y) = \frac{\pi(y|\beta, \sigma_f^2, \theta)\pi(\beta, \sigma_f^2, \theta)}{\iiint \pi(y|\beta, \sigma_f^2, \theta)\pi(\beta, \sigma_f^2, \theta)d\beta d\sigma_f^2 d\theta}. \tag{5.4}$$

When a proper prior is used, the posterior distribution can be written as

$$\pi(\beta, \sigma_f^2, \theta|y) \propto \pi(y|\beta, \sigma_f^2, \theta)\pi(\beta, \sigma_f^2, \theta).$$

The inference on β, σ_f^2, and θ can be carried out by sampling from the posterior distribution in (5.4) via the Gibbs sampler. Table 5.1 gives the steps based on the following proposition.

Table 5.1: Gibbs sampler.

Input: initial samples $\beta^{(1)}$, ${\sigma_f^2}^{(1)}$, and $\theta^{(1)}$ **Output:** samples $\left\{\beta^{(i)}, {\sigma_f^2}^{(i)}, \theta^{(i)}\right\}_{i=1}^{m}$ from joint distribution $\pi(\beta, \sigma_f^2, \theta\|y)$
1: initialize $\beta^{(1)}, {\sigma_f^2}^{(1)}, \theta^{(1)}$ 2: **for** $i = 1$ to m **do** 3: sample $\beta^{(i+1)}$ from $\pi(\beta\|{\sigma_f^2}^{(i)}, \theta^{(i)}, y)$ 4: sample ${\sigma_f^2}^{(i+1)}$ from $\pi(\sigma_f^2\|\beta^{(i+1)}, \theta^{(i)}, y)$ 5: sample $\theta^{(i+1)}$ from $\pi(\theta\|\beta^{(i+1)}, {\sigma_f^2}^{(i+1)}, y)$ 6: **end for**

Proposition 5.1.1. *For a prior distribution given in* (5.3) *with the noninformative prior on* β*, the conditional posteriors are given by*

1. $\beta|\sigma_f^2, \theta, y \sim \mathcal{N}\left(\hat{\beta}, \sigma_f^2 \Sigma_{\hat{\beta}}\right)$, *where*

$$\Sigma_{\hat{\beta}} = (F^T C^{-1} F)^{-1},$$
$$\hat{\beta} = \Sigma_{\hat{\beta}}(F^T C^{-1} y),$$

2. $\sigma_f^2|\beta, \theta, y \sim IG\left(\bar{a}, \bar{b}\right)$, *where*

$$\bar{a} = a + \frac{n+p}{2},$$
$$\bar{b} = b + \frac{1}{2}(y - F\beta)^T C^{-1}(y - F\beta),$$

3. and

$$\pi(\theta|\beta, \sigma_f^2, y) \propto \det(C)^{-1/2} \exp\left(-\frac{(y - F\beta)^T C^{-1}(y - F\beta)}{2\sigma_f^2}\right) \pi(\theta).$$

Proof. Since the noninformative prior is chosen, the posterior distribution shall be computed with $T = \alpha I$ and then let $\alpha \to \infty$.

i) For given σ_f^2, θ, and y, we have

$$\pi(\beta|\sigma_f^2, \theta, y) = \lim_{\alpha \to \infty} \frac{\pi(y|\beta, \sigma_f^2, \theta)\pi(\beta|\sigma_f^2)}{\int \pi(y|\beta, \sigma_f^2, \theta)\pi(\beta|\sigma_f^2) d\beta}.$$

Let

$$
\begin{aligned}
\texttt{num}_1 &= \pi(y|\beta,\sigma_f^2,\theta)\pi(\beta|\sigma_f^2) \\
&= \frac{\exp\left\{-\frac{1}{2\sigma_f^2}(y-F\beta)^T C^{-1}(y-F\beta)\right\}}{(2\pi\sigma_f^2)^{n/2}\det(C)^{1/2}} \frac{\exp\left\{-\frac{1}{2\sigma_f^2}\beta^T T^{-1}\beta\right\}}{(2\pi\sigma_f^2)^{p/2}\det(T)^{1/2}} \\
&= \frac{\exp\left\{-\frac{1}{2\sigma_f^2}RSS\right\}}{(2\pi\sigma_f^2)^{(n+p)/2}\det(C)^{1/2}\det(T)^{1/2}} \exp\left\{-\frac{1}{2\sigma_f^2}(\beta-\hat{\beta})^T(F^T C^{-1}F+T^{-1})(\beta-\hat{\beta})\right\},
\end{aligned}
$$

and

$$
\begin{aligned}
\texttt{den}_1 &= \frac{\exp\left\{-\frac{1}{2\sigma_f^2}RSS\right\}}{(2\pi\sigma_f^2)^{(n+p)/2}\det(C)^{1/2}\det(T)^{1/2}} \int \exp\left\{-\frac{1}{2\sigma_f^2}(\beta-\hat{\beta})^T(F^T C^{-1}F+T^{-1})(\beta-\hat{\beta})\right\} d\beta \\
&= \frac{\exp\left\{-\frac{1}{2\sigma_f^2}RSS\right\}}{(2\pi\sigma_f^2)^{(n+p)/2}\det(C)^{1/2}\det(T)^{1/2}} (2\pi\sigma_f^2)^{p/2}\det(F^T C^{-1}F+T^{-1})^{-1/2}
\end{aligned}
$$

where

$$
\begin{aligned}
RSS &= y^T\left(C^{-1}-C^{-1}F(F^T C^{-1}F+T^{-1})^{-1}F^T C^{-1}\right)y \\
&= y^T(C+FTF^T)^{-1}y.
\end{aligned}
$$

Then we have

$$\begin{aligned}\pi(\beta|\sigma_f^2,\theta,y) &= \lim_{\alpha\to\infty}\frac{\texttt{num}_1}{\texttt{den}_1}\\ &= \lim_{\alpha\to\infty}\frac{\exp\left\{-\frac{1}{2\sigma_f^2}(\beta-\hat{\beta})^T(F^TC^{-1}F+T^{-1})(\beta-\hat{\beta})\right\}}{(2\pi\sigma_f^2)^{p/2}\det(F^TC^{-1}F+T^{-1})^{-1/2}}\\ &= \frac{\exp\left\{-\frac{1}{2\sigma_f^2}(\beta-\hat{\beta})^T\Sigma_{\hat{\beta}}^{-1}(\beta-\hat{\beta})\right\}}{(2\pi\sigma_f^2)^{p/2}\det(\Sigma_{\hat{\beta}})^{1/2}}.\end{aligned}$$

Therefore, we have $\beta|\sigma_f^2,\theta,y\sim\mathcal{N}(\hat{\beta},\sigma_f^2\Sigma_{\hat{\beta}})$.

ii) For given β, θ, and y, we have

$$\begin{aligned}\pi(\sigma_f^2|\beta,\theta,y) &= \lim_{\alpha\to\infty}\frac{\pi(y|\beta,\sigma_f^2,\theta)\pi(\sigma_f^2|\beta)}{\int\pi(y|\beta,\sigma_f^2,\theta)\pi(\sigma_f^2|\beta)d\sigma_f^2}\\ &= \lim_{\alpha\to\infty}\frac{\pi(y|\beta,\sigma_f^2,\theta)\pi(\beta|\sigma_f^2)\pi(\sigma_f^2)}{\int\pi(y|\beta,\sigma_f^2,\theta)\pi(\beta|\sigma_f^2)\pi(\sigma_f^2)d\sigma_f^2}.\end{aligned}$$

Let

$$\begin{aligned}\texttt{num}_2 &= \pi(y|\beta,\sigma_f^2,\theta)\pi(\beta|\sigma_f^2)\pi(\sigma_f^2)\\ &= \frac{\exp\left\{-\frac{1}{2\sigma_f^2}(y-F\beta)^TC^{-1}(y-F\beta)\right\}}{(2\pi\sigma_f^2)^{n/2}\det(C)^{1/2}}\frac{\exp\left\{-\frac{1}{2\sigma_f^2}\beta^TT^{-1}\beta\right\}}{(2\pi\sigma_f^2)^{p/2}\det(T)^{1/2}}\frac{b^a\exp\left\{-\frac{b}{\sigma_f^2}\right\}}{\Gamma(a)(\sigma_f^2)^{a+1}}\\ &= \frac{b^a}{\Gamma(a)(2\pi)^{\bar{a}+1}\det(C)^{1/2}\det(T)^{1/2}}\frac{1}{(\sigma_f^2)^{\bar{a}+1}}\exp\left\{-\frac{\bar{b}+\frac{1}{2}\beta^TT^{-1}\beta}{\sigma_f^2}\right\},\end{aligned}$$

and

$$\begin{aligned}\texttt{den}_2 &= \frac{b^a}{\Gamma(a)(2\pi)^{\bar{a}+1}\det(C)^{1/2}\det(T)^{1/2}}\int \frac{1}{(\sigma_f^2)^{\bar{a}+1}}\exp\left\{-\frac{\bar{b}+\frac{1}{2}\beta^T T^{-1}\beta}{\sigma_f^2}\right\}d\sigma_f^2 \\ &= \frac{b^a}{\Gamma(a)(2\pi)^{\bar{a}+1}\det(C)^{1/2}\det(T)^{1/2}}\Gamma(\bar{a})\bar{b}^{-\bar{a}}.\end{aligned}$$

Then we have

$$\begin{aligned}\pi(\sigma_f^2|\beta,\theta,y) &= \lim_{\alpha\to\infty}\frac{\texttt{num}_2}{\texttt{den}_2} \\ &= \lim_{\alpha\to\infty}\frac{\bar{b}^{\bar{a}}}{\Gamma(\bar{a})(\sigma_f^2)^{\bar{a}+1}}\exp\left\{-\frac{\bar{b}+\frac{1}{2}\beta^T T^{-1}\beta}{2\sigma_f^2}\right\} \\ &= \frac{\bar{b}^{\bar{a}}}{\Gamma(\bar{a})(\sigma_f^2)^{\bar{a}+1}}\exp\left\{-\frac{\bar{b}}{2\sigma_f^2}\right\}.\end{aligned}$$

Therefore, we have $\sigma_f^2|\beta,\theta,y \sim IG\left(\bar{a},\bar{b}\right)$.

iii) For given β, σ_f^2, and y, we have

$$\begin{aligned}\pi(\theta|\beta,\sigma_f^2,y) &= \lim_{\alpha\to\infty}\frac{\pi(y|\beta,\sigma_f^2,\theta)\pi(\theta)}{\int\pi(y|\beta,\sigma_f^2,\theta)\pi(\theta)d\theta} \\ &\propto \det(C)^{-1/2}\exp\left(-\frac{(y-F\beta)^T C^{-1}(y-F\beta)}{2\sigma_f^2}\right)\pi(\theta).\end{aligned}$$

□

The posterior predictive distribution of $z_* := z(s_*,t_*)$ at location s_* and time t_* can be obtained by

$$\pi(z_*|y) = \iiint \pi(z_*|y,\beta,\sigma_f^2,\theta)\pi(\beta,\sigma_f^2,\theta|y)d\beta d\sigma_f^2 d\theta, \tag{5.5}$$

where in (5.5), the conditional distribution $\pi(z_*|\beta,\sigma_f^2,\theta,y)$, is integrated with respect to the posterior of β, σ_f^2, and θ given observations y. The conditional distribution of z_* is Gaussian,

i.e.,

$$z_*|\beta,\sigma_f^2,\theta,y \sim \mathcal{N}(\mu_{z_*|\beta,\sigma_f^2,\theta,y}, \sigma^2_{z_*|\beta,\sigma_f^2,\theta,y}),$$

with

$$\mu_{z_*|\beta,\sigma_f^2,\theta,y} = \mathrm{E}(z_*|\beta,\sigma_f^2,\theta,y) = f(x_*)^T\beta + k^T C^{-1}(y - F\beta),$$
$$\sigma^2_{z_*|\beta,\sigma_f^2,\theta,y} = \mathrm{Var}(z_*|\beta,\sigma_f^2,\theta,y) = \sigma_f^2(1 - k^T C^{-1}k),$$

where $k := \mathrm{Corr}(y, z_*|\theta) = [\mathcal{K}(x^{(i)}, x_*; \theta)] \in \mathbb{R}^n$. To obtain numerical values of $\pi(z_*|y)$, we draw m samples $\left\{\beta^{(i)}, \sigma_f^{2\,(i)}, \theta^{(i)}\right\}_{i=1}^m$ from the posterior distribution $\pi(\beta,\sigma_f^2,\theta|y)$ using the Gibbs sampler presented in Table 5.1, and then obtain the predictive distribution in (5.5) by

$$\pi(z_*|y) \approx \frac{1}{m}\sum_{i=1}^m \pi(z_*|y,\beta^{(i)},\sigma_f^{2\,(i)},\theta^{(i)}).$$

It follows that the predictive mean and variance can be obtained numerically by

$$\mu_{z_*|y} = \mathrm{E}(z_*|y) \approx \frac{1}{m}\sum_{i=1}^m \mu_{z_*|\beta^{(i)},\sigma_f^{2\,(i)},\theta^{(i)},y},$$
$$\sigma^2_{z_*|y} = \mathrm{Var}(z_*|y) \approx \frac{1}{m}\sum_{i=1}^m \sigma^2_{z_*|\beta^{(i)},\sigma_f^{2\,(i)},\theta^{(i)},y} + \frac{1}{m}\sum_{i=1}^m \left(\mu_{z_*|\beta^{(i)},\sigma_f^{2\,(i)},\theta^{(i)},y} - \mu_{z_*|y}\right)^2.$$

Remark 5.1.2. *The Gibbs sampler presented in Table 5.1 may take long time to converge, which implies that the number of samples required could be quite large depending on the initial values. This convergence rate can be monitored from a trace plot (a plot of sampled values v.s. iterations for each variable in the chain). Moreover, since C is a complicated function of σ_s and σ_t, sampling from $\pi(\theta|\beta,\sigma_f^2,y)$ in Proposition 5.1.1 is difficult. An inverse cumulative distribution function (CDF) method [19] needs to be used to generate samples, which requires*

griding on a continuous parameter space. Therefore, high computational power is needed to implement the MCMC-based approach.

In the next subsection, we present an alternative Bayesian approach which only requires drawing samples from the prior distribution $\pi(\theta)$ using a similar approach to one used in [22].

5.1.3 Importance sampling approach

The posterior predictive distribution of $z_* := z(s_*, t_*)$ can be written as

$$\pi(z_*|y) = \int \pi(z_*|\theta, y)\pi(\theta|y)d\theta, \tag{5.6}$$

where

$$\pi(\theta|y) = \frac{\pi(y|\theta)\pi(\theta)}{\int \pi(y|\theta)\pi(\theta)d\theta},$$

is the posterior distribution of θ, by integrating out analytically the parameters β and σ_f^2. We have the following proposition.

Proposition 5.1.3. *For a prior distribution given in* (5.3) *with the noninformative prior on* β*, we have*

1. $\pi(\theta|y) \propto w(\theta|y)\pi(\theta)$ *with*

$$\log w(\theta|y) = -\frac{1}{2}\log\det(C) - \frac{1}{2}\log\det(F^T C^{-1} F) - \tilde{a}\log\tilde{b}, \tag{5.7}$$

where

$$\tilde{a} = a + \frac{n}{2},$$
$$\tilde{b} = b + \frac{1}{2}y^T C^{-1} y - \frac{1}{2}(F^T C^{-1} y)^T (F^T C^{-1} F)^{-1} (F^T C^{-1} y).$$

2. *$\pi(z_*|\theta, y)$ is a shifted student's t-distribution with location parameter μ, scale parameter λ, and ν degrees of freedom,* i.e.,

$$\pi(z_*|\theta, y) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{\lambda}{\pi\nu}\right)^{\frac{1}{2}} \left(1 + \frac{\lambda(z_* - \mu)^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \tag{5.8}$$

where $\nu = 2\tilde{a}$, and

$$\mu = k^T C^{-1} y + (f(x_*) - F^T C^{-1} k)^T (F^T C^{-1} F)^{-1} (F^T C^{-1} y),$$
$$\lambda = \frac{\tilde{b}}{\tilde{a}}\big((1 - k^T C^{-1} k) + (f(x_*) - F^T C^{-1} k)^T (F^T C^{-1} F)^{-1} (f(x_*) - F^T C^{-1} k)\big).$$

Proof. i) For given θ, we have

$$\begin{aligned}
\pi(y|\theta) &= \iint \pi(y|\beta, \sigma_f^2, \theta)\pi(\beta, \sigma_f^2) d\beta d\sigma_f^2 \\
&= \iint \pi(y|\beta, \sigma_f^2, \theta)\pi(\beta|\sigma_f^2)\pi(\sigma_f^2) d\beta d\sigma_f^2 \\
&= \frac{b^a}{\Gamma(a)(2\pi)^{n/2}\det(C)^{1/2}\det(T)^{1/2}\det(F^T C^{-1} F + T^{-1})^{1/2}} \int \frac{\exp\left\{-\frac{b + \frac{RSS}{2}}{\sigma_f^2}\right\}}{(\sigma_f^2)^{n/2+a+1}} d\sigma_f^2 \\
&= \frac{\Gamma(\frac{n+2a}{2}) b^a}{\Gamma(a)(2\pi)^{n/2}\det(C)^{1/2}\det(T)^{1/2}\det(F^T C^{-1} F + T^{-1})^{1/2}} \left(b + \frac{RSS}{2}\right)^{-\frac{n+2a}{2}}
\end{aligned}$$

where

$$RSS = y^T \left(C^{-1} - C^{-1}F(F^T C^{-1} F + T^{-1})^{-1} F^T C^{-1} \right) y.$$

As $\alpha \to \infty$, we have

$$\begin{aligned} \pi(\theta|y) &= \lim_{\alpha \to \infty} \frac{\pi(y|\theta)\pi(\theta)}{\int \pi(y|\theta)\pi(\theta) d\theta} \\ &\propto \det(C)^{-1/2} \det(F^T C^{-1} F)^{-1/2} \left(b + \frac{1}{2} y^T \Sigma y \right)^{-\frac{n+2a}{2}}, \end{aligned}$$

where $\Sigma = C^{-1} - C^{-1}F(F^T C^{-1} F)^{-1} F^T C^{-1}$.

ii) For given θ and y, we have

$$\begin{aligned} \pi(z_*|\theta, y) &= \iint \pi(z_*|y, \beta, \sigma_f^2, \theta) \pi(\beta, \sigma_f^2|\theta, y) d\beta d\sigma_f^2 \\ &= \iint \pi(z_*|y, \beta, \sigma_f^2, \theta) \pi(\beta|\sigma_f^2, \theta, y) \pi(\sigma_f^2|\theta, y) d\beta d\sigma_f^2, \end{aligned}$$

where

$$\begin{aligned} z_*|y, \beta, \sigma_f^2, \theta &\sim \mathcal{N}\left(f(x_*)^T \beta + k^T C^{-1}(y - F\beta), \sigma_f^2(1 - k^T C^{-1} k) \right), \\ \beta|\sigma_f^2, \theta, y &\sim \mathcal{N}(\hat{\beta}, \sigma_f^2 \Sigma_{\hat{\beta}}), \\ \sigma_f^2|\theta, y &\sim IG\left(a + \frac{n}{2}, b + \frac{RSS}{2} \right). \end{aligned}$$

Then, it can be shown that

$$\pi(z_*|\theta, y) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{\lambda}{\pi\nu} \right)^{\frac{1}{2}} \left(1 + \frac{\lambda(z_* - \mu)^2}{\nu} \right)^{-\frac{\nu+1}{2}},$$

when $\alpha \to \infty$. $\square$

The results in Proposition 5.1.3 are different from those obtained in [22] by using a noninformative prior on β. For a special case where β and σ_f^2 are known a priori, we have the following corollary which will be exploited to derive a distributed implementation among sensor groups in Section 5.2.2.

Corollary 5.1.4. *In the case where β and σ_f^2 are known* a priori, (5.7) *and* (5.8) *can be simplified as*

$$\log w(\theta|y) = -\frac{1}{2}\log\det(C) - \frac{1}{2}(y - F\beta)^T C^{-1}(y - F\beta),$$
$$z_*|\theta, y \sim \mathcal{N}\left(f(x_*)^T\beta + k^T C^{-1}(y - F\beta), \sigma_f^2(1 - k^T C^{-1}k)\right).$$

If we draw m samples $\left\{\theta^{(i)}\right\}_{i=1}^m$ from the prior distribution $\pi(\theta)$, the posterior predictive distribution in (5.6) can then be approximated by

$$\pi(z_*|y) \approx \frac{\sum w(\theta^{(i)}|y)\pi(z_*|\theta^{(i)}, y)}{\sum w(\theta^{(i)}|y)}.$$

It follows that the predictive mean and variance can be obtained by

$$\mu_{z_*|y} = \mathrm{E}(z_*|y) \approx \frac{\sum w(\theta^{(i)}|y)\mu_{z_*|\theta^{(i)},y}}{\sum w(\theta^{(i)}|y)},$$
$$\sigma^2_{z_*|y} = \mathrm{Var}(z_*|y) \approx \frac{\sum w(\theta^{(i)}|y)\sigma^2_{z_*|\theta^{(i)},y}}{\sum w(\theta^{(i)}|y)} + \frac{\sum w(\theta^{(i)}|y)\left(\mu_{z_*|\theta^{(i)},y} - \mu_{z_*|y}\right)^2}{\sum w(\theta^{(i)}|y)},$$

where the mean and variance of the student's t-distribution $\pi(z_*|\theta, y)$ are given by

$$\mu_{z_*|\theta,y} = \mathrm{E}(z_*|\theta, y) = \mu,$$
$$\sigma^2_{z_*|\theta,y} = \mathrm{Var}(z_*|\theta, y) = \frac{\tilde{a}}{\tilde{a}-1}\lambda.$$

5.1.4 Discrete prior distribution

To further reduce the computational demands from the Monte Carlo approach, we assign discrete uniform probability distributions to σ_s and σ_t as priors instead of continuous probability distributions. Assume that we know the range of parameters in θ, *i.e.*,

$$\sigma_s \in [\underline{\sigma}_s, \overline{\sigma}_s] \text{ and } \sigma_t \in [\underline{\sigma}_t, \overline{\sigma}_t],$$

where $\underline{\sigma}$ and $\overline{\sigma}$ denote the known lower-bound and upper-bound of the random variable σ, respectively. We constrain the possible choices of θ on a finite set of grid points denoted by Θ. Hence, $\pi(\theta)$ is now a probability mass function (*i.e.*, $\sum_{\theta\in\Theta} \pi(\theta) = 1$) as opposed to a probability density. The integration in (5.6) is reduced to the following summation

$$\pi(z_*|y) = \sum_{\theta\in\Theta} \pi(z_*|\theta, y)\pi(\theta|y), \tag{5.9}$$

where the posterior distribution of θ is evaluated on the grid points in Θ by

$$\pi(\theta|y) = \frac{w(\theta|y)\pi(\theta)}{\sum_{\theta\in\Theta} w(\theta|y)\pi(\theta)}. \tag{5.10}$$

In order to obtain the posterior predictive distribution in (5.9), the computation of $\pi(z_*|\theta, y)$ and $w(\theta|y)$ for all $\theta \in \Theta$ using the results from Proposition 5.1.3 (or Corollary 5.1.4 for a

special case) are necessary. Note that these quantities are available in closed-form which reduces the computational burden significantly.

5.2 Sequential Bayesian Prediction

Although the aforementioned efforts in Sections 5.1.3 and 5.1.4 reduce the computational cost significantly, the number of observations (that mobile sensing agents collect) n increases with the time t. For each $\theta \in \Theta$, an $n \times n$ positive definite matrix C needs to be inverted which requires time $O(n^3)$ using standard methods. This motivates us to design scalable sequential Bayesian prediction algorithms by using subsets of observations.

5.2.1 Scalable Bayesian prediction algorithm

The computation of $\pi(z_*|y_{1:t})$ soon becomes infeasible as t increases. To overcome this drawback while maintaining the Bayesian framework, we propose to use subsets of all observations $y_{1:t} \in \mathbb{R}^{Nt}$. However, instead of using truncated local observations only as in [67], Bayesian inference will be drawn based on two sets of observations:

- First, a set of local observations near target points $\tilde{y}$ which will improve the quality of the prediction, and
- second, a cumulative set of observations $\bar{y}$ which will minimize the uncertainty in the estimated parameters.

Taken together, they improve the quality of prediction as the number of observations increases. We formulate this idea in detail in the following paragraph. For notational simplicity, we define $y \in \mathbb{R}^{Nt}$ as a subset of all observations $y_{1:t}$ which will be used for Bayesian

prediction. We partition y into two subsets, namely $\bar{y}$ and $\tilde{y}$. Let $\bar{F}$ and $\tilde{F}$ be the counterparts of F defined in (5.2) for $\bar{y}$ and $\tilde{y}$, respectively. The following lemma provides the conditions under which any required function of y in Proposition 5.1.3 can be decoupled.

Lemma 5.2.1. *For a given $\theta \in \Theta$, let $C = \mathrm{Corr}(y, y|\theta)$, $\bar{C} = \mathrm{Corr}(\bar{y}, \bar{y}|\theta)$, $\tilde{C} = \mathrm{Corr}(\tilde{y}, \tilde{y}|\theta)$, $k = \mathrm{Corr}(y, z_*|\theta)$, $\bar{k} = \mathrm{Corr}(\bar{y}, z_*|\theta)$, and $\tilde{k} = \mathrm{Corr}(\tilde{y}, z_*|\theta)$. If the following conditions are satisfied*

C1: $\mathrm{Corr}(\tilde{y}, \bar{y}|\theta) = 0$, i.e., *$\tilde{y}$ and $\bar{y}$ are uncorrelated, and*

C2: $\mathrm{Corr}(\bar{y}, z_*|\theta) = 0$, i.e., *$\bar{y}$ and z_* are uncorrelated,*

then we have the following results:

$$
\begin{aligned}
F^T C^{-1} F &= \bar{F}^T \bar{C}^{-1} \bar{F} + \tilde{F}^T \tilde{C}^{-1} \tilde{F} \in \mathbb{R}^{p \times p}, \\
F^T C^{-1} y &= \bar{F}^T \bar{C}^{-1} \bar{y} + \tilde{F}^T \tilde{C}^{-1} \tilde{y} \in \mathbb{R}^p, \\
y^T C^{-1} y &= \bar{y}^T \bar{C}^{-1} \bar{y} + \tilde{y}^T \tilde{C}^{-1} \tilde{y} \in \mathbb{R}, \\
\log\det C &= \log\det \bar{C} + \log\det \tilde{C} \in \mathbb{R}, \\
F^T C^{-1} k &= \tilde{F}^T \tilde{C}^{-1} \tilde{k} \in \mathbb{R}^p, \\
k^T C^{-1} k &= \tilde{k}^T \tilde{C}^{-1} \tilde{k} \in \mathbb{R}.
\end{aligned}
$$

Proof. The results follow by noting the correlation matrix C can be decoupled such that $C = \mathrm{diag}(\bar{C}, \tilde{C})$ and $\bar{k} = 0$. □

Remark 5.2.2. *In order to compute the posterior predictive distribution $\pi(z_*|y)$ (or the predictive mean and variance) in* (5.9)*, $\pi(z_*|\theta, y)$ and $\pi(\theta|y)$ for all $\theta \in \Theta$ need to be calculated. Notice that the posterior distribution of θ can be obtained by computing $w(\theta|y)$ in*

(5.7). *Suppose* $\bar{F}^T\bar{C}^{-1}\bar{F} \in \mathbb{R}^{p\times p}$, $\bar{F}^T\bar{C}^{-1}\bar{y} \in \mathbb{R}^p$, $\bar{y}^T\bar{C}^{-1}\bar{y} \in \mathbb{R}$, *and* $\log\det\bar{C} \in \mathbb{R}$ *are known for all* $\theta \in \Theta$. *If* $\tilde{F}^T\tilde{C}^{-1}\tilde{F} \in \mathbb{R}^{p\times p}$, $\tilde{F}^T\tilde{C}^{-1}\tilde{y} \in \mathbb{R}^p$, $\tilde{y}^T\tilde{C}^{-1}\tilde{y} \in \mathbb{R}$, *and* $\log\det\tilde{C} \in \mathbb{R}$ *for all* $\theta \in \Theta$ *have fixed computation times, then* (5.7) *and* (5.8) *can be computed in constant time due to decoupling results of Lemma 5.2.1.*

The following theorem provides a way to design scalable sequential Bayesian prediction algorithms.

Theorem 5.2.3. *Consider the discrete prior probability* $\pi(\theta)$ *and the compactly supported kernel function* $\phi_t(\cdot)$. *If we select* $\eta \geq \lfloor\overline{\sigma}_t\rfloor \in \mathbb{Z}_{>0}$, $\Delta \in \mathbb{Z}_{>0}$ *and define*

$$\begin{aligned} c_t &:= \max\left(\left\lfloor\frac{t-\Delta}{\Delta+\eta}\right\rfloor, 0\right) \in \mathbb{R}, \\ \xi_j &:= y_{(j-1)(\Delta+\eta)+1:(j-1)(\Delta+\eta)+\Delta} \in \mathbb{R}^{\Delta N}, \\ \bar{y} &:= (\xi_1^T, \cdots, \xi_{c_t}^T)^T \in \mathbb{R}^{\Delta N c_t}, \\ \tilde{y} &:= y_{t-\Delta+1:t} \in \mathbb{R}^{\Delta N}, \end{aligned} \tag{5.11}$$

where $\lfloor\cdot\rfloor$ *is the floor function defined by* $\lfloor x\rfloor := \max\{m \in \mathbb{Z} \,|\, m \leq x\}$, *then the posterior predictive distribution in* (5.9) *can be computed in constant time (*i.e., *does not grow with the time* t*).*

Proof. By construction, conditions C1-2 in Lemma 5.2.1 are satisfied. Hence, it follows from Remark 5.2.2 that the posterior predictive distribution can be computed in constant time. □

Remark 5.2.4. *In Theorem 5.2.3,* $\eta \geq \lfloor\overline{\sigma}_t\rfloor$ *guarantees the time distance between* ξ_j *and* ξ_{j+1} *is large enough such that the conditions in Lemma 5.2.1 are satisfied. Notice that* Δ *is a tuning parameter for users to control the trade-off between the prediction quality and*

the computation efficiency. A large value for Δ yields a small predictive variance but long computation time, and vice versa. An illustrative example with three agents sampling the spatio-temporal Gaussian process in 1-D space is shown in Fig. 5.1.

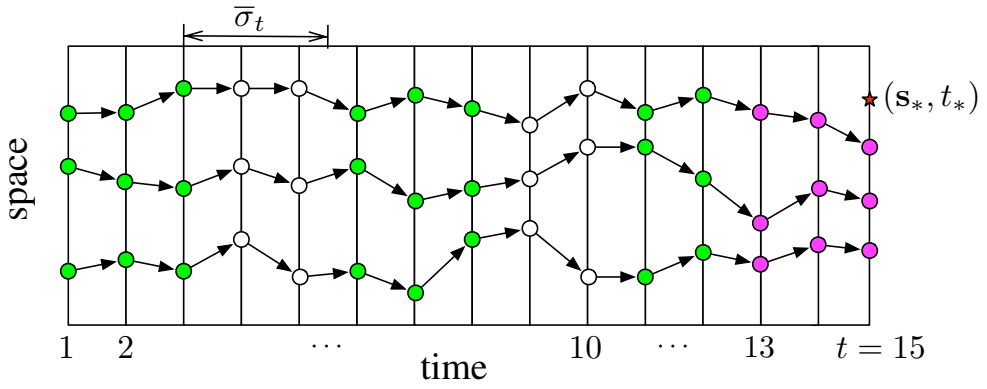


Figure 5.1: Example with three agents sampling the spatio-temporal Gaussian process in 1-D space and performing Bayesian inference. In this example, $\overline{\sigma}_t = 2.5$, $\eta = 2$, $\Delta = 3$, $t = 15$, $c_t = 2$, $\bar{y} = (y_{1:3}^T, y_{6:8}^T)^T$ and $\tilde{y} = y_{13:15}$.

Based on Theorem 5.2.3, we provide the centralized sequential Bayesian prediction algorithm as shown in Table 5.2.

5.2.2 Distributed implementation for a special case

In this subsection, we will show a distributed way (among agent groups) to implement the proposed algorithm for a special case in which β and σ_f^2 are assumed to be known *a priori*. The assumption for this special case is the exact opposite of the one made in [25] where β and σ_f^2 are unknown and θ is known *a priori*.

To develop a distributed scheme among agent groups for data fusion in Bayesian statistics, we exploit the compactly supported kernel for space. Let $C_s(h)$ in (5.1) also be a compactly supported kernel function as $C_t(h)$ so that the correlation vanishes when the spatial distance between two inputs is larger than σ_s, *i.e.*, $C_s(h) = 0, \forall h > 1$.

Consider a case in which M groups of spatially distributed agents sample a spatio-temporal Gaussian process over a large region $\mathcal{Q}$. Each group is in charge of its sub-region of $\mathcal{Q}$. The identity of each group is indexed by $\mathcal{V} := \{1, \cdots, M\}$. Each agent in group i is indexed by $\mathcal{I}^{[i]} := \{1, \cdots, N\}$. The leader of group i is referred to as leader i, which implements the centralized scheme to make prediction on its sub-region using local observations and the globally updated posterior distribution of θ. Therefore, the posterior distribution of θ shall be updated correctly using all observations from all groups (or agents) in a distributed fashion.

Let $\mathcal{G}(t) := (\mathcal{V}, \mathcal{E}(t))$ be an undirected communication graph such that an edge $(i, j) \in \mathcal{E}(t)$ if and only if leader i can communicate with leader j at time t. We define the neighborhood of leader i at time t by $\mathcal{N}_i(t) := \{j \in \mathcal{V} \,|\, (i, j) \in \mathcal{E}(t), j \neq i\}$. Let $a^{[i]}$ denote the quantity as a in the centralized scheme for group i. We then have the following theorem.

Theorem 5.2.5. *Assume that $\bar{y}^{[i]}$ and $\tilde{y}^{[i]}$ for leader i are selected accordingly to Theorem 5.2.3 in time-wise. Let $\tilde{y}$ defined by $\tilde{y} := ((\tilde{y}^{[1]})^T, \cdots, (\tilde{y}^{[M]})^T)^T$. If the following condition is satisfied*

$C3$: $\|q_\ell^{[i]}(t) - q_\nu^{[j]}(t')\| \geq \overline{\sigma}_s, \forall i \neq j, \forall \ell \in \mathcal{I}^{[i]}, \forall \nu \in \mathcal{I}^{[j]}$,

in the spatial domain, then the weights $w(\theta|y)$, based on all observations from all agents, can be obtained from

$$\log w(\theta|y) = \log w(\theta|\bar{y}) + \sum_{i=1}^{M} \log w(\theta|\tilde{y}^{[i]}). \tag{5.12}$$

Proof. The result follows by noting $\mathrm{Corr}(\tilde{y}^{[i]}, \tilde{y}^{[j]}|\theta) = 0, \forall i \neq j$, when the condition $C3$ is satisfied. □

An exemplary configuration of agents which satisfies $C3$ is shown in Fig. 5.2.

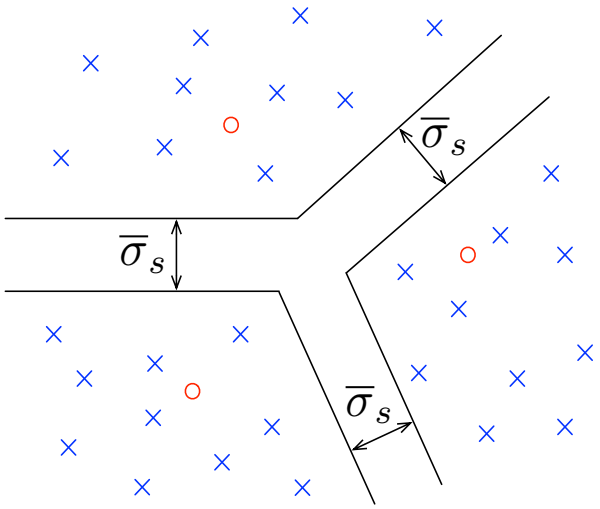


Figure 5.2: Example with three group of agents sampling the spatio-temporal Gaussian process in 2-D space and performing Bayesian prediction. The symbol 'o' denotes the position of a leader for a group and the symbol 'x' denotes the position of an agent. Distance between any two sub-regions is enforced to be greater than $\overline{\sigma}_s$ which enables the distributed Bayesian prediction.

Suppose that the communication graph $\mathcal{G}(t)$ is connected for all time t. Then the average $\frac{1}{M}\sum_{i=1}^{M}\log w(\theta|\tilde{y}^{[i]})$ can be achieved asymptotically via discrete-time average-consensus algorithm [48]:

$$\log w(\theta|\tilde{y}^{[i]}) \leftarrow \log w(\theta|\tilde{y}^{[i]}) + \epsilon \sum_{j\in\mathcal{N}_i}\left(\log w(\theta|\tilde{y}^{[j]}) - \log w(\theta|\tilde{y}^{[i]})\right),$$

with $0 < \epsilon < 1/\Delta(G)$ that depends on the maximum node degree of the network $\Delta(G) = \max_i |\mathcal{N}_i|$.

5.2.3 Adaptive sampling

At time t, the goal of the navigation of agents is to improve the quality of prediction of the field $\mathcal{Q}$ at the next sampling time $t+1$. Therefore, mobile agents should move to the most

informative sampling locations $q(t+1) = (q_1(t+1)^T, \cdots, q_N(t+1)^T)^T$ at time $t+1$ in order to reduce the prediction error [32].

Suppose at time $t+1$, agents move to a new set of positions $\tilde{q} = (\tilde{q}_1^T, \cdots, \tilde{q}_N^T)^T$. The mean squared prediction error is defined as

$$J(\tilde{q}) = \int_{s\in\mathcal{S}} \mathrm{E}\left[(z(s,t+1) - \hat{z}(s,t+1))^2\right] ds, \tag{5.13}$$

where $\hat{z}(s,t+1)$ is obtained as in (5.9). Due to the fact that θ has a distribution, the evaluation of (5.13) becomes computationally prohibitive. To simplify the optimization, we propose to utilize a maximum *a posteriori* (MAP) estimate of θ at time t, denoted by $\hat{\theta}_t$, *i.e.*,

$$\hat{\theta}_t = \arg\max_{\theta\in\Theta} \pi(\theta|y),$$

where y is the subset of all observations used up to time t. The next sampling positions can be obtained by solving the following optimization problem

$$q(t+1) = \arg\min_{\tilde{q}_i\subset\mathcal{Q}} \int_{s\in\mathcal{S}} \mathrm{Var}(z(s,t+1)|y,\hat{\theta}_t) ds. \tag{5.14}$$

This problem can be solved using standard constrained nonlinear optimization techniques (*e.g.*, the conjugate gradient algorithm), possibly taking into account mobility constraints of mobile sensors.

Remark 5.2.6. *The proposed control algorithm in* (5.14) *is truly adaptive in the sense that the new sampling positions are functions of all collected observations. On the other hand, if all parameters are known, the optimization in* (5.14) *can be performed offline without taking*

any measurements.

5.3 Simulation

In this section, we apply the proposed sequential Bayesian prediction algorithms to spatio-temporal Gaussian processes with a correlation function in (5.1). The Gaussian process was numerically generated through circulant embedding of the covariance matrix for the simulation study [20]. This technique allows us to numerically generate a large number of realizations of the Gaussian process.

5.3.1 MCMC-based approach on a 1-D scenario

We consider a scenario in which $N = 5$ agents sample the spatio-temporal Gaussian process in 1-D space and the central station performs Bayesian prediction. The surveillance region $\mathcal{Q}$ is given by $\mathcal{Q} = [0, 10]$. We consider the squared exponential function

$$C_s(h) = \exp(-\frac{1}{2}h^2),$$

for space correlation and a compactly supported correlation function [24] for time as

$$C_t(h) = \begin{cases} \frac{(1-h)\sin(2\pi h)}{2\pi h} + \frac{1-\cos(2\pi h)}{\pi \times 2\pi h}, & 0 \le h \le 1, \\ 0, & \text{otherwise}, \end{cases} \tag{5.15}$$

The signal-to-noise ratio γ is set to be 26dB which corresponds to $\sigma_w = 0.158$. The true values for the parameters used in simulating the Gaussian process are given by $(\beta, \sigma_f^2, \sigma_s, \sigma_t) = (0, 1, 2, 8)$. Notice that the mean function is assumed to be an unknown random variable,

i.e., the dimension of the regression coefficient β is 1. We assume that $\beta|\sigma_f^2$ has the noninformative prior and $\sigma_f^2 \sim IG(3, 20)$. The Gibbs sampler in Table 5.1 was used to generate samples from the posterior distribution of the parameters. A random sampling strategy was used in which agents make observations at random locations at each time $t \in \mathbb{Z}_{>0}$. The prediction was evaluated at each time step for 51 uniform grid points within $\mathcal{Q}$.

The histograms of the samples at time $t = 1$ and $t = 10$ are shown in Fig. 5.3-(a) and Fig. 5.3-(b), respectively. It is clear that the distributions of the parameters are centered around the true values with 100 observations at time $t = 20$. The prediction results at time $t = 1$ and $t = 20$ are shown in Fig. 5.4-(a) and Fig. 5.4-(b), respectively. However, with only 100 observations, the running time using the full Bayesian approach is about several minutes which will soon become intractable.

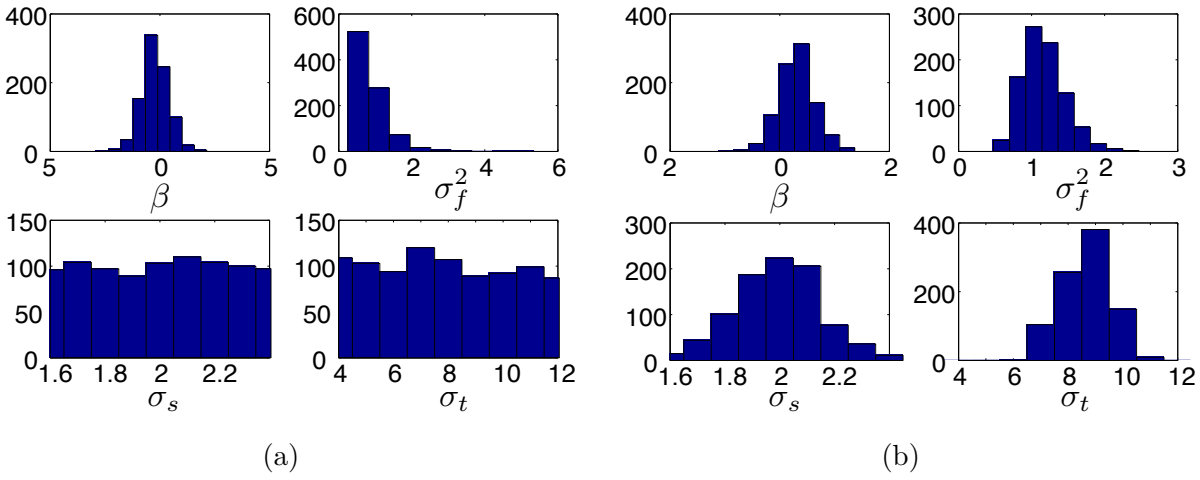
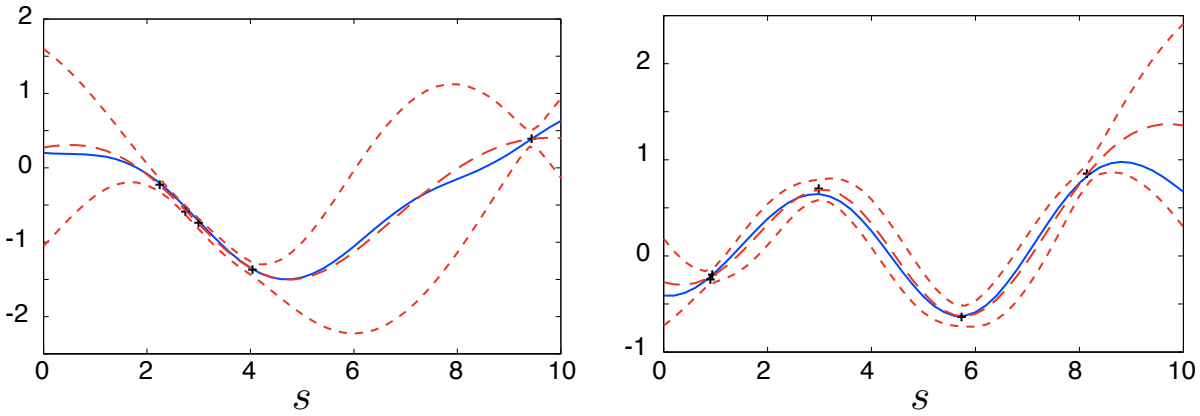


Figure 5.3: Posterior distribution of β, σ_f^2, σ_s, and σ_t at (a) $t = 1$, and (b) $t = 20$.

5.3.2 Centralized scheme on 1-D scenario

We consider the same scenario in which $N = 5$ agents sample the spatio-temporal Gaussian process in 1-D space and the central station performs Bayesian prediction. The true values



(a) (b)

Figure 5.4: Prediction at (a) $t = 1$, and (b) $t = 20$ using the MCMC-based approach. The true fields are plotted in blue solid lines. The predicted fields are plotted in red dash-dotted lines. The area between red dotted lines indicates the 95% confidence interval.

for the parameters used in simulating the Gaussian process are given by $(\beta, \sigma_f^2, \sigma_s, \sigma_t) = (20, 10, 2, 8)$. Notice that the mean function is assumed to be an unknown random variable, *i.e.*, the dimension of the regression coefficient β is 1. We assume that $\beta|\sigma_f^2$ has the non-informative prior and $\sigma_f^2 \sim IG(3, 20)$. We also assume the bounds of θ, viz. $\sigma_s \in [1.6, 2.4]$ and $\sigma_t \in [4, 12]$ are known. $\Delta = 12$ is used and $\eta = 11$ is selected satisfying the condition in Theorem 5.2.3. We use a discrete uniform probability distribution for $\pi(\theta)$ as shown in Fig. 5.6-(a). The adaptive sampling strategy was used in which agents make observations at each time $t \in \mathbb{Z}_{>0}$. The prediction was evaluated at each time step for 51 uniform grid points within $\mathcal{Q}$.

Fig. 5.5 shows the comparison between predictions at time $t = 1$ using (a) the maximum likelihood (ML) based approach, and (b) the proposed fully Bayesian approach. The ML based approach first generates a point estimate of the hyperparameters and then uses them as true ones for computing the prediction and the prediction error variance. In this simulation, a poor point estimate on θ was achieved by maximizing the likelihood function. As a result,

the prediction and the associated prediction error variance are incorrect and are far from being accurate for a small number of observations. On the other hand, the fully Bayesian approach which incorporates the prior knowledge of θ and uncertainties in θ provides a more accurate prediction and an exact confidence interval.

Using the proposed sequential Bayesian prediction algorithm along with the adaptive sampling strategy, the prior distribution was updated in a sequential manner. At time $t = 100$, the posterior distribution of θ is shown in Fig. 5.6-(b). With a larger number of observations, the support for the posterior distribution of θ becomes smaller and the peak gets closer to the true value. As shown in Fig. 5.7-(a), the quality of the prediction at time $t = 100$ is significantly improved. At time $t = 300$, the prior distribution was further updated which is shown in Fig. 5.6-(c). At this time, $\theta = (2, 8)^T$, which is the true value, has the highest probability. The prediction is also shown in Fig. 5.7-(b). This demonstrates the usefulness and correctness of our algorithm. The running time at each time step is fixed, which is around 12s using Matlab, R2008a (MathWorks) in a PC (2.4GHz Dual-Core Processor).

5.3.3 Distributed scheme on 2-D scenario

Finally, we consider a scenario in which there are 4 groups, each of which contain 10 agents sampling the spatio-temporal Gaussian process in 2-D space. The surveillance region $\mathcal{Q}$ is given by $\mathcal{Q} = [0, 10] \times [0, 10]$. The parameter values used in simulating the Gaussian process are given by $\theta = (\sigma_s, \sigma_t)^T = (2, 8)^T$, $\beta = 0$, and $\sigma_f^2 = 1$, last two values of which are assumed to be known a priori. To use the distributed scheme, we only consider compactly supported kernel functions for both space and time. In particular, we consider $C_s(h) = C_t(h)$ as in

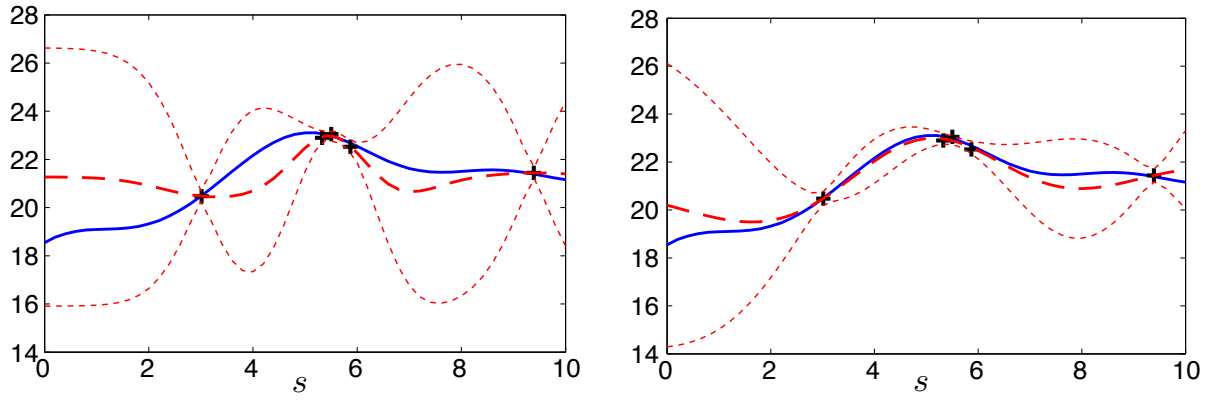
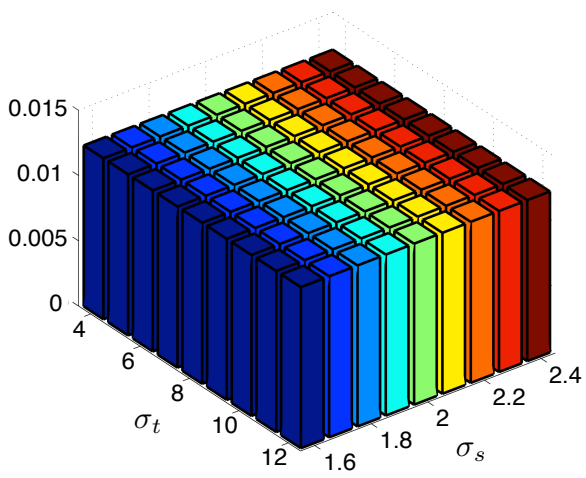


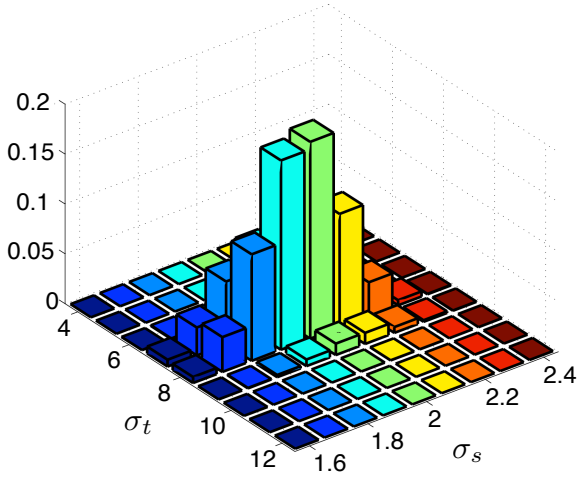
Figure 5.5: Prediction at $t = 1$ using (a) the maximum likelihood based approach, and (b) the proposed fully Bayesian approach. The true fields are plotted in blue solid lines. The predicted fields are plotted in red dash-dotted lines. The area between red dotted lines indicates the 95% confidence interval.

(5.15). We also assume the fact that $\sigma_s \in [1.6, 2.4]$ and $\sigma_t \in [4, 12]$ are known a priori. $\Delta = 12$ is used and $\eta = 11$ is selected satisfying the condition in Theorem 5.2.3. The region $\mathcal{Q}$ is divided into 4 square sub-regions with equal size areas as shown in Fig. 5.9-(a). Distance between any two sub-regions is enforced to be greater than $\bar{\sigma}_s = 2.4$, satisfying the condition in Theorem 5.2.5, which enables the distributed Bayesian prediction. The same uniform prior distribution for θ as in the centralized version (see Fig. 5.6-(a)) is used.

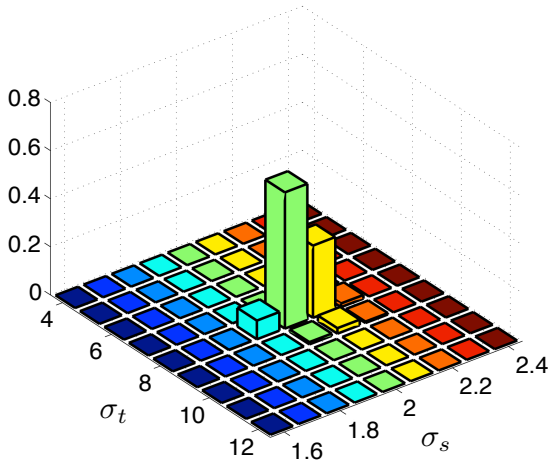
The globally updated posterior distribution of θ at time t_{100} is shown in Fig. 5.8. It has a peak near the true θ, which shows the correctness of the distributed algorithm. The predicted field compared with the true field at time t_{100} is shown in Fig. 5.9. Due to the construction of sub-regions, the interface areas between any of two sub-regions are not predicted. Notice that the prediction is not as good as in the 1-D scenario due to the effect of curse of dimensionality when we move from 1-D to 2-D spaces. The prediction quality can be improved by using more number of sensors at the cost of computational time. The running time of the distributed algorithm in this scenario is about several minutes due to



(a)



(b)



(c)

Figure 5.6: (a) Prior distribution θ, (b) posterior distribution of θ at time $t = 100$, (c) posterior distribution of θ at time $t = 300$.

the complexity of the 2-D problem under the same computational environment as the one used for the 1-D scenario. However, thanks to our proposed sequential sampling schemes, the running time does not grow with the number of measurements.

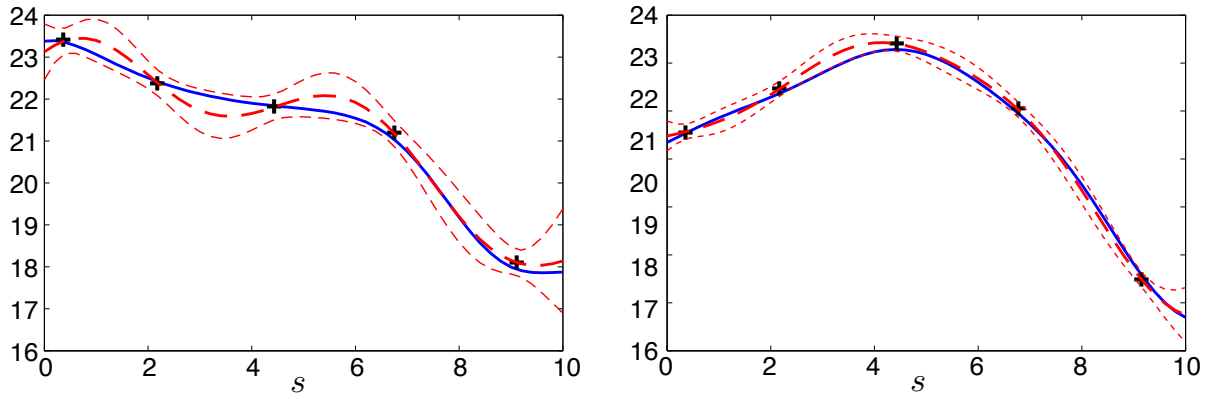


Figure 5.7: Prediction at (a) $t = 100$, and (b) $t = 300$ using the centralized sequential Bayesian approach. The true fields are plotted in blue solid lines. The predicted fields are plotted in red dash-dotted lines. The area between red dotted lines indicates the 95% confidence interval.

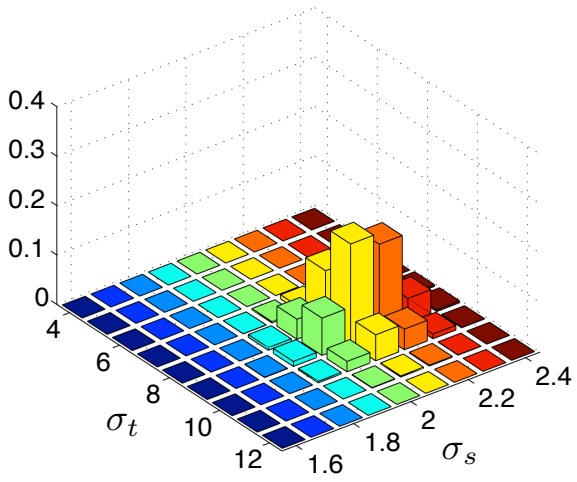


Figure 5.8: Posterior distribution of θ at time $t = 100$ using the distributed algorithm.

Table 5.2: Centralized Bayesian prediction algorithm.

Input:

(1) prior distribution on σ_f^2, *i.e.*, $\pi(\sigma_f^2) = IG(a, b)$

(2) prior distribution on $\theta \in \Theta$, *i.e.*, $\pi(\theta)$

(3) tuning variables Δ and η

(4) number of agents N

(5) $\mathcal{M}(\theta).A = 0 \in \mathbb{R}^{p\times p}$, $\mathcal{M}(\theta).B = 0 \in \mathbb{R}$, $\mathcal{M}(\theta).C = 0 \in \mathbb{R}^p$, $\mathcal{M}(\theta).D = 0 \in \mathbb{R}$, $\mathcal{M}_0(\theta) = \mathcal{M}(\theta)$, $\forall \theta \in \Theta$

Output:

(1) The predictive mean at location $s_* \in \mathcal{S}$ and time $t_* = t$, *i.e.*, $\mu_{z_*|y}$

(2) The predictive variance at location $s_* \in \mathcal{S}$ and time $t_* = t$, *i.e.*, $\sigma^2_{z_*|y}$

At time t, the central station does:

1: receive observations y_t from agents, set $\tilde{y} = y_{t-\Delta+1:t}$ and $n = N\Delta$

2: compute $\tilde{F} = (f(\tilde{x}^{(1)}), \cdots, f(\tilde{x}^{(n)}))^T$ where $\tilde{x}^{(i)}$ is the input of the i-th element in $\tilde{y}$

3: **for** each $\theta \in \Theta$ **do**

4: compute $\tilde{C} = \mathrm{Corr}(\tilde{y}, \tilde{y}) \in \mathbb{R}^{n\times n}$

5: compute the key values

$$F^T C^{-1} F = \mathcal{M}(\theta).A + \tilde{F}^T \tilde{C}^{-1} \tilde{F} \in \mathbb{R}^{p\times p},\ y^T C^{-1} y = \mathcal{M}(\theta).B + \tilde{y}^T \tilde{C}^{-1} \tilde{y} \in \mathbb{R},$$
$$F^T C^{-1} y = \mathcal{M}(\theta).C + \tilde{F}^T \tilde{C}^{-1} \tilde{y} \in \mathbb{R}^p,\ \log\det C = \mathcal{M}(\theta).D + \log\det \tilde{C} \in \mathbb{R}$$

6: compute $\tilde{a} = a + \frac{n}{2}$ and

$$\tilde{b} = b + \tfrac{1}{2} y^T C^{-1} y - \tfrac{1}{2} (F^T C^{-1} y)^T (F^T C^{-1} F)^{-1} (F^T C^{-1} y)$$

7: update weights via

$$\log w(\theta|y) = -\tfrac{1}{2} \log\det C - \tfrac{1}{2} \log\det(F^T C^{-1} F) - \tilde{a} \log \tilde{b}$$

8: **for** each $s_* \in \mathcal{S}$ **do**

9: compute $f(x_*) \in \mathbb{R}^p$, $\tilde{k} = \mathrm{Corr}(\tilde{y}, z_*) \in \mathbb{R}^n$

10: compute predictive mean and variance for given θ

$$\mu_{z_*|\theta,y} = \tilde{k} \tilde{C}^{-1} \tilde{y} + (f(x_*) - \tilde{F}^T \tilde{C}^{-1} \tilde{k})^T (F^T C^{-1} F)^{-1} (F^T C^{-1} y),$$
$$\sigma^2_{z_*|\theta,y} =$$
$$\tfrac{\tilde{b}}{\tilde{a}-1} \left((1 - \tilde{k}^T \tilde{C}^{-1} \tilde{k}) + (f(x_*) - \tilde{F}^T \tilde{C}^{-1} \tilde{k})^T (F^T C^{-1} F)^{-1} (f(x_*) - \tilde{F}^T \tilde{C}^{-1} \tilde{k}) \right)$$

11: **end for**

12: **if** $\mathrm{mod}(t, \Delta + \eta) = \Delta$ **then**

13: set $\mathcal{M}(\theta) = \mathcal{M}_0(\theta)$, then $\mathcal{M}_0(\theta).A = F^T C^{-1} F$, $\mathcal{M}_0(\theta).B = y^T C^{-1} y$, $\mathcal{M}_0(\theta).C = F^T C^{-1} y$, and $\mathcal{M}_0(\theta).D = \log\det C$

14: **end if**

15: **end for**

16: compute the posterior distribution

$$\pi(\theta|y) = \tfrac{w(\theta|y)\pi(\theta)}{\sum_\theta w(\theta|y)\pi(\theta)}$$

17: compute the predictive mean and variance

$$\mu_{z_*|y} = \textstyle\sum_\theta \mu_{z_*|\theta,y} \pi(\theta|y),$$
$$\sigma^2_{z_*|y} = \textstyle\sum_\theta \sigma^2_{z_*|\theta,y} \pi(\theta|y) + \sum_\theta \left(\mu_{z_*|\theta,y} - \mu_{z_*|y} \right)^2 \pi(\theta|y).$$

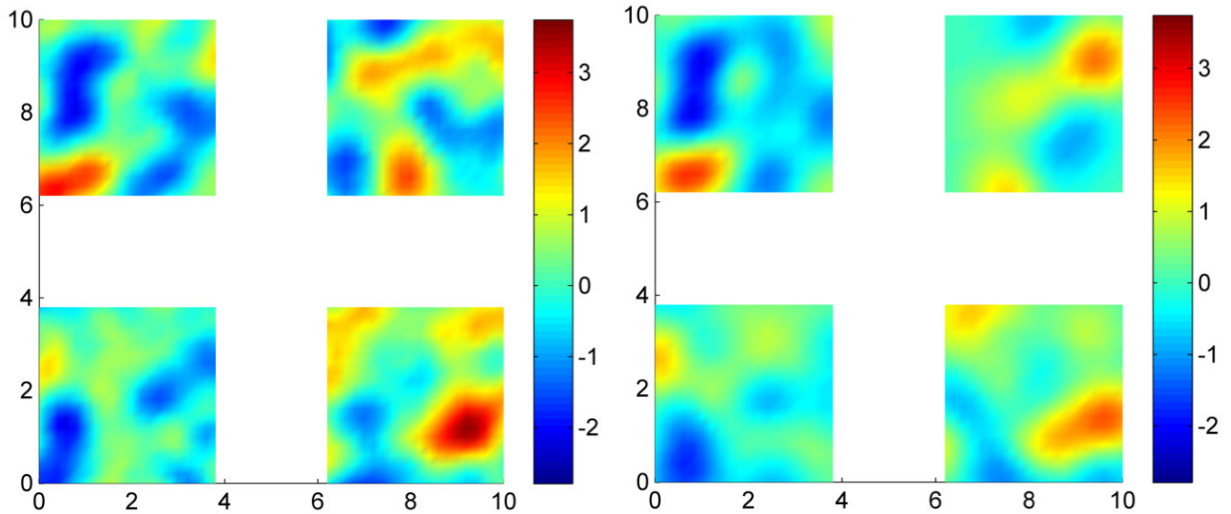


Figure 5.9: Comparison of (a) the true field at $t = 100$ and (b) the predicted field at $t = 100$ using the distributed algorithm.

Chapter 6

Gaussian Process with Built-in GMRF

Recently, there have been efforts to find a way to fit a computationally efficient Gaussian Markov random field (GMRF) on a discrete lattice to a Gaussian random field on a continuum space [17,27,56]. Such methods have been developed using a fitting with a weighted L_2-type distance [56], using a conditional-mean least-squares fitting [17], and for dealing with large data by fast Kriging [27]. It has been demonstrated that GMRFs with small neighborhoods can approximate Gaussian fields surprisingly well [56]. This approximated GMRF and its regression are very attractive for the resource-constrained mobile sensor networks due to its computational efficiency and scalability [34] as compared to the standard Gaussian process and its regression, which is not scalable as the number of observations increases.

Mobile sensing agents form an ad-hoc wireless communication network in which each agent usually operates under a short communication range, with limited memory and computational power. For resource-constrained mobile sensor networks, developing distributed

prediction algorithms for robotic sensors using only local information from local neighboring agents has been one of the most fundamental problems [4, 6, 10, 13, 25, 47].

In Section 6.1.1, a new class of Gaussian processes is proposed for resource-constrained mobile sensor networks. Such a Gaussian process builds on a GMRF [54] with respect to a proximity graph, *e.g.*, the Delaunay graph of a set of vertices over a surveillance region. The formulas for predictive statistics are derived in Section 6.1.2. We propose a sequential prediction algorithm which is scalable to deal with sequentially sampled observations in Section 6.1.3. In Section 6.2, we develop a distributed and scalable statistical inference algorithm for a simple sampling scheme by applying the Jacobi over-relaxation and discrete-time average consensus algorithms. Simulation and experimental study demonstrate the usefulness of the proposed model and algorithms in Section 6.3.

6.1 Spatial Prediction

In this section, we first propose a new class of Gaussian random fields with built-in Gaussian Markov random fields (GMRF) [54]. Then we show how to compute the prediction at any point of interest based on Gaussian process regression, and provide a sequential field prediction algorithm for mobile sensor networks.

6.1.1 Spatial model based on GMRF

Let $\gamma := (\gamma(p_1), \cdots, \gamma(p_m))^T \sim \mathcal{N}(0, Q^{-1})$ be a zero-mean GMRF [54] with respect to an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the location of vertex i is denoted by p_i in the surveillance region $\mathcal{Q}$. Such locations of vertices will be referred to as *generating points*. The inverse covariance matrix (precision matrix) $Q \succ 0$ has the property $(Q)_{ij} \neq 0 \Leftrightarrow \{i, j\} \in \mathcal{E}$.

If the graph $\mathcal{G}$ has small cardinalities of the neighbor sets, its precision matrix Q becomes sparse with many zeros in its entries. This plays a key role in computation efficiency of a GMRF which can be greatly exploited by the resource-constrained mobile sensor network.

The spatial field is modeled by a Gaussian process with a built-in GMRF defined as

$$z(s) = \mu(s) + \sum_{j=1}^{m} \lambda(s, p_j)\gamma(p_j), \tag{6.1}$$

where $\lambda(\cdot, \cdot)$ is a weighting function. The new class of Gaussian processes is capable of representing a wide range of non-stationary Gaussian fields, by selecting

1. different number of generating points m,

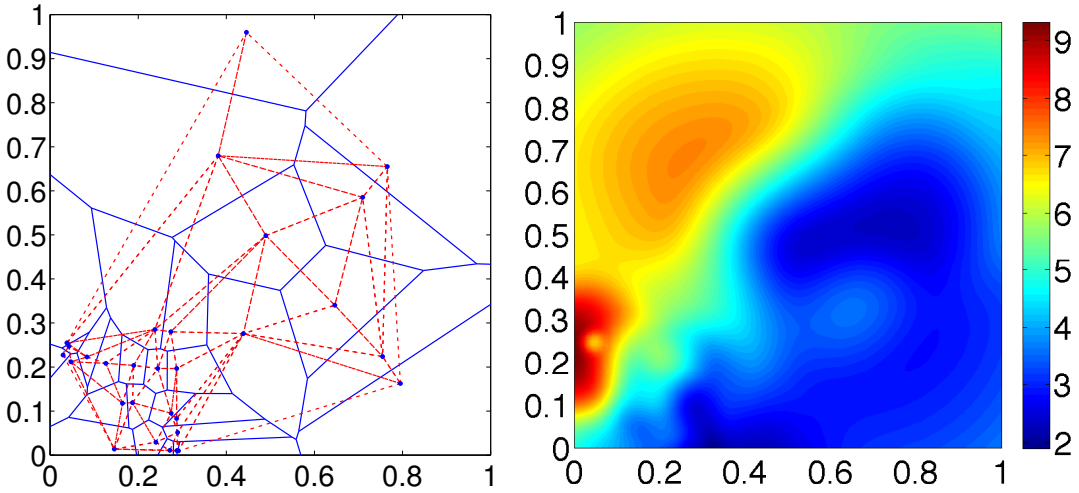
2. different locations of generating points $\{p_j \,|\, j = 1, \cdots, m\}$ over $\mathcal{Q}$,

3. a different structure of the precision matrix Q, and

4. different weighting functions $\{\lambda(\cdot, p_j) \,|\, j = 1, \cdots, m\}$.

Remark 6.1.1. *The number of generating points could be determined by a model selection criterion such as the Akaike information criterion [1]. Similar to hyperparameter estimation in the standard Gaussian process regression, one can estimate all other parameters using maximum likelihood (ML) optimization [51, 65]. This is non-convex optimization and so the initial conditions need to be chosen carefully to avoid local minima. In our approach, we use basic structures for weighting functions and the precision matrix, however, we make them as functions of the locations of generating points. Different spatial resolutions can be obtained by a suitable choice of locations of generating points. As an example shown in Fig. 6.1, higher resolution can be obtained by higher density of generating points (see lower left corner). In*

this way, we only need to determine the locations of generating points. This approach will be demonstrated with real-world data in Section 6.3.4.



(a) (b)

Figure 6.1: (a) Generating points in blue dots and the associated Delaunay graph with edges in red dotted lines. The Voronoi partition is also shown in blue solid lines. (b) Gaussian random field with a built-in GMRF with respect to the Delaunay graph in (a).

6.1.2 Gaussian process regression

Suppose we have a collection of observations $y := (y_1, \cdots, y_n)^T$ whose entries are sampled at the corresponding points $s_1, \cdots, s_n$. The noise corrupted measurement $y_i \in \mathbb{R}$ is given by

$$y_i = z(s_i) + \epsilon_i,$$

where $\epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_w^2)$ is an independent and identically distributed (i.i.d.) Gaussian white noise. We then have the following results.

Proposition 6.1.2. *Let $\Lambda \in \mathbb{R}^{n \times m}$ be a matrix obtained by $(\Lambda)_{ij} = \lambda(s_i, p_j)$ and let $\lambda \in \mathbb{R}^m$ be a vector obtained by $(\lambda)_i = \lambda(s_0, p_i)$, where s_0 is a point of interest. Then the covariance*

matrix of y and the covariance between y and $z(s_0)$ are given by

$$C := \mathrm{E}[(y - \mathrm{E}\, y)(y - \mathrm{E}\, y)^T] = \Lambda Q^{-1} \Lambda^T + \sigma_w^2 I,$$
$$k := \mathrm{E}[(y - \mathrm{E}\, y) z(s_0)] = \Lambda Q^{-1} \lambda,$$

where $Q \in \mathbb{R}^{m \times m}$ is the precision matrix of the GMRF $\gamma \in \mathbb{R}^m$.

Proof. The (i, j)-th element of the covariance matrix C, *i.e.*, the covariance between y_i and y_j, can be obtained by

$$\begin{aligned}
C_{ij} &= \mathrm{Cov}(z(s_i), z(s_j)) + \sigma_w^2 \delta_{ij} \\
&= \mathrm{E}(z(s_i) - \mu(s_i))(z(s_j) - \mu(s_j)) + \sigma_w^2 \delta_{ij} \\
&= \mathrm{E}\left(\sum_k \lambda(s_i, p_k) \gamma(p_k) \right) \left(\sum_l \lambda(s_j, p_l) \gamma(p_l) \right) + \sigma_w^2 \delta_{ij} \\
&= \mathrm{E}\left(\sum_{k,l} \lambda(s_i, p_k) \gamma(p_k) \gamma(p_l) \lambda(s_j, p_l) \right) + \sigma_w^2 \delta_{ij} \\
&= \sum_{k,l} \lambda(s_i, p_k)\, \mathrm{E}(\gamma(p_k)\gamma(p_l)) \lambda(s_j, p_l) + \sigma_w^2 \delta_{ij} \\
&= \sum_{k,l} \lambda(s_i, p_k) (Q^{-1})_{kl} \lambda(s_j, p_l) + \sigma_w^2 \delta_{ij}.
\end{aligned}$$

The i-th element of the covariance vector k, *i.e.*, the covariance between y_i and $z(s_0)$,

can be obtained by

$$
\begin{aligned}
k_i &= \mathrm{Cov}(z(s_i), z(s_0)) \\
&= \mathrm{E}(z(s_i) - \mu(s_i))(z(s_0) - \mu(s_0)) \\
&= \mathrm{E}\left(\sum_k \lambda(s_i, p_k)\gamma(p_k)\right)\left(\sum_l \lambda(s_0, p_l)\gamma(p_j)\right) \\
&= \mathrm{E}\left(\sum_{k,l} \lambda(s_i, p_k)\gamma(p_k)\gamma(p_l)\lambda(s_0, p_l)\right) \\
&= \sum_{k,l} \lambda(s_i, p_k)\,\mathrm{E}(\gamma(p_k)\gamma(p_l))\lambda(s_0, p_l) \\
&= \sum_{k,l} \lambda(s_i, p_k)(Q^{-1})_{kl}\lambda(s_0, p_l),
\end{aligned}
$$

whose matrix form completes the proof. □

By Propositions 6.1.2, we can make prediction at the point of interest s_0 using Gaussian process regression [51]. This is summarized by the following theorem.

Theorem 6.1.3. *For given y, the prediction of $z_0 := z(s_0)$ at any location $s_0 \in \mathcal{Q}$ is given by the conditional distribution*

$$
z_0|y \sim \mathcal{N}\left(\mu_{z_0|y}, \sigma^2_{z_0|y}\right),
$$

where the predictive mean and variance are obtained by

$$
\begin{aligned}
\mu_{z_0|y} &= \mu(s_0) + \lambda^T \hat{Q}^{-1}\hat{y}, \\
\sigma^2_{z_0|y} &= \lambda^T \hat{Q}^{-1}\lambda,
\end{aligned}
\tag{6.2}
$$

with

$$\hat{Q} = Q + \sigma_w^{-2}\Lambda^T\Lambda \in \mathbb{R}^{m\times m},$$

$$\hat{y} = \sigma_w^{-2}\Lambda^T(y-\mu) \in \mathbb{R}^m.$$

Proof. By using the Woodbury matrix identity (see Appendix A.2.1), the prediction mean can be obtained by

$$\begin{aligned}
\mu_{z_0|y} &= \mu(s_0) + k^T C^{-1}(y-\mu) \\
&= \mu(s_0) + (\Lambda Q^{-1}\lambda)^T(\Lambda Q^{-1}\Lambda^T + \sigma_w^2 I)^{-1}(y-\mu) \\
&= \mu(s_0) + \lambda^T Q^{-1}\Lambda^T(\Lambda Q^{-1}\Lambda^T + \sigma_w^2 I)^{-1}(y-\mu) \\
&= \mu(s_0) + \lambda^T Q^{-1}\Lambda^T(\sigma_w^{-2}I - \\
&\quad \sigma_w^{-2}\Lambda(Q + \sigma_w^{-2}\Lambda^T\Lambda)^{-1}\Lambda^T\sigma_w^{-2})(y-\mu) \\
&= \mu(s_0) + \lambda^T(\sigma_w^{-2}Q^{-1} - \\
&\quad \sigma_w^{-4}Q^{-1}\Lambda^T\Lambda(Q + \sigma_w^{-2}\Lambda^T\Lambda)^{-1})\Lambda^T(y-\mu) \\
&= \mu(s_0) + \lambda^T\Xi\Lambda^T(y-\mu),
\end{aligned}$$

where

$$\begin{aligned}
\Xi &= \sigma_w^{-2}Q^{-1} - \sigma_w^{-4}Q^{-1}\Lambda^T\Lambda(Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1} \\
&= \sigma_w^{-2}Q^{-1}(Q+\sigma_w^{-2}\Lambda^T\Lambda)(Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1} \\
&\quad - \sigma_w^{-4}Q^{-1}\Lambda^T\Lambda(Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1} \\
&= (\sigma_w^{-2}I + \sigma_w^{-4}Q^{-1}\Lambda^T\Lambda)(Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1} \\
&\quad - \sigma_w^{-4}Q^{-1}\Lambda^T\Lambda(Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1} \\
&= \sigma_w^{-2}(Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1}.
\end{aligned}$$

Similarly, the prediction error variance can be obtained by

$$\begin{aligned}
\sigma^2_{z_0|y} &= \lambda^T Q^{-1}\lambda - k^T C^{-1} k \\
&= \lambda^T Q^{-1}\lambda - (\Lambda Q^{-1}\lambda)^T(\Lambda Q^{-1}\Lambda^T + \sigma_w^2 I)^{-1}(\Lambda Q^{-1}\lambda) \\
&= \lambda^T \left(Q^{-1} - Q^{-1}\Lambda^T(\Lambda Q^{-1}\Lambda^T + \sigma_w^2 I)^{-1}\Lambda Q^{-1}\right)\lambda \\
&= \lambda^T (Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1}\lambda,
\end{aligned}$$

where $\mathrm{Cov}(z(s_0), z(s_0)) = \lambda^T Q^{-1}\lambda$ is obtained similarly as in Proposition 6.1.2. □

Remark 6.1.4. *When the generating points $\{p_1, p_2, \cdots, p_m\}$ are not known a priori, they can be estimated by maximizing the likelihood function. Given n observations $y = (y_1, y_2, \cdots, y_n)^T$ sampled at $\{s_1, s_2, \cdots, s_n\}$, the log likelihood of y is given by*

$$\log \pi(y) = -\frac{1}{2}(y-\mu)^T C^{-1}(y-\mu) - \frac{1}{2}\log\det C - \frac{n}{2}\log 2\pi,$$

where $C = \Lambda Q^{-1}\Lambda^T + \sigma_w^2 I$ is the covariance matrix of y. the maximum likelihood estimate

of the generating points can be obtained via solving the following optimization problem.

$$\hat{p}_{ML} = \arg\max_{p} \log \pi(y). \tag{6.3}$$

Remark 6.1.5. *Note that the number of generating points m is fixed and the number of observations n may grow in time, and so in general we consider $m \ll n$. Theorem 6.1.3 shows that only the inversion of an $m \times m$ matrix $\hat{Q} = Q + \sigma_w^{-2}\Lambda^T\Lambda$ is required in order to compute the predictive distribution of the field at any point. The computational complexity grows linearly with the number of observations,* i.e., *$O(nm^2)$, compare to the standard Gaussian process regression which requires $O(n^3)$. Moreover, it enables a scalable prediction algorithm for sequential measurements.*

In what follows, we present a sequential field prediction algorithm for sequential observations by exploiting the results of Theorem 6.1.3.

6.1.3 Sequential prediction algorithm

Consider a sensor network consisting of N mobile sensing agents distributed in the surveillance region $\mathcal{Q}$. The index of the robotic sensors is denoted by $\mathcal{I} := \{1, \cdots, N\}$. The sensing agents sample the environmental field at time $t \in \mathbb{Z}_{>0}$ and send the observations to a central station which is in charge of the data fusion.

At time t, agent i makes an observation $y_i(t)$ at location $s_i(t)$. Denote the collection of observations at time t by $y_t := (y_1(t), \cdots, y_N(t))^T$. We have the following proposition.

Proposition 6.1.6. *At time $t \in \mathbb{Z}_{>0}$, the predictive mean and variance at any point of*

interest can be obtained via (6.2) *with*

$$\hat{Q}_t = \hat{Q}_{t-1} + \sigma_w^{-2}\Lambda_t^T\Lambda_t, \quad \hat{Q}_0 = Q$$

$$\hat{y}_t = \hat{y}_{t-1} + \sigma_w^{-2}\Lambda_t^T(y_t - \mu_t), \quad \hat{y}_0 = 0,$$

where $(\Lambda_t)_{ij} = \lambda(s_i(t), s_j(t))$, *and* $(\mu_t)_i = \mu(s_i(t))$.

Proof. The result can be obtained easily by noting that $A^TA = A_1^TA_1 + A_2^TA_2$, where $A = (A_1^T, A_2^T)^T$. □

Based on Proposition 6.1.6, we present a sequential field prediction algorithm using mobile sensor networks in Table 6.1.

Table 6.1: Sequential algorithm for field prediction.

Input: a set of target points $\mathcal{S}$ **Output:** (1) prediction mean $\{\hat{z}(s_0) \,\vert\, s_0 \in \mathcal{S}\}$ (2) prediction error variance $\left\{\sigma^2(s_0) \,\vert\, s_0 \in \mathcal{S}\right\}$ **Assumption:** (1) the central station knows p, Q, and $\lambda(\cdot,\cdot)$ (2) the central station initially has $\hat{Q} \leftarrow Q$, $\hat{y} \leftarrow 0$
At time t, agent $i \in \mathcal{I}$ in the network does: 1: take measurement y_i from its current location s_i 2: send the measurement (s_i, y_i) to the central station At time t, the central station does: 1: obtain measurements $\{(s_\ell, y_\ell) \,\vert\, \forall \ell \in \mathcal{I}\}$ from mobile sensors 2: compute Λ via $(\Lambda)_{ij} = \lambda(s_i, p_j)$ 3: update $\hat{Q} \leftarrow \hat{Q} + \sigma_w^{-2}\Lambda^T\Lambda$ 4: update $\hat{y} \leftarrow \hat{y} + \sigma_w^{-2}\Lambda^T(y - \mu)$, where $\mu_i = \mu(s_i)$ 5: **for** $s_0 \in \mathcal{S}$ **do** 6: compute $(\lambda)_i$ via $\lambda(s_0, p_i)$ 7: compute $\hat{z}(s_0) = \mu(s_0) + \lambda^T\hat{Q}^{-1}\hat{y}$ 8: compute $\sigma^2(s_0) = \lambda^T\hat{Q}^{-1}\lambda$ 9: **end for**

6.2 Distributed Spatial Prediction

In this section, we propose a distributed approach, in which robotic sensors exchange only local information between neighbors, to implement the field prediction effectively fusing all observations collected by all sensors correctly. This distributed approach can be implemented for a class of weighting functions $\lambda(\cdot,\cdot)$ in (6.1) that have compact supports. In particular, we consider the weighting function defined by

$$\lambda(s, p_j) = \lambda(\left\|s - p_j\right\| / r), \tag{6.4}$$

where

$$\lambda(h) := \begin{cases} (1-h)\cos(\pi h) + \frac{1}{\pi}\sin(\pi h), & h \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Notice that the weighting function $\lambda(\cdot,\cdot)$ in (6.4) has a compact support, *i.e.*, $\lambda(s, p_j)$ is non-zero if and only if the distance $\left\|s - p_j\right\|$ is less than the support $r \in \mathbb{R}_{>0}$.

6.2.1 Distributed computation

We first briefly introduce distributed algorithms for solving linear systems and computing the averages. They will be used as major tools for distributed implementation of field prediction.

- **Jacobi over-relaxation method:** The Jacobi over-relaxation (JOR) [4] method provides an iterative solution of a linear system $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ is a nonsingular matrix and $x, b \in \mathbb{R}^n$. If agent i knows the $\text{row}_i(A) \in \mathbb{R}^n$ and b_i, and $a_{ij} = (A)_{ij} = 0$

if agent i and agent j are not neighbors, then the recursion is given by

$$x_i^{(k+1)} = (1-h)x_i^{(k)} + \frac{h}{a_{ii}}\left(b_i - \sum_{j\in\mathcal{N}_i} a_{ij}x_j^{(k)}\right). \tag{6.5}$$

This JOR algorithm converges to the solution of $Ax = b$ from any initial condition if $h < 2/n$ [13]. At the end of the algorithm, agent i knows the i-th element of $x = A^{-1}b$.

- **Discrete-time average consensus:** The Discrete-time average consensus (DAC) provides a way to compute the arithmetic mean of elements in the a vector $c \in \mathbb{R}^n$. Assume the graph is connected. If agent i knows the i-th element of c, the network can compute the arithmetic mean via the following recursion [47]

$$x_i^{(k+1)} = x_i^{(k)} + \epsilon\sum_{j\in\mathcal{N}_i} a_{ij}(x_j^{(k)} - x_i^{(k)}), \tag{6.6}$$

with initial condition $x(0) = c$, where $a_{ij} = 1$ if $j \in \mathcal{N}_i$ and 0 otherwise, $0 < \epsilon < 1/\Delta$, and $\Delta = \max_i(\sum_{j\neq i} a_{ij})$ is the maximum degree of the network. After the algorithm converges, all node in the network know the average of c, *i.e.*, $\sum_{i=1}^n c_i/n$.

6.2.2 Distributed prediction algorithm

Consider a GMRF with respect to a proximity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that generates a Gaussian random field in (6.1). The index of the generating points is denoted by $\mathcal{V} := \{1, \cdots, n\}$. The location of the i-th generating point is p_i. The edges of the graph are considered to be $\mathcal{E} := \big\{\{i,j\} \mid \|p_i - p_j\| \leq R\big\}$, where R is a constant that ensures the graph is connected.

Consider a mobile sensor network consisting of N mobile sensing agents distributed in the surveillance region $\mathcal{Q}$. For simplicity, we assume that the number of agents is equal to

the number of generating points, *i.e.*, $N = m$. The index of the robotic sensors is denoted by $\mathcal{I} := \{1, \cdots, m\}$. The location of agent i is denoted by s_i.

The assumptions made for the resource-constrained mobile sensor networks are listed as follows.

A.1 Agent i is in charge of sampling at point s_i within a r-disk centered at p_i, *i.e.*, $\|s_i - p_i\| < r$.

A.2 r is the radius of the support of the weighting function in (6.4) and also satisfies that $0 < r < \frac{R}{2}$.

A.3 Agent i can only locally communicate with neighbors in $\mathcal{N}_i := \{j \in \mathcal{I} \mid \{i, j\} \in \mathcal{E}\}$ defined by the connected proximity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

A.4 Agent i knows $\text{row}_i(Q)$, *i.e.*, the i-th row of Q, where $(Q)_{ij} \neq 0$ if and only if $j \in \{i\} \cup \mathcal{N}_i$.

Remark 6.2.1. *As in A.1, it is reasonable to have at least one agent collect measurements that are correlated with a random variable from a single generating point. This sampling rule may be modified such that a single agent dynamically samples for multiple generating points or more number of agents samples for a generating point depending on available resources. Since there is at least one agent in charge of a generating point by A.1, it is natural to have A.3 and A.4 taking advantage of the proximity graph for the GMRF. Notice that each agent only knows local information of Q as described in A.4.*

An illustration of agent ℓ sampling a measurement at point s_ℓ in the intersection of the supports of the weighting functions of p_i and p_j is shown in Fig. 6.2.

From A.1 and A.2, since $R > 2r$, we have $\lambda(s_\ell, p_i) = 0$ if $\ell \notin \mathcal{N}_i$. Thus the matrix $\hat{Q} = Q + \sigma_w^{-2} \Lambda^T \Lambda \in \mathbb{R}^{m \times m}$ and the vector $\hat{y} = \sigma_w^{-2} \Lambda^T (y - \mu) \in \mathbb{R}^m$ can be obtained in the

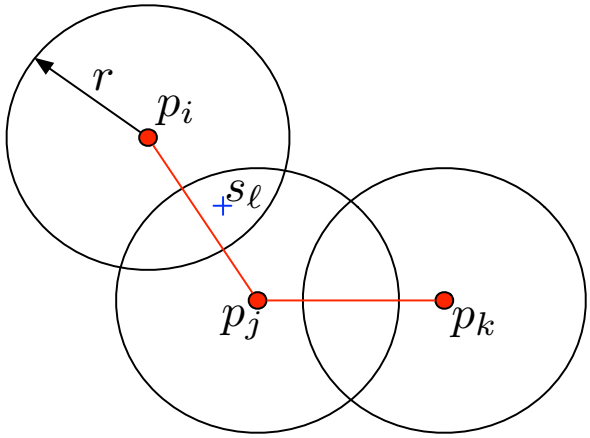


Figure 6.2: Example of computing $(\Lambda^T\Lambda)_{ij} = \lambda(s_\ell, p_i)\lambda(s_\ell, p_j)$.

following form.

$$\begin{aligned}(\hat{Q})_{ij} &= (Q)_{ij} + \sigma_w^{-2} \sum_{\ell \in \{\{i\} \cup \mathcal{N}_i\} \cap \left\{\{j\} \cup \mathcal{N}_j\right\}} \lambda(s_\ell, p_i)\lambda(s_\ell, p_j), \\ (\hat{y})_i &= \sigma_w^{-2} \sum_{\ell \in \{i\} \cup \mathcal{N}_i} \lambda(s_\ell, p_i)(y_\ell - \mu_\ell).\end{aligned} \tag{6.7}$$

Notice that $\hat{Q}$ has the same sparsity as Q. From (6.7), A.3 and A.4, agent i can compute $\text{row}_i(\hat{Q})$ and $(\hat{y})_i$ by using only local information from neighbors. Using $\text{row}_i(\hat{Q})$ and $(\lambda)_i$, agent i can obtain the i-th element in the vector $\hat{Q}^{-1}\lambda = (Q+\sigma_w^{-2}\Lambda^T\Lambda)^{-1}\lambda$ via JOR by using only local information. Finally, using $(\hat{y})_i$ and $(\lambda)_i$ the prediction mean and variance can be obtained via the discrete-time average consensus algorithm. Notice that the sequential update of $\hat{Q}$ and $\hat{y}$ for sequential observations proposed in Section 6.1.3 can be also applied to the distributed algorithm. The distributed algorithm for sequential field prediction under assumptions A.1-4 is summarized in Table 6.2.

The number of robotic sensors and the sampling rule can be modified or optimized to maintain a better quality of the prediction and the corresponding distributed algorithm may

Table 6.2: Distributed algorithm for sequential field prediction.

Input:
(1) a set of target points $\mathcal{S}$
(2) the topology of sensor network $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ in which $\mathcal{E} := \big\{\{i,j\} \mid \|p_i - p_j\| \le R\big\}$
Output:
(1) prediction mean $\Big\{\mu_{z_0|y} \,|\, s_0 \in \mathcal{S}\Big\}$
(2) prediction error variance $\Big\{\sigma^2_{z_0|y} \,|\, s_0 \in \mathcal{S}\Big\}$
Assumption:
(A1) agent $i \in \mathcal{I}$ is in charge of sampling at point s_i within a r-disk centered at p_i, *i.e.*, $\|s_i - p_i\| < r$
(A2) the radius of the support of the weighting function satisfies $0 < r < \frac{R}{2}$
(A3) agent $i \in \mathcal{I}$ can only locally communicate with neighbors $\mathcal{N}_i := \{j \in \mathcal{I} \mid \{i,j\} \in \mathcal{E}\}$ defined by the connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
(A4) agent $i \in \mathcal{I}$ initially has $\mathrm{row}_i(\hat{Q}) \leftarrow \mathrm{row}_i(Q)$, $(\hat{y})_i \leftarrow 0$

At time t, agent $i \in \mathcal{I}$ in the network does the following concurrently:

1: take measurement y_i from its current location s_i
2: update $\mathrm{row}_i(\hat{Q}) \leftarrow \mathrm{row}_i(\hat{Q}) + \mathrm{row}_i(\sigma_w^{-2}\Lambda^T\Lambda)$ by exchanging information from neighbors $\mathcal{N}_i$
3: update $(\hat{y})_i \leftarrow (\hat{y})_i + (\sigma_w^{-2}\Lambda^T(y-\mu))_i$ by exchanging information from neighbors $\mathcal{N}_i$
4: **for** $s_0 \in \mathcal{S}$ **do**
5: compute $(\lambda)_i = \lambda(s_0, p_i)$
6: compute $(\hat{Q}^{-1}\lambda)_i$ via JOR
7: compute $\mu_{z_0|y} = \mu(s_0) + \lambda^T\hat{Q}^{-1}\hat{y}$ via DAC
8: compute $\sigma^2_{z_0|y} = \lambda^T\hat{Q}^{-1}\lambda$ via DAC
9: **end for**

be derived in a same way accordingly.

6.3 Simulation and Experiment

In this section, we apply the proposed schemes to both simulation and experimental study.

6.3.1 Simulation

We first apply our proposed prediction algorithms to a numerically generated Gaussian random field $z(\cdot)$ based on a GMRF with respect to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ defined in (6.1). The mean function $\mu(\cdot)$ is assumed to be constant and $\mu = 5$ is used in the simulation. We assume the generating points of the GMRF, indexed by $\mathcal{V} = \{1, \cdots, n\}$ where $n = 30$, are located at $\{p_1, \cdots, p_n\}$ in a 2-D unit area $\mathcal{Q}$. The edges of the graph are assumed to be $\mathcal{E} := \big\{\{i, j\} \mid \big\|p_i - p_j\big\| \leq R\big\}$, where $R = 0.4$.

The GMRF $\gamma = (\gamma(p_1), \cdots, \gamma(p_n))^T$ has a zero-mean and the precision matrix Q is given by

$$(Q)_{ij} = \begin{cases} |\mathcal{N}(i)| + c_0, & \text{if } j = i, \\ -1, & \text{if } j \in \mathcal{N}(i), \\ 0, & \text{otherwise,} \end{cases}$$

where $|\mathcal{N}(i)|$ denotes the degree of node i, *i.e.*, the number of connections it has to other nodes, $c_0 = 0.1$ is used to ensure Q is positive definite since a Hermitian diagonally dominant matrix with real non-negative diagonal entries is positive semi-definite [54]. We use compactly supported weighting functions defined in (6.4) for both centralized and distributed schemes with different support r. The sensor noise level is given by $\sigma_w = 0.5$. Since the optimal sampling is beyond the scope of this chapter, in the simulation, we use a random sampling strategy in which robotic sensors sample at random locations at each time instance.

6.3.2 Centralized scheme

We first consider a scenario in which $N = 5$ agents take samples in the surveillance region $\mathcal{D}$ at certain time instance $t \in \mathbb{Z}_{>0}$ and send the observations to a central station in which

the prediction of the field is made.

The Gaussian random field $z(\cdot)$ is shown in Fig. 6.3-(a) with the $n = 30$ generating points of the built-in GMRF shown in black circles. The predicted field at times $t = 1$, $t = 5$, and $t = 20$ are shown in Figs. 6.3-(b), (c), and (d), respectively. The sampling locations are shown in black crosses. Clearly, the predicted field gets closer to the true field as the number of observations increases. The computational time for field prediction at each time instance remains fixed due to the nice structure of the proposed Gaussian field in (6.1) and its consequent results from Theorem 6.1.3.

6.3.3 Distributed scheme

Next, we consider a scenario in which prediction is implemented in a distributed fashion (Table 6.2) under assumptions A.1-4 for the resource-constrained mobile sensor network in Section 6.2.2. In particular, $N = 30$ robotic sensors are distributed according to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is connected. Agent i is in charge of the sampling with in a r-disk centered at p_i, where the support $r = 0.2$ is used. Agent i has a fixed neighborhood, *i.e.*, $\mathcal{N}(i) = \{j \mid \{i, j\} \in \mathcal{E}\}$. In the simulation, $h = 0.02$ in (6.5) and $\epsilon = 0.02$ in (6.6) are chosen to ensure the convergence of the JOR algorithm and the DAC algorithm.

Fig. 6.4-(a) shows the underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for the GMRF with the generating points denoted by black circles and the edges in red lines. The sparsity of the precision matrix Q is shown in Fig. 6.4-(b). Notice that only 316 out of 900 elements in Q are non-zero which enables the efficient distributed computation. The true and the predicted fields at time $t = 5$ are shown in Figs. 6.5-(a) and (b), respectively. The normalized RMS error computed over about 10000 grid points at time $t = 5$ is 7.8%. The computational time at

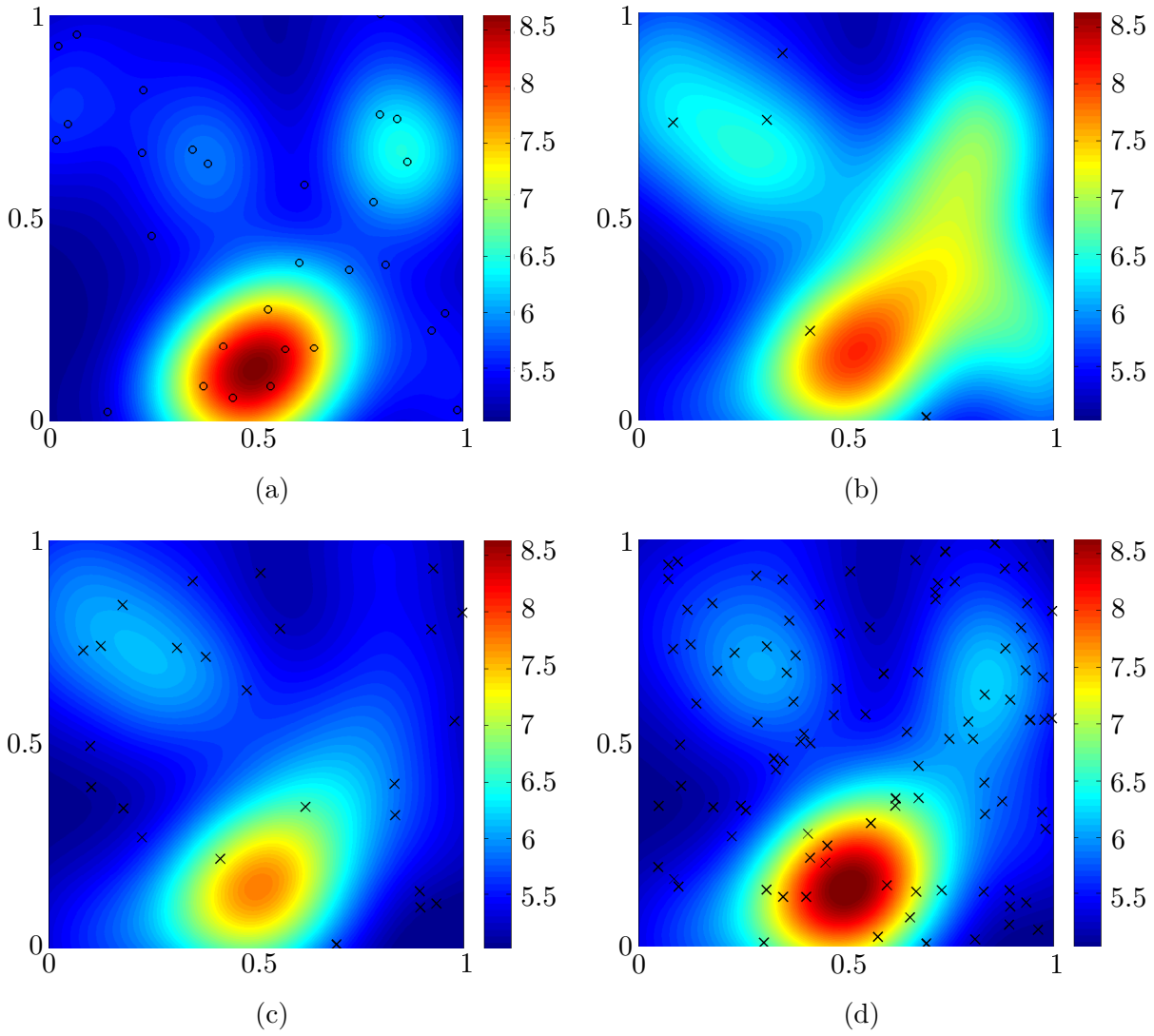
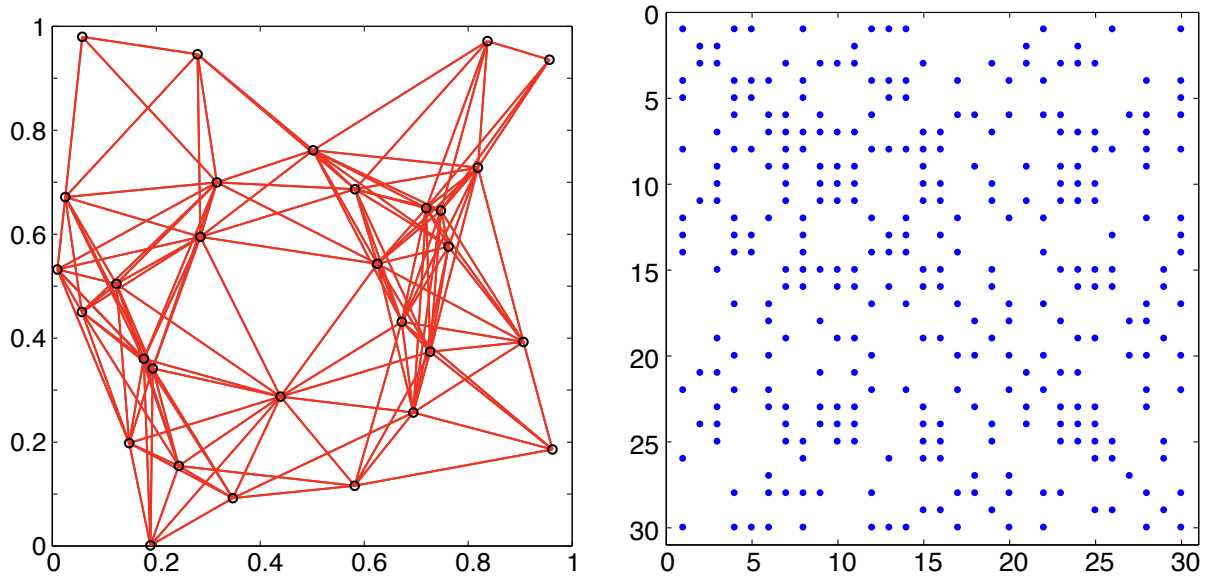


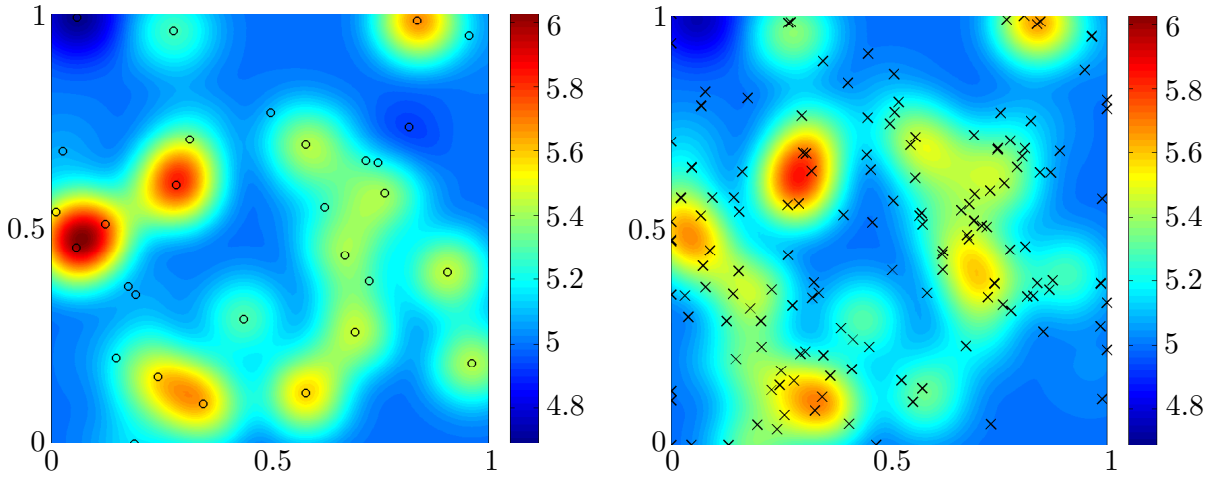
Figure 6.3: Simulation results for the centralized scheme. (a) The true field, (b) the predicted field at time $t = 1$, (c) the predicted field at time $t = 5$, (d) the predicted field at time $t = 20$. The generating points are shown in black circles, and the sampling locations are shown in black crosses.

each time instance remains fixed due to the nice structure of the proposed Gaussian field in (6.1) and its consequent results from Theorem 6.1.3.



(a) (b)

Figure 6.4: (a) Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. (b) Sparsity structure of the precision matrix Q.



(a) (b)

Figure 6.5: Simulation results for the distributed scheme. (a) The true field, (b) the predicted field at time $t = 5$. The generating points are shown in circles, and the sampling locations are shown in crosses.

6.3.4 Experiment

In order to show the practical usefulness of the proposed approach, we apply the centralized scheme in Theorem 6.1.3 on an experimentally obtained observations. We first measured

depth values of a terrain on grid points by using a Microsoft Kinect sensor [12] as shown in Fig. 6.6-(a). As pointed out in Remark 6.1.1, we make the structures of weighting functions and the precision matrix as functions of the locations of generating points. In particular, two generating points are neighbors if and only if their corresponding Voronoi cells intersect. The individual weighting function takes the same form as in (6.4) and its support size r_i is selected to be the largest distrance between the generating point i and it's neighbors. We then predict the field by our model with 20 estimated generating points given by the ML estimator in (6.3) using a subset of experimental observations, *i.e.*, 200 randomly sampled observations denoted by crosses in Fig. 6.6-(a). The estimated positions of generating points along with the predicted field are shown in Fig. 6.6-(b). In this experiment, it is clear to see that our approach effectively produces the predicted field, which is very close to the true field for the case of unknown generating points.

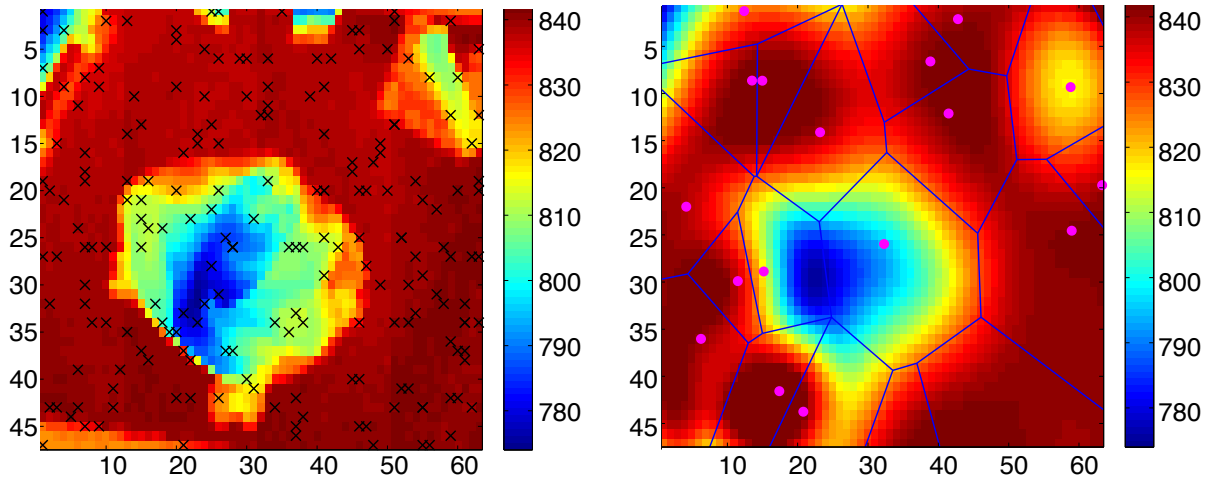


Figure 6.6: (a) True field on grid positions obtained by the Kinect sensor and randomly sampled positions indicated in black crosses. (b) The fitted Gaussian random field with a build-in GMRF with respect to the Delaunay graph.

Chapter 7

Bayesian Spatial Prediction Using GMRF

In this chapter, we consider the problem of predicting a large scale spatial field using successive noisy measurements obtained by mobile sensing agents. The physical spatial field of interest is discretized and modeled by a Gaussian Markov random field (GMRF) with unknown hyperparameters. From a Bayesian perspective, we design a sequential prediction algorithm to exactly compute the predictive inference of the random field. The main advantages of the proposed algorithm are: (1) the computational efficiency due to the sparse structure of the precision matrix, and (2) the scalability as the number of measurements increases. Thus, the prediction algorithm correctly takes into account the uncertainty in hyperparameters in a Bayeisan way and also is scalable to be usable for the mobile sensor networks with limited resources. An adaptive sampling strategy is also designed for mobile sensing agents to find the most informative locations in taking future measurements in order to minimize the prediction error and the uncertainty in hyperparameters. The effectiveness

of the proposed algorithms is illustrated by a numerical experiment.

In Chapter 5, we designed a sequential Bayesian prediction algorithm to deal with unknown bandwidths by using a compactly supported kernel and selecting a subset of collected measurements. In this Chapter, we instead seek a fully Bayesian approach over a discretized surveillance region such that the Bayesian spatial prediction utilizes all collected measurements in a scalable fashion.

In Section 7.1, we model the physical spatial field as a GMRF with unknown hyperparameters and formulate the estimation problem from a Bayesian point of view. In Section 7.2, we design an sequential Bayesian estimation algorithm to effectively and efficiently compute the exact predictive inference of the spatial field. The proposed algorithm often takes only seconds to run even for a very large spatial field, as will be demonstrated in this chapter. Moreover, the algorithm is scalable in the sense that the running time does not grow as the number of observations increases. In particular, the scalable prediction algorithm does not rely on the subset of samples to obtain scalability (as was done in Chapter 5), correctly fusing all collected measurements. In Section 7.4, an adaptive sampling strategy for mobile sensor networks is designed to largely improve the quality of prediction and to reduce the uncertainty in the hyperparameter estimation simultaneously. We demonstrate the effectiveness through a simulation study in Section 7.5.

7.1 Problem Setup

In what follows, we specify the models for the spatial field and the mobile sensor network. Notice that in this Chapter, we slightly change notation for notational simplicity.

7.1.1 Spatial field model

Let $\mathcal{Q}_* \subset \mathbb{R}^D$ denote the spatial field of interest. We discretize the field into n_* spatial sites $\mathcal{S}_* := \{s_1, \cdots, s_{n_*}\}$ and let $z_* = (z_1, \cdots, z_{n_*})^T \in \mathbb{R}^{n_*}$ be the value of the field (*e.g.*, the temperature). Due to the irregular shape a spatial field may have, we extend the field such that $n \geq n_*$ sites denoted by $\mathcal{S} := \{s_1, \cdots, s_n\}$ are on a regular grid. The latent variable $z_i := z(s_i) \in \mathbb{R}$ is modeled by

$$z_i = \mu(s_i) + \eta_i, \quad \forall 1 \leq i \leq n, \tag{7.1}$$

where $s_i \in \mathcal{S} \subset \mathbb{R}^D$ is the i-th site location. The mean function $\mu : \mathbb{R}^D \to \mathbb{R}$ is defined as

$$\mu(s_i) = f(s_i)^T \beta,$$

where $f(s_i) = (f_1(s_i), \cdots, f_p(s_i))^T \in \mathbb{R}^p$ is a known regression function, and $\beta = (\beta_1, \cdots, \beta_p)^T \in \mathbb{R}^p$ is an unknown vector of regression coefficients. We define $\eta = (\eta_1, \cdots, \eta_n)^T \in \mathbb{R}^n$ as a zero-mean Gaussian Markov random field (GMRF) [54] denoted by

$$\eta \sim \mathcal{N}\left(0, Q_{\eta|\theta}^{-1}\right),$$

where the inverse covariance matrix (or precision matrix) $Q_{\eta|\theta} \in \mathbb{R}^{n \times n}$ is a function of a hyperparameter vector $\theta \in \mathbb{R}^m$.

There exists many different choices of the GMRF (*i.e.*, the precision matrix $Q_{\eta|\theta}$) [54]. For instance, we can choose one with the full conditionals in (7.2) (with obvious notation as shown in [54]).

$$\mathrm{E}(\eta_i|\eta_{-i},\theta) = \frac{1}{4+a^2}\begin{pmatrix} & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \bullet & \circ & \circ \\ & \circ & \circ & \bullet & \circ & \circ & & \circ & \bullet & \circ & \bullet & \circ & & \circ & \circ & \circ & \circ & \circ \\ 2a & \circ & \bullet & \circ & \bullet & \circ & -2 & \circ & \circ & \circ & \circ & \circ & -1 & \bullet & \circ & \circ & \circ & \bullet \\ & \circ & \circ & \bullet & \circ & \circ & & \circ & \bullet & \circ & \bullet & \circ & & \circ & \circ & \circ & \circ & \circ \\ & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \bullet & \circ & \circ \end{pmatrix},$$

$$\mathrm{Var}(\eta_i|\eta_{-i},\theta) = (4+a^2)\kappa. \tag{7.2}$$

Fig. 7.1 displays the elements of the precision matrix related to a single location that explains (7.2). The hyperparameter vector is defined as $\theta = (\kappa, \alpha)^T \in \mathbb{R}^2_{>0}$, where $\alpha = a - 4$. The resulting GMRF accurately represents a Gaussian random field with the Matérn covariance function [36]

$$C(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{\ell}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}r}{\ell}\right),$$

where $K_\nu(\cdot)$ is a modified Bessel function [51], with order $\nu = 1$, a bandwidth $\ell = 1/\sqrt{\alpha}$, and vertical scale $\sigma_f^2 = 1/4\pi\alpha\kappa$. The hyperparameter $\alpha > 0$ guarantees the positive definiteness of the precision matrix $Q_{\eta|\theta}$. In the case where $\alpha = 0$, the resulting GMRF is a second-order polynomial intrinsic GMRF [54, 55]. Notice that the precision matrix is sparse which contains only small number of non-zero elements. This property will be exploited for fast computation in the following sections.

Example 7.1.1. *Consider a spatial field of interest* $\mathcal{Q}_* \in [0, 100] \times [0, 50]$. *We first divide*

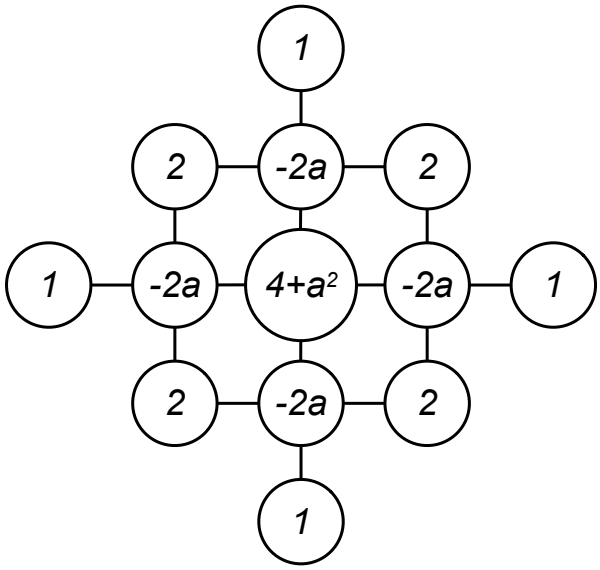
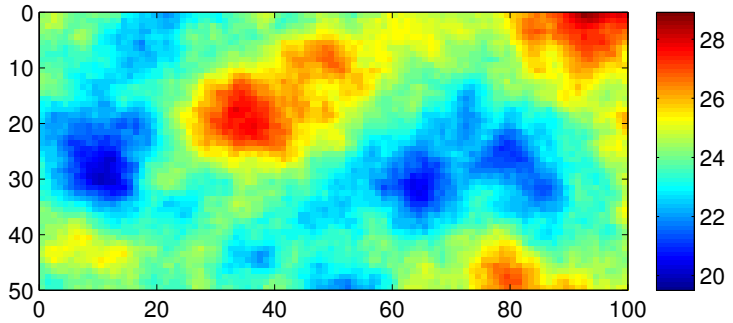


Figure 7.1: Elements of the precision matrix Q related to a single location.

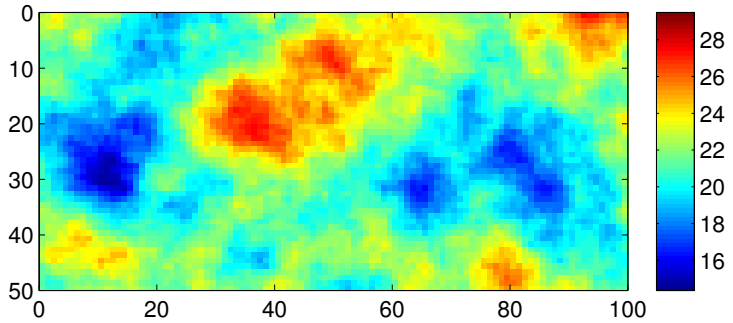
the spatial field into a 100×50 *regular grid with equal areas* 1*, which makes* $n_* = 5000$*. We then extend the the field such that* 120×70 *grids (*i.e.*,* $n = 8400$*) are constructed on the extended field* $\mathcal{Q} = [-10, 110] \times [-10, 60]$*. The precision matrix* $Q_{\eta|\theta}$ *introduced above is chosen with the regular lattices wrapped on a torus [54]. In this case, only* 0.15% *elements in the sparse matrix* $Q_{\eta|\theta}$ *are non-zero. The numerically generated fields with the mean function* $\mu(s_i) = \beta = 20$*, and the hyperparameter vector* $\theta = (\kappa, \alpha)^T$ *being different values are shown in Fig. 7.2.*

7.1.2 Mobile sensor network

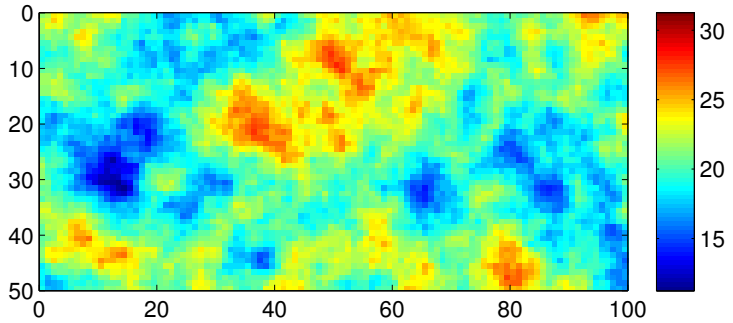
Consider N spatially distributed mobile sensing agents indexed by $i \in \mathcal{I} = \{1, \cdots, N\}$ sampling from n_* spatial sites in $\mathcal{S}_*$. Agents are equipped with identical sensors and sample at time $t \in \mathbb{Z}_{>0}$. At time t, agent i takes a noisy corrupted measurement at it's current



(a)



(b)



(c)

Figure 7.2: Numerically generated spatial fields defined in (7.1) with $\mu(s_i) = \beta = 20$, and $Q_{\eta|\theta}$ constructed using (7.2) with hyperparameters being (a) $\theta = (4, 0.0025)^T$, (b) $\theta = (1, 0.01)^T$, and (c) $\theta = (0.25, 0.04)^T$.

location $q_{t,i} \in \mathcal{S}_*$, *i.e.*,

$$y_{t,i} = z(q_{t,i}) + \epsilon_{t,i}, \quad \epsilon_{t,i} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_w^2),$$

where measurement errors are assumed to be independent and identically distributed (i.i.d.). The noise level $\sigma_w^2 > 0$ is assumed to be known. For notational simplicity, we denote all agents' locations at time t by $q_t = (q_{t,1}^T, \cdots, q_{t,N}^T)^T$ and the observations made by all agents at time t by $y_t = (y_{t,1}, \cdots, y_{t,N})^T$. Furthermore, we denote the collection of agents' locations and the collective observations from time 1 to t by $q_{1:t} = (q_1^T, \cdots, q_t^T)^T$, and $y_{1:t} = (y_1, \cdots, y_N)^T$, respectively.

7.2 Bayesian Predictive Inference

In this section, we propose a Bayesian inference approach to make predictive inferences of a spatial field $z_* \in \mathbb{R}^{n*}$.

First, we assign the vector of regression coefficients $\beta \in \mathbb{R}^p$ with a Gaussian prior, namely $\beta \sim \mathcal{N}\left(0, T^{-1}\right)$, where the precision matrix $T \in \mathbb{R}^{p \times p}$ is often chosen as a diagonal matrix with small diagonal elements when no prior information is available. Hence, the distribution of latent variables z given β and the hyperparameter vector θ is Gaussian, *i.e.*,

$$z|\beta, \theta \sim \mathcal{N}\left(F\beta, Q_{\eta|\theta}^{-1}\right),$$

where $F = (f(s_1), \cdots, f(s_n))^T \in \mathbb{R}^{n \times p}$. For notational simplicity, we denote the full latent field of dimension $n + p$ by $x = (z^T, \beta^T)^T$. Then, for a given hyperparameter vector θ, the

distribution $\pi(x|\theta)$ is Gaussian obtained by

$$\begin{aligned}\pi(x|\theta) &= \pi(z|\beta,\theta)\pi(\beta)\\ &\propto \exp\left(-\frac{1}{2}(z-F\beta)^T Q_{\eta|\theta}(z-F\beta)-\frac{1}{2}\beta^T T\beta\right)\\ &= \exp\left(-\frac{1}{2}x^T Q_{x|\theta}x\right),\end{aligned}$$

where the precision matrix $Q_{x|\theta}\in\mathbb{R}^{(n+p)\times(n+p)}$ is defined by

$$Q_{x|\theta}=\begin{bmatrix} Q_{\eta|\theta} & -Q_{\eta|\theta}F\\ -F^T Q_{\eta|\theta} & F^T Q_{\eta|\theta}F+T\end{bmatrix}.$$

By the matrix inversion lemma, the covariance matrix $\Sigma_{x|\theta}\in\mathbb{R}^{(n+p)\times(n+p)}$ can be obtained by

$$\Sigma_{x|\theta}=Q_{x|\theta}^{-1}=\begin{bmatrix} Q_{\eta|\theta}^{-1}+FT^{-1}F^T & FT^{-1}\\ (FT^{-1})^T & T^{-1}\end{bmatrix}.$$

At time $t\in\mathbb{Z}_{>0}$, we have a collection of observational data $y_{1:t}\in\mathbb{R}^{Nt}$ obtained by the mobile sensing agents over time. Let $A_{1:t}=(A_1,\cdots,A_t)\in\mathbb{R}^{(n+p)\times Nt}$, where $A_\tau\in\mathbb{R}^{(n+p)\times N}$ is defined by

$$(A_\tau)_{ij}=\begin{cases}1, & \text{if } s_i=q_{\tau,j},\\ 0, & \text{otherwise.}\end{cases}$$

Then the covariance matrix of $y_{1:t}$ can be obtained by

$$R_{1:t}=A_{1:t}^T\Sigma_{x|\theta}A_{1:t}+P_{1:t},$$

where $P_{1:t}=\sigma_w^2 I\in\mathbb{R}^{Nt\times Nt}$. By Gaussian process regression [51], the full conditional

distribution of x is also Gaussian, *i.e.*,

$$x|\theta, y_{1:t} \sim \mathcal{N}(\mu_{x|\theta,y_{1:t}}, \Sigma_{x|\theta,y_{1:t}}),$$

where

$$\begin{aligned} \Sigma_{x|\theta,y_{1:t}} &= \Sigma_{x|\theta} - \Sigma_{x|\theta} A_{1:t} R_{1:t}^{-1} A_{1:t}^T \Sigma_{x|\theta}, \\ \mu_{x|\theta,y_{1:t}} &= \Sigma_{x|\theta} A_{1:t} R_{1:t}^{-1} y_{1:t}. \end{aligned} \tag{7.3}$$

The posterior distribution of the hyperparameter vector θ can be obtained via

$$\pi(\theta|y_{1:t}) \propto \pi(y_{1:t}|\theta)\pi(\theta),$$

where the log likelihood function is defined by

$$\log \pi(y_{1:t}|\theta) = -\frac{1}{2} y_{1:t}^T R_{1:t}^{-1} y_{1:t} - \frac{1}{2} \log \det R_{1:t} - \frac{Nt}{2} \log 2\pi. \tag{7.4}$$

If a discrete prior on the hyperparameter vector θ is chosen with a support $\Theta = \{\theta_1, \cdots, \theta_L\}$, the posterior predictive distribution $\pi(x|y_{1:t})$ can be obtained by

$$\pi(x|y_{1:t}) = \sum_{\ell} \pi(x|\theta_\ell, y_{1:t}) \pi(\theta_\ell|y_{1:t}). \tag{7.5}$$

The predictive mean and variance then follow as

$$\begin{aligned}\mu_{x_i|y_{1:t}} &= \sum_{\ell} \mu_{x_i|\theta_\ell,y_{1:t}} \pi(\theta_\ell|y_{1:t}),\\ \sigma^2_{x_i|y_{1:t}} &= \sum_{\ell} \sigma^2_{x_i|\theta_\ell,y_{1:t}} \pi(\theta_\ell|y_{1:t}) + \sum_{\ell} (\mu_{x_i|\theta_\ell,y_{1:t}} - \mu_{x_i|y_{1:t}})^2 \pi(\theta_\ell|y_{1:t}),\end{aligned} \tag{7.6}$$

where $\mu_{x_i|\theta_\ell,y_{1:t}}$ is the i-th element in $\mu_{x|\theta_\ell,y_{1:t}}$, and $\sigma^2_{x_i|\theta_\ell,y_{1:t}}$ is the i-th diagonal element in $\Sigma_{x|\theta_\ell,y_{1:t}}$.

Remark 7.2.1. *The discrete prior $\pi(\theta)$ greatly reduced the computational complexity in that it enables summation in* (7.5) *instead of numerical integration which has to be performed with a choice of continuous prior distribution. However, the computation of the full conditional distribution $\pi(x|\theta, y_{1:t})$ in* (7.3) *and the likelihood $\pi(y_{1:t}|\theta)$* (7.4) *requires the inversion of the covariance matrix $R_{1:t}$, whose size grows as the time t increases. Thus, the running time grows fast as new observations are collected and it will soon become intractable.*

7.3 Sequential Bayesian Inference

In this section, we exploit the sparsity of the precision matrix, and propose a sequential Bayesian prediction algorithm which can be performed in constant time and fast enough even for a very large spatial field.

7.3.1 Update full conditional distribution

First, we rewrite the full conditional distribution $\pi(x|\theta, y_{1:t})$ in terms of the sparse precision matrix $Q_{x|\theta}$ as follows

$$x|\theta, y_{1:t} \sim \mathcal{N}(\mu_{x|\theta,y_{1:t}}, Q^{-1}_{x|\theta,y_{1:t}}),$$

where

$$\begin{aligned} Q_{x|\theta,y_{1:t}} &= Q_{x|\theta} + A_{1:t}P_{1:t}^{-1}A_{1:t}^T \\ \mu_{x|\theta,y_{1:t}} &= Q_{x|\theta,y_{1:t}}^{-1}A_{1:t}P_{1:t}^{-1}y_{1:t}. \end{aligned} \tag{7.7}$$

From here on, we will use $Q_{t|\theta} = Q_{x|\theta,y_{1:t}}$ and $\mu_{t|\theta} = \mu_{x|\theta,y_{1:t}}$, for notational simplicity. Notice that (7.7) can be represented by the following recursion

$$\begin{aligned} Q_{t|\theta} &= Q_{t-1|\theta} + \frac{1}{\sigma_w^2}\sum_{i=1}^{N} u_{t,i}u_{t,i}^T, \\ b_t &= b_{t-1} + \frac{1}{\sigma_w^2}\sum_{i=1}^{N} u_{t,i}y_{t,i}, \end{aligned} \tag{7.8}$$

where $b_t = Q_{t|\theta}\mu_{t|\theta}$ with initial conditions

$$Q_{0|\theta} = Q_{x|\theta,y_{1:0}} = Q_{x|\theta}, \text{and } b_0 = 0.$$

In (7.8), we have defined $u_{t,i} \in \mathbb{R}^{n+p}$ as

$$(u_{t,i})_j = \begin{cases} 1, & \text{if } s_j = q_{t,i}, \\ 0, & \text{otherwise.} \end{cases}$$

Lemma 7.3.1. *For a given $\theta \in \Theta$, the full conditional mean and variance,* i.e., *$\mu_{t|\theta}$ and $Q_{t|\theta}$, can be updated in short constant time given $Q_{t-1|\theta}$ and b_{t-1}.*

Proof. The update of $Q_{t|\theta}$ and b_t can be obviously computed in constant time. Hence $\mu_{t|\theta}$ can be obtained by solving a linear equation $Q_{t|\theta}\mu_{t|\theta} = b_t$. Due to the sparse structure of $Q_{t|\theta}$, this operation can be done in a very short time. Moreover, notice that $Q_{t|\theta}$ and $Q_{t-1|\theta}$

have the same sparsity structure and hence the computational complexity remains fixed. □

From Lemma 7.3.1, we can compute $\mu_{x_i|\theta,y_{1:t}}$ in (7.6) in constant time. In order to find $\sigma^2_{x_i|\theta,y_{1:t}}$ in (7.6), we need to compute $\Sigma_{x|\theta,y_{1:t}}$ which requires the inversion of $Q_{t|\theta}$. The inversion of a big matrix (even a sparse matrix) is undesirable. However, notice that only the diagonal elements in $Q^{-1}_{t|\theta}$ are needed. Following the Sherman-Morrison formula (see Appendix A.2.2) and using (7.8), $\sigma^2_{x_i|\theta,y_{1:t}}$ can be obtained exactly via

$$\begin{aligned}
\operatorname{diag}(Q^{-1}_{t|\theta}) &= \operatorname{diag}\left(\left[Q_{t-1|\theta} + \sum_{i=1}^{N} u_{t,i}u_{t,i}^T\right]^{-1}\right) \\
&= \operatorname{diag}(Q^{-1}_{t-1|\theta}) - \sum_{i=1}^{N} \frac{h_{t,i|\theta} \circ h_{t,i|\theta}}{\sigma_w^2 + u_{t,i}^T h_{t,i|\theta}}, \qquad (7.9)\\
h_{t,i|\theta} &= B^{-1}_{t,i|\theta} u_{t,i}, \\
B_{t,i|\theta} &= Q_{t-1|\theta} + \frac{1}{\sigma_w^2}\sum_{j=1}^{i} u_{t,j}u_{t,j}^T,
\end{aligned}$$

where $\circ$ denotes the element-wise produce. By this way, the computation can be done efficiently in constant time.

7.3.2 Update likelihood

Next, we derive the update rule for the log likelihood function. We have the following proposition.

Proposition 7.3.2. *The log likelihood function* $\log \pi(y_{1:t}|\theta)$ *in* (7.4) *can be obtained by*

$$\log \pi(y_{1:t}|\theta) = c_t + g_{t,\theta} + \frac{1}{2} b_t^T \mu_{t|\theta} - \frac{Nt}{2}\log(2\pi\sigma_w^2) \qquad (7.10)$$

where

$$c_t = c_{t-1} - \frac{1}{2\sigma_w^2}\sum_{i=1}^{N} y_{t,i}^2, \quad c_0 = 0,$$
$$g_{t|\theta} = g_{t-1|\theta} - \frac{1}{2}\sum_{i=1}^{N} \log\left(1 + \frac{1}{\sigma_w^2} u_{t,i}^T h_{t,i|\theta}\right), \quad g_{0|\theta} = 0,$$

with $h_{t,i|\theta}$ *defined in* (7.9).

Proof. The inverse of the covariance matrix $R_{1:t}$ can be obtained by

$$\begin{aligned} R_{1:t}^{-1} &= (A_{1:t}^T Q_{0|\theta}^{-1} A_{1:t} + P_{1:t})^{-1} \\ &= P_{1:t}^{-1} - P_{1:t}^{-1} A_{1:t}^T (Q_{0|\theta} + A_{1:t} P_{1:t}^{-1} A_{1:t}^T)^{-1} A_{1:t} P_{1:t}^{-1} \\ &= P_{1:t}^{-1} - P_{1:t}^{-1} A_{1:t}^T Q_{t|\theta}^{-1} A_{1:t} P_{1:t}^{-1}. \end{aligned}$$

Similarly, the log determinant of the covariance matrix $\Sigma_{1:t}$ can be obtained by

$$\begin{aligned} \log\det R_{1:t} &= \log\det(A_{1:t}^T Q_{0|\theta}^{-1} A_{1:t} + P_{1:t}) \\ &= \log\det(I + \frac{1}{\sigma_w^2} A_{1:t}^T Q_{0|\theta}^{-1} A_{1:t}) + Nt\log\sigma_w^2 \\ &= \log\det(Q_{0|\theta} + \frac{1}{\sigma_w^2}\sum_{\tau=1}^{t}\sum_{i=1}^{N} u_{\tau,i} u_{\tau,i}^T) - \log\det(Q_{0|\theta}) + Nt\log\sigma_w^2 \\ &= \sum_{\tau=1}^{t} \log(1 + u_\tau^T Q_{\tau-1|\theta}^{-1} u_\tau) + Nt\log\sigma_w^2. \end{aligned}$$

Hence, we have

$$\begin{aligned}
&\log \pi(y_{1:t}|\theta) \\
&= -\frac{1}{2} y_{1:t}^T R_{1:t}^{-1} y_{1:t} - \frac{1}{2} \log\det R_{1:t} - \frac{Nt}{2} \log 2\pi \\
&= -\frac{1}{2} y_{1:t}^T P_{1:t}^{-1} y_{1:t} + \frac{1}{2} b_t^T \mu_{t|\theta} - \frac{1}{2} \sum_{\tau=1}^{t} \sum_{i=1}^{N} \log(1 + u_{\tau,i}^T B_{\tau,i|\theta}^{-1} u_{\tau,i}) - \frac{Nt}{2} \log(2\pi\sigma_w^2).
\end{aligned}$$

□

Lemma 7.3.3. *For a given $\theta \in \Theta$, the log likelihood function,* i.e., $\log \pi(y_{1:t}|\theta)$ *can be computed in short constant time.*

Proof. The result follows directly from Proposition 7.3.2. □

7.3.3 Update predictive distribution

Combining the results in Lemmas 7.3.1, 7.3.3, and (7.5), (7.6), we summarize our results in the following theorem.

Theorem 7.3.4. *The predictive distribution in* (7.5) *(or the predictive mean and variance in* (7.6)*) can be obtained in constant time as time t increases.*

We summarize the proposed sequential Bayesian prediction algorithm in Table 7.1.

7.4 Adaptive Sampling

In the previous section, we have designed a sequential Bayesian prediction algorithm for estimating the scalar field at time t. In this section, we propose an adaptive sampling strategy for finding most informative sampling locations at time $t + 1$ for mobile sensing

agents in order to improve the quality of prediction and reduce the uncertainty in hyper parameters simultaneously.

In our previous work [66], we have proposed to use the conditional entropy $H(z_*|\theta = \hat{\theta}_t, y_{1:t+1})$ as an optimality criterion, where

$$\hat{\theta}_t = \arg\max_{\theta} \pi(\theta|y_{1:t}),$$

is the maximum *a posterior* (MAP) estimate based on the cumulative observations up to current time t. Although this approach greatly simplifies the computation, it does not count for the uncertainty in estimating the hyperparameter vector θ.

In this chapter, we propose to use the conditional entropy $H(z_*, \theta|y_{t+1}, y_{1:t})$ which represents the uncertainty remained in both random vectors z_* and θ by knowing future measurements in the random vector y_{t+1}. Notice that the measurements $y_{1:t}$ have been observed and treated as constants. It can be obtained by

$$\begin{aligned} H(z_*, \theta|y_{t+1}, y_{1:t}) &= H(z_*|\theta, y_{t+1}, y_{1:t}) + H(\theta|y_{t+1}, y_{1:t}) \\ &= H(z_*|\theta, y_{t+1}, y_{1:t}) + H(y_{t+1}|\theta, y_{1:t}) + H(\theta|y_{1:t}) - H(y_{t+1}|y_{1:t}). \end{aligned}$$

Notice that we have the following Gaussian distributions (the means will not be exploited and hence not shown here):

$$\begin{aligned} z_*|\theta, y_{t+1}, y_{1:t} &\sim \mathcal{N}(\cdot, \Sigma_{x_*|\theta, y_{1:t+1}}), \\ y_{t+1}|\theta, y_{1:t} &\sim \mathcal{N}(\cdot, \Sigma_{y_{t+1}|\theta, y_{1:t}} + \sigma_w^2 I), \\ y_{t+1}|y_{1:t} &\overset{\text{approx}}{\sim} \mathcal{N}(\cdot, \Sigma_{y_{t+1}|y_{1:t}} + \sigma_w^2 I), \end{aligned}$$

in which the last one is approximated using (7.6). Notice that the approximation is used here to avoid numerical integration over the random vector y_{t+1} which needs to be done using Monte Carlo methods. Moreover, the entropy $H(\theta|y_{1:t}) = c$ is a constant since $y_{1:t}$ is known. Since the entropy for a multivariate Gaussian distribution has a closed-from expression [14], we have

$$\begin{aligned} H(z_*, \theta|y_{t+1}, y_{1:t}) = &\sum_\ell \frac{1}{2} \log \left((2\pi e)^{n_*} \det(\Sigma_{x_*|\theta_\ell, y_{1:t+1}}) \right) \pi(\theta_\ell|y_{1:t}) \\ &+ \sum_\ell \frac{1}{2} \log \left((2\pi e)^{N} \det(\Sigma_{q_{t+1}|\theta_\ell, y_{1:t}}) \right) \pi(\theta_\ell|y_{1:t}) \\ &- \frac{1}{2} \log \left((2\pi e)^{N} \det(\Sigma_{q_{t+1}|y_{1:t}}) \right) + c. \end{aligned}$$

It can also be shown that

$$\begin{aligned} \log\det(\Sigma_{x_*|\theta_\ell, y_{1:t+1}}) &= \log\det(Q^{-1}_{t+1|\theta_\ell})_{(\mathcal{S}_*)} \\ &= \log\det(Q_{t+1|\theta_\ell})_{(-\mathcal{S}_*)} - \log\det(Q_{t+1|\theta_\ell}), \end{aligned}$$

where $A_{(\mathcal{S}_*)}$ denotes the submatrix of A formed by the first 1 to n_* rows and columns (recall that $\mathcal{S}_* = \{s_1, \cdots, s_{n_*}\}$). Notice that the term $\log\det(Q_{t+1|\theta_\ell})_{(-\mathcal{S}_*)}$ is a constant since agents only sample at $\mathcal{S}_*$. Hence, the optimal sampling locations at time $t+1$ can be determined by solving the following optimization problem

$$\begin{aligned} q^*_{t+1} &= \arg \min_{\left\{q_{t+1,i} \in \mathcal{R}_{t,i}\right\}} H(z_*, \theta|y_{t+1}, y_{1:t}) \\ &= \arg \min_{\left\{q_{t+1,i} \in \mathcal{R}_{t,i}\right\}} \sum_\ell - \log\det(Q_{t+1|\theta_\ell}) \pi(\theta_\ell|y_{1:t}) \\ &\quad + \sum_\ell \log\det(\Sigma_{y_{t+1}|\theta_\ell, y_{1:t}}) \pi(\theta_\ell|y_{1:t}) - \log\det(\Sigma_{y_{t+1}|y_{1:t}}), \end{aligned}$$

where $\mathcal{R}_{t,i} = \left\{ s \mid \left\| s - q_{t,i} \right\| \leq r, s \in \mathcal{S}_* \right\}$ (in which $r \in \mathbb{R}_{>0}$ is the maximum distance an agent can move between time instances) is the reachable set at time t. This combinatorial optimization problem can be solved using a greedy algorithm, *i.e.*, finding the sub-optimal sampling locations for agents in sequence.

7.5 Simulation

In this section, we demonstrate the effectiveness of the proposed sequential Bayesian inference algorithm and the adaptive sampling strategy through a numerical experiment.

Consider a spatial field introduced in Example 7.1.1. The mean function is a constant $\beta = 20$. We choose the precision matrix $Q_{x|\theta}$ with hyperparameters $\alpha = 0.01$ equivalent to a bandwidth $\ell = 1/\sqrt{\alpha} = 10$, and $\kappa = 1$ equivalent to a vertical scale $\sigma_f^2 = 1/4\pi\alpha\kappa \approx 8$. The numerically generated field is shown in Fig. 7.2-(b). The precision matrix T of β is chosen to be 10^{-4}. The measurement noise level $\sigma_w = 0.2$ is assumed to be known. A discrete uniform distribution is selected with a support shown in Fig. 7.3. $N = 5$ mobile sensing agents take measurements at time $t \in \mathbb{Z}_{>0}$, starting from locations shown in Fig. 7.4-(b) (in white dots). The maximum distance each agent can travel between time instances is chosen to be $r = 5$.

Fig. 7.4 shows the predicted fields and the prediction error variances at times $t = 1, 5, 10, 20$. The trajectories of agents are shown in white circles with the current locations shown in white dots. It can be seen that agents try to cover the field of interest as time evolves. The predicted field (the predictive mean) gets closer to the true field (see Fig. 7.2-(b)) and the prediction error variances become smaller as more observations are collected. Fig. 7.3 shows the posterior distribution of the hyperparameters in θ. Clearly, as more measurements are obtained, this posterior distribution becomes peaked at the true

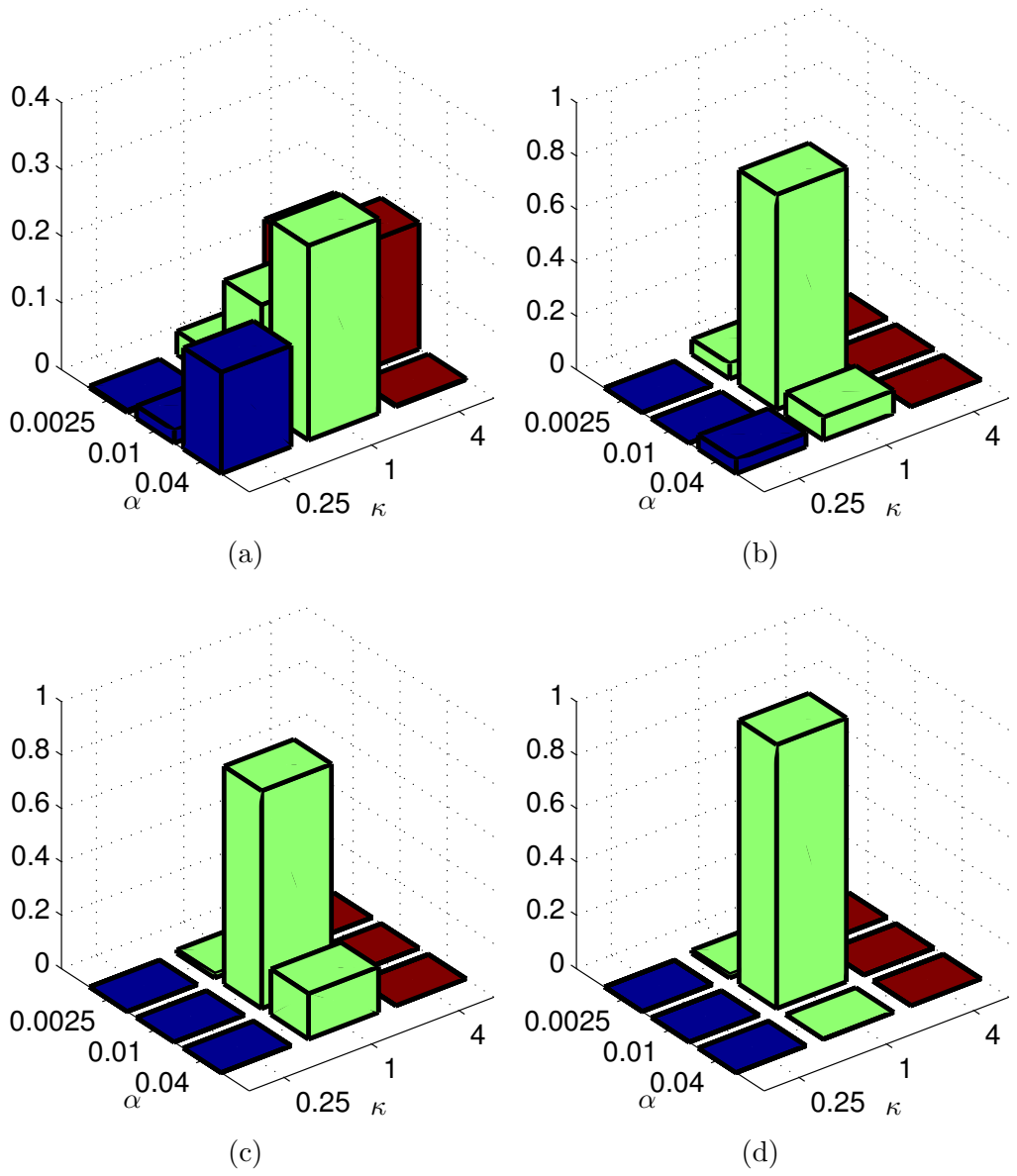


Figure 7.3: Posterior distributions of θ, *i.e.*, $\pi(\theta|y_{1:t})$, at (a) $t = 1$, (b) $t = 5$, (c) $t = 10$, and (d) $t = 20$.

value $(1, 0.01)$. Fig. 7.5-(a) shows the predicted distribution of the estimated mean β as time evolves. In Fig. 7.5-(b), we can see that the RMS error computed via

$$\text{rms}(t) = \sqrt{\frac{1}{n_*}\sum_{i=1}^{n_*}(\mu_{z_i|y_{1:t}} - z_i)^2},$$

decreases as time increases, which shows the effectiveness of the proposed scheme.

The most important contribution is that the computation time at each time step does not grow as the number of measurements increases. This fixed running time using Matlab, R2009b (MathWorks) in a Mac (2.4 GHz Intel Core 2 Duo Processor) is about 10 seconds which is fast enough for real-world implementation.

Table 7.1: Sequential Bayesian predictive inference.

Input:
(1) prior distribution of $\theta \in \Theta$, *i.e.*, $\pi(\theta)$
Output:
(1) predictive mean $\left\{\mu_{x_i|y_{1:t}}\right\}_{i=1}^{n_*}$
(2) predictive variance $\left\{\sigma^2_{x_i|y_{1:t}}\right\}_{i=1}^{n_*}$

Initialization:

initialize $b = 0$, $c = 0$
for $\theta \in \Theta$ **do**
  initialize Q_θ, $g_\theta = 0$
  compute $\mathrm{diag}(Q_\theta^{-1})$
end for

At time $t \in \mathbb{Z}_{>0}$, do:

for $1 \le i \le N$ **do**
  obtain new observations $y_{t,i}$ collected at current locations $q_{t,i}$
  find the index k corresponding to $q_{t,i}$, and set $u = e_k$
  update $b = b + \frac{y_{t,i}}{\sigma_w^2} u$
  update $c = c - \frac{1}{2\sigma_w^2} y_{t,i}^2$
  for $\theta \in \Theta$ **do**
    compute $h_\theta = Q_\theta^{-1} u$
    update $\mathrm{diag}(Q_\theta^{-1}) = \mathrm{diag}(Q_\theta^{-1}) - \frac{h_\theta \circ h_\theta}{\sigma_w^2 + u^T h_\theta}$
    update Q_θ via $Q_\theta = Q_\theta + \frac{1}{\sigma_w^2} u u^T$
    update $g_\theta = g_\theta - \frac{1}{2}\log(1 + \frac{1}{\sigma_w^2} u^T h)$
  end for
end for
for $\theta \in \Theta$ **do**
  compute $\mu_\theta = Q_\theta^{-1} b$
  compute the likelihood via

$$\log \pi(\theta|y_{1:t}) = c + g_\theta + \tfrac{1}{2} b^T \mu_\theta$$

end for
compute the posterior distribution via

$$\pi(\theta|y_{1:t}) \propto \pi(y_{1:t}|\theta)\pi(\theta)$$

compute the predictive mean via

$$\mu_{x_i|y_{1:t}} = \textstyle\sum_\ell (\mu_{\theta_\ell})_i \pi(\theta_\ell|y_{1:t})$$

compute the predictive variance via

$$\sigma^2_{x_i|y_{1:t}} = \textstyle\sum_\ell \left((\mathrm{diag}(Q_{\theta_\ell}))_i + ((\mu_{\theta_\ell})_i - \mu_{x_i|y_{1:t}})^2 \right) \pi(\theta_\ell|y_{1:t})$$

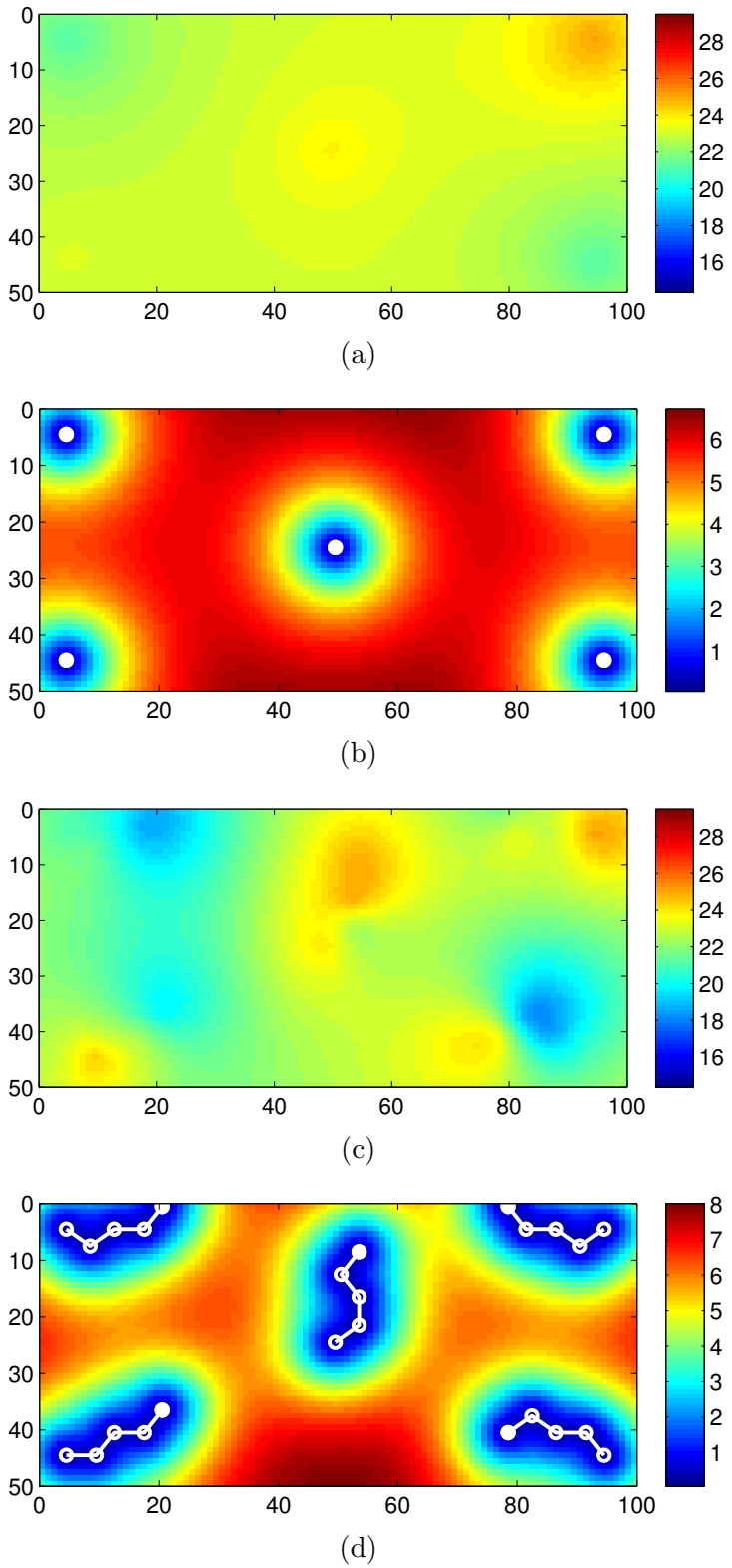
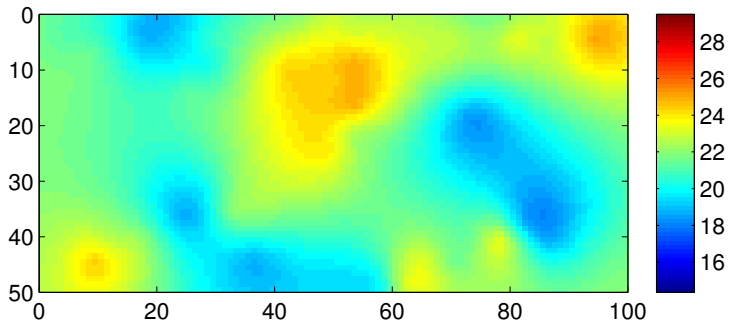
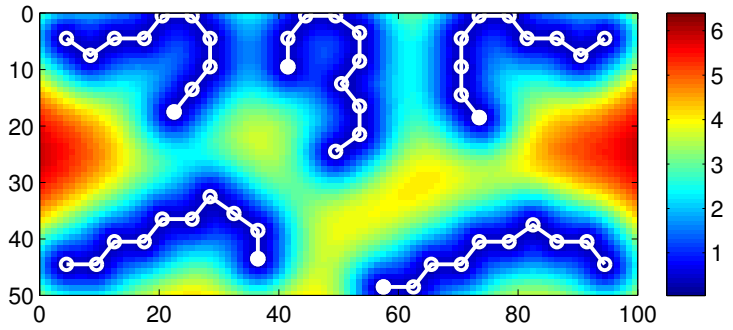


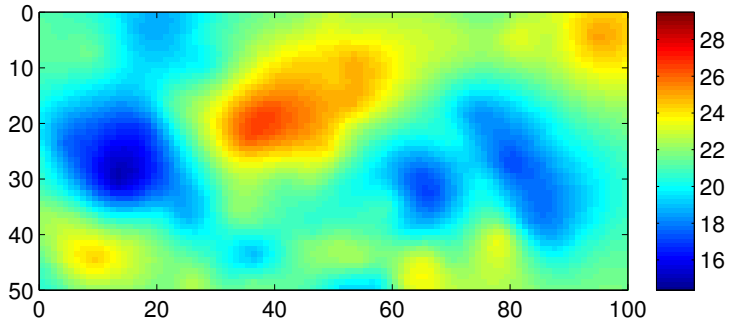
Figure 7.4: Predicted fields at (a) $t = 1$, (c) $t = 5$, (e) $t = 10$, and (g) $t = 20$. Prediction error variances at (b) $t = 1$, (d) $t = 5$, (f) $t = 10$, and (h) $t = 20$.



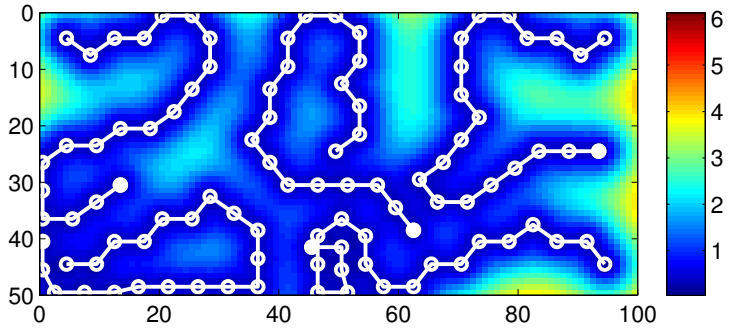
(e)



(f)

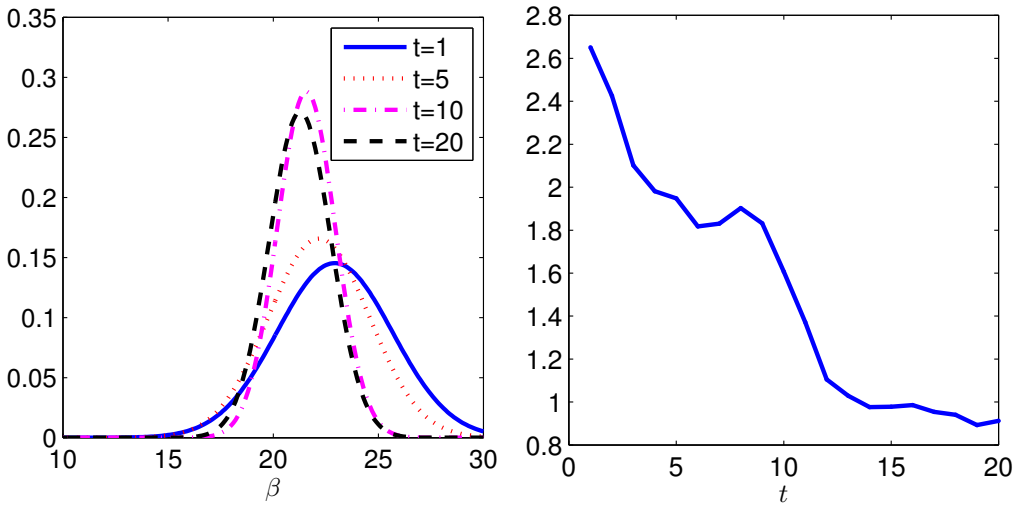


(g)



(h)

Figure 7.4: Predicted fields at (a) $t = 1$, (c) $t = 5$, (e) $t = 10$, and (g) $t = 20$. Prediction error variances at (b) $t = 1$, (d) $t = 5$, (f) $t = 10$, and (h) $t = 20$ (cont'd).



(a) (b)

Figure 7.5: (a) Estimated β, and (b) root mean square error.

Chapter 8

Conclusion and Future Work

In this chapter, we briefly summarize the key contributions presented in this dissertation and propose some promising directions for future work.

8.1 Conclusion

In Chapter 3, we presented a novel class of self-organizing sensing agents that learn an anisotropic, spatio-temporal Gaussian process using noisy measurements and move in order to improve the quality of the estimated covariance function. The ML estimator was used to estimate the hyperparameters in the unknown covariance function and the prediction of the field of interest was obtained based on the ML estimates. An optimal navigation strategy was proposed to minimize the information-theoretic cost function of the Fisher Information Matrix for the estimated hyperparameters. The proposed scheme was applied to both a spatio-temporal Gaussian process and a true advection-diffusion field. Simulation study indicated the effectiveness of the proposed scheme and the adaptability to time-varying covariance functions.

In Chapter 4, for spatio-temporal Gaussian processes, we justified prediction based on truncated observations for mobile sensor networks. In particular, we presented a theoretical foundation of Gaussian processes with truncated observations. Centralized and distributed navigation strategies were proposed to minimize the average of prediction error variances at target points that can be arbitrarily chosen by a user. Simulation results demonstrated that mobile sensing agents under the distributed navigation strategy produce an emergent, collective behavior for communication connectivity, and are coordinated to improve the quality of the collective prediction capability.

In Chapter 5, we formulated a fully Bayesian approach for spatio-temporal Gaussian process regression under practical conditions. We designed sequential Bayesian prediction algorithms to compute exact predictive distributions in constant time as the number of observations increases. An adaptive sampling strategy was also provided to improve the quality of prediction. Simulation results showed the practical usefulness of the proposed theoretically-correct algorithms in the context of environmental monitoring by mobile sensor networks.

In Chapter 6, we introduced a new class of Gaussian processes with built-in GMRFs for modeling a wide range of environmental fields. The Gaussian process regression for the predictive statistics at any point of interest was provided and a sequential field prediction algorithm with fixed complexity was proposed to deal with sequentially sampled observations. For a special case with compactly supported weighting functions, we proposed a distributed field prediction algorithm in which the prediction can be computed via Jacobi over-relaxation algorithm and discrete-time average consensus.

In Chapter 7, we have discussed the problem of predicting a large scale spatial field

using successive noisy measurements obtained by a multi-agent system. We modeled the spatial field of interest using a GMRF and designed a sequential prediction algorithm for computing the exact predictive inference from a Bayesian point of view. The proposed algorithm is computationally efficient and scalable as the number of measurements increases. We also designed an adaptive sampling algorithm for agents to find the sub-optimal locations in order to minimize the prediction error and reduce the uncertainty in hyperparameters simultaneously.

8.2 Future Work

In the long term, we plan to expanding our current work and exploring on the following directions:

- consider the optimal sampling strategies for mobile sensing agents with complicated vehicle dynamics;
- consider the optimal coordination of the mobile senor network subject to energy constraints;
- develop the approximated Bayesian prediction algorithms for resource-constraint mobile robots, such as using integrated nested Laplace approximations;
- expand the work on spatial modeling using GMRF to deal with the more general spato-temporal process;
- consider the effects of localization error on the posterior predictive distribution;

- implement the developed algorithms in experiments using robotic boats under development.

Appendix A

Mathematical Background

A.1 Gaussian Identities

The multivariate Gaussian distribution of a random vector $x \in \mathbb{R}^n$ (i.e., $x \sim \mathcal{N}(\mu, \Sigma)$) has a joint probability density function (pdf) given by

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{-n/2}|\Sigma|^{-1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right),$$

where $\mu \in \mathbb{R}^n$ is the mean vector, and $\Sigma \in \mathbb{R}^{n\times n}$ is the covariance matrix.

Now, suppose x consists of two disjoint subsets x_a and x_b, i.e.,

$$x = \begin{bmatrix} x_a \\ \\ x_b \end{bmatrix}.$$

The corresponding mean vector μ and covariance matrix Σ can be written as

$$\mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix},$$

where $\Sigma_{ab} = \Sigma_{ba}^T$ due to the symmetry of Σ. Then, the marginal distribution of x_a is given by

$$x_a \sim \mathcal{N}(\mu_a, \Sigma_{aa}),$$

and the conditional distribution of x_a given x_b is given by

$$x_a | x_b \sim \mathcal{N}(\mu_{a|b}, \Sigma_{a|b}),$$

where

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b)$$
$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}.$$

A.2 Matrix Inversion Lemma

Matrices can be inverted blockwise by using the following analytic inversion formula:

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(C - B^TA^{-1}B)^{-1}B^TA^{-1} & -A^{-1}B(C - B^TA^{-1}B)^{-1} \\ -(C - B^TA^{-1}B)^{-1}B^TA^{-1} & (C - B^TA^{-1}B)^{-1} \end{bmatrix},$$

where A, B and C are matrix sub-blocks of arbitrary size. Matrices A and $C - B^TA^{-1}B$ must be non-singular.

A.2.1 Woodbury identity

The Woodbury matrix identity is

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1}+VA^{-1}U\right)^{-1}VA^{-1},$$

where A, U, C and V denote matrices with appropriate size.

A.2.2 Sherman-Morrison formula

Suppose $A \in \mathbb{R}^{n\times n}$ is invertible and $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$ are vectors. Assume that $1+v^TA^{-1}u \neq 0$, the Sherman-Morrison formular states that

$$(A+uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}.$$

A.3 Generating Gaussian processes

In order to implement algorithms in simulation studies, we need to generate multivariate Gaussian samples from $\mathcal{N}(\mu,\Sigma)$ with arbitrary mean μ and covariance matrix Σ. In what follows, we introduce two approaches.

A.3.1 Cholesky decomposition

Given an arbitrary mean μ and a positive definite covariance matrix Σ, the algorithm generates multivariate Gaussian samples is shown in Tab. A.1.

Table A.1: Generating multivariate Gaussian samples by Cholesky decomposition.

1: compute the Cholesky decomposition of the positive definite symmetric covariance matrix $\Sigma = LL^T$, where L is a lower triangular matrix
2: generate $u \sim \mathcal{N}(0, I)$ by multiple separate calls to the scalar Gaussian generator
3: compute $x = \mu + Lu$ which has desired normal distributed with mean μ and covariance matrix $L\,\mathrm{E}[uu^T]L^T = LL^T = \Sigma$

A.3.2 Circulant embedding

Consider a 1-D zero-mean stationary Gaussian process $z(x)$ with a covariance function $C(x, x')$. The covariance matrix Σ of $z(x)$ sampled on the equispaced grids $\Omega = \left\{x^{(1)}, \cdots, x^{(n)}\right\}$ has entries $(\Sigma)_{pq} = C(|x^{(p)} - x^{(q)}|)$. Notice that the covariance matrix Σ is a positive semi-definite symmetric Toeplitz matrix which can be characterized by its first row $r = \mathrm{row}_1(\Sigma)$.

The key idea behind circulant embedding method is to construct a circulant matrix S that contains Σ as its upper-left submatrix. The reason for seeking a circulant embedding is the fact that, being a $m \times m$ circulant matrix, S has an eigendecomposition $S = (1/m)F\Lambda F^H$, where F is the standard FFT matrix of size m with entries $(F)_{pq} = \exp(2\pi ipq/m)$, F^H is the conjugate transpose of F, and Λ is a diagonal matrix whose diagonal entries form the vector $\tilde{s} = Fs$ (s is the first row of S).

Given a positive semi-definite circulant extension S of Σ, the algorithm generates the realization of $z(x)$ sampled on Ω is shown in Tab. A.2. Extension to multidimensional cases can be found in [20].

Table A.2: Generating multivariate Gaussian samples by circulant embedding.

1: compute via the FFT the discrete Fourier transform of $\tilde{s} = Fs$ and form the vector $(\tilde{s}/m)^{1/2}$
2: generate a vector $\epsilon = \epsilon_1 + i\epsilon_2$ of dimension m with $\epsilon_1 \sim \mathcal{N}(0, I)$ and $\epsilon_2 \sim \mathcal{N}(0, I)$ being independent and real random variables
3: compute a vector $\tilde{e} = \epsilon \circ (\tilde{s}/m)^{1/2}$
4: compute via FFT the discrete Fourier transform $e = F\tilde{e}$. The real and imaginary parts of the first n entries in e yield two independent realizations of $z(x)$ on Ω

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[3] D. P. Bertsekas, W. W. Hager, and O. L. Mangasarian. *Nonlinear programming*. Athena Scientific, Belmont, Mass., 1999.

[4] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice Hall, Englewood Cliffs, 1999.

[5] C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006.

[6] F. Bullo, J. Cortés, and S. Martínez. *Distributed control of robotic networks*. Applied Mathematics Series. Princeton University Press, 2009.

[7] C. G. Cassandras and W. Li. Sensor networks and cooperative control. *European Journal of Control*, 11(4-5):436–463, 2005.

[8] J. Choi, J. Lee, and S. Oh. Biologically-inspired navigation strategies for swarm intelligence using spatial gaussian processes. In *Proceedings of the 17th International Federation of Automatic Control (IFAC) World Congress*, 2008.

[9] J. Choi, J. Lee, and S. Oh. Swarm intelligence for achieving the global maximum using spatio-temporal gaussian processes. In *Proceedings of the 27th American Control Conference (ACC)*, 2008.

[10] J. Choi, S. Oh, and R. Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45:2802–2814, 2009.

[11] V. N. Christopoulos and S. Roumeliotis. Adaptive sensing for instantaneous gas release parameter estimation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4450–4456, 2005.

[12] Microsoft Corporation. Official website of Kinect for Xbox 360. http://www.xbox.com/en-US/kinect.

[13] J. Cortés. Distributed kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827, 2009.

[14] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, Lnc, Hoboken, New Jersey, 2nd edition, 2006.

[15] N. Cressie. Kriging nonstationary data. *Journal of the American Statistical Association*, 81(395):625–634, 1986.

[16] N. Cressie. *Statistics for spatial data*. A Wiley-Interscience Publication, John Wiley and Sons, Inc., 1991.

[17] N. Cressie and N. Verzelen. Conditional-mean least-squares fitting of Gaussian Markov random fields to Gaussian fields. *Computational Statistics & Data Analysis*, 52(5):2794–2807, 2008.

[18] N Cressie and C K Wikle. Space-time kalman filter. *Encyclopedia of Environmetrics*, 4:2045–2049, 2002.

[19] L. Devroye. *Non-uniform random variate generation*. Springer-Verlag, New York, 1986.

[20] C. R. Dietrich and G. N. Newsam. Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, 1997.

[21] A. F. Emery and A. V. Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864–76, 1998.

[22] M. Gaudard, M. Karson, E. Linder, and D. Sinha. Bayesian spatial prediction. *Environmental and Ecological Statistics*, 6(2):147–171, 1999.

[23] M. Gibbs and D. J. C. MacKay. Efficient implementation of gaussian processes. *Available electronically from http://www. cs. toronto. edu/mackay/gpros. ps. gz. Preprint*, 1997.

[24] T. Gneiting. Compactly supported correlation functions. *Journal of Multivariate Analysis*, 83(2):493–508, 2002.

[25] R. Graham and J. Cortés. Cooperative adaptive sampling of random fields with partially known covariance. *International Journal of Robust and Nonlinear Control*, 1:1–2, 2009.

[26] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006.

[27] L. Hartman and O. Hössjer. Fast kriging of large data sets with Gaussian Markov random fields. *Computational Statistics & Data Analysis*, 52(5):2331–2349, 2008.

[28] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[29] P. Kathirgamanathan, R. McKibbin, and R. I. McLachlan. Source term estimation of pollution from an instantaneous point source. *Research Letters in the Information and Mathmatical Sciences*, 3(1):59–67, 2002.

[30] S. M. Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice Hall, Inc., 1993.

[31] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 2–10, 2006.

[32] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[33] N Lawrence, M Seeger, and R Herbrich. Fast sparse gaussian process methods: The informative vector machine. *Advances in neural information ...*, Jan 2003.

[34] J. Le Ny and G.J. Pappas. On trajectory optimization for active sensing in Gaussian process models. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese*

Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, pages 6286–6292, 2010.

[35] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R.E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, January 2007.

[36] F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B*, 73(4):423–498, 2011.

[37] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724, June 2008.

[38] D. J. C. MacKay. Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–165, 1998.

[39] M. Mandic and E. Franzzoli. Efficient sensor coverage for acoustic localization. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 3597–3602, Dec 2007.

[40] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso. The Kriged Kalman filter. *Test*, 7(2):217–282, 1998.

[41] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[42] G. M. Mathews, H. Durrant-Whyte, and M. Prokopenko. Decentralised decision making in heterogeneous teams using anonymous optimisation. *Robotics and Autonomous Systems*, 57(3):310–320, 2009.

[43] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 1999.

[44] D. J. Nott and W. T. M. Dunsmuir. Estimation of nonstationary spatial covariance structure. *Biometrika*, 89(4):819–829, 2002.

[45] S. Oh, Y. Xu, and J. Choi. Explorative navigation of mobile sensor networks using sparse Gaussian processes. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, 2010.

[46] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[47] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.

[48] R. Olfati-Saber, R. Franco, E. Frazzoli, and J. S. Shamma. Belief consensus and distributed hypothesis testing in sensor networks. *Networked Embedded Sensing and Control*, pages 169–182, 2006.

[49] F. Pukelsheimi. *Optimal design of experiments*. New York: Wiley, 1993.

[50] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[51] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, Cambridge, Massachusetts, London, England, 2006.

[52] W. Ren and R. W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005.

[53] W. Rudin. *Principles of mathematical analysis*. McGraw-Hill New York, 1976.

[54] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. Chapman & Hall, 2005.

[55] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009.

[56] H. Rue and H. Tjelmeland. Fitting Gaussian Markov random fields to Gaussian fields. *Scandinavian Journal of Statistics*, 29(1):31–49, 2002.

[57] M. Seeger. *Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations*. PhD thesis, School of Informatics, University of Edinburgh, 2003.

[58] A Singh, A Krause, C Guestrin, and W Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34(1):707–755, 2009.

[59] A. J. Smola and P. Bartlett. Sparse greedy gaussian process regression. In *Advances in Neural Information Processing Systems 13*, 2001.

[60] P. Sollich and A. Halees. Learning curves for Gaussian process regression: approximations and bounds. *Neural Computation*, 14(6):1393–1428, 2002.

[61] V. Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

[62] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8:514–520, 1996.

[63] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, 2001.

[64] C .K. I. Williams and F. Vivarelli. Upper and lower bounds on the learning curve for Gaussian processes. *Machine Learning*, 40(1):77–102, 2000.

[65] Y. Xu and J. Choi. Adaptive sampling for learning Gaussian processes using mobile sensor networks. *Sensors*, 11(3):3051–3066, 2011.

[66] Y. Xu, J. Choi, S. Dass, and T. Maiti. Bayesian prediction and adaptive sampling algorithms for mobile sensor networks. In *Proceedings of the 2011 American Control Conference (ACC)*, pages 4095–4200, 2011.

[67] Y. Xu, J. Choi, and S. Oh. Mobile sensor setwork navigation using Gaussian processes with truncated observations. *IEEE Transactions on on Robotics*, 27(5):1–14, 2011. to appear.

[68] M. M. Zavlanos and G. J. Pappas. Distributed connectivity control of mobile networks. *IEEE Transactions on Robotics*, 24(6):1416–1428, December 2008.

[69] M. M. Zavlanos and G. J. Pappas. Dynamic assignment in distributed motion planning with local coordination. *IEEE Transactions on Robotics*, 24(1):232–242, 2008.