GAUSSIAN PROCESS-BASED DECENTRALIZED DATA FUSION AND ACTIVE SENSING AGENTS:

Towards Large-Scale Modeling and Prediction of Spatiotemporal Traffic Phenomena

CHEN JIE

NATIONAL UNIVERSITY OF SINGAPORE

2013

GAUSSIAN PROCESS-BASED DECENTRALIZED DATA FUSION AND ACTIVE SENSING AGENTS:

Towards Large-Scale Modeling and Prediction of Spatiotemporal Traffic Phenomena

CHEN JIE
(M.Eng, Zhejiang University)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2013

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Chen Jie
16 August 2013

ACKNOWLEDGEMENTS

I appreciate and thank both my advisors Dr. Bryan Kian Hsiang Low and Dr. Colin Keng-Yan Tan for the support, guidance, and advice throughout my PhD candidature.

I am thankful to all friends from MapleCG group. My research benefited a lot from the discussions with you.

I thank my colleague Cao Nannan for helping me in the implementation of parallel Gaussian process together.

Many thanks to Professor Patrick Jaillet (MIT), Professor Lee Wee Sun (NUS), Professor Leong Tze Yun (NUS), Professor Tan Chew Lim (NUS), Professor David Hsu (NUS) and Professor Geoff Hollinger (OSU) for providing invaluable feedbacks that improved my work.

I acknowledge Future Urban Mobility (FM) research group of Singapore-MIT Alliance for Research and Technology (SMART) for sharing the high quality datasets and funding my research[1].

I appreciate School of Computing, National University of Singapore for providing the facilities to run all my experiments.

Last, but not least, I would like to thank my wife Orange for the love, understanding, and support you gave me all these years. To my parents and family, thank you for the encouragement, concern, and care.

[1] Singapore-MIT Alliance Research and Technology (SMART) Subaward Agreement 14 R-252-000-466-592

PUBLICATIONS

Parts of the thesis have been published in

1. Parallel Gaussian Process Regression with Low-Rank Covariance Matrix Approximations. Jie Chen, Nannan Cao, Kian Hsiang Low, Ruofei Ouyang, Colin Keng-Yan Tan & Patrick Jaillet. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence* (UAI-13), pages 152-161, Bellevue, WA, Jul 11-15, 2013.

2. Gaussian Process-Based Decentralized Data Fusion and Active Sensing for Mobility-on-Demand System. Jie Chen, Kian Hsiang Low, & Colin Keng-Yan Tan. In *Proceedings of the Robotics: Science and Systems* (RSS-13), Berlin, Germany, Jun 24-28, 2013.

3. Decentralized Data Fusion and Active Sensing with Mobile Sensors for Modeling and Predicting Spatiotemporal Traffic Phenomena. Jie Chen, Kian Hsiang Low, Colin Keng-Yan Tan, Ali Oran, Patrick Jaillet, John M. Dolan & Gaurav S. Sukhatme. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence* (UAI-12), pages 163-173, Catalina Island, CA, Aug 15-17, 2012.

The other published work during my course of study:

4. Decentralized Active Robotic Exploration and Mapping for Probabilistic Field Classification in Environmental Sensing. Kian Hsiang Low, Jie Chen, John M. Dolan, Steve Chien & David R. Thompson. In *Proceedings of the 11th International Conference on Autonomous Agents and MultiAgent Systems* (AAMAS-12), pages 105-112, Valencia, Spain, June 4-8, 2012.

Contents

Summary

Knowing and understanding the environmental phenomena is important to many real world applications. This thesis is devoted to study large-scale modeling and prediction of spatiotemporal environmental phenomena (i.e., urban traffic phenomena). Towards this goal, our proposed approaches rely on a class of Bayesian non-parametric models: *Gaussian processes* (GP).

To accurately model spatiotemporal urban traffic phenomena in real world situation, a novel relational GP taking into account both the road segment features and road network topology information is proposed to model real world traffic conditions over road network. Additionally, a GP variant called log-Gaussian process (ℓGP) is exploited to model an urban mobility demand pattern which contains skewness and extremity in demand measurements.

To achieve efficient and scalable urban traffic phenomenon prediction given a large phenomenon data, we propose three novel parallel GPs: *parallel partially independent training conditional* (pPITC), *parallel partially independent conditional*(pPIC) and *parallel incomplete Cholesky factorization* (pICF)-based approximations of GP model, which can distribute their computational load into a cluster of parallel/multi-core machines, thereby achieving time efficiency. The predictive performances of such parallel GPs are theoretically guaranteed to be equivalent to that of some centralized approaches to approximate full/exact GP regression. The proposed parallel GPs are implemented using the *message passing interface* (MPI) framework and tested on two large real world datasets. The theoretical and empirical results show that our parallel GPs achieve significantly

better time efficiency and scalability than that of full GP, while achieving comparable accuracy. They also achieve fine speedup performance that is the ratio of time required by the parallel algorithms and their centralized counterparts.

To exploit active mobile sensors to perform decentralized perception of the spatiotemporal urban traffic phenomenon, we propose a decentralized algorithm framework: *Gaussian process-based decentralized data fusion and active sensing* (D^{2}FAS) which is composed of a *decentralized data fusion* (DDF) component and a *decentralized active sensing* (DAS) component. The DDF component includes a novel *Gaussian process-based decentralized data fusion* (GP-DDF) algorithm that can achieve remarkably efficient and scalable prediction of phenomenon and a novel *Gaussian process-based decentralized data fusion with local augmentation* (GP-DDF$^+$) algorithm that can achieve better predictive accuracy while preserving time efficiency of GP-DDF. The predictive performances of both GP-DDF and GP-DDF$^+$ are theoretically guaranteed to be equivalent to that of some sophisticated centralized sparse approximations of exact/full GP. For the DAS component, we propose a novel *partially decentralized active sensing* (PDAS) algorithm that exploits property in correlation structure of GP-DDF to enable mobile sensors cooperatively gathering traffic phenomenon data along a near-optimal joint walk with theoretical guarantee, and a *fully decentralized active sensing* (FDAS) algorithm that guides each mobile sensor gather phenomenon data along its locally optimal walk.

Lastly, to justify the practicality of the D^{2}FAS framework, we develop and test D^{2}FAS algorithms running with active mobile sensors on real world datasets for monitoring traffic conditions and sensing/servicing urban mobility demands. Theoretical and empirical results show that the proposed algorithms are significantly more time-efficient, more scalable in the size of data and in the number of sensors than the state-of-the-art centralized approaches, while achieving comparable predictive accuracy.

List of Tables

List of Figures

List of Symbols

Abbreviations

D^{2}FAS	Gaussian process-based decentralized data fusion and active sensing
DAS	decentralized active sensing
FDAS	fully decentralized active sensing
PDAS	partially decentralized active sensing
DDF	decentralized data fusion
GP-DDF	Gaussian process-based decentralized data fusion
GP-DDF$^+$	Gaussian process-based decentralized data fusion with local augmentation
GP	Gaussian process
ℓGP	log-Gaussian process
FGP	full/exact Gaussian process
PITC	partially independent training conditional approximation of GP model
pPITC	parallel partially independent training conditional approximation of GP regression
PIC	partially independent conditional approximation of GP model

pPIC	parallel partially independent conditional approximation of GP regression
ICF	incomplete Cholesky factorization
pICF	parallel incomplete Cholesky factorization
SoD	subset of data approximation of GP
RMSE	root mean square error
KLD	Kullback-Leibler divergence
MoD	mobility-on-demand

Numbers

$\mathbb{R}$	set of all reals
$\mathbb{R}^+$	set of all positive reals
$\mathbb{R}^p$	p-dimensional Euclidean space
K	number of mobile agents
$\mathcal{K}$	number of connected components
κ	size of the largest connected component
M	number of parallel machines of a cluster
C	number of users in a MoD system
H	horizon of a planned walk
L	total length of an agent's walk
R	the reduced rank
ε	a user-defined constant

Data

$\mathcal{X}$	input domain
V	domain of road segments / regions
$\mathcal{D}$	a set of observed inputs
$\mathcal{U}$	a set of unobserved inputs
$\mathcal{S}$	a set of support inputs / a subset of observed inputs

$\mathcal{D}_k$	a set of observed inputs that is local to agent k
Y_s	random output variable of input s
y_s	realized output value (measurement) of input s
Z_s	log of random output variable of input s
z_s	log of realized output value (measurement) of input s
$p(p')$	dimension of inputs
r_i	range of i-th feature of inputs

Functions

$k(.,.)$	positive definite kernel function
$m(.)$	standardized Manhattan distance of an edge
$d(.,.)$	shortest path distance between two vertex
$g(.)$	mapping from domain of road segments to Euclidean space
$\tau(.)$	assignment function
$\log$	logarithm to base e
$\mathbb{H}[.]$	entropy of a probabilistic distribution
$\overline{\mathbb{H}}[.]$	approximation of Gaussian entropy
$\widetilde{\mathbb{H}}[.]$	approximation of log-Gaussian entropy
$\max$	maximum value of a function
$\arg\max$	argument of the maximum of a function
$\delta_{ss'}$	Kronecker delta

Vectors or Matrices

$\mathbf{1}$	vector with all entries equal to one
I	identity matrix
A^T	transpose of matrix A
A^{-1}	inverse of matrix A
$[\cdot]_i$	the i-th element of a vector
$[.]_{i,j}$	the element in row i and column j of a matrix

$\lvert . \rvert$	number of elements of a vector / determinant of a matrix

Gaussian Process

$\mathcal{N}(.,.)$	a Gaussian distribution
$\mathbb{E}[.]$	prior mean of a random variable
$\mathrm{cov}[.,.]$	covariance function
σ_s	signal variance
σ_n	noise variance
ℓ_i	characteristic length-scale of i-th feature of inputs
$\sigma_{ss'}$	covariance
$\mathcal{N}(\mu_{\mathcal{U}\mid\mathcal{D}}, \Sigma_{\mathcal{U}\mathcal{U}\mid\mathcal{D}})$	posterior distribution of a full/exact GP
$\mathcal{N}(\mu_{\mathcal{U}\mid\mathcal{S}}, \Sigma_{\mathcal{U}\mathcal{U}\mid\mathcal{S}})$	posterior distribution of SoD approximation of GP
$\mathcal{N}(\mu^{\text{PITC}}_{\mathcal{U}\mid\mathcal{D}}, \Sigma^{\text{PITC}}_{\mathcal{U}\mathcal{U}\mid\mathcal{D}})$	posterior distribution of a PITC approximation of GP model
$\mathcal{N}(\mu^{\text{PIC}}_{\mathcal{U}\mid\mathcal{D}}, \Sigma^{\text{PIC}}_{\mathcal{U}\mathcal{U}\mid\mathcal{D}})$	posterior distribution of a PIC approximation of GP model
$\mathcal{N}(\mu^{\text{ICF}}_{\mathcal{U}\mid\mathcal{D}}, \Sigma^{\text{ICF}}_{\mathcal{U}\mathcal{U}\mid\mathcal{D}})$	posterior distribution of a PIC approximation of GP model
$\mu^{\ell\text{GP}}_{s\mid\mathcal{D}}$	ℓGP posterior mean

Decentralized Perception

G	graph representing road network or service area
E	edges of graph G
V	vertex of graph G
w_k	walk of agent k
W_k	set of all possible walk of agent k
w^*_k	optimal walk of agent k
w	joint walk
w^*	optimal joint walk
$\widehat{w}$	optimal joint walk obtained from PDAS
$\mathcal{U}_w$	set of inputs induced by walk w
$\mathcal{G}$	coordination graph

$\mathcal{V}$	vertex of $\mathcal{G}$ representing agents
$\mathcal{E}$	edges of $\mathcal{G}$ representing coordination dependencies among agents
$\mathcal{J}_k$	adjacency between agents
a_k	adjacency vector
$A_{\mathcal{G}}$	adjacency matrix representing coordination graph
P_c	fleet distribution
P_d	historic demand distribution
$\mathcal{N}(\widehat{\mu}_{\mathcal{U}}, \widehat{\Sigma}_{\mathcal{U}\mathcal{U}})$	predictive distribution of GP-DDF / pPITC
$\mathcal{N}(\widehat{\mu}^{+}_{\mathcal{U}}, \widehat{\Sigma}^{+}_{\mathcal{U}\mathcal{U}})$	predictive distribution of GP-DDF^{+} / pPIC
$(\dot{y}^{k}_{\mathcal{S}}, \dot{\Sigma}^{k}_{\mathcal{S}\mathcal{S}})$	local summary of agent k
$(\ddot{y}_{\mathcal{S}}, \ddot{\Sigma}_{\mathcal{S}\mathcal{S}})$	global summary in pPITC/pPIC/GP-DDF/GP-DDF^{+}
$\mathcal{N}(\widetilde{\mu}_{\mathcal{U}}, \widetilde{\Sigma}_{\mathcal{U}\mathcal{U}})$	predictive distribution of pICF-based GP
$(\dot{y}_m, \dot{\Sigma}_m, \Phi_m)$	local summary of machine m in pICF-based GP
$(\ddot{y}, \ddot{\Sigma})$	global summary in pICF-based GP
F	upper triangular incomplete Cholesky factor

Chapter 1

Introduction

1.1 Motivation

Our modern world faces global issues such as non-renewable energy resources depletion, human population explosion, and ecological environmental degradation. Confronted by these issues, in the Millennium Campaign [UNS, 2010], the United Nations called for the worldwide effort in reversing the loss of natural resources and reducing the loss of biodiversity to ensure environmental sustainability. Crucial to achieving this ambitious goal is the need to study, analyze and understand the environmental phenomena spatiotemporally distributed over our urban cities and natural habitats, such as

i. Urban Traffic Phenomena Sensing: The traffic phenomena such as traffic speeds and volumes [Min and Wynter, 2011], travel time along road segments [Hofleitner *et al.*, 2012a; Herring *et al.*, 2010], congestion patterns [Hofleitner *et al.*, 2012b], or travel demand [Powell *et al.*, 2011] are studied in urban transportation domain (Figures 6.1 & 7.1 illustrate real-world examples of traffic speeds over road networks and mobility demand patterns, respectively). Knowing and using these phenomena at network level or user level, drivers can reduce the time wasted (e.g., waiting time during congestion, cruising time of taxicabs seeking customers) on traffic network, and consequently reduce the wastage of fossil fuel and emission of air pollutants.

ii. Natural Phenomena Sensing: The natural phenomena such as the ocean and fresh water phenomena (e.g., plankton bloom, anoxic zones, temperature, salinity) [Low *et al.*, 2012; Low *et al.*, 2009c; Podnar *et al.*, 2010; Dolan *et al.*, 2009], forest ecosystems, rare species, pollution, or contamination are monitored by environmental sensing applications. These environmental phenomena can be used to predict thresholds and indicators that detect unsustainable situation endangering ecosystems [Srebotnjak *et al.*, 2010].

This research will focus on the urban traffic phenomena sensing. We believe our work would be more promising in urban traffic domains as the traditional solutions to urban traffic are becoming unsustainable in increasingly denser populated urban cities. For example, Hong Kong and Singapore have, respectively, experienced 27.6% and 37% increase in private vehicles from 2003 to 2011 [RPT, 2012]. However, their road networks have only expanded less than 10% in size. Without implementing sustainable measures, traffic congestions and delays will grow more severe and frequent, especially during peak hours. According to a 2011 urban mobility report [Schrank *et al.*, 2011], the traffic congestions in the USA have caused 1.9 billion gallons of extra fuel, 4.8 billion hours of travel delay, and $\$101$ billion of delay and fuel cost. Such huge resource wastage can be potentially mitigated if the spatiotemporally varying traffic phenomena (e.g., speeds and travel times along road segments, mobility demand in a region) are predicted accurately enough in real time to detect and forecast the congestion hotspots; network-level (e.g., ramp metering, road pricing) and user-level (e.g., route replanning, on-demand mobility servicing) measures can then be taken to relieve these congestions, so as to improve the overall efficiency of road networks. In addition, a large quantity of *in situ* high-resolution (meter-level) urban traffic data[1] is available, which is valuable to justify the practicality of our work. Moreover, the proposed techniques can also be applied to natural phenomena sensing where the model has to be modified to represent phenomena with respect to geographic locations and time.

The urban traffic phenomena are spatiotemporally varying (e.g., traffic conditions over road networks can vary between peak business hour and off-peak hour, and vary between central business district and residency district at certain time) and happening in large-scale domain (Figure 1.1 illustrates the road network of Singapore). To accurately understand such large-scale spatiotemporal urban traffic phenomena, the sensors deployed to collect phenomena data tend

[1] The traffic flow & taxicabs trajectory datasets collected from Singapore road network are supported by future urban mobility (FM) research group of Singapore-MIT Alliance for Research and Technology (SMART).

to be in large number which is proportional to the domain size. Moreover, the proliferation of the use of static and mobile sensors within urban city enables a large traffic phenomena data to be gathered over space and time. Such large phenomena data can be exploited to understand the large-scale spatiotemporally varying urban traffic phenomena.

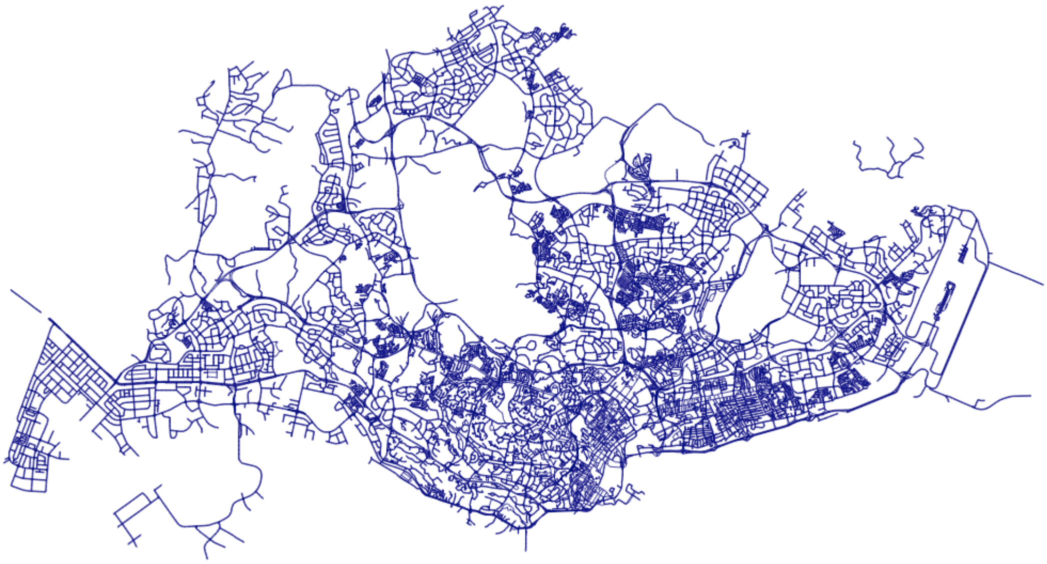


Figure 1.1: The road network of Singapore with a large number 57848 of road segments.

1.2 Objectives

1.2.1 Accurate Traffic Modeling and Prediction

Towards understanding the spatiotemporally varying urban traffic phenomena (e.g., traffic speeds or mobility demand patterns), the first question to ask is

Question one: How can a model be built to accurately represent and predict a spatiotemporal traffic phenomena within real-world situation?

To address this question, the modeling approach should be capable of representing and capturing the properties and characteristics (e.g., complex correlation structure over road networks, or extremity and skewness in measurements) of urban traffic phenomena. Existing methods (Section 2.1) failed to account for both segment features and network topology in traffic phenomena modeling . In this thesis, we investigate a class of data-driven models which can exploit the phenomena data for flexibly modeling and predicting spatiotemporal phenomena.

1.2.2 Efficiency and Scalability

Time efficiency and scalability are important factors for practical employment of a proposed model. With a large traffic phenomena data available, the next question to ask is

Question two: How can a model be built to achieve real-time and scalable prediction on the unobserved area given a large observations?

The key to addressing the above question is to alleviate the high computation overheads caused by a large phenomena data. To achieve this goal, this thesis explore along two directions: exploiting more computing resources (parallel/multi-core machines) or using a smaller, more informative phenomena data; the former direction requires parallel/decentralized techniques to speed up learning the model and the latter direction needs active sensing techniques to only collect data that matters. The existing literatures pertaining to these two directions are discussed in Section 2.2 and Section 2.4, respectively.

1.2.3 Decentralized Perception

In practice, it is advantageous to exploit active mobile sensors to gather traffic phenomena data (e.g., traffic speeds over road networks). Traditionally, static sensors such as loop detectors [Krause *et al.*, 2008a; Wang and Papageorgiou, 2005] are placed at designated locations in a road network to collect data for predicting the traffic phenomena. However, they provide sparse coverage (i.e., many road segments are not observed, thus leading to data sparsity), incur high installation and maintenance costs, and cannot reposition by themselves in response to changes in the traffic phenomena. Low-cost GPS technology allows the collection of traffic data using passive mobile probes [Work *et al.*, 2010] (e.g., taxis/cabs). Unlike static sensors, they can directly measure the travel times along road segments. But, they provide fairly sparse coverage due to low GPS sampling frequency (i.e., often imposed by taxi/cab companies) and no control over their routes. In addition, they also incur high initial implementation cost, pose privacy issues, and produce highly-varying speeds and travel times while traversing the same road segment due to inconsistent driving behaviors. A critical mass of probes is needed on each road segment to ease the severity of the last drawback [Srinivasan and Jovanis, 1996] but is often hard to achieve on non-highway segments due to sparse coverage. In contrast, we proposed the use of active mobile probes[2] [Turner *et al.*, 1998] to overcome the limitations of static and passive mobile probes. In particular, they can be directed to explore any segment of a road network to gather traffic data at a desired GPS sampling rate while enforcing consistent driving behavior.

Towards understanding the spatiotemporal traffic phenomena with active mobile sensors, the third question to ask is

Question three: How do the mobile sensors actively explore an urban network to gather and assimilate the most informative phenomenon data for predicting a spatiotemporal traffic phenomenon?

We can gain some perspectives from addressing the previous two questions. First, mobile sensors can also exploit phenomena data to model and predict the spatiotemporal traffic phenomena. Second, as each mobile sensor stores some

[2] In this thesis, mobile probes, mobile sensors and vehicles will be used interchangeable as they are essentially mobile agents with capability of actively collecting traffic phenomena data.

local phenomena data and has certain (usually not so high) computing power, the parallel/decentralized techniques can be adapted for mobile sensors to cooperatively assimilate the phenomena data to predict the traffic phenomena. Third, since each individually mobile sensor can actively explore the traffic network and decide which phenomena data to gather, then distributed active sensing techniques are needed to coordinate the mobile sensors ensuring the most "informative" phenomena data is gathered. The related literatures are discussed in Sections 2.3 & 2.4.

1.3 Contributions

Towards large-scale modeling and prediction of spatiotemporal traffic phenomena, the contributions of this thesis address three research questions raised in previous section.

1.3.1 Accurate Traffic Modeling and Prediction

Answering question one, the spatiotemporal traffic phenomena modeling relies on a class of Bayesian non-parametric (data-driven) models: *Gaussian Processes* (GP) described in Section 3.1. Based on GP, a novel relational GP model [Chen *et al.*, 2012] is proposed to model real world traffic conditions over road network. The correlation structure of such relation GP model takes into account both the road segment features and road network topology information (Section 3.3).

1.3.2 Efficiency and Scalability

Along the first direction of question two, which aims to exploit parallel/multicore machines to achieve real-time prediction given a large phenomena data, this thesis presents three novel parallel GPs: *parallel partially independent training conditional* (pPITC), *parallel partially independent conditional*(pPIC) and *parallel incomplete Cholesky factorization* (pICF)-based approximations of GP model [Chen *et al.*, 2013a]. The predictive performances of these parallel GPs are theoretically guaranteed to be equivalent to that of some centralized approaches to approximate GP regression (Sections 4.1 & 4.2). By analytically comparing the time, space, and communication complexity of the proposed parallel GPs, it is showed that the parallel GPs improves the scalability of their centralized counterparts (Section 4.3). Furthermore, the proposed parallel GPs are implemented using the *message passing interface* (MPI) framework to run in a cluster of 20 computing nodes, and their performances (i.e., predictive accuracy, time efficiency, scalability, and speedups) are empirically evaluated on two large real-world datasets (Section 4.4). The results show that our parallel GPs achieve significantly better time efficiency than that of full GP while achieving comparable accuracy; the parallel GPs also achieve fine speedups to their centralized counterparts (Section 4.5).

1.3.3 Decentralized Perception

The second direction of question two is investigated together with question three in the context of traffic phenomena sensing with active mobile sensors. Here, we propose a decentralized algorithm framework [Chen *et al.*, 2012; Chen *et al.*, 2013b]: *Gaussian process-based decentralized data fusion and active sensing* (D^2FAS) which is composed of a *decentralized data fusion* (DDF) component that can cooperatively assimilate the distributed traffic phenomena data into a globally consistent predictive model and a *decentralized active sensing* (DAS) component that can guide mobile sensors to cooperatively collect the most informative phenomena data.

The DDF component [Chen *et al.*, 2012; Chen *et al.*, 2013b] includes a novel *Gaussian process-based decentralized data fusion* (GP-DDF) algorithm (Section 5.1.1) that can achieve remarkably efficient and scalable prediction of phenomena and a novel *Gaussian process-based decentralized data fusion with local augmentation* (GP-DDF$^+$) algorithm (Section 5.1.2) that can achieve better predictive accuracy while preserving time efficiency of GP-DDF. The predictive performances of both GP-DDF and GP-DDF$^+$ are theoretically guaranteed to be equivalent to that of some sophisticated centralized sparse approximations of exact/full GP.

For the DAS component [Chen *et al.*, 2012; Chen *et al.*, 2013b], we first propose a novel *partially decentralized active sensing* (PDAS) algorithm which exploits property in correlation structure of GP-DDF to enable mobile sensors cooperatively selecting a joint walk of approximated maximum posterior Gaussian entropy. The performance of PDAS is theoretically guaranteed, and various practical environment conditions can be established to ensure it be comparably well (Section 5.2.3). To alleviate the issue that PDAS algorithm cannot perform or perform poorly (in terms of time) in certain situations, a *fully decentralized active sensing* (FDAS) algorithm is proposed to make each mobile sensor gather phenomena data along its locally optimal walk (Section 5.2.4).

Lastly, the practicality of D^{2}FAS framework is justified in two real-world applications: traffic condition monitoring [Chen *et al.*, 2012] (Chapter 6) and mobility-on-demand systems [Chen *et al.*, 2013b] (Chapter 7). We propose D^{2}FAS algorithms running with active mobile sensors for monitoring traffic conditions (Section 6.2) and sensing/servicing urban mobility demands (Section 7.2), respectively. By analysing the time and communication overheads of these D^{2}FAS algorithms, it is showed that the D^{2}FAS algorithms scale better with a large phenomena data and number of sensors than state-of-the-art centralized approaches (Section 6.2 & 7.2). Then, we simulate the D^{2}FAS algorithms on two real-world datasets (Sections 6.3 & 7.3) and empirically evaluate their performance; the results show that the proposed algorithms are significantly more time-efficient, more scalable in the size of data and number of sensors than the state-of-the-art centralized approaches, while achieving comparable predictive accuracy (Sections 6.4 & 7.4). Therefore, the proposed D^{2}FAS framework is of significant value in practical deployment of active mobile sensors to monitor traffic conditions over road networks and to sense/service urban mobility demands.

Chapter 2

Related Works

This chapter reviews existing literatures related to the three research questions raised in Section 1.2. First, Section 2.1 investigates modeling approaches in terms of the capability of accounting for properties and characteristics (e.g., space, time, road features, and road network topology etc.) pertaining to urban traffic phenomena, and the capability of quantifying predictive uncertainty. Second, Section 2.1 reviews the techniques (i.e. approximation and parallelize computation) of scaling up the GP model, which are related to the purpose of achieving efficient and scalable prediction of traffic phenomena. As active mobile sensors are exploited to explore road networks and gather phenomenon data for prediction of the urban traffic phenomena, Section 2.3 discusses the related techniques of assimilating distributed data into predictive models and Section 2.4 focuses on the active sensing strategies that can guide mobile agents to collect the most informative data. Decentralization for both kinds of techniques is also tightly related when a large size of mobile sensors are involved.

2.1 Spatiotemporal Phenomena Modeling

The spatiotemporal correlation structure of a traffic phenomenon can be exploited to predict the traffic conditions of any unobserved road segment at any time using the observations taken along the sensors paths. To achieve this, existing Bayesian filtering frameworks [Chen *et al.*, 2011; Wang and Papageorgiou, 2005; Work *et al.*, 2010] utilize various handcrafted parametric models to predict traffic flow along a highway stretch that only correlates adjacent segments of the highway. As such, their predictive performance will be compromised when the current observations are sparse and/or the actual spatial correlation spans multiple segments. Their strong Markov assumption further exacerbates this problem. It is also not shown how these models can be generalized to work for arbitrary road network topologies and more complex correlation structures. On the other hand, existing multivariate parametric traffic prediction models [Kamarianakis and Prastacos, 2003; Min and Wynter, 2011] do not quantify uncertainty estimates of the predictions and impose rigid spatial locality assumptions that do not adapt to the true underlying correlation structures.

In contrast, we assume the traffic phenomenon over an urban road network (i.e., comprising full range of road types like highways, arterials, slip roads, etc.) can be be realized from a rich class of Bayesian non-parametric models called the *Gaussian process* (GP) (Section 3.1) that can formally characterize its spatiotemporal correlation structure and be refined with a growing number of observations. The GP models have been used in modelling various complex phenomena, for example, ocean-geographic phenomena [Low *et al.*, 2012], large scale terrain [Vasudevan *et al.*, 2009], deformation cost of planning a robot trajectory in a deformable environment [Frank *et al.*, 2011], and surface of 3D structure for ship hull inspection [Hollinger *et al.*, 2012]. An important feature of GP is that it can provide formal measures of predictive uncertainty (e.g., based on variance or entropy criterion) for directing the sensors to explore highly uncertain areas of the road network. Krause et al. in [Krause *et al.*, 2008a] used GP to represent the traffic phenomenon over a network of only highways and defined the correlation of speeds between highway segments to depend only on the geodesic (i.e., shortest path) distance of these segments with respect to the network topology. However, the features of road segments are not considered. Neumann et al. in [Neumann *et al.*, 2009] maintained a mixture of two independent GPs for flow prediction such that the correlation structure of one GP utilized road segment features while that of the other GP depended on manually specified relations (instead of geodesic distance) between segments with respect to an undirected network topology. In other words, the existing works on GP failed to account for both types of information (segment features and network topology). To address the above limitations, we propose a relational GP (Section 3.3) whose correlation structure exploits the geodesic distance between segments based on the topology of a directed road network with vertices denoting road segments and edges indicating adjacent segments weighted by dissimilarity of their features, hence tightly integrating the features and relational information.

2.2 Scaling Up Gaussian Process

The exact/full GP prediction (Section 3.1) cannot be performed well in real time due to its cubic time complexity. To reduce the computational cost, two classes of approximate GP regression methods have been proposed: (a) Low-rank covariance matrix approximation methods [Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2005; Williams and Seeger, 2000] are especially suitable for modeling smoothly-varying functions with high correlation (i.e., long length-scales) and they utilize all the data for predictions like the exact/full GP; and (b) localized regression methods (e.g., local GPs [Das and Srivastava, 2010; Choudhury *et al.*, 2002; Park *et al.*, 2011] and compactly supported covariance functions [Furrer *et al.*, 2006]) are capable of modeling highly-varying functions with low correlation (i.e., short length-scales) but they use only local data for predictions, hence predicting poorly in input regions with sparse data. Recent approximate GP regression methods of [Snelson, 2007] and [Vanhatalo and Vehtari, 2008] have attempted to combine the best of both worlds.

Another idea to achieve efficient and scalable predictions in real time is to distribute computational loads to clusters of parallel machines. Such an idea of scaling up machine learning techniques (e.g., clustering, support vector machines, graphical models) has recently attracted widespread interest in the machine learning community [Bekkerman *et al.*, 2011]. For the case of Gaussian process regression, the local GPs method [Das and Srivastava, 2010; Choudhury *et al.*, 2002] appears most straightforward to be "embarrassingly" parallelized but they suffer from discontinuities in predictions on the boundaries of different local GPs. The work of [Park *et al.*, 2011] rectifies this problem by imposing continuity constraints along the boundaries in a centralized manner. But, its use is restricted strictly to data with 1- and 2-dimensional input features.

2.3 Data Fusion

The phenomenon data is distributed among mobile sensors, therefore has to be assimilated into a predictive model for spatiotemporal phenomenon prediction.

Existing decentralized and distributed Bayesian filtering frameworks for addressing nontraffic related problems [Chung *et al.*, 2004; Coates, 2004; Olfati-Saber and Shamma, 2005; Rosencrantz *et al.*, 2003; Sukkarieh *et al.*, 2003] face the same difficulties as their centralized counterparts described above if applied to predict traffic phenomena, thus resulting in loss of predictive performance. Distributed regression algorithms [Guestrin *et al.*, 2004; Paskin and Guestrin, 2004] for static sensor networks gain efficiency from spatial locality assumptions. However, such methods cannot be exploited by mobile sensors whose paths are not constrained by locality. Cortes in [Cortes, 2009] proposed a distributed data fusion approach to approximate GP prediction based on an iterative Jacobi overrelaxation algorithm, which incurs some critical limitations: (a) the past observations taken along the sensors paths are assumed to be uncorrelated, which greatly undermines its predictive performance when they are in fact correlated and/or the current observations are sparse; (b) when the number of sensors grows large, it converges very slowly; (c) it assumes that the range of positive correlation has to be bounded by some factor of the communication range. Our proposed decentralized data fusion algorithms (Sections 5.1.1 and 5.1.2) do not suffer from these limitations and can be computed exactly with efficient time bounds.

2.4 Active Sensing

Towards sensing and predicting environmental phenomena with active mobile sensors, one branch of active sensing strategies [Leonard *et al.*, 2007; Zhang and Sukhatme, 2007; Singh *et al.*, 2007] focus on collecting phenomenon data from sparsely sampled regions considering the unobserved phenomenon in these regions are of high uncertainty. In addition, another branch [Popa and Lewis, 2008; Choi *et al.*, 2007; Singh *et al.*, 2006; Bryan *et al.*, 2005] emphasize on collecting phenomenon data from feature regions (e.g., hotspots) that have highly varying measurements, as more observations in these regions are needed for predicting the phenomenon. For certain environmental phenomena, such as the ocean phenomena (e.g., temperature, plankton density) [Low *et al.*, 2012] which contain multiple hotspots, active sensing strategies need to balance between sensing the feature regions (i.e., tracking hotspot boundary) and exploring sparsely sampled regions to search for other hotspots. However, this strategy can only be applied to boundary tracking and works in greedy fashion. Existing parametric approaches [Rahimi *et al.*, 2005; Choi and Oh, 2008] combine different criteria (e.g., for avoidance, tracing, or exploration) with trade-off coefficients, thereby achieving such balance. However, it is not showed how the optimal coefficients of these parameterized active sensing strategies can be automatically obtained in online manner. To address this issue, Low et al. exploit a principled approach *log-Gaussian process* (ℓGP) to model the phenomena containing hotspot [Low *et al.*, 2008a], and based on which develop an information-theoretic sampling strategy [Low *et al.*, 2009a] that can collect phenomenon data from sparsely sampled regions and hotspot regions simultaneously without tuning any coefficients. This active sensing strategy provides an important insight on designing strategies for actively sensing an urban mobility pattern containing extremity and skewness. This strategy requires centralized computation that is a major limitation hindering it from performing efficiently.

Existing centralized active sensing algorithms [Low *et al.*, 2008a; Low *et al.*, 2009a; Low *et al.*, 2011] scale poorly with a large number of data and sensors, therefore, are not suitable for providing online information. The active sensing strategy in [Low *et al.*, 2012] is decentralized in the sensing that each mobile sensor selects their locally optimal walk. However, a centralized data fusion method is needed to assimilate the phenomenon data to compute the strategy, which is inefficient when the phenomenon data is large. [Graham and Cortés, 2009; Graham and Cortes, 2010; Graham and Cortes, 2011] present efficient cooperative active sensing by partitioning the field into Voronoi configuration. Then, only the static and mobile senors in correlated Voronoi cells have to be coordinated. However, their approaches assume the availability of static sensors that are deployed under near-independence assumption. Additionally, they can only work in geospatia domain with 2-dimensional input features. [Stranders *et al.*, 2009] present a decentralized coordination algorithm for mobile sensors performing active sensing based on GP model. However, this algorithm still suffers from computation and communication issues: (a) direct employment of max-sum message passing algorithm for decentralized algorithm is prohibitive due to enormous computation of messages, therefore, pruning algorithm is a necessity; (b) the online joint action pruning algorithm relies on partial joint moves to reduce the size of action space, which is not effective in large scale, in the worse case, is still exponential in the number of agents and length of planning horizon; (c) the run-time efficiency is extremely sensitive to the connectivity and latency of network due to message passing; (d) a centralized fusion center is required to assimilate all the measurements.

Chapter 3

Modeling Spatiotemporal Traffic Phenomena

This chapter starts by providing an overview of *Gaussian process* (GP) model (Section 3.1). Then, we introduce a simple *subset of data* (SoD) approximation of GP model (Section 3.2) to alleviate the cubic time complexity of full/exact GP. Based on GP, a novel relational GP [Chen *et al.*, 2012] is proposed to model real world traffic conditions (speeds) over road network. The correlation structure of such relational GP model takes into account both the road segment features and road network topology information (Section 3.3). Additionally, another GP variant called *log-Gaussian process* (ℓGP) [Chen *et al.*, 2013b] is exploited to model an urban mobility demand pattern which contains skewness and extremity in demand measurements (Section 3.4).

3.1 Gaussian Process

The *Gaussian processes* (GP) which are Bayesian non-parametric models can be used to perform probabilistic regression as follows: Let $\mathcal{X}$ be a set representing the input domain such that each input $x \in \mathcal{X}$ denotes a p-dimensional feature vector and is associated with a realized output value y_x (random output variable Y_x) if it is observed (unobserved). Let $\{Y_x\}_{x\in\mathcal{X}}$ denote a GP, that is, every finite subset of $\{Y_x\}_{x\in\mathcal{X}}$ follows a multivariate Gaussian distribution [Rasmussen and Williams, 2006]. Then, the GP is fully specified by its *prior* mean $\mu_x \triangleq \mathbb{E}[Y_x]$ and covariance $\sigma_{xx'} \triangleq \text{cov}[Y_x, Y_{x'}]$ for all $x, x' \in \mathcal{X}$, the latter of which is usually defined by a specific covariance function.

Given that a column vector $y_{\mathcal{D}}$ of realized outputs is observed for some set $\mathcal{D} \subset \mathcal{X}$ of inputs, the GP can exploit this data $(\mathcal{D}, y_{\mathcal{D}})$ to provide predictions of the unobserved outputs for any set $\mathcal{U} \subseteq \mathcal{X} \setminus \mathcal{D}$ of inputs and their corresponding predictive uncertainties using the posterior Gaussian distribution $\mathcal{N}(\mu_{\mathcal{U}|\mathcal{D}}, \Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}})$ specified by *posterior* mean vector $\mu_{\mathcal{U}|\mathcal{D}}$ and covariance matrix $\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}$ defined as

$$\mu_{\mathcal{U}|\mathcal{D}} \triangleq \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{D}}\Sigma_{\mathcal{D}\mathcal{D}}^{-1}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) \tag{3.1}$$

$$\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}} \triangleq \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{D}}\Sigma_{\mathcal{D}\mathcal{D}}^{-1}\Sigma_{\mathcal{D}\mathcal{U}} \tag{3.2}$$

where $\mu_{\mathcal{U}}$ ($\mu_{\mathcal{D}}$) is a column vector with mean components μ_x for all $x \in \mathcal{U}$ ($x \in \mathcal{D}$), $\Sigma_{\mathcal{U}\mathcal{D}}$ ($\Sigma_{\mathcal{D}\mathcal{D}}$) is a covariance matrix with covariance components $\sigma_{xx'}$ for all $x \in \mathcal{U}, x' \in \mathcal{D}$ ($x, x' \in \mathcal{D}$), and $\Sigma_{\mathcal{D}\mathcal{U}}$ is the transpose of $\Sigma_{\mathcal{U}\mathcal{D}}$.

The posterior covariance matrix $\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}$ (3.2), which is independent of the measurements $y_{\mathcal{D}}$, can be processed in two ways to quantify the uncertainty of these predictions: (a) the trace of $\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}$ yields the sum of posterior variances $\Sigma_{xx|\mathcal{D}}$ over all $x \in \mathcal{U}$; (b) the determinant of $\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}$ is used in calculating the Gaussian posterior joint entropy

$$\mathbb{H}[Y_{\mathcal{U}}|Y_{\mathcal{D}}] \triangleq \frac{1}{2}\log(2\pi e)^{|\mathcal{U}|}\left|\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}\right| . \tag{3.3}$$

In contrast to the first measure of uncertainty that assumes conditional independence between measurements in the set $\mathcal{U}$ of unobserved inputs, the entropy-based measure (3.3) accounts for their correlation, thereby not overestimating their uncertainty. Hence, this thesis will focus on using the entropy-based measure of uncertainty.

3.2 Subset of Data Approximation

Although the GP is an effective predictive model, it faces a practical limitation of cubic time complexity in the number $|\mathcal{D}|$ of observations; this can be observed from computing the posterior distribution (i.e., (3.1) and (3.2)), which requires inverting covariance matrix $\Sigma_{\mathcal{D}\mathcal{D}}$ that incurs $\mathcal{O}\big(|\mathcal{D}|^3\big)$ time. If $|\mathcal{D}|$ is expected to be large, GP prediction cannot be performed in real time. For practical usage, we have to resort to computationally cheaper approximate GP prediction.

A simple method of approximation is to select only a subset $\mathcal{S}$ of the entire set $\mathcal{D}$ of observed inputs (i.e., $\mathcal{S} \subset \mathcal{D}$) to compute the posterior distribution of the measurements at any set $\mathcal{U} \subseteq \mathcal{X} \setminus \mathcal{D}$ of unobserved inputs. Such a sparse *subset of data* (SoD) approximation method produces the following predictive Gaussian distribution $\mathcal{N}(\mu_{\mathcal{U}|\mathcal{S}}, \Sigma_{\mathcal{U}\mathcal{U}|\mathcal{S}})$, which closely resembles that of the full GP model (i.e., by simply replacing $\mathcal{D}$ in (3.1) and (3.2) with $\mathcal{S}$):

$$\mu_{\mathcal{U}|\mathcal{S}} = \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}(y_{\mathcal{S}} - \mu_{\mathcal{S}}) \tag{3.4}$$

$$\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{S}} = \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{U}} \ . \tag{3.5}$$

Notice that the covariance matrix $\Sigma_{\mathcal{S}\mathcal{S}}$ to be inverted only incurs $\mathcal{O}\big(|\mathcal{U}|^3\big)$ time, which is independent of $|\mathcal{D}|$.

The predictive performance of SoD approximation is sensitive to the selection of subset $\mathcal{S}$. In practice, random subset selection often yields poor performance. This issue can be resolved by actively selecting an informative subset $\mathcal{S}$ in an iterative greedy manner: Firstly, $\mathcal{S}$ is initialized to be an empty set. Then, all inputs in $\mathcal{D} \setminus \mathcal{S}$ are scored based on a criterion that can be chosen from, for example, the works of [Krause *et al.*, 2008b; Lawrence *et al.*, 2003; Seeger and Williams, 2003]. The highest-scored input is selected for inclusion into $\mathcal{S}$ and removed from $\mathcal{D}$. This greedy selection procedure is iterated until $\mathcal{S}$ reaches a pre-defined size. Among the various criteria introduced in the literatures, the differential entropy score [Lawrence *et al.*, 2003] is reported to perform well [Oh *et al.*, 2010]; it is a monotonic function of the posterior variance $\Sigma_{xx|\mathcal{S}}$ (3.5), thus resulting in the greedy selection of a segment $x \in \mathcal{D} \setminus \mathcal{S}$ with the largest variance in each iteration.

3.3 Modeling a Traffic Condition over Road Network

The *Gaussian process* (GP) can be used to model a spatiotemporal traffic phenomenon (e.g., traffic speeds in Figure 6.1) over a road network as follows: The traffic phenomenon is defined to vary as a realization of a GP. Let V be a set of road segments representing the domain of the road network such that each road segment $s \in V$ is specified by a p-dimensional vector of features and is associated with a realized (random) measurement y_s (Y_s) of the traffic condition such as speed if s is observed (unobserved).

If the observations are noisy (i.e., by assuming additive independent identically distributed Gaussian noise with variance σ_n^2), then their prior covariance $\sigma_{ss'}$ can be expressed as $\sigma_{ss'} = k(s, s') + \sigma_n^2 \delta_{ss'}$ where $\delta_{ss'}$ is a Kronecker delta that is 1 if $s = s'$ and 0 otherwise, and k is a kernel function measuring the pairwise "similarity" of road segments.

For a traffic phenomenon (e.g., road speeds), the correlation of measurements between pairs of road segments depends not only on their features (e.g., length, number of lanes, speed limit, direction) but also the road network topology. So, the kernel function is defined to exploit both the features and topology information. To achieve this aim, we present a relational Gaussian process model with a graph-based kernel in the subsequent section.

3.3.1 Relational Gaussian Process

The key to developing a relational GP is to specify a graph-based kernel that can take into account the road segment features and the topology information of road network. In the following, we first define the road network as

Definition 1 (Road Network)**.** *Let the road network be represented as a weighted directed graph $G \triangleq (V, E, m)$ that consists of*

- *a set V of vertices denoting the domain of all possible road segments,*
- *a set $E \subseteq V \times V$ of edges such that there is a edge (s, s') from $s \in V$ to $s' \in V$ iff the end of segment s connects to the start of segment s' in the road network, and*
- *a weight function $m : E \to \mathbb{R}^+$ measuring the standardized Manhattan distance [Borg and Groenen, 2005] $m((s, s')) \triangleq \sum_{i=1}^{p} |[s]_i - [s']_i|/r_i$ of each edge (s, s') where $[s]_i$ ($[s']_i$) is the i-th component of the feature vector specifying road segment s (s'), and r_i is the range of the i-th feature. The weight function m serves as a dissimilarity measure between adjacent road segments.*

The next step is to compute the shortest path distance $d(s, s')$ between all pairs of road segments $s, s' \in V$ (i.e., using Floyd-Warshall or Johnson's algorithm) with respect to the topology of the weighted directed graph G. Such a distance function is again a measure of dissimilarity, rather than one of similarity, as required by a kernel function. Furthermore, a valid GP kernel needs to be positive semidefinite and symmetric [Schölkopf and Smola, 2002], which are clearly violated by d because $d(s, s')$ and $d(s', s)$ may not be equal.

To construct a valid GP kernel from d, multi-dimensional scaling [Borg and Groenen, 2005] is applied to embed the domain of road segments into the p'-dimensional Euclidean space $\mathbb{R}^{p'}$. Specifically, a mapping $g : V \to \mathbb{R}^{p'}$ is determined by minimizing the squared loss $g^* = \arg\min_g \sum_{s,s' \in V}(d(s, s') - \|g(s) - g(s')\|)^2$.

With a small squared loss, the Euclidean distance $\|g^*(s) - g^*(s')\|$ between $g^*(s)$ and $g^*(s')$ is expected to closely approximate the shortest path distance $d(s, s')$ between any pair of road segments s and s'. After embedding into Euclidean space, a conventional kernel function such as the squared exponential one [Rasmussen and Williams, 2006] can be used:

$$k(s, s') = \sigma_s^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{p'}\left(\frac{[g^*(s)]_i - [g^*(s')]_i}{\ell_i}\right)^2\right)$$

where $[g^*(s)]_i$ ($[g^*(s')]_i$) is the i-th component of the p'-dimensional vector $g^*(s)$ ($g^*(s')$), and the hyperparameters σ_s, $\ell_1, \ldots, \ell_{p'}$ are, respectively, signal variance and length-scales that can be learned using maximum likelihood estimation [Rasmussen and Williams, 2006]. The resulting kernel function k[1] is guaranteed to be valid. Then, a standard GP specified by this Graph-based kernel can be used to model the spatiotemporal traffic phenomenon over road network.

3.4 Modeling an Urban Mobility Demand Pattern

The GP can also be used to model a spatiotemporal urban mobility demand pattern. First, the service area in an urban city can be represented as a directed graph $G \triangleq (V, E)$ where V denotes a set of all regions generated by gridding

[1] For spatiotemporal traffic modeling, the kernel function k can be extended to account for the temporal dimension. Refer to Section 4.4 for the details that such kernel function is applied to model spatiotemporal traffic speeds (i.e., AIMPEAK dataset).

the service area, and $E \subseteq V \times V$ denotes a set of edges such that there is an edge (s, s') from $s \in V$ to $s' \in V$ iff at least one road segment in the road network starts in s and ends in s'. Each region $s \in V$ is associated with a p-dimensional feature vector x_s representing its context information (e.g., location, time, precipitation), and a measurement y_s quantifying its mobility demand[2]. Since it is often impractical in terms of sensing resource cost to determine the actual mobility demand of a region, a common practice is to use the pickup count[3] of the region as a surrogate measure.

To elaborate, the user pickups made by vacant vehicles cruising in a region contribute to its pickup count. Since we do not assume a data center to be available to keep track of the pickup count, a fully distributed gossip-based protocol [Jelasity *et al.*, 2005] is utilized to aggregate these pickup information from the vehicles in the region that are connected via an ad hoc wireless communication network. The gossip-based protocol supports distributed aggregation, such as, counting number of nodes (vehicles) and summing up distributed local values (pickup counts); moreover, these operations are robust with respect to changing topology, crashing node, and link failure. Consequently, any vehicle entering the region can access its pickup count simply by joining its ad hoc network.

3.4.1 Log-Gaussian Process

As observed in [Chang *et al.*, 2010; Li *et al.*, 2012] and our real-world data (see Figure 7.1a), a mobility demand pattern over a large service area in an urban city is typically characterized by spatiotemporally correlated demand measurements and contains a few small-scale hotspots exhibiting extreme measurements and much higher spatiotemporal variability than the rest of the demand pattern. That is, if the measurements are put together into a 1D sample frequency distribution, a positive skew results. We like to consider using a rich class of Bayesian nonparametric models called *Gaussian process* (GP) [Rasmussen and Williams, 2006] to model the demand pattern. But, the GP covariance structure is sensitive to strong positive skewness and easily destabilized by a few extreme measure-

[2] At the segment level, we observe a lower degree of spatial correlation across segments because many road segments do not allow vehicles to stop, hence disrupting the smoothness of demand measurements.

[3] In our experiments, a pickup point can be identified when a taxi's status is changed from free to passenger-on-board. Then, the pickups in the same region are aggregated into a pickup count.

ments [Webster and Oliver, 2007]. In practice, this can cause reconstructed patterns to display large hotspots centered about a few extreme measurements and predictive variances to be unrealistically small in hotspots [Hohn, 1998], which are undesirable. So, if the GP is used to model a demand pattern directly, it may not predict well. To resolve this, a standard statistical practice is to take the log of the measurements (i.e., $z_s = \log y_s$) to remove skewness and extremity, and use the GP to model the demand pattern in the *log-scale* instead.

Since our ultimate interest is to predict the mobility demand in the *original scale*, GP's predicted log-measurements of these unobserved regions must be transformed back *unbiasedly*. To achieve this, we utilize a widely-used variant of GP in geostatistics called the Log-Gaussian Process (ℓGP) that can model the demand pattern in the original scale. Let $\{Y_s\}_{s \in V}$ denote a ℓGP: If $Z_s \triangleq \log Y_s$, then $\{Z_s\}_{s \in V}$ is a GP. So, $Y_s = \exp\{Z_s\}$ denotes the original random demand measurement of unobserved region s and is predicted using the log-Gaussian posterior mean (i.e., best unbiased predictor)

$$\mu^{\ell GP}_{s|\mathcal{D}} \triangleq \exp(\mu_{s|\mathcal{D}} + \Sigma_{ss|\mathcal{D}}/2) \tag{3.6}$$

where $\mu_{s|\mathcal{D}}$ and $\Sigma_{ss|\mathcal{D}}$ are simply the Gaussian posterior mean (3.1) and variance (3.2) of GP, respectively.

The uncertainty of predicting the measurements of any set $\mathcal{U} \subset V$ of unobserved regions can be quantified by the following log-Gaussian posterior joint entropy:

$$\mathbb{H}[Y_{\mathcal{U}}|Y_{\mathcal{D}}] \triangleq \frac{1}{2}\log(2\pi e)^{|\mathcal{U}|}\left|\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}\right| + \mu_{\mathcal{U}|\mathcal{D}} \cdot \mathbf{1} \tag{3.7}$$

where $\mu_{\mathcal{U}|\mathcal{D}}$ and $\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}$ are the Gaussian posterior mean vector (3.1) and covariance matrix (3.2) of GP, respectively.

Chapter 4

Parallel Gaussian Process

In chapter 3, it has been shown that spatiotemporal traffic phenomenon can be modeled using *Gaussian process* (GP)-based models. In general, GP are Bayesian non-parametric models for performing nonlinear regression, which offer an important advantage of providing fully probabilistic predictive distributions with formal measures of the uncertainty of the predictions. The key limitation hindering the practical use of GP for large phenomenon data is the high computational cost: It incurs cubic time and quadratic memory in the size of the data. Despite various efforts to scale up GP (Section 2.2), it remains computationally impractical for performing real-time predictions necessary in many time-critical applications and decision support systems (e.g., ocean sensing, traffic monitoring, geographical information systems) that need to process and analyze huge quantities of data collected over short time durations (e.g., in astronomy, internet traffic, meteorology, surveillance).

To resolve this, this chapter considers exploiting clusters of parallel/multi-core machines to achieve efficient and scalable predictions in real time [Chen *et al.*, 2013a]. Section 4.1 presents two novel parallel GPs: *parallel partially independent training conditional* (pPITC) and *parallel partially independent conditional* (pPIC) approximation of full GP (FGP) model. Such parallel GPs exploit the notion of a support set. In addition, Section 4.2 presents another novel parallel GP based on *parallel incomplete Cholesky factorization* (pICF). Then, the properties (i.e., time, space, and communication complexity, online learning, and structural assumptions) of all proposed parallel GPs are analysed in comparison with their centralized counterparts and FGP (Section 4.3). Lastly, Sections 4.4 & 4.5 empirically evaluate the predictive performances, time efficiency, scalability, and speedups of our proposed parallel GPs against their centralized counterparts and FGP on two real-world datasets.

4.1 Parallel Gaussian Process Regression using Support Set

In this section, we will present a class of parallel Gaussian processes (*parallel partially independent training conditional* (pPITC) and *parallel partially independent conditional* (pPIC)) which can distribute the computational load into a cluster of parallel machines to achieve efficient and scalable approximate GP regression by exploiting the notion of a support set.

4.1.1 Parallel Gaussian Process: pPITC

The key idea of pPITC is as follows: After distributing the data evenly among M machines (Step 1), each machine encapsulates its local data, based on a common prior support set $\mathcal{S} \subset X$ where $|\mathcal{S}| \ll |\mathcal{D}|$, into a local summary that is communicated to the master[1] (Step 2). The master assimilates the local summaries into a global summary (Step 3), which is then sent back to the M machines to be used for predictions distributed among them (Step 4). These steps are detailed below:

STEP 1: DISTRIBUTE DATA AMONG M MACHINES.

The data $(\mathcal{D}, y_\mathcal{D})$ is partitioned evenly into M blocks, each of which is assigned to a machine, as defined below:

Definition 2 (Local Data). *The local data of machine m is defined as a tuple $(\mathcal{D}_m, y_{\mathcal{D}_m})$ where $\mathcal{D}_m \subseteq \mathcal{D}$, $\mathcal{D}_m \bigcap \mathcal{D}_i = \emptyset$ and $|\mathcal{D}_m| = |\mathcal{D}_i| = |\mathcal{D}|/M$ for $i \neq m$.*

STEP 2: EACH MACHINE CONSTRUCTS AND SENDS LOCAL SUMMARY TO MASTER.

The local data of each machine is summarized into a local summary defined below:

[1] One of the M machines can be assigned to be the master.

Definition 3 (Local Summary). *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all M machines and the local data $(\mathcal{D}_m, y_{\mathcal{D}_m})$, the local summary of machine m is defined as a tuple $(\dot{y}^m_{\mathcal{S}}, \dot{\Sigma}^m_{\mathcal{S}\mathcal{S}})$ where*

$$\dot{y}^m_{\mathcal{B}} \triangleq \Sigma_{\mathcal{B}\mathcal{D}_m} \Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}} (y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}) \tag{4.1}$$

$$\dot{\Sigma}^m_{\mathcal{B}\mathcal{B}'} \triangleq \Sigma_{\mathcal{B}\mathcal{D}_m} \Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}} \Sigma_{\mathcal{D}_m\mathcal{B}'} \tag{4.2}$$

such that $\Sigma_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}$ is defined in a similar manner as (3.2) and $\mathcal{B}$ denotes any subset of input domain $\mathcal{X}$.

Remark. Since the local summary is independent of the outputs $y_{\mathcal{S}}$, they need not be observed. As a result, the support set $\mathcal{S}$ does not have to be a subset of $\mathcal{D}$ and can be selected prior to data collection. The predictive performance of pPITC (and pPIC) is sensitive to the selection of $\mathcal{S}$. An informative support set $\mathcal{S}$ can be selected from domain $\mathcal{X}$ using an iterative greedy active selection procedure [Krause *et al.*, 2008b; Lawrence *et al.*, 2003; Seeger and Williams, 2003] prior to observing data. For example, the differential entropy score criterion [Lawrence *et al.*, 2003] can be used to greedily select an input $x \in \mathcal{X} \setminus \mathcal{S}$ with the largest posterior variance $\Sigma_{xx|\mathcal{S}}$ (3.2) to be included in $\mathcal{S}$ in each iteration.

STEP 3: MASTER CONSTRUCTS AND SENDS GLOBAL SUMMARY TO M MACHINES.

The local summaries are assimilated into a global summary defined by

Definition 4 (Global Summary). *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all M machines and the local summary $(\dot{y}^m_{\mathcal{S}}, \dot{\Sigma}^m_{\mathcal{S}\mathcal{S}})$ of every machine $m = 1, \ldots, M$, the global summary is defined as a tuple $(\ddot{y}_{\mathcal{S}}, \ddot{\Sigma}_{\mathcal{S}\mathcal{S}})$ where*

$$\ddot{y}_{\mathcal{S}} \triangleq \sum_{m=1}^{M} \dot{y}^m_{\mathcal{S}} \tag{4.3}$$

$$\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} \triangleq \Sigma_{\mathcal{S}\mathcal{S}} + \sum_{m=1}^{M} \dot{\Sigma}^m_{\mathcal{S}\mathcal{S}} \ . \tag{4.4}$$

STEP 4: DISTRIBUTE PREDICTIONS AMONG M MACHINES.

To predict the unobserved outputs for any set $\mathcal{U}$ of inputs, $\mathcal{U}$ is partitioned evenly into disjoint subsets $\mathcal{U}_1, \ldots, \mathcal{U}_M$ to be assigned to the respective machines $1, \ldots, M$. So, $|\mathcal{U}_m| = |\mathcal{U}|/M$ for $m = 1, \ldots, M$.

Definition 5 (pPITC)**.** *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all M machines and the global summary $(\ddot{y}_{\mathcal{S}}, \ddot{\Sigma}_{\mathcal{SS}})$, each machine m computes a predictive Gaussian distribution $\mathcal{N}(\widehat{\mu}_{\mathcal{U}_m}, \widehat{\Sigma}_{\mathcal{U}_m\mathcal{U}_m})$ of the unobserved outputs for the set $\mathcal{U}_m$ of inputs where*

$$\widehat{\mu}_{\mathcal{U}_m} \triangleq \mu_{\mathcal{U}_m} + \Sigma_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}_{\mathcal{SS}}^{-1}\ddot{y}_{\mathcal{S}} \tag{4.5}$$

$$\widehat{\Sigma}_{\mathcal{U}_m\mathcal{U}_m} \triangleq \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \Sigma_{\mathcal{U}_m\mathcal{S}}\left(\Sigma_{\mathcal{SS}}^{-1} - \ddot{\Sigma}_{\mathcal{SS}}^{-1}\right)\Sigma_{\mathcal{S}\mathcal{U}_m}\ . \tag{4.6}$$

Though pPITC scales very well with large data (Table 4.1), it can predict poorly due to (a) loss of information caused by summarizing the realized outputs and correlation structure of the original data; and (b) sparse coverage of $\mathcal{U}$ by the support set. To resolve this issue, subsequently, we propose a novel parallel Gaussian process regression method called *parallel partially independent conditional* (pPIC) approximation of GP that can improve the predictive accuracy of pPITC, at the same time, preserve its efficiency.

4.1.2 Parallel Gaussian Process: pPIC

pPIC is based on the following intuition: A machine can exploit its local data to improve the predictions of the unobserved outputs that are highly correlated with its data. At the same time, pPIC can preserve the time efficiency of pPITC by exploiting its idea of encapsulating information into local and global summaries. pPIC differs from pPITC only in Step 4 when computes the prediction.

Definition 6 (pPIC)**.** *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all M machines, the global summary $(\ddot{y}_{\mathcal{S}}, \ddot{\Sigma}_{\mathcal{SS}})$, the local summary $(\dot{y}_{\mathcal{S}}^{m}, \dot{\Sigma}_{\mathcal{SS}}^{m})$, and the local data $(\mathcal{D}_m, y_{\mathcal{D}_m})$, each machine m computes a predictive Gaussian distribution $\mathcal{N}(\widehat{\mu}_{\mathcal{U}_m}^{+}, \widehat{\Sigma}_{\mathcal{U}_m\mathcal{U}_m}^{+})$ of the unobserved outputs for the set $\mathcal{U}_m$ of inputs where*

$$\widehat{\mu}_{\mathcal{U}_m}^{+} \triangleq \mu_{\mathcal{U}_m} + \left(\Phi_{\mathcal{U}_m\mathcal{S}}^{m}\ddot{\Sigma}_{\mathcal{SS}}^{-1}\ddot{y}_{\mathcal{S}} - \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma_{\mathcal{SS}}^{-1}\dot{y}_{\mathcal{S}}^{m}\right) + \dot{y}_{\mathcal{U}_m}^{m} \tag{4.7}$$

$$\begin{aligned}\widehat{\Sigma}^{+}_{\mathcal{U}_m\mathcal{U}_m} \triangleq \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \Big(&\Phi^{m}_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{U}_m} - \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\dot{\Sigma}^{m}_{\mathcal{S}\mathcal{U}_m} \\ &- \Phi^{m}_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^{m}_{\mathcal{S}\mathcal{U}_m}\Big) - \dot{\Sigma}^{m}_{\mathcal{U}_m\mathcal{U}_m}\end{aligned} \tag{4.8}$$

such that

$$\Phi^{m}_{\mathcal{U}_m\mathcal{S}} \triangleq \Sigma_{\mathcal{U}_m\mathcal{S}} + \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\dot{\Sigma}^{m}_{\mathcal{S}\mathcal{S}} - \dot{\Sigma}^{m}_{\mathcal{U}_m\mathcal{S}} \tag{4.9}$$

and $\Phi^{m}_{\mathcal{S}\mathcal{U}_m}$ *is the transpose of* $\Phi^{m}_{\mathcal{U}_m\mathcal{S}}$.

Remark 1. The predictive Gaussian mean $\widehat{\mu}^{+}_{\mathcal{U}_m}$ and covariance $\widehat{\Sigma}^{+}_{\mathcal{U}_m\mathcal{U}_m}$ (see (4.7) and (4.8)) of pPIC exploit both summary information (i.e., bracketed term) and local information (i.e., last term). In contrast, pPITC only exploits the global summary (see (5.1) and (5.2)).

Remark 2. To improve the predictive performance of pPIC, $\mathcal{D}$ and $\mathcal{U}$ should be partitioned into tuples of $(\mathcal{D}_1, \mathcal{U}_1), \ldots, (\mathcal{D}_M, \mathcal{U}_M)$ such that the outputs $y_{\mathcal{D}_m}$ and $Y_{\mathcal{U}_m}$ are as highly correlated as possible for $m = 1, \ldots, M$. To achieve this, we employ a simple parallelized clustering scheme in our experiments: Each machine m randomly selects a cluster center from its local data $\mathcal{D}_m$ and informs the other machines about its chosen cluster center. Then, each input in $\mathcal{D}_m$ and $\mathcal{U}_m$ is simply assigned to the "nearest" cluster center i and sent to the corresponding machine i while being subject to the constraints of the new $\mathcal{D}_i$ and $\mathcal{U}_i$ not exceeding $|\mathcal{D}|/M$ and $|\mathcal{U}|/M$, respectively. More sophisticated clustering schemes can be utilized at the expense of greater time and communication complexity.

Remark 3. The predictive performances of pPITC and pPIC can be improved by increasing the size of $\mathcal{S}$ at the expense of greater time, space, and communication complexity (Table 4.1).

4.1.3 Performance Guarantee

Theorem 1. *Let a common support set $\mathcal{S} \subset \mathcal{X}$ be known to all M machines. Let $\mathcal{N}(\mu^{\text{PITC}}_{\mathcal{U}|\mathcal{D}}, \Sigma^{\text{PITC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}})$ be the predictive Gaussian distribution computed by the centralized partially independent training conditional (PITC) approximation of FGP model [Quiñonero-Candela and Rasmussen, 2005] where*

$$\mu^{\text{PITC}}_{\mathcal{U}|\mathcal{D}} \triangleq \mu_{\mathcal{U}} + \Gamma_{\mathcal{U}\mathcal{D}}\left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1}\left(y_{\mathcal{D}} - \mu_{\mathcal{D}}\right) \tag{4.10}$$

$$\Sigma^{\text{PITC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}} \triangleq \Sigma_{\mathcal{U}\mathcal{U}} - \Gamma_{\mathcal{U}\mathcal{D}} \left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1} \Gamma_{\mathcal{D}\mathcal{U}} \tag{4.11}$$

such that

$$\Gamma_{\mathcal{B}\mathcal{B}'} \triangleq \Sigma_{\mathcal{B}\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{B}'} \tag{4.12}$$

and Λ is a block-diagonal matrix constructed from the M diagonal blocks of $\Sigma_{\mathcal{D}\mathcal{D}|\mathcal{S}}$, each of which is a matrix $\Sigma_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}$ for $m = 1, \ldots, M$ where $\mathcal{D} = \bigcup_{m=1}^{M} \mathcal{D}_m$. Then, $\widehat{\mu}_{\mathcal{U}} = \mu^{\text{PITC}}_{\mathcal{U}|\mathcal{D}}$ and $\widehat{\Sigma}_{\mathcal{U}\mathcal{U}} = \Sigma^{\text{PITC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}}$.

Theorem 2. *Let a common support set $\mathcal{S} \subset \mathcal{X}$ be known to all M machines. Let $\mathcal{N}(\mu^{\text{PIC}}_{\mathcal{U}|\mathcal{D}}, \Sigma^{\text{PIC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}})$ be the predictive Gaussian distribution computed by the centralized partially independent conditional (PIC) approximation of FGP model [Snelson, 2007] where*

$$\mu^{\text{PIC}}_{\mathcal{U}|\mathcal{D}} \triangleq \mu_{\mathcal{U}} + \widetilde{\Gamma}_{\mathcal{U}\mathcal{D}} \left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1} \left(y_{\mathcal{D}} - \mu_{\mathcal{D}}\right) \tag{4.13}$$

$$\Sigma^{\text{PIC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}} \triangleq \Sigma_{\mathcal{U}\mathcal{U}} - \widetilde{\Gamma}_{\mathcal{U}\mathcal{D}} \left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1} \widetilde{\Gamma}_{\mathcal{D}\mathcal{U}} \tag{4.14}$$

and $\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}}$ is the transpose of $\widetilde{\Gamma}_{\mathcal{U}\mathcal{D}}$ such that

$$\widetilde{\Gamma}_{\mathcal{U}\mathcal{D}} \triangleq \left(\widetilde{\Gamma}_{\mathcal{U}_i\mathcal{D}_m}\right)_{i,m=1,\ldots,M} \tag{4.15}$$

$$\widetilde{\Gamma}_{\mathcal{U}_i\mathcal{D}_m} \triangleq \begin{cases} \Sigma_{\mathcal{U}_i\mathcal{D}_m} & \textit{if } i = m, \\ \Gamma_{\mathcal{U}_i\mathcal{D}_m} & \textit{otherwise}. \end{cases} \tag{4.16}$$

Then, $\widehat{\mu}^{+}_{\mathcal{U}} = \mu^{\text{PIC}}_{\mathcal{U}|\mathcal{D}}$ and $\widehat{\Sigma}^{+}_{\mathcal{U}\mathcal{U}} = \Sigma^{\text{PIC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}}$.

The proofs of Theorems 1 and 2 are given in Appendix A and B, respectively.

Remark 1. The equivalence results of Theorems 1 and 2 imply that the computational load of the centralized PITC and PIC approximations of FGP can be distributed among the M parallel machines, hence improving the time efficiency and scalability of approximate GP regression (Table 4.1).

Remark 2. The equivalence results also shed some light on the underlying properties of pPITC and pPIC based on the structural assumptions of PITC and PIC, respectively: pPITC assumes that $Y_{\mathcal{D}_1}, \ldots, Y_{\mathcal{D}_M}, Y_{\mathcal{U}_1}, \ldots, Y_{\mathcal{U}_M}$ are conditionally independent given $Y_{\mathcal{S}}$. In contrast, pPIC can predict the unobserved outputs $Y_{\mathcal{U}}$ better since it imposes a less restrictive assumption of conditional independence between $Y_{\mathcal{D}_1 \bigcup \mathcal{U}_1}, \ldots, Y_{\mathcal{D}_M \bigcup \mathcal{U}_M}$ given $Y_{\mathcal{S}}$. This assumption further supports an earlier remark (i.e., just before Theorem 1) on clustering inputs $\mathcal{D}_m$ and $\mathcal{U}_m$ whose corresponding outputs are highly correlated for improving the predictive performance of pPIC. Experimental results on two real-world datasets (Section 4.4) show that pPIC achieves predictive accuracy comparable to FGP and significantly better than pPITC, thus justifying the practicality of such an assumption.

Remark 3. Since PITC generalizes the Bayesian Committee Machine (BCM) of [Schwaighofer and Tresp, 2002], pPITC generalizes parallel BCM [Ingram and Cornford, 2010], the latter of which assumes the support set $\mathcal{S}$ to be $\mathcal{U}$ [Quiñonero-Candela and Rasmussen, 2005]. As a result, parallel BCM does not scale well with large $\mathcal{U}$.

4.2 Parallel Gaussian Process Regression using Incomplete Cholesky Factorization

In this section, we will present another parallel Gaussian process called *parallel incomplete Cholesky factorization* (pICF)-based GP approximation that can distribute the computational load into a cluster of parallel machines to achieve efficient and scalable approximate GP regression by exploiting *incomplete Cholesky factorization* (ICF) technique.

4.2.1 Parallel Incomplete Cholesky Factorization

Incomplete Cholesky Factorization (ICF) can approximate a rank-N symmetric and positive semidefinite (PSD) matrix $\Sigma \in \mathcal{R}^{N\times N}$ by a low-rank PSD matrix LL^T where $L \in \mathcal{R}^{N\times R}$ is the lower triangular incomplete Cholesky factor ($R \ll N$). L can be obtained with an iterative ICF algorithm, the k-th iteration of which is as follows:

$$\begin{array}{lcl} [L]_{i_k,k} & \leftarrow & [v]_{i_k}^{-1/2} \\ [L]_{J_k,k} & \leftarrow & [\Sigma]_{J_k,k} - \sum_{j=1}^{k-1}[L]_{J_k,j}[L]_{i_k,j}/[L]_{i_k,k} \\ [v]_{J_k} & \leftarrow & [v]_{J_k} - [L]^2_{J_k,k}\ , \end{array} \tag{4.17}$$

where vector v is the diagonal of Σ, i_k is the index of pivot element in iteration k and J_k denotes $\{1, \cdots, N\} \backslash \{i_1, \cdots, i_k\}$. Note that, running the above ICF algorithm can generate the complete Cholesky factor of Σ.

ICF can in fact be parallelized: Instead of using a column-based parallel implementation [Golub and Van Loan, 1996], our proposed pICF-based GP employs a row-based parallel implementation [Chang *et al.*, 2007], the latter of which incurs lower time, space, and communication complexity.

4.2.2 pICF-based Parallel Gaussian Process

A fundamental step of pICF-based GP is to use ICF to approximate the covariance matrix $\Sigma_{\mathcal{DD}}$ in (3.1) and (3.2) of FGP by a low-rank symmetric positive semidefinite matrix: $\Sigma_{\mathcal{DD}} \approx F^\top F + \sigma_n^2 I$ where $F \in \mathbb{R}^{R \times |\mathcal{D}|}$ denotes the upper triangular incomplete Cholesky factor and $R \ll |\mathcal{D}|$ is the reduced rank. The steps of performing pICF-based GP are detailed as follows:

STEP 1: DISTRIBUTE DATA AMONG M MACHINES.

This step is the same as that of pPITC and pPIC in Section 4.1.

STEP 2: RUN PARALLEL ICF TO PRODUCE INCOMPLETE CHOLESKY FACTOR AND DISTRIBUTE ITS STORAGE.

The pICF-based GP exploits parallel ICF (Section 4.2.1) to produce an upper triangular incomplete Cholesky factor $F \triangleq (F_1 \cdots F_M)$ and each submatrix $F_m \in \mathbb{R}^{R \times |\mathcal{D}_m|}$ is stored distributedly on machine m for $m = 1, \ldots, M$.

STEP 3: EACH MACHINE CONSTRUCTS AND SENDS LOCAL SUMMARY TO MASTER.

Definition 7 (Local Summary). *Given the local data $(\mathcal{D}_m, y_{\mathcal{D}_m})$ and incomplete Cholesky factor F_m, the local summary of machine m is defined as a tuple $(\dot{y}_m, \dot{\Sigma}_m, \Phi_m)$ where*

$$\dot{y}_m \triangleq F_m(y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}) \tag{4.18}$$

$$\dot{\Sigma}_m \triangleq F_m \Sigma_{\mathcal{D}_m \mathcal{U}} \tag{4.19}$$

$$\Phi_m \triangleq F_m F_m^\top \, . \tag{4.20}$$

Step 4: Master constructs and sends global summary to M machines.

Definition 8 (Global Summary)**.** *Given the local summary $(\dot{y}_m, \dot{\Sigma}_m, \Phi_m)$ of every machine $m = 1, \ldots, M$, the global summary is defined as a tuple $(\ddot{y}, \ddot{\Sigma})$ where*

$$\ddot{y} \triangleq \Phi^{-1} \sum_{m=1}^{M} \dot{y}_m \tag{4.21}$$

$$\ddot{\Sigma} \triangleq \Phi^{-1} \sum_{m=1}^{M} \dot{\Sigma}_m \tag{4.22}$$

such that $\Phi \triangleq I + \sigma_n^{-2} \sum_{m=1}^{M} \Phi_m$.

Remark. If $|\mathcal{U}|$ is large, the computation of (4.22) can be parallelized by partitioning $\mathcal{U}$: Let $\dot{\Sigma}_m \triangleq (\dot{\Sigma}^1_m \cdots \dot{\Sigma}^M_m)$ where $\dot{\Sigma}^i_m \triangleq F_m \Sigma_{\mathcal{D}_m \mathcal{U}_i}$ is defined in a similar way as (4.19) and $|\mathcal{U}_i| = |\mathcal{U}|/M$. So, in Step 3, instead of sending $\dot{\Sigma}_m$ to the master, each machine m sends $\dot{\Sigma}^i_m$ to machine i for $i = 1, \ldots, M$. Then, each machine i computes and sends $\ddot{\Sigma}_i \triangleq \Phi^{-1} \sum_{m=1}^{M} \dot{\Sigma}^i_m$ to every other machine to obtain $\ddot{\Sigma} = (\ddot{\Sigma}_1 \cdots \ddot{\Sigma}_M)$.

Step 5: Each machine constructs and sends predictive component to master.

Definition 9 (Predictive Component)**.** *Given the local data $(\mathcal{D}_m, y_{\mathcal{D}_m})$, a component $\dot{\Sigma}_m$ of the local summary, and the global summary $(\ddot{y}, \ddot{\Sigma})$, the predictive component of machine m is defined as a tuple $(\widetilde{\mu}^m_{\mathcal{U}}, \widetilde{\Sigma}^m_{\mathcal{U}\mathcal{U}})$ where*

$$\widetilde{\mu}^m_{\mathcal{U}} \triangleq \sigma_n^{-2} \Sigma_{\mathcal{U}\mathcal{D}_m} (y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}) - \sigma_n^{-4} \dot{\Sigma}_m^{\top} \ddot{y} \tag{4.23}$$

$$\widetilde{\Sigma}^m_{\mathcal{U}\mathcal{U}} \triangleq \sigma_n^{-2} \Sigma_{\mathcal{U}\mathcal{D}_m} \Sigma_{\mathcal{D}_m \mathcal{U}} - \sigma_n^{-4} \dot{\Sigma}_m^{\top} \ddot{\Sigma}\ . \tag{4.24}$$

Step 6: Master performs predictions.

Definition 10 (pICF-based GP)**.** *Given the predictive component $(\widetilde{\mu}^m_{\mathcal{U}}, \widetilde{\Sigma}^m_{\mathcal{U}\mathcal{U}})$ of every machine $m = 1, \ldots, M$, the master computes a predictive Gaussian distribution $\mathcal{N}(\widetilde{\mu}_{\mathcal{U}}, \widetilde{\Sigma}_{\mathcal{U}\mathcal{U}})$ of the unobserved outputs for any set $\mathcal{U}$ of inputs where*

$$\widetilde{\mu}_{\mathcal{U}} \triangleq \mu_{\mathcal{U}} + \sum_{m=1}^{M} \widetilde{\mu}^{m}_{\mathcal{U}} \tag{4.25}$$

$$\widetilde{\Sigma}_{\mathcal{U}\mathcal{U}} \triangleq \Sigma_{\mathcal{U}\mathcal{U}} - \sum_{m=1}^{M} \widetilde{\Sigma}^{m}_{\mathcal{U}\mathcal{U}} \ . \tag{4.26}$$

Remark. The predictive performance of pICF-based GP can be improved by increasing the rank R at the expense of greater time, space, and communication complexity (Table 4.1).

4.2.3 Performance Guarantee

Theorem 3. *Let $\mathcal{N}(\mu^{\text{ICF}}_{\mathcal{U}|\mathcal{D}}, \Sigma^{\text{ICF}}_{\mathcal{U}\mathcal{U}|\mathcal{D}})$ be the predictive Gaussian distribution computed by the centralized ICF approximation of FGP model where*

$$\mu^{\text{ICF}}_{\mathcal{U}|\mathcal{D}} \triangleq \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{D}}(F^{\top}F + \sigma_n^2 I)^{-1}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) \tag{4.27}$$

$$\Sigma^{\text{ICF}}_{\mathcal{U}\mathcal{U}|\mathcal{D}} \triangleq \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{D}}(F^{\top}F + \sigma_n^2 I)^{-1}\Sigma_{\mathcal{D}\mathcal{U}} \ . \tag{4.28}$$

Then, $\widetilde{\mu}_{\mathcal{U}} = \mu^{\text{ICF}}_{\mathcal{U}|\mathcal{D}}$ and $\widetilde{\Sigma}_{\mathcal{U}\mathcal{U}} = \Sigma^{\text{ICF}}_{\mathcal{U}\mathcal{U}|\mathcal{D}}$.

The proof of Theorem 3 is given in Appendix C.

Remark 1. The equivalence result of Theorem 3 implies that the computational load of the centralized ICF approximation of FGP can be distributed among the M parallel machines, hence improving the time efficiency and scalability of approximate GP regression (Table 4.1).

Remark 2. By approximating the covariance matrix $\Sigma_{\mathcal{D}\mathcal{D}}$ in (3.1) and (3.2) of FGP with $F^{\top}F + \sigma_n^2 I$, $\widetilde{\Sigma}_{\mathcal{U}\mathcal{U}} = \Sigma^{ICF}_{\mathcal{U}\mathcal{U}|\mathcal{D}}$ is not guaranteed to be positive semidefinite, hence rendering such a measure of predictive uncertainty not very useful. However, it is observed in our experiments (Section 4.5) that this problem can be alleviated by choosing a sufficiently large rank R.

4.3 Analytical Comparison

This section compares and contrasts the properties of the proposed parallel GPs analytically (A summary of the empirical comparison is presented in Section 4.5.4).

4.3.1 Time, Space, and Communication Complexity

Table 4.1 shows an analytical comparison of the time, space, and communication complexity between *parallel partially independent training conditional* (pPITC), *parallel partially independent conditional* (pPIC), *parallel incomplete Cholesky factorization* (pICF)-based GP, *partially independent training conditional* (PITC), *partially independent conditional* (PIC), *incomplete Cholesky factorization* (ICF)-based GP, and *full Gaussian process* (FGP) based on the following assumptions: (a)These respective methods compute the predictive means (i.e., $\widehat{\mu}_{\mathcal{U}}$ (5.1), $\widehat{\mu}^{+}_{\mathcal{U}}$ (4.7), $\widetilde{\mu}_{\mathcal{U}}$ (4.25), $\mu^{\text{PITC}}_{\mathcal{U}|\mathcal{D}}$ (4.10), $\mu^{\text{PIC}}_{\mathcal{U}|\mathcal{D}}$ (4.13), $\mu^{\text{ICF}}_{\mathcal{U}|\mathcal{D}}$ (4.27), and $\mu_{\mathcal{U}|\mathcal{D}}$ (3.1)) and their corresponding predictive variances (i.e., $\widehat{\Sigma}_{xx}$ (5.2), $\widehat{\Sigma}^{+}_{xx}$ (4.8), $\widetilde{\Sigma}_{xx}$ (4.26), $\Sigma^{\text{PITC}}_{xx|\mathcal{D}}$ (4.11), $\Sigma^{\text{PIC}}_{xx|\mathcal{D}}$ (4.14), $\Sigma^{\text{ICF}}_{xx|\mathcal{D}}$ (4.28), and $\Sigma_{xx|\mathcal{D}}$ (3.2) for all $x \in \mathcal{U}$); (b) $|\mathcal{U}| < |\mathcal{D}|$ and recall $|\mathcal{S}|, R \ll |\mathcal{D}|$; (c) the data is already distributed among the M parallel machines for pPITC, pPIC, and pICF-based GP; and (d) for MPI, a broadcast operation in the communication network of M machines incurs $\mathcal{O}(\log M)$ messages [Pjesivac-Grbovic *et al.*, 2007]. The observations are as follows:

(a) Our pPITC, pPIC, and pICF-based GP improve the scalability of their centralized counterparts (respectively, PITC, PIC, and ICF-based GP) in the size $|\mathcal{D}|$ of data by distributing their computational loads among the M parallel machines.

(b) The speedups of pPITC, pPIC, and pICF-based GP over their centralized counterparts deviate further from the ideal speedup with an increasing number M of machines due to their additional $\mathcal{O}(|\mathcal{S}|^2 M)$ or $\mathcal{O}(R^2 M)$ time.

(c) The speedups of pPITC and pPIC grow with increasing size $|\mathcal{D}|$ of data because, unlike the additional $\mathcal{O}(|\mathcal{S}|^2|\mathcal{D}|)$ time of PITC and PIC that increase with more data, they do not have corresponding $\mathcal{O}(|\mathcal{S}|^2|\mathcal{D}|/M)$ terms.

(d) pPIC incurs additional $\mathcal{O}(|\mathcal{D}|)$ time and $\mathcal{O}((|\mathcal{D}|/M)\log M)$-sized messages over pPITC due to its parallelized clustering (see Remark 2 after Definition 6).

(e) Keeping the other variables fixed, an increasing number M of machines reduces the time and space complexity of pPITC and pPIC at a faster rate than pICF-based GP while increasing size $|\mathcal{D}|$ of data raises the time and space complexity of pICF-based GP at a slower rate than pPITC and pPIC.

(f) pICF-based GP distributes the memory requirement of ICF-based GP among the M parallel machines.

(g) The communication complexity of pICF-based GP depends on the number $|\mathcal{U}|$ of predictions whereas that of pPITC and pPIC are independent of it.

Table 4.1: Comparison of time & space complexity between pPITC, pPIC, pICF-based GP, PITC, PIC, ICF, and FGP. (Note that PITC, PIC, and ICF-based GP are, respectively, the centralized counterparts of pPITC, pPIC, and pICF, as proven in Theorems 1, 2 and 3.)

GP	Time complexity	Space complexity
p**PITC**	$\mathcal{O}\left(\|\mathcal{S}\|^2\left(\|\mathcal{S}\|+M+\frac{\|\mathcal{U}\|}{M}\right)+\left(\frac{\|\mathcal{D}\|}{M}\right)^3\right)$	$\mathcal{O}\left(\|\mathcal{S}\|^2+\left(\frac{\|\mathcal{D}\|}{M}\right)^2\right)$
p**PIC**	$\mathcal{O}\left(\|\mathcal{S}\|^2\left(\|\mathcal{S}\|+M+\frac{\|\mathcal{U}\|}{M}\right)+\left(\frac{\|\mathcal{D}\|}{M}\right)^3+\|\mathcal{D}\|\right)$	$\mathcal{O}\left(\|\mathcal{S}\|^2+\left(\frac{\|\mathcal{D}\|}{M}\right)^2\right)$
p**ICF**	$\mathcal{O}\left(R^2\left(R+M+\frac{\|\mathcal{D}\|}{M}\right)+R\|\mathcal{U}\|\left(M+\frac{\|\mathcal{D}\|}{M}\right)\right)$	$\mathcal{O}\left(R^2+R\frac{\|\mathcal{D}\|}{M}\right)$
PITC	$\mathcal{O}\left(\|\mathcal{S}\|^2\|\mathcal{D}\|+\|\mathcal{D}\|\left(\frac{\|\mathcal{D}\|}{M}\right)^2\right)$	$\mathcal{O}\left(\|\mathcal{S}\|^2+\left(\frac{\|\mathcal{D}\|}{M}\right)^2\right)$
PIC	$\mathcal{O}\left(\|\mathcal{S}\|^2\|\mathcal{D}\|+\|\mathcal{D}\|\left(\frac{\|\mathcal{D}\|}{M}\right)^2+M\|\mathcal{D}\|\right)$	$\mathcal{O}\left(\|\mathcal{S}\|^2+\left(\frac{\|\mathcal{D}\|}{M}\right)^2\right)$
ICF	$\mathcal{O}\left(R^2\|\mathcal{D}\|+R\|\mathcal{U}\|\|\mathcal{D}\|\right)$	$\mathcal{O}(R\|\mathcal{D}\|)$
FGP	$\mathcal{O}\left(\|\mathcal{D}\|^3\right)$	$\mathcal{O}\left(\|\mathcal{D}\|^2\right)$

4.3.2 Online/Incremental Learning

Supposing new data $(\mathcal{D}', y_{\mathcal{D}'})$ becomes available, pPITC and pPIC do not have to run Steps 1 to 4 (Section 4.1) on the entire data $(\mathcal{D}\bigcup\mathcal{D}', y_{\mathcal{D}\bigcup\mathcal{D}'})$. The lo-

Table 4.2: Comparison of communication complexity between parallel GP algorithms: pPITC, pPIC, pICF-based GP

GP	Communication complexity
p**PITC**	$\mathcal{O}(\|\mathcal{S}\|^2 \log M)$
p**PIC**	$\mathcal{O}\left(\left(\|\mathcal{S}\|^2 + \frac{\|\mathcal{D}\|}{M}\right) \log M\right)$
p**ICF**	$\mathcal{O}((R^2 + R\|\mathcal{U}\|) \log M)$

cal and global summaries of the old data $(\mathcal{D}, y_{\mathcal{D}})$ can in fact be reused and assimilated with that of the new data, thus saving the need of recomputing the computationally expensive matrix inverses in (4.1) and (4.2) for the old data. The exact mathematical details are omitted due to lack of space. As a result, the time complexity of pPITC and pPIC can be greatly reduced in situations where new data is expected to stream in at regular intervals. In contrast, pICF-based GP does not seem to share this advantage.

4.3.3 Structural Assumptions

The above advantage of online learning for pPITC and pPIC results from their assumptions of conditional independence given the support set. With fewer machines, such an assumption is violated less, thus potentially improving their predictive performances. In contrast, the predictive performance of pICF-based GP is not affected by varying the number of machines. However, it suffers from a different problem: Utilizing a reduced-rank matrix approximation of $\Sigma_{\mathcal{D}\mathcal{D}}$, its resulting predictive covariance matrix $\widetilde{\Sigma}_{\mathcal{U}\mathcal{U}}$ is not guaranteed to be positive semidefinite (see Remark 2 after Theorem 3), thus being problematic in measuring predictive uncertainty. However, as our experiments (Section 4.5) show, this problem can be alleviated by choosing a sufficiently large rank R.

4.4 Experimental Setup

This section empirically evaluates the predictive performances, time efficiency, scalability, and speedups of our proposed parallel GPs against their centralized counterparts and FGP.

4.4.1 Settings

The experiments are performed on two real-world datasets: (a) The AIMPEAK dataset of size $|\mathcal{D}| = 41850$ contains traffic speeds (km/h) along 775 road segments of an urban road network (including highways, arterials, slip roads, etc.) during the morning peak hours (6-10:30 a.m.) on April 20, 2011. The traffic speeds are the outputs. The mean speed is 49.5 km/h and the standard deviation is 21.7 km/h. Each input (i.e., road segment) is specified by a 5-dimensional vector of features: length, number of lanes, speed limit, direction, and time. The time dimension comprises 54 five-minute time slots. This spatiotemporal traffic phenomenon is modeled using a relational GP (Section 3.3) whose correlation structure can exploit both the road segment features and road network topology information; (b) The SARCOS dataset [Vijayakumar *et al.*, 2005] of size $|\mathcal{D}| = 48933$ pertains to an inverse dynamics problem for a seven degrees-of-freedom SARCOS robot arm. Each input denotes a 21-dimensional vector of features: 7 joint positions, 7 joint velocities, and 7 joint accelerations. Only one of the 7 joint torques is used as the output. The mean torque is 13.7 and the standard deviation is 20.5.

In our setting, both datasets are modeled as GP specified by a squared exponential covariance function[2]:

$$\sigma_{xx'} \triangleq \sigma_s^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i - x'_i}{\ell_i}\right)^2\right) + \sigma_n^2 \delta_{xx'}$$

where x_i (x'_i) is the i-th component of the input feature vector x (x'), the hyperparameters $\sigma_s^2, \sigma_n^2, \ell_1, \ldots, \ell_d$ are, respectively, signal variance, noise variance, and length-scales; and $\delta_{xx'}$ is a Kronecker delta that is 1 if $x = x'$ and 0 otherwise. The hyperparameters are learned using randomly selected data of size 10000 via maximum likelihood estimation [Rasmussen and Williams, 2006].

For each dataset, 10% of the data is randomly selected as test data for predictions (i.e., as $\mathcal{U}$). From the remaining data, training data of varying sizes $|\mathcal{D}| = 8000$, 16000, 24000, and 32000 are randomly selected. The training data are distributed into M machines according to the simple parallelized clustering scheme (see in Remark 2 after Definition (5)). In addition, pPITC and pPIC are

[2] In AIMPEAK dataset, the domain of road segments is embedded into the Euclidean space using multi-dimensional scaling [Borg and Groenen, 2005] so that a squared exponential covariance function can be applied.

evaluated using support sets of varying sizes $|\mathcal{S}| = 256$, 512, 1024, and 2048. These support sets are selected via differential entropy score criterion (see in Remark after Definition (3)). pICF-based GP is evaluated using varying reduced ranks R of the same values as $|\mathcal{S}|$ in the AIMPEAK domain and twice the values of $|\mathcal{S}|$ in the SARCOS domain.

Our experimental platform is a cluster of 20 computing nodes connected via gigabit links: Each node runs a Linux system with Intel® Xeon® CPU E5520 at 2.27 GHz and 20 GB memory. Our parallel GPs are tested with different number $M = 4$, 8, 12, 16, and 20 of computing nodes.

4.4.2 Performance Metrics

The tested GP regression methods are evaluated with four different performance metrics: (a) Root mean square error (RMSE) $\sqrt{|\mathcal{U}|^{-1}\sum_{x\in\mathcal{U}}\left(y_x - \mu_{x|\mathcal{D}}\right)^2}$; (b) mean negative log probability (MNLP)[3] $0.5|\mathcal{U}|^{-1}\sum_{x\in\mathcal{U}}\left((y_x - \mu_{x|\mathcal{D}})^2/\Sigma_{xx|\mathcal{D}} + \log(2\pi\Sigma_{xx|\mathcal{D}})\right)$ [Rasmussen and Williams, 2006]; (c) incurred time; and (d) speedup is defined as the incurred time of a sequential/centralized algorithm divided by that of its corresponding parallel algorithm. For the first two metrics, the tested methods have to plug their predictive mean and variance into $\mu_{x|\mathcal{D}}$ and $\Sigma_{xx|\mathcal{D}}$, respectively.

4.5 Results and Analysis

In this section, we analyze the results that are obtained by averaging over 5 random instances.

4.5.1 Varying Size of Data

Figure 4.1 demonstrates the results obtained from experiments by varying size of training data $|\mathcal{D}| = 8000$, 16000, 24000, and 32000.

Figures 4.1a-b and 4.1e-f show that the predictive performances of our parallel GPs improve with more data and are comparable to that of FGP, hence justifying the practicality of their inherent structural assumptions.

[3] MNLP and RMSE are both commonly used as metrics to evaluate the predictive accuracy. Unlike RMSE that uses squared residual to evaluate predictive mean at test points, MNLP exploits negative log probability to evaluate both predictive variance and predictive mean.

From Figures 4.1e-f, it can be observed that the predictive performance of pICF-based GP is very close to that of FGP when $|\mathcal{D}|$ is relatively small (i.e., $|\mathcal{D}| = 8000$, 16000). But, its performance approaches that of pPIC as $|\mathcal{D}|$ increases further because the reduced rank $R = 4096$ of pICF-based GP is not large enough (relative to $|\mathcal{D}|$) to maintain its close performance to FGP. In addition, pPIC achieves better predictive performance than pPITC since the former can exploit local information (see Remark 1 after Definition 6).

Figures 4.1c and 4.1g indicate that our parallel GPs are significantly more time-efficient and scalable than FGP (i.e., 1-2 orders of magnitude faster) while achieving comparable predictive performance. Among the three parallel GPs, pPITC and pPIC are more time-efficient and thus more capable of meeting the real-time prediction requirement of a time-critical application/system.

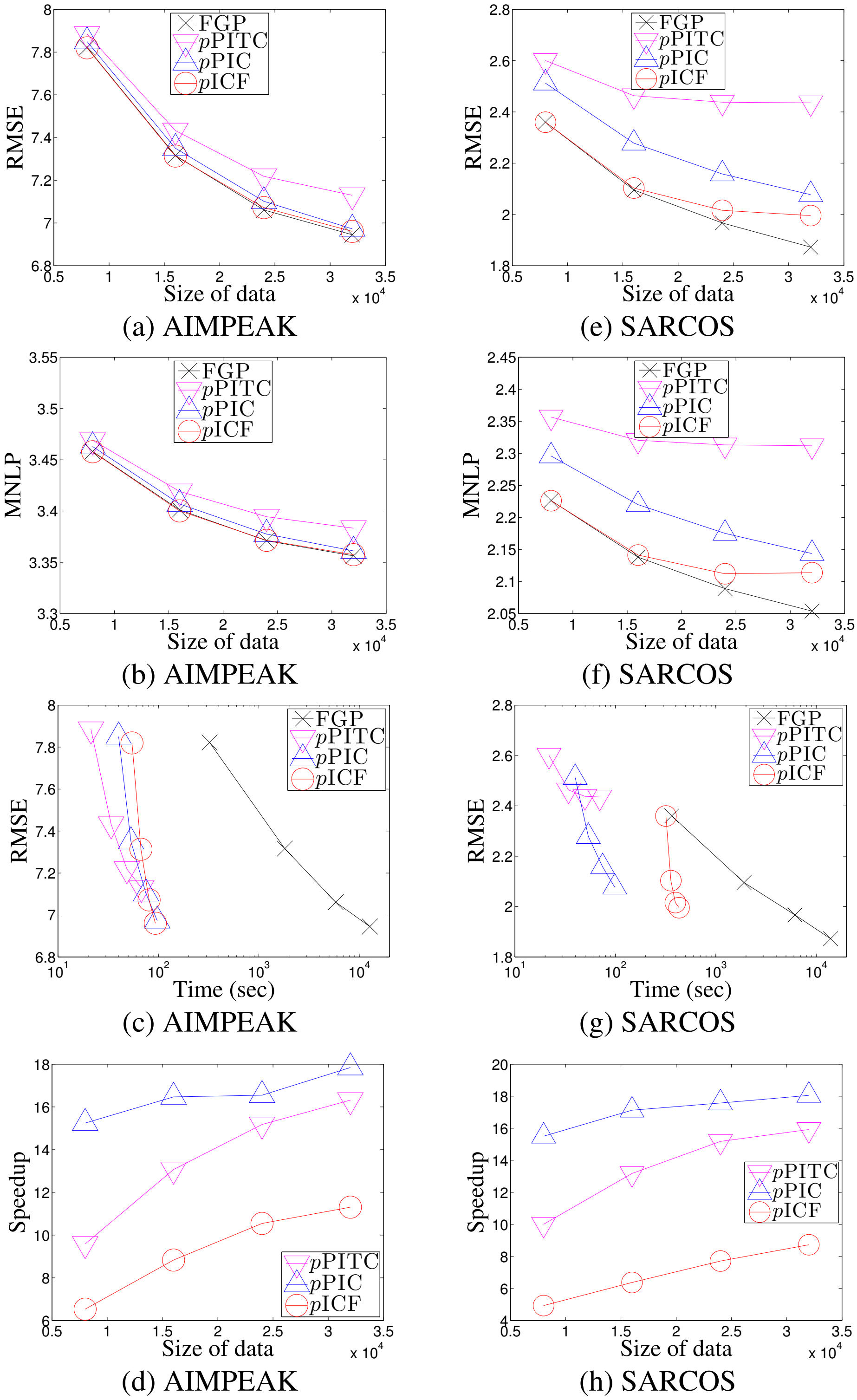
Figures 4.1d and 4.1h show that the speedups of our parallel GPs over their centralized counterparts increase with more data, which agree with observation c in Section 4.3.1. pPITC and pPIC achieve better speedups than pICF-based GP.

4.5.2 Varying Number of Machines

Figure 4.2 demonstrates the results obtained from experiments by varying number $M = 4$, 8, 12, 16, and 20 of computing nodes.

Figures 4.2a-b and 4.2e-f show that pPIC and pICF-based GP achieve predictive performance comparable to that of FGP with different number M of machines. pPIC achieves better predictive performance than pPITC due to its use of local information (see Remark 1 after Definition 6).

From Figures 4.2e-f, it can be observed that as the number M of machines increases, the predictive performance of pPIC drops slightly due to smaller size of local data $\mathcal{D}_m$ assigned to each machine. In contrast, the predictive performance of pPITC improves: If the number M of machines is small as compared to the actual number of clusters in the data, then the clustering scheme (see Remark 2 after Definition 6) may assign data from different clusters to the same machine or data from the same cluster to multiple machines. Consequently, the conditional independence assumption is violated. Such an issue is mitigated by increasing the number M of machines to achieve better clustering, hence resulting in better predictive performance.



(a) AIMPEAK

(e) SARCOS

(b) AIMPEAK

(f) SARCOS

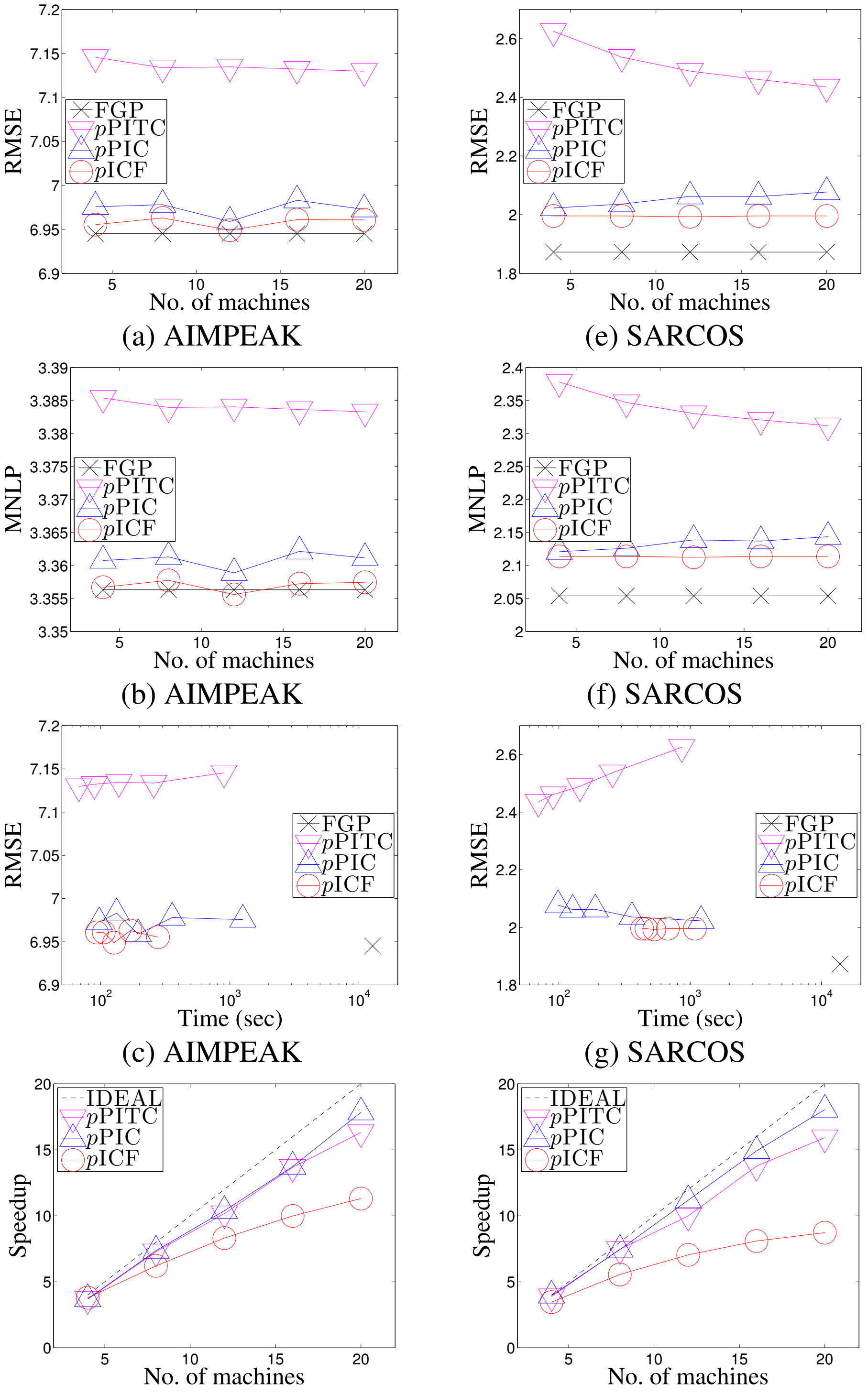
(c) AIMPEAK

(g) SARCOS

(d) AIMPEAK

(h) SARCOS

Figure 4.1: Performance of parallel GPs with varying data sizes $|\mathcal{D}| = 8000$, 16000, 24000, and 32000, number $M = 20$ of machines, support set size $|\mathcal{S}| = 2048$, and reduced rank $R = 2048$ (4096) in the AIMPEAK (SARCOS) domain.



(a) AIMPEAK (e) SARCOS

(b) AIMPEAK (f) SARCOS

(c) AIMPEAK (g) SARCOS

(d) AIMPEAK (h) SARCOS

Figure 4.2: Performance of parallel GPs with varying number $M = 4$, 8, 12, 16, 20 of machines, data size $|\mathcal{D}| = 32000$, support set size $\mathcal{S} = 2048$, and reduced rank $R = 2048$ (4096) in the AIMPEAK (SARCOS) domain. The ideal speedup of a parallel algorithm equals to the number M of machines running the algorithm.

Figures 4.2c and 4.2g show that pPIC and pICF-based GP are significantly more time-efficient than FGP (i.e., 1-2 orders of magnitude faster) while achieving comparable predictive performance. This is previously explained in the analysis of their time complexity (Table 4.1).

Figures 4.2c and 4.2g also reveal that as the number M of machines increases, the incurred time of pPITC and pPIC decreases at a faster rate than that of pICF-based GP, which agree with observation e in Section 4.3.1. Hence, we expect pPITC and pPIC to be more time-efficient than pICF-based GP when the number M of machines increases beyond 20.

Figures 4.2d and 4.2h show that the speedups of our parallel GPs over their centralized counterparts deviate further from the ideal speedup with a greater number M of machines, which agree with observation b in Section 4.3.1. The speedups of pPITC and pPIC are closer to the ideal speedup than that of pICF-based GP.

4.5.3 Varying Size of Support Set/Reduced Rank

Figure 4.3 demonstrates the results obtained from experiments by varying size of support set (reduced rank) of pPITC/pPIC (pICF-based GP) algorithms.

Figures 4.3a and 4.3e show that the predictive performance of pICF-based GP is extremely poor when the reduced rank R is not large enough (relative to $|\mathcal{D}|$), thus resulting in a poor ICF approximation of the covariance matrix $\Sigma_{\mathcal{D}\mathcal{D}}$. In addition, it can be observed that the reduced rank R of pICF-based GP needs to be much larger than the support set size $|\mathcal{S}|$ of pPITC and pPIC in order to achieve comparable predictive performance. These results also indicate that the heuristic $R = \sqrt{|\mathcal{D}|}$, which is used by [Chang *et al.*, 2007] to determine the reduced rank R, fails to work well in both our datasets (e.g., $R = 1024 > \sqrt{32000} \approx 179$).

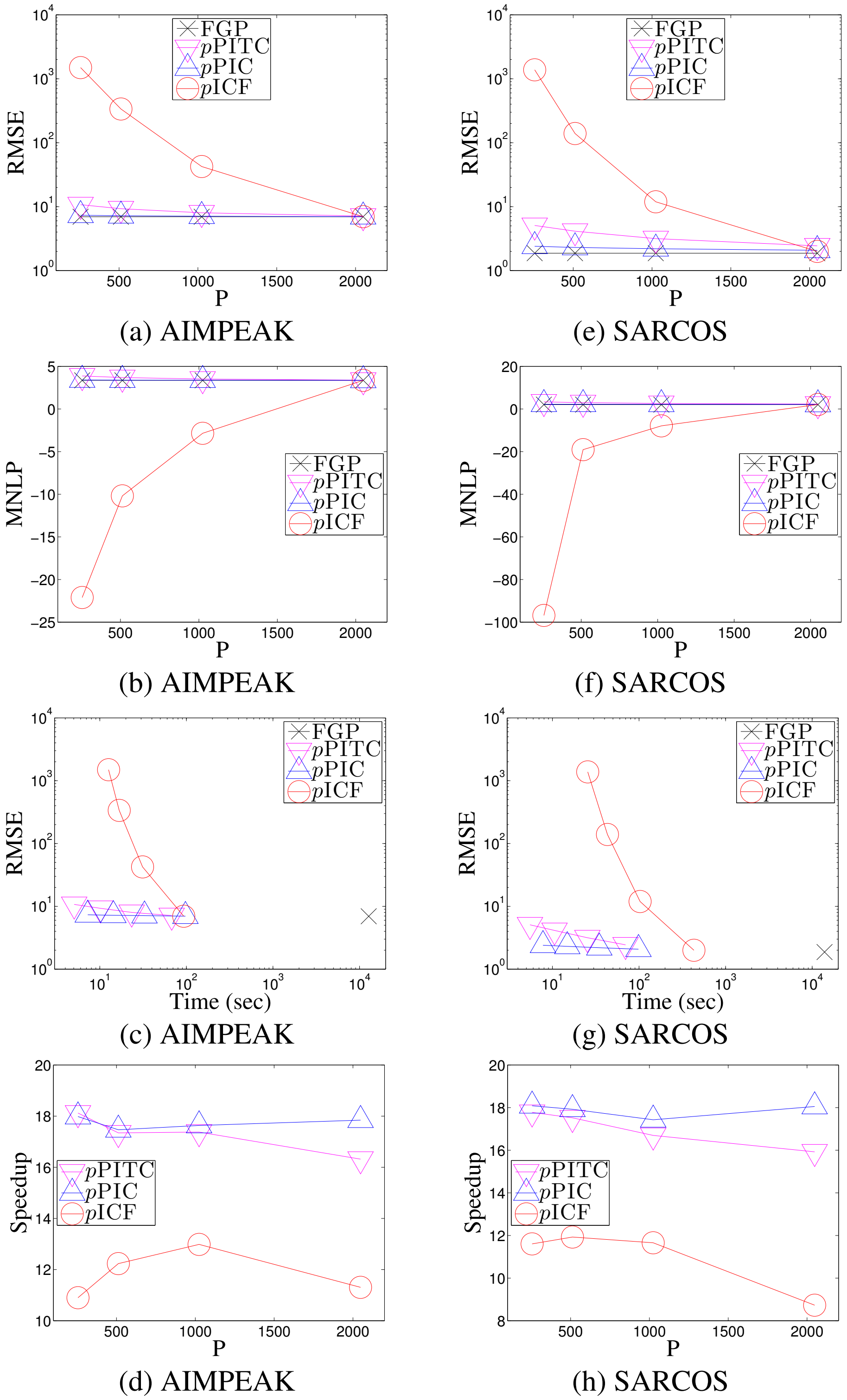
From Figures 4.3b and 4.3f, it can be observed that pICF-based GP incurs negative MNLP for $R \leq 1024$ ($R \leq 2048$) in the AIMPEAK (SARCOS) domain. This is because pICF-based GP cannot guarantee positivity of predictive variance, as explained in Remark 2 after Theorem 3. But, it appears that when R is sufficiently large (i.e., $R = 2048$ ($R = 4096$) in the AIMPEAK (SARCOS) domain), this problem can be alleviated.

It can be observed in Figures 4.3c and 4.3g that pPITC and pPIC are significantly more time-efficient than FGP (i.e., 2-4 orders of magnitude faster) while achieving comparable predictive performance. To ensure high predictive performance, pICF-based GP has to select a large enough rank $R = 2048$ ($R = 4096$) in the AIMPEAK (SARCOS) domain, thus making it less time-efficient than pPITC and pPIC. But, it can still incur 1-2 orders of magnitude less time than FGP. These results indicate that pPITC and pPIC are more capable than pICF-based GP of meeting the real-time prediction requirement of a time-critical application/system.

Figures 4.3d and 4.3h show that pPITC and pPIC achieve better speedups than pICF-based GP.

4.5.4 Summary of Results

pPIC and pICF-based GP are significantly more time-efficient and scalable than FGP (i.e., 1-4 orders of magnitude faster) while achieving comparable predictive performance, hence justifying the practicality of their structural assumptions. pPITC and pPIC are expected to be more time-efficient than pICF-based GP with an increasing number M of machines because their incurred time decreases at a faster rate than that of pICF-based GP. Since the predictive performances of pPITC and pPIC drop slightly (i.e., more stable) with smaller $|\mathcal{S}|$ as compared to that of pICF-based GP dropping rapidly with smaller R, pPITC and pPIC are more capable than pICF-based GP of meeting the real-time prediction requirement of a time-critical application/system. The speedups of our parallel GPs over their centralized counterparts improve with more data but deviate further from the ideal speedup with a larger number of machines.



(a) AIMPEAK

(e) SARCOS

(b) AIMPEAK

(f) SARCOS

(c) AIMPEAK

(g) SARCOS

(d) AIMPEAK

(h) SARCOS

Figure 4.3: Performance of parallel GPs with data size $|\mathcal{D}| = 32000$, number $M = 20$ of machines, and varying parameter $P = 256, 512, 1024, 2048$ where $P = |\mathcal{S}| = R$ ($P = |\mathcal{S}| = R/2$) in the AIMPEAK (SARCOS) domain.

Chapter 5

Decentralized Data Fusion & Active Sensing

Towards understanding the large-scale spatiotemporal traffic phenomenon with active mobile sensors, this chapter aims to address the question: how do the mobile sensors actively explore an urban network to gather and assimilate the most informative phenomenon data for predicting a spatiotemporal traffic phenomena? To achieve this goal, first, we can represent the spatiotemporal traffic phenomena with Gaussian Process models (Chapter 3); then, the mobile sensors can distributedly gather and assimilate the traffic data to model and predict the traffic phenomena. However, the full Gaussian process model suffers from cubic time complexity in the size of data. To alleviate this limitation, we adapt the parallel GP techniques (Chapter 4) into *decentralized data fusion* (DDF) algorithms (Section 5.1) that can distribute computational load among mobile sensors; thereby achieving better efficiency and scalability in assimilating distributed traffic data into a globally consistent model. Furthermore, a set of *decentralized active sensing* (DAS) algorithms (Section 5.2) are developed to guild mobile sensors to cooperatively collect the most informative traffic data; thereby reducing the size of data required to achieve comparable predictive accuracy. The DDF algorithms coupled with DAS algorithms form our decentralized algorithm framework: *Gaussian process-based decentralized data fusion and active sensing* (D^{2}FAS) [Chen *et al.*, 2012; Chen *et al.*, 2013b].

5.1 Decentralized Data Fusion

Since the gathered traffic phenomenon data (e.g., traffic speeds on road segments, urban mobility demand data) are distributed among the mobile sensors, data fusion techniques are required to assimilate these distributed phenomenon data into a global predictive model. A straightforward approach to data fusion is to fully communicate all the data to every vehicle, each of which then performs the same exact GP prediction (3.1) separately. These approaches, which we call *full Gaussian process* (FGP) [Low *et al.*, 2008b; Low *et al.*, 2009b], unfortunately cannot scale well and be performed in real time due to its cubic time complexity in the size of the data. To alleviate this issue, the parallel GP techniques (Chapter 4) provide a valuable perspective that the computational load can be distributed to each mobile sensor; thereby achieving better efficiency and scalability than that of the centralized GP models. Therefore, we adapt pPITC (Section 4.1.1) and pPIC (Section 4.1.2) into two decentralized data fusion algorithms that can be applied in mobile sensor network; they are *Gaussian process-based decentralized data fusion* (GP-DDF) algorithm [Chen *et al.*, 2012] (Section 5.1.1) and *Gaussian Process-based Decentralized Data Fusion with Local Augmentation* (GP-DDF$^+$) algorithm [Chen *et al.*, 2013b] (Section 5.1.2).

5.1.1 Gaussian Process-based Decentralized Data Fusion

The intuition to our GP-DDF algorithm is as follows: Each of the K mobile sensors constructs a local summary of the observations taken along its own path in an environmental field and communicates its local summary to every other sensor. Then, it assimilates the local summaries received from the other sensors into a globally consistent summary, which is exploited for predicting the traffic phenomenon as well as active sensing. This intuition will be formally realized and described in subsequent sections.

5.1.1.1 Local & Global Summary

While exploring the field, each mobile sensor summarizes its local observations taken along its path based on a common support set $\mathcal{S} \subset V$ known to all the other sensors. Its local summary is defined as follows:

Definition 11 (Local Summary)**.** *Given a common support set $\mathcal{S} \subset V$ known to all K mobile sensors, a set $\mathcal{D}_k \subset V$ of observed inputs and a column vector $y_{\mathcal{D}_k}$ of corresponding measurements local to mobile sensor k, its local summary is defined as a tuple $(\dot{y}^k_{\mathcal{S}}, \dot{\Sigma}^k_{\mathcal{SS}})$ where $\dot{y}^k_{\mathcal{S}}$ and $\dot{\Sigma}^k_{\mathcal{SS}}$ are defined in the same manner to (4.1) & (4.2).*

Remark. Unlike SoD (Section 3.2), the support set $\mathcal{S}$ of inputs does not have to be observed since the local summary (i.e., (4.1) and (4.2)) is independent of the corresponding measurements $y_{\mathcal{S}}$. So, $\mathcal{S}$ does not need to be a subset of $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$, while the support set of SoD has to be selected from $\mathcal{D}$. To select an informative support set $\mathcal{S}$ from the set V of all possible segments in the road network, an offline active selection procedure similar to that in Section 3.2 can be performed just once prior to observing data to determine $\mathcal{S}$. In contrast, SoD has to perform online active selection every time when new observations are being gathered.

By communicating its local summary to every other sensor, each mobile sensor can then construct a globally consistent summary from the received local summaries:

Definition 12 (Global Summary)**.** *Given a common support set $\mathcal{S} \subset V$ known to all K mobile sensors and the local summary $(\dot{y}^k_{\mathcal{S}}, \dot{\Sigma}^k_{\mathcal{SS}})$ of every mobile sensor $k = 1, \ldots, K$, the global summary is defined as a tuple $(\ddot{y}_{\mathcal{S}}, \ddot{\Sigma}_{\mathcal{SS}})$ where $\ddot{y}_{\mathcal{S}}$ and $\ddot{\Sigma}_{\mathcal{SS}}$ are defined in the same manner to (4.3) & (4.4).*

Remark. This thesis assumes all-to-all communication between the K mobile sensors. Supposing this is not possible and each sensor can only communicate locally with its neighbors, the summation structure of the global summary (specifically, (4.3) and (4.4)) makes it amenable to be constructed using distributed consensus filters [Olfati-Saber and Shamma, 2005]. We omit these details since they are beyond the scope of this thesis.

5.1.1.2 Global Predictive Model

GP-DDF algorithm can exploit the global summary to compute a globally consistent predictive Gaussian distribution detailed as below:

Definition 13 (GP-DDF)**.** *Given a common support set $\mathcal{S} \subset V$ known to all K mobile sensors, the global summary $(\ddot{y}_{\mathcal{S}}, \ddot{\Sigma}_{\mathcal{S}\mathcal{S}})$, each mobile sensor computes a globally consistent predictive Gaussian distribution $\mathcal{N}(\widehat{\mu}_{\mathcal{U}}, \widehat{\Sigma}_{\mathcal{U}\mathcal{U}})$ of the measurements at any set $\mathcal{U}$ of unobserved inputs where*

$$\widehat{\mu}_{\mathcal{U}} \triangleq \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}} \ddot{y}_{\mathcal{S}} \tag{5.1}$$

$$\widehat{\Sigma}_{\mathcal{U}\mathcal{U}} \triangleq \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{S}} (\Sigma^{-1}_{\mathcal{S}\mathcal{S}} - \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}) \Sigma_{\mathcal{S}\mathcal{U}} \, . \tag{5.2}$$

According to Appendix A, the predictive distribution $\mathcal{N}(\widehat{\mu}_{\mathcal{U}}, \widehat{\Sigma}_{\mathcal{U}\mathcal{U}})$ computed by GP-DDF is proved to be equivalent to the predictive Gaussian distribution $\mathcal{N}(\mu^{\textit{PITC}}_{\mathcal{U}|\mathcal{D}}, \Sigma^{\textit{PITC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}})$ computed by the centralized partially independent training conditional (PITC) approximation of GP model [Quiñonero-Candela and Rasmussen, 2005] where $\mu^{\textit{PITC}}_{\mathcal{U}|\mathcal{D}}$ and $\Sigma^{\textit{PITC}}_{\mathcal{U}\mathcal{U}|\mathcal{D}}$ are defined in (4.10) and (4.11), respectively. The equivalence result bears two implications:

First, the computational load of the centralized PITC approximation of GP model can be distributed among K mobile sensors, thereby improving the time efficiency of prediction. Specifically, supposing $|\mathcal{U}| \leq |\mathcal{S}|$ for simplicity, the $\mathcal{O}\big(|\mathcal{D}|((|\mathcal{D}|/K)^2 + |\mathcal{S}|^2)\big)$ time incurred by PITC can be reduced to $\mathcal{O}\big((|\mathcal{D}|/K)^3 + |\mathcal{S}|^3 + |\mathcal{S}|^2 K\big)$ time of running our decentralized algorithm on each of the K sensors, the latter of which scales better with increasing number $|\mathcal{D}|$ of observations.

Second, we can draw insights from PITC to elucidate an underlying property of our decentralized algorithm: It is assumed that $Y_{\mathcal{D}_1}, \ldots, Y_{\mathcal{D}_K}, Y_{\mathcal{U}}$ are conditionally independent given the measurements at the support set $\mathcal{S}$. To potentially reduce the degree of violation of this assumption, an informative support set $\mathcal{S}$ is actively selected, as described earlier in this section. Furthermore, the experimental results on real-world urban road network data[1] (Section 6.4) show that GP-DDF can achieve predictive performance comparable to that of the full GP model while enjoying significant computational gain over it, thus demonstrating the practicality of such an assumption for predicting traffic phenomena. The predictive performance of GP-DDF can be improved by increasing the size of $\mathcal{S}$ at the expense of greater time and communication overhead.

[1] [Quiñonero-Candela and Rasmussen, 2005] only illustrated the predictive performance of PITC on a simulated toy example.

5.1.2 Gaussian Process-based Decentralized Data Fusion with Local Augmentation

Though GP-DDF scales very well with large data, it can predict poorly due to (a) loss of information caused by summarizing the measurements and correlation structure of the original data; and (b) sparse coverage of the hotspots (i.e., with higher spatiotemporal variability) by the support set.

To address this issue, this section proposes a novel DDF algorithm called *Gaussian process-based decentralized data fusion with local augmentation* (GP-DDF$^+$) that can achieve better predictive accuracy while preserving the efficiency of GP-DDF; GP-DDF$^+$ is based on the intuition that a mobile sensor can exploit its local data to improve the predictions for unobserved inputs "close" to its data (in the correlation sense).

5.1.2.1 Local Predictive Model

Using GP-DDF (Section 5.1.1), each mobile sensor exploits the global summary to compute a globally consistent predictive Gaussian distribution of the measurements of any set of unobserved inputs. To improve the predictive power of GP-DDF, we develop the following GP-DDF^+_k algorithm that is further augmented by local information of mobile sensor k.

Definition 14 (GP-DDF^+_k)**.** *Given a common support set $\mathcal{U} \subset V$ known to all K mobile sensors, the global summary $(\ddot{y}_{\mathcal{S}}, \ddot{\Sigma}_{\mathcal{SS}})$, the local summary $(\dot{y}^k_{\mathcal{S}}, \dot{\Sigma}^k_{\mathcal{SS}})$, a set $\mathcal{D}_k \subset V$ of observed inputs and a column vector $y_{\mathcal{D}_k}$ of corresponding measurements local to mobile sensor k, its GP-DDF^+_k algorithm computes a predictive Gaussian distribution $\mathcal{N}(\widehat{\mu}^k_{\mathcal{U}}, \widehat{\Sigma}^k_{\mathcal{UU}})$ of the measurements of any set $\mathcal{U} \subset V$ of unobserved inputs where $\widehat{\mu}^k_{\mathcal{U}} \triangleq \left(\widehat{\mu}^k_s\right)_{s\in\mathcal{U}}$ and $\widehat{\Sigma}^k_{\mathcal{UU}} \triangleq \left(\widehat{\sigma}^k_{ss'}\right)_{s,s'\in\mathcal{U}}$ such that*

$$\widehat{\mu}^k_s \triangleq \mu_s + \left(\Phi^k_{s\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{SS}}\ddot{y}_{\mathcal{S}} - \Sigma_{s\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}}\dot{y}^k_{\mathcal{S}}\right) + \dot{y}^k_s \tag{5.3}$$

$$\begin{aligned}\widehat{\sigma}^k_{ss'} \triangleq \sigma_{ss'} - \Big(&\Phi^k_{s\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{S}s'} - \Sigma_{s\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}}\dot{\Sigma}^k_{\mathcal{S}s'} \\ &- \Phi^k_{s\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{SS}}\Phi^k_{\mathcal{S}s'}\Big) - \dot{\Sigma}^k_{ss'}\end{aligned} \tag{5.4}$$

and

$$\Phi^k_{s\mathcal{S}} \triangleq \Sigma_{s\mathcal{S}} + \Sigma_{s\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}}\dot{\Sigma}^k_{\mathcal{SS}} - \dot{\Sigma}^k_{s\mathcal{S}}\ . \tag{5.5}$$

Remark 1. Both the predictive Gaussian mean $\widehat{\mu}^k_s$ (5.3) and covariance $\widehat{\sigma}^k_{ss'}$(5.4) of GP-DDF^+_k exploit summary information (i.e., bracketed term) contributed from exchanged summaries among mobile sensors and local information (i.e., last term) contributed from local data.

Remark 2. Since different mobile sensors exploit different local data, their GP-DDF^+_k algorithms provide inconsistent predictions of the measurements.

5.1.2.2 Assignment Function

It is often desirable to achieve a globally consistent prediction of measurements among all mobile sensors. To do this, each unobserved input is simply assigned to the mobile sensor that predicts its measurement best, which can be performed in a decentralized way:

Definition 15 (Assignment Function)**.** *An assignment function $\tau : V \mapsto \{1 \ldots K\}$ is defined as*

$$\tau(s) \triangleq \mathop{\arg\min}_{k \in \{1 \ldots K\}} \widehat{\sigma}^k_{ss} \tag{5.6}$$

for all $s \in \mathcal{U}$ where the predictive variance $\widehat{\sigma}^k_{ss}$ is defined in (5.4). From now on, let $\tau_s \triangleq \tau(s)$ for notational simplicity.

5.1.2.3 Global Predictive Model

Using the assignment function τ, each mobile sensor can now compute a globally consistent predictive Gaussian distribution, as detailed in Definition 16 below:

Definition 16 (GP-DDF$^+$)**.** *Given a common support set $\mathcal{S} \subset V$ and a common assignment function τ be known to all K mobile sensors. The GP-DDF$^+$ algorithm of each mobile sensor computes a globally consistent predictive Gaussian distribution $\mathcal{N}(\widehat{\mu}^+_{\mathcal{U}}, \widehat{\Sigma}^+_{\mathcal{UU}})$ of the measurements of any set $\mathcal{U} \subset V$ of unobserved inputs where $\widehat{\mu}^+_{\mathcal{U}} \triangleq (\widehat{\mu}^{\tau_s}_s)_{s \in \mathcal{U}}$ (5.3) and $\widehat{\Sigma}^+_{\mathcal{UU}} \triangleq \left(\widehat{\sigma}^+_{ss'}\right)_{s,s' \in \mathcal{U}}$ such that*

$$\widehat{\sigma}^+_{ss'} \triangleq \begin{cases} \widehat{\sigma}^{\tau_s}_{ss'} & \textit{if } \tau_s = \tau_{s'}, \\ \Sigma_{ss'|\mathcal{S}} + \Phi^{\tau_s}_{s\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{SS}} \Phi^{\tau_{s'}}_{\mathcal{S}s'} & \textit{otherwise} \ , \end{cases} \tag{5.7}$$

and $\Phi^{\tau_{s'}}_{\mathcal{S}s'}$ is the transpose of $\Phi^{\tau_{s'}}_{s'\mathcal{S}}$(5.5).

Remark 1. In Definition 16, if $\tau_s = \tau_{s'} = k$, then mobile sensor k can compute $\widehat{\mu}_s^{\tau_s}$ (5.3) and $\widehat{\sigma}_{ss'}^{k}$ (5.4) locally and send them to the other mobile sensors that request them. Otherwise, $\tau_s \neq \tau_{s'}$ and mobile sensor k has to request $|\mathcal{S}|$-sized vectors $\Phi_{s\mathcal{S}}^{\tau_s}$ and $\Phi_{s'\mathcal{S}}^{\tau_{s'}}$ from the respective mobile sensors τ_s and $\tau_{s'}$ to compute $\widehat{\sigma}_{ss'}^{+}$ (5.7).

According Appendix B, the predicitve distribution $\mathcal{N}(\widehat{\mu}_{\mathcal{U}}^{+}, \widehat{\Sigma}_{\mathcal{U}\mathcal{U}}^{+})$ computed by GP-DDF$^+$ is proved to be equivalent to the predictive Gaussian distribution $\mathcal{N}(\mu_{\mathcal{U}|\mathcal{D}}^{\text{PIC}}, \Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}^{\text{PIC}})$ computed by the centralized partially independent conditional (PIC) approximation of GP model [Snelson, 2007] where $\mu_{\mathcal{U}|\mathcal{D}}^{\text{PIC}}$ and $\Sigma_{\mathcal{U}\mathcal{U}|\mathcal{D}}^{\text{PIC}}$ are defined in (4.13) and (4.14), respectively. The equivalence result bears two implications:

First, the equivalence result implies that the computational load of the centralized PIC approximation of GP can be distributed among K mobile sensors, hence improving the time efficiency of demand prediction. Supposing $|\mathcal{U}| \leq |\mathcal{S}|$ and $|\mathcal{U}| \leq |\mathcal{D}|/K$ for simplicity, the $\mathcal{O}\big(|\mathcal{D}|((|\mathcal{D}|/K)^2 + |\mathcal{S}|^2)\big)$ time incurred by PIC can be reduced to $\mathcal{O}\big((|\mathcal{D}|/K)^3 + |\mathcal{S}|^3 + |\mathcal{S}|^2 K\big)$ time of running GP-DDF$^+$ on each of the K mobile sensors. Hence, GP-DDF$^+$ scales better with increasing size $|\mathcal{D}|$ of data.

Second, the equivalence result also sheds some light on an important property of GP-DDF$^+$ based on the structure of PIC: It is assumed that $Y_{\mathcal{D}_1 \bigcup \mathcal{U}_1}, \ldots,$ $Y_{D_K \bigcup \mathcal{U}_K}$ are conditionally independent given the support set $\mathcal{S}$. As compared to GP-DDF that assumes conditional independence of $Y_{\mathcal{D}_1}, \ldots, Y_{\mathcal{D}_K}, Y_{\mathcal{U}_1}, \ldots, Y_{\mathcal{U}_K}$, GP-DDF$^+$ can predict $Y_{\mathcal{U}}$ better since it imposes a weaker conditional independence assumption. Experimental results on real-world mobility demand data (Section 7.4) also show that GP-DDF$^+$ achieves predictive accuracy comparable to FGP and significantly better than GP-DDF, thus justifying the practicality of such an assumption for predicting a mobility demand pattern.

5.2 Decentralized Active Sensing

This chapter aims to develop techniques for *decentralized active sensing* component of *Gaussian process-based decentralized data fusion and active sensing* (D^{2}FAS) framework. First, Section 5.2.1 formulate the active sensing problem with mobile sensors; It is showed that deriving the most informative (maximum posterior Gaussian entropy) joint walk is not scalable in the size of phenomenon data and in the number of mobile sensors. To address former scalability issue due to a big phenomenon data, Section 5.2.2 exploits the decentralized data fusion algorithms(Section 5.1) to provide efficient and scalable computation of a posterior Gaussian entropy / a posterior log-Gaussian entropy. To overcome the latter scalability issue due to a large number of mobile sensors, Section 5.2.3 presents a novel *partially decentralized active sensing* (PDAS) strategy [Chen *et al.*, 2012] whose performance can be theoretically guaranteed, and Section 5.2.4 presents a *fully decentralized active sensing* (FDAS) strategy [Chen *et al.*, 2013b] to alleviate situation in which PDAS strategy perform poorly when its partitioning heuristic tends to form large subsets of agents.

5.2.1 Problem Formulation

The problem of active sensing with K mobile sensors is formulated as follows: Given the set $\mathcal{D}_k \subset V$ of observed inputs (e.g., road segments/regions) and the currently observed inputs $s_k \in V$ of every mobile sensor $k = 1, \ldots, K$, the mobile sensors have to coordinate to select the most informative walks $w_1^*, \ldots, w_K^*$ of length H each and with respective origins $s_1, \ldots, s_K$ in the environmental field:

$$(w_1^*, \ldots, w_K^*) = \underset{(w_1,\ldots,w_K)}{\arg\max}\, \mathbb{H}\left[Y_{\bigcup_{k=1}^K \mathcal{U}_{w_k}} \middle| Y_{\bigcup_{k=1}^K \mathcal{D}_k}\right] \tag{5.8}$$

where $\mathcal{U}_{w_k}$ denotes the set of unobserved inputs induced by the walk w_k. To ease notations, let a joint walk be denoted by $w \triangleq (w_1, \ldots, w_K)$ (similarly, for w^*) and its induced set of unobserved inputs be $\mathcal{U}_w \triangleq \bigcup_{k=1}^K \mathcal{U}_{w_k}$ from now on. Interestingly, it can be shown using the chain rule for entropy that these maximum-entropy walks w^* minimize the posterior joint entropy (i.e., $\mathbb{H}[Y_{V\setminus(\mathcal{D}\bigcup\mathcal{U}_{w^*})}|Y_{\mathcal{D}\bigcup\mathcal{U}_{w^*}}]$) of the measurements at the remaining unobserved inputs (i.e., $V \setminus (\mathcal{D} \bigcup \mathcal{U}_{w^*})$) in the field. After executing the walk w_k^*, each mobile sensor k observes the set $\mathcal{U}_{w_k^*}$ of the field and updates its local information:

$$\mathcal{D}_k \leftarrow \mathcal{D}_k \bigcup \mathcal{U}_{w_k^*}\,,\, y_{\mathcal{D}_k} \leftarrow y_{\mathcal{D}_k \bigcup \mathcal{U}_{w_k^*}}\,,\, s_k \leftarrow \text{terminus of } w_k^*\,. \tag{5.9}$$

To derive the most informative joint walk w^*, the posterior entropy (5.8) of every possible joint walk w has to be evaluated[2]. Such a centralized strategy

[2] Solving (5.8) is an NP-hard problem.

cannot be performed in real time due to the following two issues: (a) It relies on all the phenomenon data that are gathered distributedly by the mobile sensors, thus incurring huge time and communication overheads with large data, and (b) it involves evaluating a prohibitively large number of joint walks (i.e., exponential in the number of mobile sensors).

5.2.2 Decentralized Posterior Gaussian Entropy Strategy

Evaluating the Gaussian posterior entropy term in (5.8) involves computing a posterior covariance matrix (3.3) using one of the data fusion methods described earlier: If (3.2) of full GP (FGP) model (Section 3.1) or (3.5) of SoD (Section 3.2) is to be used, then the observations that are gathered distributedly by the sensors have to be fully communicated to a central data fusion center. In contrast, GP-DDF algorithm (Section 5.1.1) only requires communicating local summaries (Definition 11) to compute (4.11) for solving the active sensing problem (5.8):

$$w^* = \arg\max_w \overline{\mathbb{H}}[Y_{\mathcal{U}_w}] \ , \tag{5.10}$$

$$\overline{\mathbb{H}}[Y_{\mathcal{U}_w}] \triangleq \frac{1}{2} \log (2\pi e)^{|\mathcal{U}_w|} \left| \widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w} \right| \ . \tag{5.11}$$

In (5.11), $\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ is defined by (5.2). To exploit GP-DDF$^+$ model (Section 5.1.2), $\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ can be replaced by $\widehat{\Sigma}^+_{\mathcal{U}_w\mathcal{U}_w}$ defined in Definition 16.

If a ℓGP model (Section 3.4) is exploited to model an environmental phenomeon, then the active sensing problem (5.8) can be approximated by

$$w^* = \arg\max_w \widetilde{\mathbb{H}}[Y_{\mathcal{U}_w}] \ , \tag{5.12}$$

$$\widetilde{\mathbb{H}}\left[Y_{\mathcal{U}_w}\right] \triangleq \frac{1}{2} \log (2\pi e)^{|\mathcal{U}_w|} \left| \widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w} \right| + \widehat{\mu}_{\mathcal{U}_w} \cdot \mathbf{1} \ . \tag{5.13}$$

To obtain $\widetilde{\mathbb{H}}\left[Y_{\mathcal{U}_w}\right]$ (5.13) using GP-DDF (Section 5.1.1), $\Sigma_{\mathcal{U}_w\mathcal{U}_w|\mathcal{D}}$ and $\mu_{\mathcal{U}_w|\mathcal{D}}$ in $\mathbb{H}[Y_{\mathcal{U}_w}|Y_{\mathcal{D}}]$ ((3.7) & (5.8)) can be replaced by $\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ (5.2) and $\widehat{\mu}_{\mathcal{U}_w}$ (5.1), respectively. To exploit GP-DDF$^+$ model (Section 5.1.2) instead, $\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ and $\widehat{\mu}_{\mathcal{U}_w}$ have to be replaced by $\widehat{\Sigma}^+_{\mathcal{U}_w\mathcal{U}_w}$ and $\widehat{\mu}^+_{\mathcal{U}_w}$ (Definition 16).

5.2.3 Partially Decentralized Active Sensing

Without imposing any structural assumption, solving the active sensing problem (5.10) will be prohibitively expensive due to the space of possible joint walks w that grows exponentially in the number K of mobile sensors. To overcome this scalability issue for decentralized active sensing with mobile sensors, our key idea is to construct a block-diagonal matrix whose log-determinant closely approximates that of $\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ (5.2) and exploit the property that the log-determinant of such a block-diagonal matrix can be decomposed into a sum of log-determinants of its diagonal blocks, each of which depends only on the walks of a disjoint subset of the K mobile sensors. Consequently, the active sensing problem can be partially decentralized leading to a reduced space of possible joint walks to be searched, as detailed in the rest of this section.

Firstly, we extend an earlier assumption in Section 5.1.1: $Y_{\mathcal{D}_1}, \ldots, Y_{\mathcal{D}_K}$, $Y_{\mathcal{U}_{w_1}}, \ldots, Y_{\mathcal{U}_{w_K}}$ are conditionally independent given the measurements at the support set $\mathcal{S}$. Then, it can be shown via the equivalence to PITC (Theorem 1) that $\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ (5.2) comprises diagonal blocks of the form $\widehat{\Sigma}_{\mathcal{U}_{w_k}\mathcal{U}_{w_k}}$ for $k = 1, \ldots, K$ and off-diagonal blocks of the form $\Sigma_{\mathcal{U}_{w_k}\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{U}_{w_{k'}}}$ for $k, k' = 1, \ldots, K$ and $k \neq k'$. In particular, each off-diagonal block of $\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ represents the correlation of measurements between the unobserved inputs $\mathcal{U}_{w_k}$ and $\mathcal{U}_{w_{k'}}$ along the respective walks w_k of sensor k and $w_{k'}$ of sensor k'. If the correlation between some pair of their possible walks is high enough, then their walks have to be coordinated. This is formally realized by the following coordination graph over the K sensors:

Definition 17 (Coordination Graph)**.** *Define the coordination graph to be an undirected graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ that comprises*

- *a set $\mathcal{V}$ of vertices denoting the K mobile sensors, and*
- *a set $\mathcal{E}$ of edges denoting coordination dependencies between sensors such that there exists an edge $\{k, k'\}$ incident with sensors $k \in \mathcal{V}$ and $k' \in \mathcal{V} \setminus \{k\}$ iff*

$$\max_{s\in\mathcal{U}_{W_k}, s'\in\mathcal{U}_{W_{k'}}} \left|\Sigma_{s\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}s'}\right| > \varepsilon \qquad (5.14)$$

for a predefined constant $\varepsilon > 0$ where W_k denotes the set of possible walks of length H of mobile sensor k from origin s_k in the environmental field and $\mathcal{U}_{W_k} \triangleq \bigcup_{w_k\in W_k} \mathcal{U}_{w_k}$.

Remark. The construction of $\mathcal{G}$ can be decentralized as follows: Since $\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}$ is symmetric and positive definite, it can be decomposed by Cholesky factorization into $\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} = \Psi\Psi^\top$ where Ψ is a lower triangular matrix and $\Psi^\top$ is the transpose of Ψ. Then, $\Sigma_{s\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}s'} = (\Psi\backslash\Sigma_{\mathcal{S}s})^\top\Psi\backslash\Sigma_{\mathcal{S}s'}$ where $\Psi\backslash B$ denotes the column vector ϕ solving $\Psi\phi = B$. That is, $\Sigma_{s\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}s'}$ (5.14) can be expressed as a dot product of two vectors $\Psi\backslash\Sigma_{\mathcal{S}s}$ and $\Psi\backslash\Sigma_{\mathcal{S}s'}$; this property is exploited to determine adjacency between sensors in a decentralized manner:

Definition 18 (Adjacency)**.** *Let*

$$\mathcal{J}_k \triangleq \{\Psi\backslash\Sigma_{\mathcal{S}s}\}_{s\in\mathcal{U}_{W_k}} \tag{5.15}$$

for $k = 1, \ldots, K$. A sensor $k \in \mathcal{V}$ is adjacent to sensor $k' \in \mathcal{V} \setminus \{k\}$ in coordination graph $\mathcal{G}$ iff

$$\max_{\phi\in\mathcal{J}_k, \phi'\in\mathcal{J}_{k'}} \left|\phi^\top\phi'\right| > \varepsilon\ . \tag{5.16}$$

It follows from the above definition that if each sensor k constructs $\mathcal{J}_k$ and exchanges it with every other sensor, then it can determine its adjacency to all the other sensors and store this information in a column vector a_k of length K with its k'-th component being defined as follows:

$$[a_k]_{k'} = \begin{cases} 1 & \text{if sensor } k \text{ is adjacent to sensor } k', \\ 0 & \text{otherwise.} \end{cases} \tag{5.17}$$

By exchanging its adjacency vector a_k with every other sensor, each sensor can construct a globally consistent adjacency matrix $A_{\mathcal{G}} \triangleq (a_1 \ldots a_K)$ to represent coordination graph $\mathcal{G}$.

Next, by computing the connected components (say, $\mathcal{K}$ of them) of coordination graph $\mathcal{G}$, their resulting vertex sets partition the set $\mathcal{V}$ of K sensors into $\mathcal{K}$ disjoint subsets $\mathcal{V}_1, \ldots, \mathcal{V}_{\mathcal{K}}$ such that the sensors within each subset have to coordinate their walks. Each sensor can determine its residing connected component in a decentralized way by performing a depth-first search in $\mathcal{G}$ starting from it as root.

Finally, construct a block-diagonal matrix $\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ to comprise diagonal blocks of the form $\widehat{\Sigma}_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{U}_{w_{\mathcal{V}_n}}}$ for $n = 1, \ldots, \mathcal{K}$ where $w_{\mathcal{V}_n} \triangleq (w_k)_{k\in\mathcal{V}_n}$ and $\mathcal{U}_{w_{\mathcal{V}_n}} \triangleq$

$\bigcup_{k\in\mathcal{V}_n}\mathcal{U}_{w_k}$. The active sensing problem (5.10) is then approximated by

$$\begin{array}{l}\displaystyle\max_{w}\ \frac{1}{2}\log(2\pi e)^{|\mathcal{U}_w|}\left|\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right|\\ \displaystyle\equiv\max_{(w_{\mathcal{V}_1},\ldots,w_{\mathcal{V}_\mathcal{K}})}\sum_{n=1}^{\mathcal{K}}\log(2\pi e)^{\left|\mathcal{U}_{w_{\mathcal{V}_n}}\right|}\left|\widehat{\Sigma}_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{U}_{w_{\mathcal{V}_n}}}\right|\\ \displaystyle=\sum_{n=1}^{\mathcal{K}}\max_{w_{\mathcal{V}_n}}\ \log(2\pi e)^{\left|\mathcal{U}_{w_{\mathcal{V}_n}}\right|}\left|\widehat{\Sigma}_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{U}_{w_{\mathcal{V}_n}}}\right|,\end{array}\tag{5.18}$$

which can be solved in a partially decentralized manner by each disjoint subset $\mathcal{V}_n$ of mobile sensors:

$$\widehat{w}_{\mathcal{V}_n}=\arg\max_{w_{\mathcal{V}_n}}\ \log(2\pi e)^{\left|\mathcal{U}_{w_{\mathcal{V}_n}}\right|}\left|\widehat{\Sigma}_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{U}_{w_{\mathcal{V}_n}}}\right|.\tag{5.19}$$

Our active sensing algorithm becomes fully decentralized if ε is set to be sufficiently large: more sensors become isolated in $\mathcal{G}$, consequently decreasing the size $\kappa\triangleq\max_n|\mathcal{V}_n|$ of its largest connected component to 1. As shown in Section 6.2.1, decreasing κ improves its time efficiency. On the other hand, it tends to a centralized behavior (5.10) by setting $\varepsilon\rightarrow 0^+$: $\mathcal{G}$ becomes near-complete, thus resulting in $\kappa\rightarrow K$.

5.2.3.1 Performance Guarantee

This section theoretically guarantees performance of the proposed PDAS algorithm. Let

$$\xi\triangleq\max_{n,w_{\mathcal{V}_n},i,i'}\left|\left[\left(\widehat{\Sigma}_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{U}_{w_{\mathcal{V}_n}}}\right)^{-1}\right]_{ii'}\right|\tag{5.20}$$

and $\epsilon\triangleq 0.5\log 1\Big/\left(1-\left(K^{1.5}H^{2.5}\kappa\xi\varepsilon\right)^2\right)$. In the result below, we prove that the joint walk $\widehat{w}\triangleq(\widehat{w}_{\mathcal{V}_1},\ldots,\widehat{w}_{\mathcal{V}_\mathcal{K}})$ partially decentralized active sensing problem (5.19) is guaranteed to achieve an entropy $\overline{\mathbb{H}}\left[Y_{\mathcal{U}_{\widehat{w}}}\right]$ (i.e., by plugging $\widehat{w}$ into (5.11)) that is not more than ϵ from the maximum entropy $\overline{\mathbb{H}}[Y_{\mathcal{U}_{w^*}}]$ achieved by joint walk w^* (5.10):

Theorem 4 (Performance Guarantee)**.**

If $K^{1.5}H^{2.5}\kappa\xi\varepsilon<1$, *then* $\overline{\mathbb{H}}[Y_{\mathcal{U}_{w^*}}]-\overline{\mathbb{H}}\left[Y_{\mathcal{U}_{\widehat{w}}}\right]\leq\epsilon\,.$

The proof of Theorem 4 is given in Appendix D. The implication of Theorem 4 is that our partially decentralized active sensing algorithm can perform comparatively well (i.e., small ϵ) under the following favorable environmental conditions: (a) the network of K sensors is not large, (b) length H of each sensor's walk to be optimized is not long, (c) the largest subset of κ sensors being formed to coordinate their walks (i.e., largest connected component in $\mathcal{G}$) is reasonably small, and (d) the minimum required correlation ε between walks of adjacent sensors is kept low.

5.2.4 Fully Decentralized Active Sensing

The PDAS algorithm proposed in Section 5.2.3 partitions the vehicles into several small groups such that each group of vehicles selects its joint walk independently. This partitioning heuristic performs poorly when the largest group formed still contains many vehicles. This is indeed the case if the posterior log-Gaussian entropy (3.7) or its approximation (5.13) is exploited as active sensing strategy, because many vehicles tend to cluster within hotspots due to the $\mu_{\mathcal{U}|\mathcal{D}} \cdot \mathbf{1}$ term. To scale well in the fleet size, we therefore adopt a fully decentralized active sensing (FDAS) strategy by assuming that the joint walk $w_1^* \ldots w_K^*$ is derived by selecting the locally optimal walk of each vehicle k:

$$w_k^* = \arg\max_{w_k} \widetilde{\mathbb{H}}\left[Y_{\mathcal{U}_{w_k}}\right] \tag{5.21}$$

where $\widetilde{\mathbb{H}}\left[Y_{\mathcal{U}_{w_k}}\right]$ is defined in the same way as (5.13). Then, each vehicle can select its locally optimal walk independently of the other vehicles, thus significantly reducing the search space of joint walks. A consequence of such an assumption is that, without coordinating their walks, the vehicles may select suboptimal joint walks (e.g., two vehicles' locally optimal walks are highly correlated). In practice, this assumption becomes less restrictive when the size $|\mathcal{D}|$ of data increases to potentially reduce the degree of violation of conditional independence of $Y_{\mathcal{U}_{w_1}}, \ldots, Y_{\mathcal{U}_{w_K}}$.

Chapter 6

Decentralized Solution to Traffic Condition Monitoring

Up to this point, it has been shown that the spatiotemporal urban traffic phenomena can be modeled based on Gaussian process-based models (Chapter 3); To work with active mobile sensors, a class of efficient and scalable techniques have been proposed for data fusion (Section 5.1) and active sensing (Section 5.2) in decentralized manner.

This chapter demonstrates that how such techniques can be exploited to address a real-world traffic condition monitoring problem [Chen *et al.*, 2012]. Section 6.1 discusses the practical importance of monitoring traffic conditions over road network . To address this problem, Section 6.2 presents a novel *Gaussian process-based decentralized data fusion and active sensing* (D^2FAS) algorithm which is a *Gaussian process-based decentralized data fusion* (GP-DDF) algorithm (Section 5.1.1) coupled with a *partially decentralized active sensing* (PDAS) algorithm (Section 5.2.3); Then, the time and communication complexity of this D^2FAS algorithm are analysed in Sections 6.2.1 & 6.2.2. Subsequently, The performance of our D^2FAS algorithm are empirically evaluated in a real-world traffic phenomenon over an urban road network (Section 6.3). The results in Section 6.4 show that our D^2FAS algorithm is significantly more time efficient and scalable than state-of-the-art centralized approaches and achieves comparable predictive performance.

6.1 Motivation

Knowing and understanding the traffic conditions and phenomena over road networks has become increasingly important to the goal of achieving smooth-flowing, congestion-free traffic, especially in densely-populated urban cities. To achieve this goal, this thesis exploits a fleet of mobile sensors operating over road network to actively collect and assimilate traffic phenomena data (e.g., traffic speeds). Briefly, each mobile sensor runs a *Gaussian process-based decentralized data fusion and active sensing* (D^2FAS) algorithm which is comprised of two components: decentralized data fusion (see Section 5.1) and decentralized active sensing (see Section 5.2). In the following, we present a novel D^2FAS algorithm to address the specific problem of monitoring traffic condition over road networks.

6.2 D^2FAS Algorithm

The key operations of our D^2FAS algorithm that is run on each mobile sensor k are presented in Algorithm 1. In this algorithm, the data fusion component implements the *Gaussian process-based decentralized data fusion* (GP-DDF) (Sections 5.1.1), and the active sensing component employs *partially decentralized actively sensing* (PDAS) (Section 5.2.3).

In the next, the time and communication complexity of our D^2FAS algorithm are analyzed and compared to that of centralized active sensing (5.10) coupled with the data fusion methods: Full GP (FGP) (Section 3.1) and SoD (Section 3.2).

6.2.1 Time Complexity

The GP-DDF (Section 5.1.1) involves computing the local and global summaries and the predictive Gaussian distribution.

To construct the local summary using (4.1) and (4.2), each sensor has to evaluate $\Sigma_{\mathcal{D}_k\mathcal{D}_k|\mathcal{S}}$ in $\mathcal{O}\big(|\mathcal{S}|^3 + |\mathcal{S}|(|\mathcal{D}|/K)^2\big)$ time and invert it in $\mathcal{O}((|\mathcal{D}|/K)^3)$ time, after which the local summary is obtained in $\mathcal{O}\big(|\mathcal{S}|^2|\mathcal{D}|/K + |\mathcal{S}|(|\mathcal{D}|/K)^2\big)$ time.

The global summary is computed in $\mathcal{O}\big(|\mathcal{S}|^2 K\big)$ by (4.3) and (4.4).

Algorithm 1: PDAS+GP-DDF($\mathcal{S}, K, H, k, \mathcal{D}_k, y_{\mathcal{D}_k}, s_k$)

```
/* $\mathcal{S}$: support set */
/* $K$: number of mobile sensors */
/* $H$: length of planned steps of a walk */
/* $k$: index of the mobile sensor */
/* ($\mathcal{D}_k$, $y_{\mathcal{D}_k}$): data gathered by sensor $k$ */
/* $s_k$: initial location of mobile sensor $k$ */
while true do
    /* Data fusion (Section 5.1.1) */
    Construct local summary by (4.1) & (4.2)
    Exchange local summary with every sensor $i \neq k$
    Construct global summary by (4.3) & (4.4)
    Predict measurements at unobserved road segments by (5.1) & (5.2)
    /* Active Sensing (Section 5.2.3) */
    Construct $\mathcal{J}_k$ by (5.15)
    Exchange $\mathcal{J}_k$ with every sensor $i \neq k$
    Compute adjacency vector $a_k$ by (5.16) & (5.17)
    Exchange adjacency vector with every sensor $i \neq k$
    Construct adjacency matrix of coordination graph
    Find vertex set $\mathcal{V}_n$ of its residing connected component
    Compute maximum-entropy joint walk $\widehat{w}_{\mathcal{V}_n}$ by (5.19)
    Execute walk $\widehat{w}_k$ and observe its road segments $\mathcal{U}_{\widehat{w}_k}$
    Update local information $\mathcal{D}_k$, $y_{\mathcal{D}_k}$, and $s_k$ by (5.9)
```

Finally, the predictive Gaussian distribution is derived in $\mathcal{O}\big(|\mathcal{S}|^3 + |\mathcal{S}||\mathcal{U}|^2\big)$ time using (5.1) and (5.2).

Supposing $|\mathcal{U}| \leq |\mathcal{S}|$ for simplicity, the time complexity of data fusion is then $\mathcal{O}\big((|\mathcal{D}|/K)^3 + |\mathcal{S}|^3 \ + |\mathcal{S}|^2 K\big)$.

Let the maximum out-degree of road network G (Definition 1) be denoted by δ. Then, each sensor has to consider $\Delta \triangleq \delta^H$ possible walks of length H. The PDAS algorithm involves computing $\mathcal{J}_k$ in $\mathcal{O}\big(\Delta H|\mathcal{S}|^2\big)$ time, a_k in $\mathcal{O}(\Delta^2 H^2 |\mathcal{S}| K)$ time, its residing connected component in $\mathcal{O}(\kappa^2)$ time, and the maximum-entropy joint walk by (5.2) and (5.19) with the following incurred time: The largest connected component of κ sensors in $\mathcal{G}$ has to consider Δ^κ possible joint walks.

Note that $\widehat{\Sigma}_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{U}_{w_{\mathcal{V}_n}}} = \text{diag}\big((\Sigma_{\mathcal{U}_{w_k}\mathcal{U}_{w_k}|\mathcal{S}})_{k\in\mathcal{V}_n}\big) + \Sigma_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{U}_{w_{\mathcal{V}_n}}}$ where $\text{diag}(B)$ constructs a diagonal matrix by placing vector B on its diagonal.

By exploiting $\mathcal{J}_k$, the diagonal and latter matrix terms for all possible joint walks can be computed in $\mathcal{O}\big(\kappa\Delta(H|\mathcal{S}|^2 + H^2|\mathcal{S}|)\big)$ and $\mathcal{O}(\kappa^2\Delta^2 H^2|\mathcal{S}|)$ time, respectively.

For each joint walk $w_{\mathcal{V}_n}$, evaluating the determinant of $\widehat{\Sigma}_{\mathcal{U}_{w_{\mathcal{V}_n}}\mathcal{U}_{w_{\mathcal{V}_n}}}$ incurs $\mathcal{O}((\kappa H)^3)$ time.

Therefore, the time complexity of active sensing is $\mathcal{O}\big(\kappa\Delta H|\mathcal{S}|^2 +$
$\Delta^2 H^2 |\mathcal{S}|(K + \kappa^2) + \Delta^\kappa(\kappa H)^3)$.

Hence, the time complexity of our D^{2}FAS algorithm is $\mathcal{O}((|\mathcal{D}|/K)^3 + |\mathcal{S}|^2(|\mathcal{S}|$ $+K + \kappa\Delta H) + \Delta^2 H^2|\mathcal{S}|(K + \kappa^2) + \Delta^\kappa(\kappa H)^3)$. In contrast, the time incurred by centralized active sensing coupled with FGP and SoD are, respectively, $\mathcal{O}\big(|\mathcal{D}|^3 + \Delta^K KH(|\mathcal{D}|^2 + (KH)^2)\big)$ and $\mathcal{O}\big(|\mathcal{S}|^3|\mathcal{D}| + \Delta^K KH(|\mathcal{S}|^2 + (KH)^2)\big)$. It can be observed that D^{2}FAS can scale better with large $|\mathcal{D}|$ (i.e., number of observations) and K (i.e., number of sensors). The scalability of D^{2}FAS vs. FGP and SoD will be further evaluated empirically in Section 6.3.

6.2.2 Communication Complexity

Let the communication overhead be defined as the size of each broadcast message. Recall the GP-DDF in Algorithm 1 that, in each iteration, each sensor broadcasts a $\mathcal{O}\big(|\mathcal{S}|^2\big)$-sized summary encapsulating its local observations, which is robust against communication failure. In contrast, FGP and SoD require each sensor to broadcast, in each iteration, a $\mathcal{O}(|\mathcal{D}|/K)$-sized message comprising exactly its local observations to handle communication failure. If the number of local observations grows to be larger in size than a local summary of predefined size, then the GP-DDF of D^{2}FAS is more scalable than FGP and SoD in terms of communication overhead. For the PDAS of D^{2}FAS, each sensor broadcasts $\mathcal{O}(\Delta H|\mathcal{S}|)$-sized $\mathcal{J}_k$ and $\mathcal{O}(K)$-sized a_k messages.

6.2.3 Summary of Theoretical Results

The time overheads show that both data fusion component (GP-DDF) and active sensing component (PDAS) of the proposed D^{2}FAS algorithm scale better than that of the centralized GP models (i.e., FGP and SoD) in size of data when number of agents is large. The communication overheads indicate that the proposed D^{2}FAS algorithm is more scalable than that of the centralized algorithms, because the broadcast messages of D^{2}FAS are independent of the size of data while the centralized models have to broadcast all the data.

6.3 Experimental Setup

In this section, we evaluate the predictive performance, time efficiency, and scalability of our D^{2}FAS algorithm.

6.3.1 Settings

We introduce a real-world traffic phenomenon[1] (i.e., speeds (km/h) of road segments) over an urban road network (Figure 6.1) in Tampines area, Singapore during evening peak hours on April 20, 2011. It comprises 775 road segments including highways, arterials, slip roads, etc. The mean speed is 48.8 km/h and the standard deviation is 20.5 km/h.

The performance of D^{2}FAS is compared to that of centralized active sensing (5.10) coupled with the state-of-the-art data fusion methods: full GP (FGP)

[1] The traffic flow dataset over Singapore road network is provided by Land Transport Authority (LTA) of Singapore and future urban mobility (FM) research group of Singapore-MIT Alliance for Research and Technology (SMART).

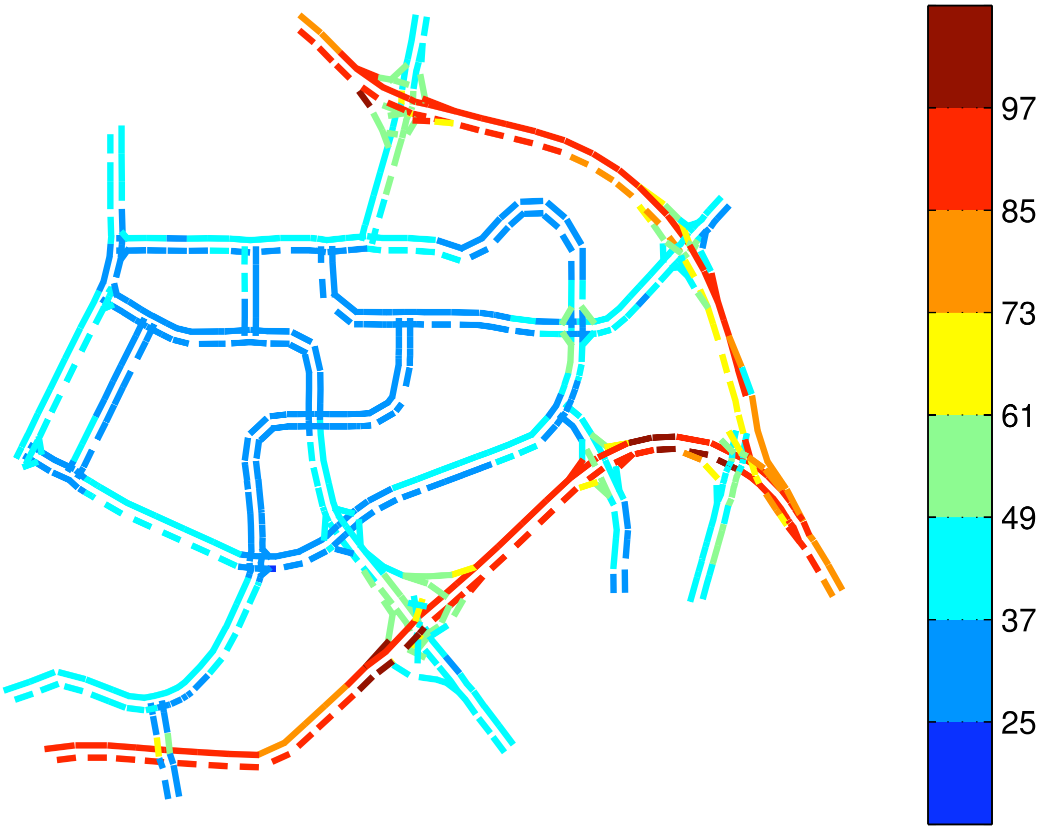


Figure 6.1: A real-world traffic phenomenon (speeds) over an urban road network

(Section 3.1) and SoD (Section 3.2). A network of K mobile sensors is deployed with the initial location of each mobile sensor is randomly distributed; then, it is tasked to explore the road network to gather a total of up to 960 observations. To reduce computational time, each sensor repeatedly computes and executes maximum-entropy walks of length $H = 2$ (instead of computing a very long walk), unless otherwise stated. For D^{2}FAS and SoD, $\mathcal{S}$ is set to 64 . For the active sensing component of D^{2}FAS, ε is set to 0.1, unless otherwise stated. The experiments are run on a Linux PC with Intel® Core™2 Quad CPU Q9550 at 2.83 GHz.

6.3.2 Performance Metrics

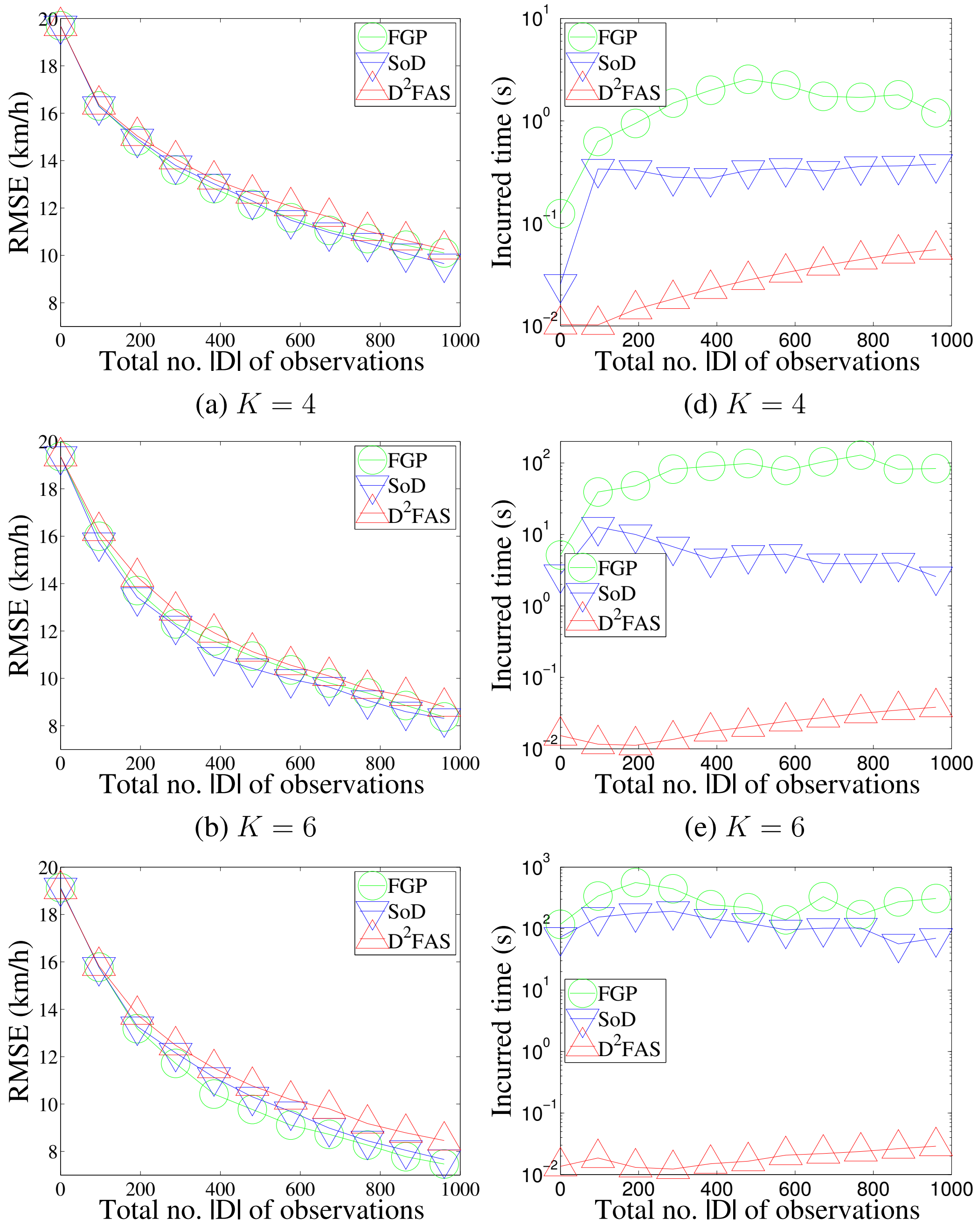
The first metric evaluates the predictive performance of a tested algorithm: It measures the *root mean squared error* (RMSE) $\sqrt{|V|^{-1}\sum_{s\in V}\left(y_s - \mu_{s|\mathcal{D}}\right)^2}$ over the entire domain V of the road network that is incurred by the predictive mean $\mu_{s|\mathcal{D}}$ of the tested algorithm, specifically, plugging in (3.1) of FGP, (3.4) of SoD, or (5.1) of D^{2}FAS. The second metric evaluates the time efficiency and scalability of a tested algorithm by measuring its incurred time; for D^{2}FAS, the maximum of the time incurred by all subsets $\mathcal{V}_1, \ldots, \mathcal{V}_K$ of sensors is recorded.

6.4 Results and Analysis

This section demonstrates and analyzes the results which are averaged over 40 randomly generated starting sensor locations.

6.4.1 Predictive Performance & Time Efficiency

Figure 6.2 shows results of the performance of the tested algorithms with varying number $K = 4, 6, 8$ of sensors. It can be observed that D^{2}FAS is significantly more time-efficient and scales better with increasing number $|\mathcal{D}|$ of observations (Figures 6.2d to 6.2f) while achieving predictive performance close to that of centralized active sensing coupled with FGP and SoD (Figures 6.2a to 6.2c). Specifically, D^{2}FAS is about $1, 2, 4$ orders of magnitude faster than centralized active sensing coupled with FGP and SoD for $K = 4, 6, 8$ sensors, respectively.



(a) $K = 4$

(d) $K = 4$

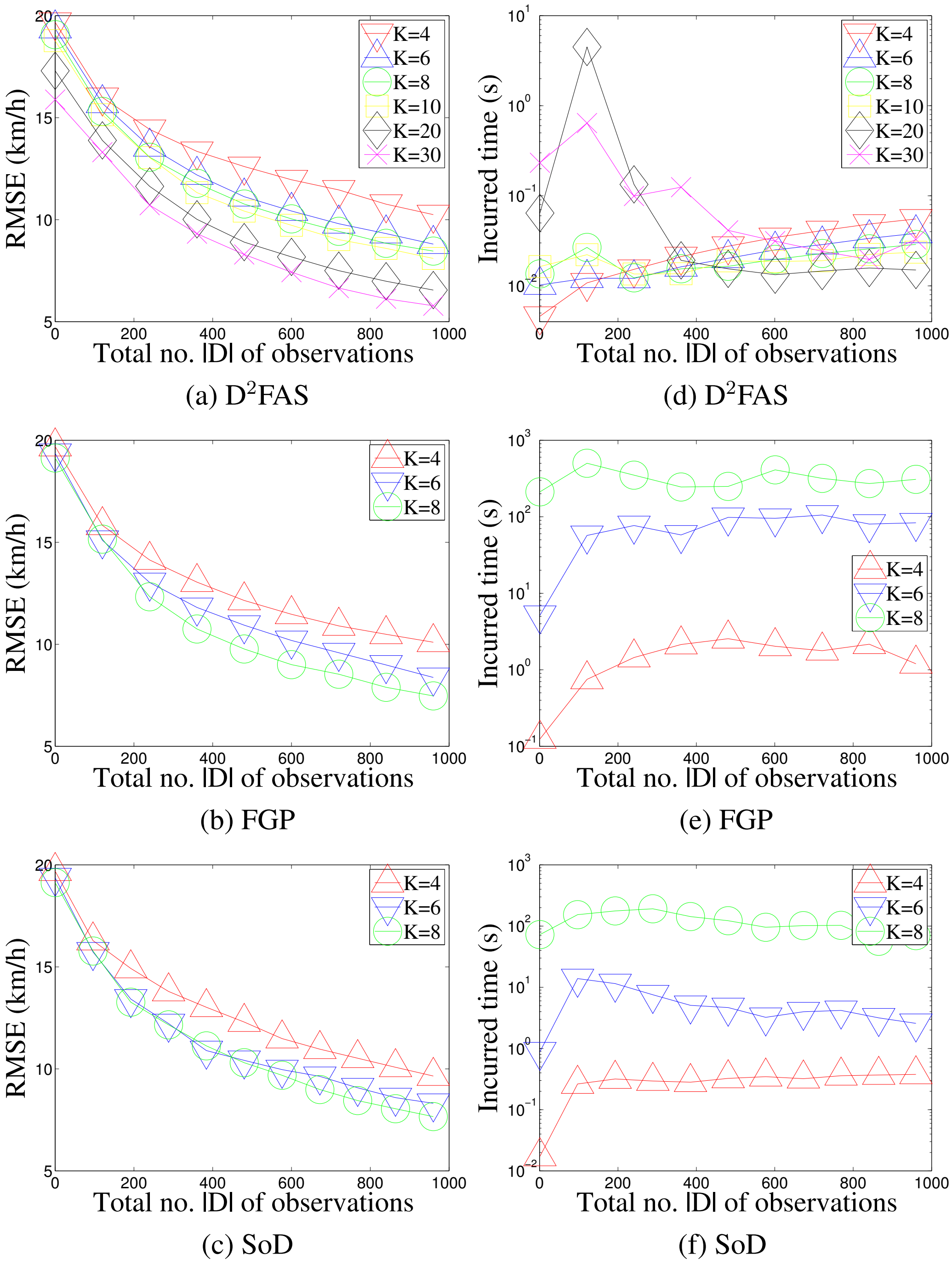
(b) $K = 6$

(e) $K = 6$

(c) $K = 8$

(f) $K = 8$

Figure 6.2: Predictive performance (a-c) & time efficiency (d-f) vs. total no. $|D|$ of observations gathered by varying number K of mobile sensors.



(a) D^{2}FAS

(d) D^{2}FAS

(b) FGP

(e) FGP

(c) SoD

(f) SoD

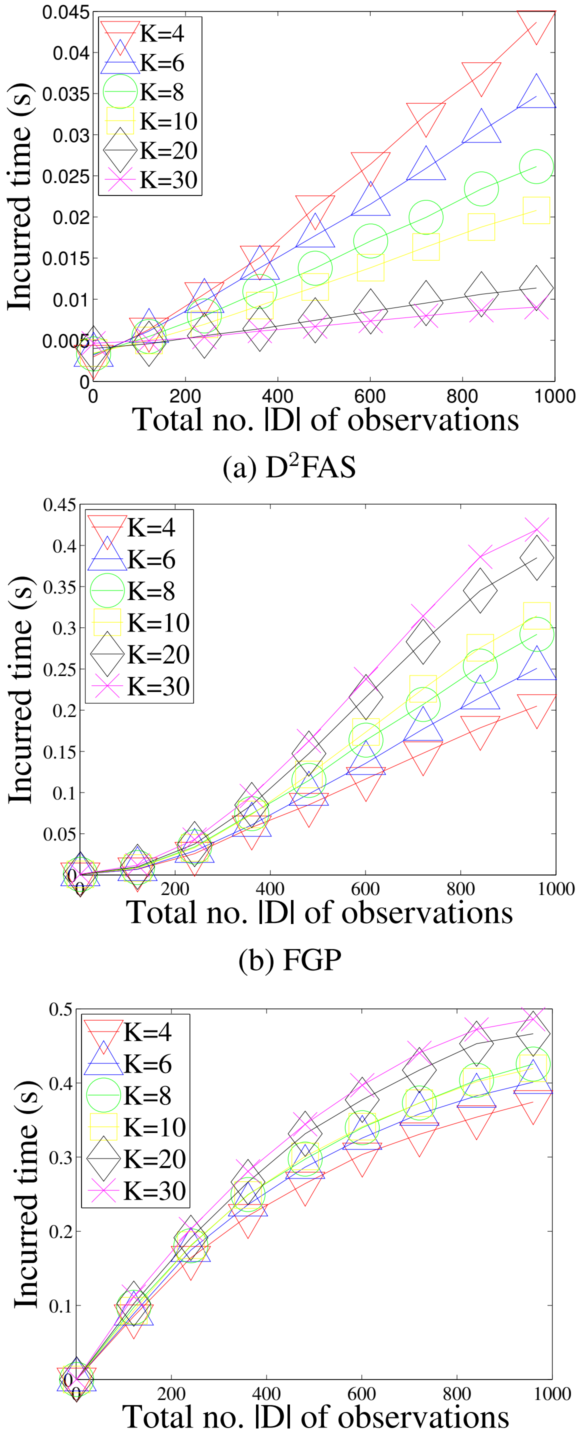
Figure 6.3: Predictive performance (a-c) & time efficiency (d-f) vs. total no. $|D|$ of observations gathered by varying number K of mobile sensors.

6.4.2 Scalability

Scalability of D^{2}FAS algorithm: Using the same results as that in Figure 6.2, Figure 6.3 plots them differently to reveal the scalability of the tested algorithms with increasing number K of sensors. Additionally, we provide results of the performance of D^{2}FAS for $K = 10, 20, 30$ sensors; such results are not available for centralized active sensing coupled with FGP and SoD due to extremely long incurred time. It can be observed from Figures 6.3a to 6.3c that the predictive performance of all tested algorithms improve with a larger number of sensors because each sensor needs to execute fewer number of walks and its performance is therefore less adversely affected by its myopic selection (i.e., $H = 2$) of maximum-entropy walks. As a result, more informative unobserved road segments are explored.

As shown in Figure 6.3d, when the randomly placed sensors gather their initial observations (i.e., $|\mathcal{D}| < 400$), the time incurred by D^{2}FAS is higher for greater K due to larger subsets of sensors being formed to coordinate their walks (i.e., larger κ). As more observations are gathered (i.e., $|\mathcal{D}| \geq 400$), the PDAS algorithm directs the sensors to explore further apart from each other in order to maximize the entropy of their walks. This consequently decreases κ leading to a reduction in incurred time. Furthermore, as K increases from 4 to 20, the incurred time decreases due to its decentralized data fusion component that can distribute the computational load among a greater number of sensors. When the road network becomes more crowded from $K = 20$ to $K = 30$ sensors, the incurred time increases slightly due to slightly larger κ. In contrast, Figures 6.3e and 6.3f show that the time taken by FGP and SoD increase significantly primarily due to their centralized active sensing incurring exponential time in K. Hence, the scalability of our D^{2}FAS algorithm in the number of sensors allows the deployment of a larger-scale mobile sensor network (i.e., $K \geq 10$) to achieve more accurate traffic modeling and prediction (Figures 6.3a to 6.3c).

Scalability of DDF Component: Figure 6.4 shows results of the scalability of the tested data fusion methods with increasing number K of sensors. In order to produce meaningful results for fair comparison, the same active sensing component has to be coupled with the data fusion methods and its incurred time kept to a minimum. As such, we impose the use of fully decentralized active sensing to be performed by each mobile sensor k: $w_k^* = \arg\max_{w_k} \mathbb{H}[Y_{\mathcal{U}_{w_k}}|Y_{\mathcal{D}}]$. For



(a) D^{2}FAS

(b) FGP

(c) SoD

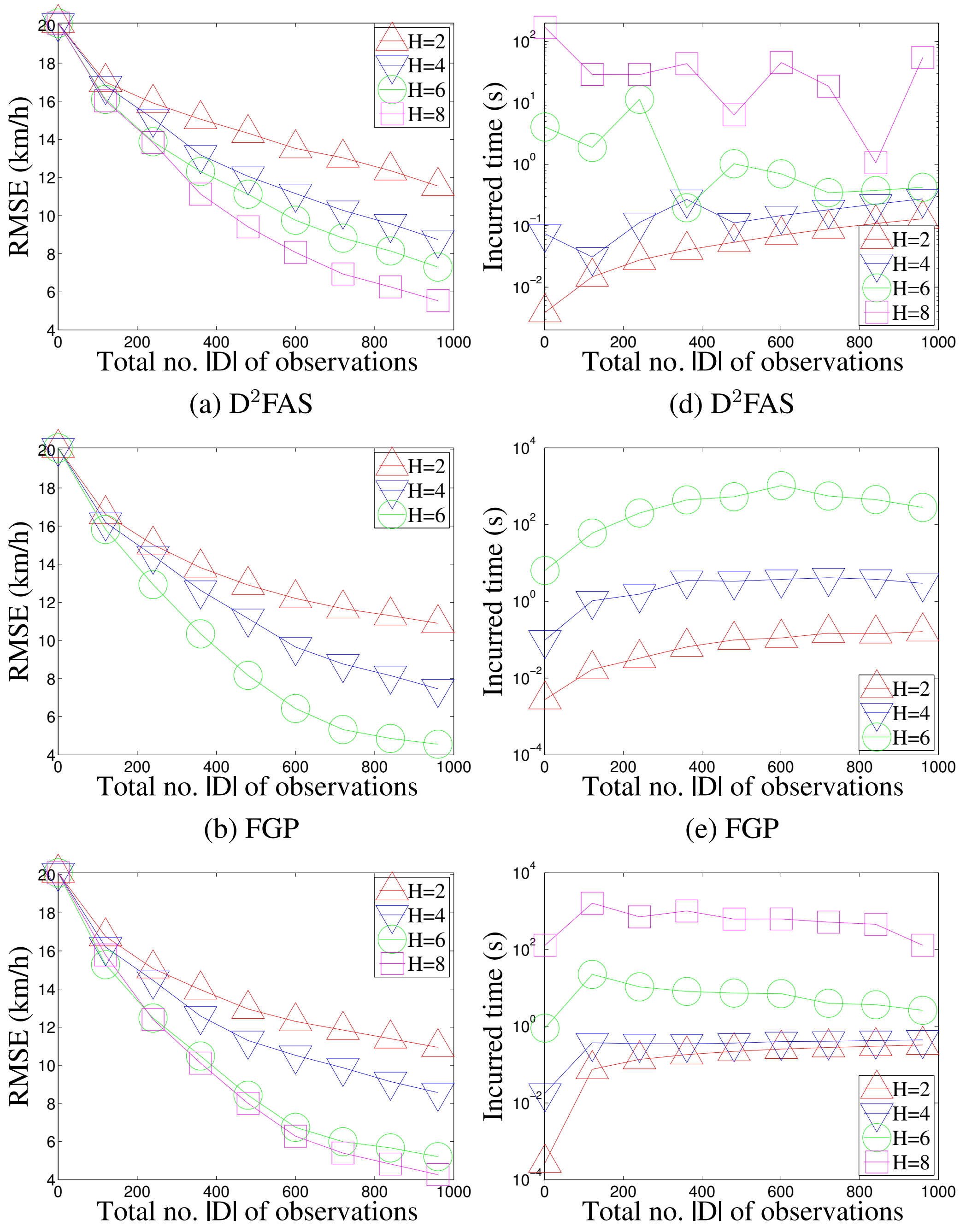
Figure 6.4: Time efficiency vs. total no. $|D|$ of observations gathered by varying number K of sensors.

D^{2}FAS, this corresponds exactly to (5.19) by setting a large enough ε (in our experiments, $\varepsilon = 2$) to yield $\kappa = 1$; consequently, computational and communicational operations pertaining to the coordination graph can be omitted.

It can be seen from Figure 6.4a that the time incurred by the GP-DDF of D^{2}FAS decreases with increasing K, as explained previously. In contrast, the time incurred by FGP and SoD increase (Figure 6.4b and 6.4c): As discussed above, a larger number of sensors result in a greater quantity of more informative unique observations to be gathered (i.e., fewer repeated observations), which increase the time needed for data fusion. When $K \geq 10$, D^{2}FAS is at least 1 order of magnitude faster than FGP and SoD. It can also be observed that D^{2}FAS scales better with increasing number of observations. So, the real-time performance and scalability of D^{2}FAS's decentralized data fusion enable it to be used for persistent large-scale traffic modeling and prediction where a large number of observations and sensors (including static and passive ones) are expected to be available.

6.4.3 Varying length of walk

Figure 6.5 shows results of the performance of the tested algorithms with varying length $H = 2, 4, 6, 8$ of maximum-entropy joint walks; we choose to experiment with just 2 sensors since Figures 6.3 and 6.4 reveal that a smaller number of sensors produce poorer predictive performance and higher incurred time with large number of observations for D^{2}FAS. It can be observed that the predictive performance of all tested algorithms improve with increasing walk length H because the selection of maximum-entropy joint walks is less myopic. The time incurred by D^{2}FAS increases due to larger κ but grows more slowly and is lower than that incurred by centralized active sensing coupled with FGP and SoD. Specifically, when $H = 8$, D^{2}FAS is at least 1 order of magnitude faster (i.e., average of 60 s) than centralized active sensing coupled with SoD (i.e., average of > 732 s) and FGP (i.e., not available due to excessive incurred time). Also, notice from Figures 6.3a and 6.3d that if a large number of sensors (i.e., $K = 30$) is available, D^{2}FAS can select shorter walks of $H = 2$ to be significantly more time-efficient (i.e., average of > 3 orders of magnitude faster) while achieving predictive performance comparable to that of SoD with $H = 8$ and FGP with $H = 6$.



(a) D^{2}FAS

(d) D^{2}FAS

(b) FGP

(e) FGP

(c) SoD

(f) SoD

Figure 6.5: Predictive performance (a-c) & time efficiency (d-f) vs. total no. $|D|$ of observations gathered by 2 mobile sensors with varying length H of maximum-entropy joint walks.

6.4.4 Summary of Empirical Result

With a larger size of data, the proposed D^{2}FAS algorithm is significantly more time efficient (i.e., 1-4 orders of magnitude faster) and scales better than the centralized FGP and SoD models, while its predictive performance is close to that of the centralized models. Hence, a larger number of mobile sensors can be deployed to achieve more accurate traffic modeling and prediction. With an increasing number of agents, the incurred time of GP-DDF decreases while that of FGP and SoD increase, since GP-DDF can distribute the computational load among a larger number of agents. With a longer walk length, the time incurred by D^{2}FAS grows more slowly and is lower than that incurred by centralized models. This indicates that D^{2}FAS can exploit a longer walking length than the centralized models in terms of reducing the effect caused by myopic selection.

Chapter 7

Decentralized Solution to Mobility-on-Demand Systems

In previous chapter, we demonstrate that a network of mobile sensors can employ our D^{2}FAS framework to model and predict traffic condition that is constrained by road network. This chapter [Chen *et al.*, 2013b] will further show that the D^{2}FAS framework can be applied to a fleet of autonomous vehicles to accurately model and predict spatiotemporally varying mobility demand patterns in mobility-on-demand (MoD) systems (Section 7.1), meanwhile, redistributing the fleet to achieve balance between mobility demand and supply. In particular, the proposed D^{2}FAS algorithm (Section 7.2) is designed as a *Gaussian process-based decentralized data fusion with local augmentation* (GP-DDF$^+$) algorithm (Section 5.1.2) coupled with a *fully decentralized active sensing* (FDAS) algorithm (Section 5.2.4). With theoretical evaluation of time & communication complexity (Sections 7.2.1 & 7.2.2) and empirical justification in a real world dataset (Section 7.3 & Section 7.4), the results show that our proposed D^{2}FAS algorithm (1) can achieve a better balance between predictive accuracy and time efficiency in sensing and predicting mobility demand patterns; (2) can achieve a better performance in servicing the mobility demands than the D^{2}FAS algorithm based on GP-DDF (Section 5.1.1).

7.1 Motivation

Private automobiles are becoming unsustainable personal mobility solutions in densely populated urban cities because the addition of parking and road spaces cannot keep pace with their escalating numbers due to limited urban land. For example, Hong Kong and Singapore have, respectively, experienced 27.6% and 37% increase in private vehicles from 2003 to 2011 [RPT, 2012]. However, their road networks have only expanded less than 10% in size. Without implementing sustainable measures, traffic congestions and delays will grow more severe and frequent, especially during peak hours.

Mobility-on-demand (MoD) systems: [Mitchell *et al.*, 2010] (e.g., Vélib system of over 20000 shared bicycles in Paris, experimental car-sharing systems described in [Pavone *et al.*, 2012]) have recently emerged as a promising paradigm of one-way vehicle sharing for sustainable personal urban mobility, specifically, to tackle the problems of low vehicle utilization rate and parking space caused by private automobiles. Conventionally, a MoD system provides stacks and racks of light electric vehicles distributed throughout a city: When a user wants to go somewhere, he simply walks to the nearest rack, swipes a card to pick up a vehicle, drives it to the rack nearest to his destination, and drops it off. In this thesis, we enhance the capability of a MoD system by deploying robotic shared vehicles (e.g., General Motors Chevrolet EN-V 2.0 prototype [GM, 2012]) that can autonomously drive and cruise the streets of a densely populated urban city to be hailed by users (like taxis) instead of just waiting at the racks to be picked up. Compared to the conventional MoD system, the fleet of autonomous robotic vehicles provides greater accessibility to users who can be picked up and dropped off at any location in the road network. As a result, it can service regions of high mobility demand but with poor coverage of stacks and racks due to limited space for their installation.

The key factors in the success of a MoD system are the costs to the users and system latencies, which can be minimized by managing the MoD system effectively. To achieve this, two main technical challenges need to be addressed [Mitchell, 2008]: (a) Real-time, fine-grained mobility demand sensing and prediction, and (b) real-time active fleet management to balance vehicle supply and demand and satisfy latency requirements at sustainable operating costs. Existing works on load balancing in MoD systems [Pavone *et al.*, 2012], dynamic traffic assignment problems [Peeta and Ziliaskopoulos, 2001], dynamic one-to-one pickup and delivery problems [Berbeglia *et al.*, 2010], and location recommendation and dispatch for cruising taxis [Agussurja and Lau, 2012; Chang *et al.*, 2010; Ge *et al.*, 2010; Li *et al.*, 2012; Yuan *et al.*, 2012] have tackled variants of the second challenge by assuming the necessary input of mobility demand information to be perfectly or accurately known using prior knowledge or offline processing of historic data. Such an assumption does not hold for densely populated urban cities because their mobility demand patterns are often subject to short-term random fluctuations and perturbations, in particular, due to frequent special events (e.g., storewide sales, exhibitions), unpredictable weather conditions, or emergencies (e.g., breakdowns in public transport services). So, in order for the active fleet management strategies to perform well, they require accurate, fine-grained information of the spatiotemporally varying mobility demand patterns in real time, which is the desired outcome of addressing the first challenge. To the best of our knowledge, there is little progress in the algorithmic development of the first challenge, which will be the focus of our work in this thesis.

7.2 D^{2}FAS Algorithm

To address the above challenges, we propose a novel D^{2}FAS algorithm (Algorithm 2) that is run by each MoD vehicles k. In this algorithm, the data fusion component employs the *Gaussian process-based decentralized data fusion with local augmentation* (GP-DDF$^+$) presented in Section 5.1.2, and the active sensing component uses the *fully decentralized active sensing* (FDAS) algorithm in Section 5.2.4. As we can observe from (5.13) and (5.21), FDAS exhibits a cruising behavior trades off between exploring sparsely sampled regions with high predictive uncertainty (i.e., by maximizing the log-determinant of Gaussian posterior covariance matrix $\Sigma_{\mathcal{U}_{w_k}\mathcal{U}_{w_k}}$ term) and hotspots (i.e., by maximizing the Gaussian posterior mean vector $\mu_{\mathcal{U}_{w_k}}$ term). As a result, it redistributes vacant MoD vehicles to regions with high likelihood of picking up users. Hence, besides gathering the most informative data for predicting the mobility demand pattern, FDAS is able to achieve a dual effect of fleet rebalancing to service mobility demands.

In the subsequent sections, the time and communication overheads of the

Algorithm 2: FDAS+GP-DDF$^+$($\mathcal{S}, K, H, k, \mathcal{D}_k, y_{\mathcal{D}_k}, s_k$)

```
/* $\mathcal{S}$: support set */
/* $K$: number of vehicles */
/* $H$: length of planned steps of a walk */
/* $k$: index of the vehicle */
/* ($\mathcal{D}_k$, $y_{\mathcal{D}_k}$): data gathered by vehicle $k$ */
/* $s_k$: initial location of vehicle $k$ */
while true do
    /* Data fusion (Section 5.1.2) */
    Construct local summary by (4.1) & (4.2)
    Exchange local summary with every vehicle $i \neq k$
    Construct global summary by (4.3) & (4.4)
    Construct assignment function by (5.6)
    Predict demand measurements of unobserved regions by (4.7) & (5.7)
    /* Active Sensing (Section 5.2.4) */
    Compute local maximum-entropy walk $w_k^*$ by (5.21) & (5.13)
    Execute walk $w_k^*$ and observe its demand measurements $\mathcal{U}_{w_k^*}$
    Update local information $\mathcal{D}_k$, $y_{\mathcal{D}_k}$ and $s_k$
```

propose D^{2}FAS algorithm are analyzed, and in comparison with that of both *full Gaussian process* (FGP) algorithm (Section 3.4) and *Gaussian process-based decentralized data fusion* (GP-DDF) algorithm (Section 5.1.1) coupled with a FDAS algorithm.

7.2.1 Time Complexity

Firstly, each vehicle k has to evaluate $\Sigma_{\mathcal{D}_k\mathcal{D}_k|\mathcal{S}}$ in $\mathcal{O}\big(|\mathcal{S}|^3 + |\mathcal{S}|(|\mathcal{D}|/K)^2\big)$ time and invert it in $\mathcal{O}((|\mathcal{D}|/K)^3)$ time.

After that, the GP-DDF$^+$ constructs the local summary in $\mathcal{O}\big(|\mathcal{S}|^2|\mathcal{D}|/K + |\mathcal{S}|(|\mathcal{D}|/K)^2\big)$ time by (4.1) and (4.2), and subsequently the global summary in $\mathcal{O}\big(|\mathcal{S}|^2 K\big)$ time by (4.3) and (4.4).

To construct the assignment function for any unobserved set $S \subset V$, vehicle k first computes $|\mathcal{U}|$ number of $\Phi^k_{s\mathcal{S}}$ for all unobserved regions $s \in \mathcal{U}$ in $\mathcal{O}\big(|\mathcal{U}||\mathcal{S}|^2 + |\mathcal{U}|(|\mathcal{D}|/K)^2\big)$ time by (4.9).

Then, after inverting $\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}$ in $\mathcal{O}(|\mathcal{S}|^3)$, the predictive means and variances for all $s \in S$ are computed in $\mathcal{O}\big(|\mathcal{U}||\mathcal{S}|^2 + |\mathcal{U}|(|\mathcal{D}|/K)^2\big)$ time by (4.7) and (5.7), respectively.

Let $\Delta \triangleq \delta^H$ denote the number of possible walks of length H where δ is the maximum out-degree of graph G. In the active sensing component, to obtain the locally optimal walk, the log-Gaussian posterior entropies (5.21) of all possible walks are derived from (4.7) and (5.7), respectively, in $\mathcal{O}\big(\Delta H|\mathcal{S}|^2\big)$ and $\mathcal{O}(\Delta(H|\mathcal{S}|)^2)$ time.

In FDAS algorithm, we assume $|\mathcal{U}| < \delta\Delta$ where S denotes $\bigcup_{w_k} \mathcal{U}_{w_k}$ the set of regions covered by any vehicle k's all possible walks of length H. Then, the time complexity for our GP-DDF$^+$ coupled with FDAS algorithm is $\mathcal{O}((|\mathcal{D}|/K)^3$ $+|\mathcal{S}|^3 + |\mathcal{S}|^2 K + \Delta(H^3 + (H|\mathcal{S}|)^2 + (|\mathcal{D}|/K)^2)\big)$.

In contrast, the time incurred by FGP and GP-DDF coupled with FDAS algorithms are, respectively, $\mathcal{O}\big(|\mathcal{D}|^3 + \Delta(H^3 + (H|\mathcal{D}|)^2)\big)$ and $\mathcal{O}((|\mathcal{D}|/K)^3$ $+|\mathcal{S}|^3 + |\mathcal{S}|^2 K + \Delta(H^3 + (H|\mathcal{S}|)^2)\big)$. It can be observed that our GP-DDF$^+$ coupled with FDAS algorithm can scale better with large size $|\mathcal{D}|$ of data and fleet size K than FGP coupled with FDAS algorithm, and its increased computational load, as compared to GP-DDF coupled with FDAS algorithm, is well distributed among K vehicles.

7.2.2 Communication complexity

In each iteration, each vehicle of the system running our GP-DDF$^+$ coupled with FDAS algorithm has to broadcast a $\mathcal{O}(|\mathcal{S}|^2)$-sized local summary for constructing the global summary, exchange $\mathcal{O}(\Delta)$ scalar values for constructing the assignment function, and request $\mathcal{O}(\Delta)$ number of $\mathcal{O}(|\mathcal{S}|)$-sized $\Phi^k_{s\mathcal{S}}$ components for evaluating the entropies of all possible local walks. In contrast, FGP coupled with FDAS algorithm needs to broadcast $\mathcal{O}(|\mathcal{D}|/K)$-sized message comprising all its local data to handle communication failure, and GP-DDF coupled with FDAS algorithm only needs to broadcast a $\mathcal{O}(|\mathcal{S}|^2)$-sized local summary.

7.2.3 Summary of Theoretical Result

The time complexity of GP-DDF$^+$ is the same as that of GP-DDF and the communication overhead of GP-DDF$^+$ is also independent of the size of data. Hence, GP-DDF$^+$ coupled with FDAS algorithm can scale better with large size of data and fleet size than FGP coupled with FDAS algorithm.

7.3 Experimental Setup

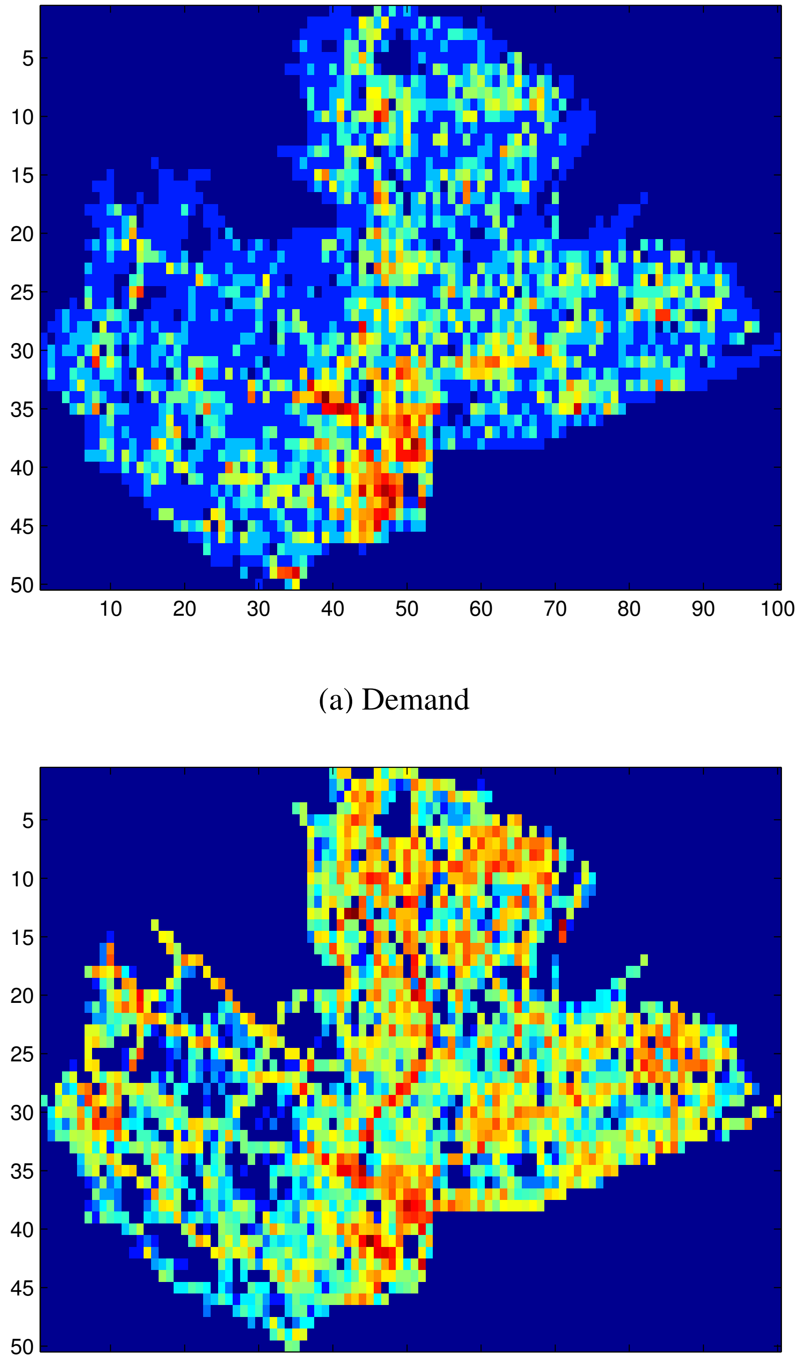
This section evaluates the performance of our proposed algorithm in terms of sensing the mobility demand pattern, servicing the real world mobility demands, time efficiency and scalability.

7.3.1 Settings

We use a real world taxi trajectory dataset[1] taken from the central business district of Singapore between $9{:}30$ p.m. and 10 p.m. on August 2, 2010. The service area is gridded into 50×100 regions such that 2506 regions are included into the dataset as the remaining regions contain no road segment for cruising vehicles to access. The maximum out-degree of graph imposed on these regions is 8. In our experimental setting, the input feature of each region is specified by its corresponding location. In any region, the demand (supply) measurement is obtained by counting the number of pickups (taxis cruising by) from all historic taxi trajectories generated by a major taxi company in a 30-minute time slot. After processing the taxi trajectories, the historic demand and supply distributions are obtained, as shown in Figure 7.1. Then, a number C of users are randomly distributed over the service area with their locations drawn from the demand distribution (Figure 7.1a). Similarly, a fleet of K vacant MoD vehicles are initialized at locations drawn from the supply distribution (Figure 7.1b).

In our simulation, when a vehicle enters a region with users, it picks up one of them randomly. Then, the MoD system removes this vehicle from the fleet of vacant cruising vehicles and introduce a new vacant vehicle drawn from the supply distribution. Similarly, a new customer appears at a random location drawn from the demand distribution. The MoD system operates for L time steps and each vehicle plans a walk of length 4 at each time step, with all vehicles running a data fusion algorithm coupled with our FDAS strategy. We will compare the performance of our GP-DDF$^+$ algorithm with that of FGP and GP-DDF algorithms when coupled with our FDAS strategy. The experiments are conducted on a Linux system with Intel® Xeon® CPU E5520 at 2.27 GHz.

[1] The taxi trajectory dataset in Singapore is provided by Comfort Transportation Pte Ltd (CTPL) and future urban mobility (FM) research group of Singapore-MIT Alliance for Research and Technology (SMART).



(a) Demand

(b) Supply

Figure 7.1: Historic demand and supply distributions obtained from a real world taxi trajectory dataset in central business district of Singapore.

7.3.2 Performance Metrics

The tested algorithms are evaluated with two sets of performance metrics. The performance of sensing and predicting mobility demands is evaluated using (a) root mean square error (RMSE) $\sqrt{|V|^{-1}\sum_{s\in V}\left(y_s-\mu^{\ell\text{GP}}_{s|\mathcal{D}}\right)^2}$ where y_s is the demand measurement and $\mathcal{D}$ is the set of regions observed by the MoD vehicles, and (b) incurred time of the algorithms.

The performance of servicing mobility demands is evaluated by comparing the Kullback-Leibler divergence (KLD) $\sum_{s\in V} P_c(s)\log\left(P_c(s)/P_d(s)\right)$ between the fleet distribution P_c of vacant MoD vehicles controlled by the tested algorithms and historic demand distribution P_d (i.e., lower KLD implies better balance between fleet and demand), average cruising length of MoD vehicles, average waiting time of users, and total number of pickups resulting from the tested algorithms.

7.4 Results and Analysis

For notational simplicity, we will use GP-DDF$^+$, FGP, and GP-DDF to represent the algorithms of their corresponding data fusion components coupled with FDAS strategy in this section.

7.4.1 Performance

The MoD system comprises $K = 20$ vehicles running three tested algorithms for $L = 960$ time steps in a service area with $C = 200$ users. All results are taken from the average of 40 random instances.

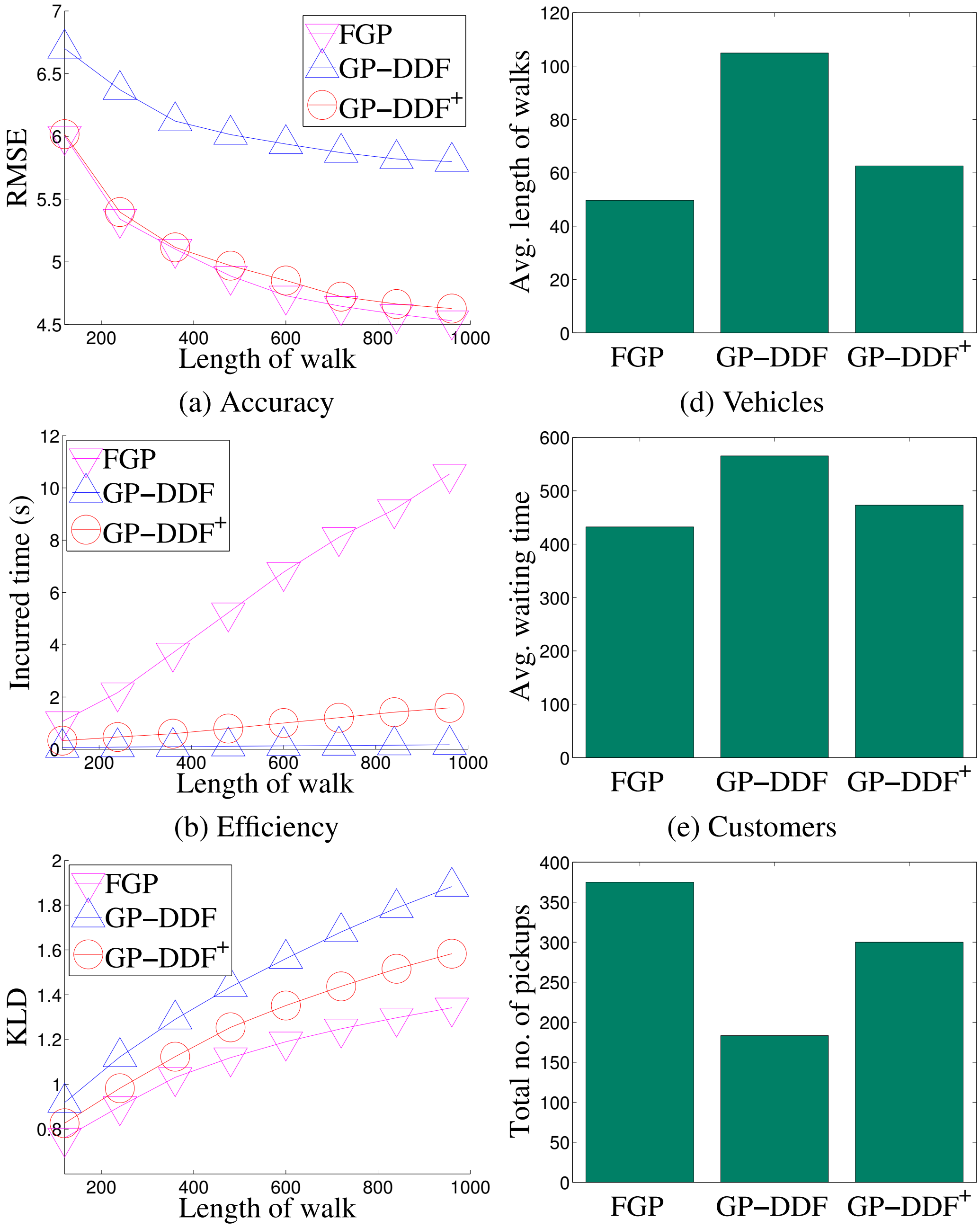
The performance of MoD systems in sensing and predicting mobility demands is illustrated in Figures 7.2a-7.2b. Figure 7.2a shows that the demand data collected by MoD vehicles using GP-DDF$^+$ can achieve predictive accuracy comparable to that of using FGP and significantly better than that of using GP-DDF. This indicates that exploiting the local data of vehicles for predicting demands of nearby unobserved regions can improve the prediction of the mobility demand pattern. Figure 7.2b shows the average incurred time of each vehicle using three algorithms. GP-DDF$^+$ is significantly more time-efficient (i.e., one order of magnitude) than FGP, and only slightly less time-efficient than GP-DDF. This can be explained by the time analysis in Section 7.2.1. The above results indicate that GP-DDF$^+$ is more practical for real-world deployment due to a better balance between predictive accuracy and time efficiency.

The performance of MoD systems in servicing the mobility demands is illustrated in Figures 7.2c-7.2f. Figure 7.2c shows that a MoD system using GP-DDF$^+$ can achieve better fleet rebalancing of vehicles to service mobility demands than GP-DDF, but worse rebalancing than FGP. This implies that a better prediction of the underlying mobility demand pattern (Figure 7.2a) can lead to better fleet rebalancing. Note that KLD (i.e., imbalance between mobility demand and fleet) increases over time because we assume that when a vehicle picks up a user, its local data is removed from the fleet of cruising vehicles, and a new vehicle is introduced at a random location that may be distant from a demand hotspot, hence worsening the imbalance between demand and fleet. It can also be observed that an algorithm generating a better balance between fleet and demand will also perform better in servicing the mobility demands, that is, shorter average cruising trajectories of vehicles (Figure 7.2d), shorter average waiting time of users (Figure 7.2e), and larger total number of pickups (Figure 7.2f). These observations imply that exploiting an active sensing strategy to collect the most informative demand data for predicting the mobility pattern achieves a dual effect of improving performance in servicing the mobility demands since these vehicles have higher chance of picking up users in demand hotspots or sparsely sampled regions (Section 5.2.4).

7.4.2 Scalability

We vary the number $K = 10, 20, 30$ of vehicles in the MoD system, and keep the total length of walks of all the vehicles to be the same, that is, these vehicles will walk for $L = 960, 480, 320$ steps, respectively. All three algorithms are tested in a service area with $C = 600$ customers.

From Figures 7.3a-7.3c, it can be observed that all three algorithms can improve their prediction accuracy with an increasing number of vehicles in the MoD system because more vehicles indicate less walks when the total length of walks are the same, thus suffering less from the myopic planning ($H = 4$) and gathering more informative demand data. Figures 7.3d-7.3f show that, with more MoD vehicles, GP-DDF$^+$ and GP-DDF incur less time, while FGP incurs more time. This is because the computational load in decentralized data fusion



(a) Accuracy

(d) Vehicles

(b) Efficiency

(e) Customers

(c) Balance

(f) Pickups

Figure 7.2: Performance of MoD systems in predicting and servicing mobility demands.

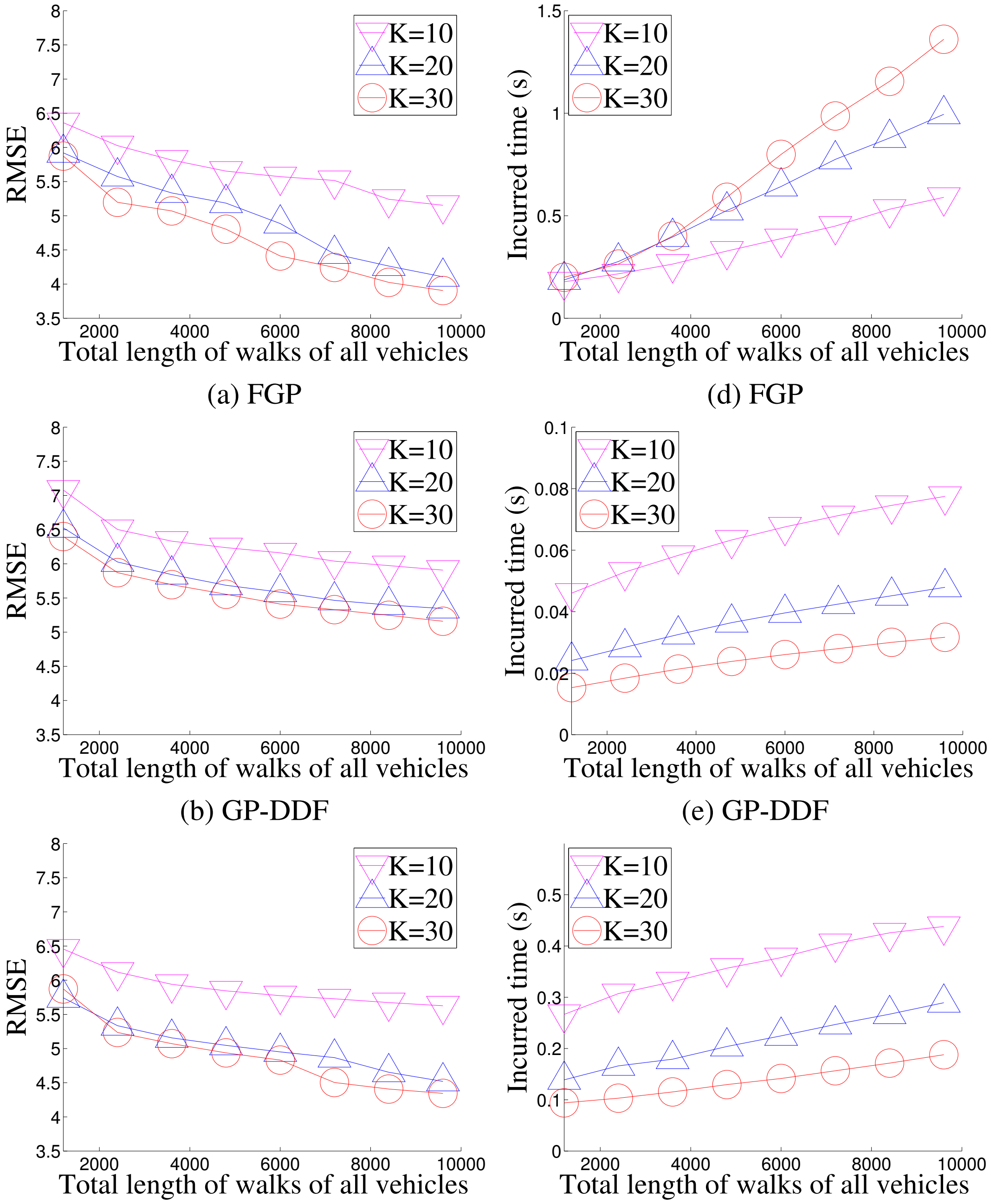
algorithms are distributed among all vehicles, thus reducing the incurred time with more vehicles.

Figures 7.4a-7.4c show that all three algorithms can achieve better balance between mobility demand and fleet with larger number of vehicles. It can also be observed that all three algorithms can improve the performance of servicing the mobility demand with more vehicles, that is, shorter average cruising trajectories of vehicles (Figure 7.4d), shorter average waiting time of users (Figure 7.4e), and larger total number of pickups (Figure 7.4f). This is because MoD vehicles can collect more informative demand data with larger number of vehicles sampling demand hotspots or sparsely sampled regions, which are the regions with higher chance of picking up users than the rest of the service area.

The above results indicate that more vehicles in MoD system result in better accuracy in predicting the mobility demand pattern, and achieve a dual effect of better performance in servicing mobility demands.

7.4.3 Summary of Empirical Result

GP-DDF$^+$ can predict mobility demand pattern more accurate than GP-DDF and closely to FGP. In addition, GP-DDF$^+$ scales significantly better than FGP in size of data with a large fleet size, and is close to GP-DDF. This indicates that GP-DDF$^+$ achieves a better balance between predictive accuracy and time efficiency than GP-DDF and FGP. GP-DDF$^+$ achieves a closer balance between fleet and mobility demand to centralized FGP than GP-DDF; thus, GP-DDF$^+$ achieves better performance in servicing mobility demand (i.e., shorter cruising time, shorter waiting time, and more pickups) than GP-DDF. When fleet size increases, GP-DDF$^+$, GP-DDF and FGP all improve the performance in demand sensing and servicing. However, GP-DDF$^+$ becomes more time-efficient while FGP is less time-efficient, since the computational load of GP-DDF$^+$ is distributed to a larger number of agents. To sum up, GP-DDF$^+$ is more practical for real-world deployment than GP-DDF and FGP.



(a) FGP

(d) FGP

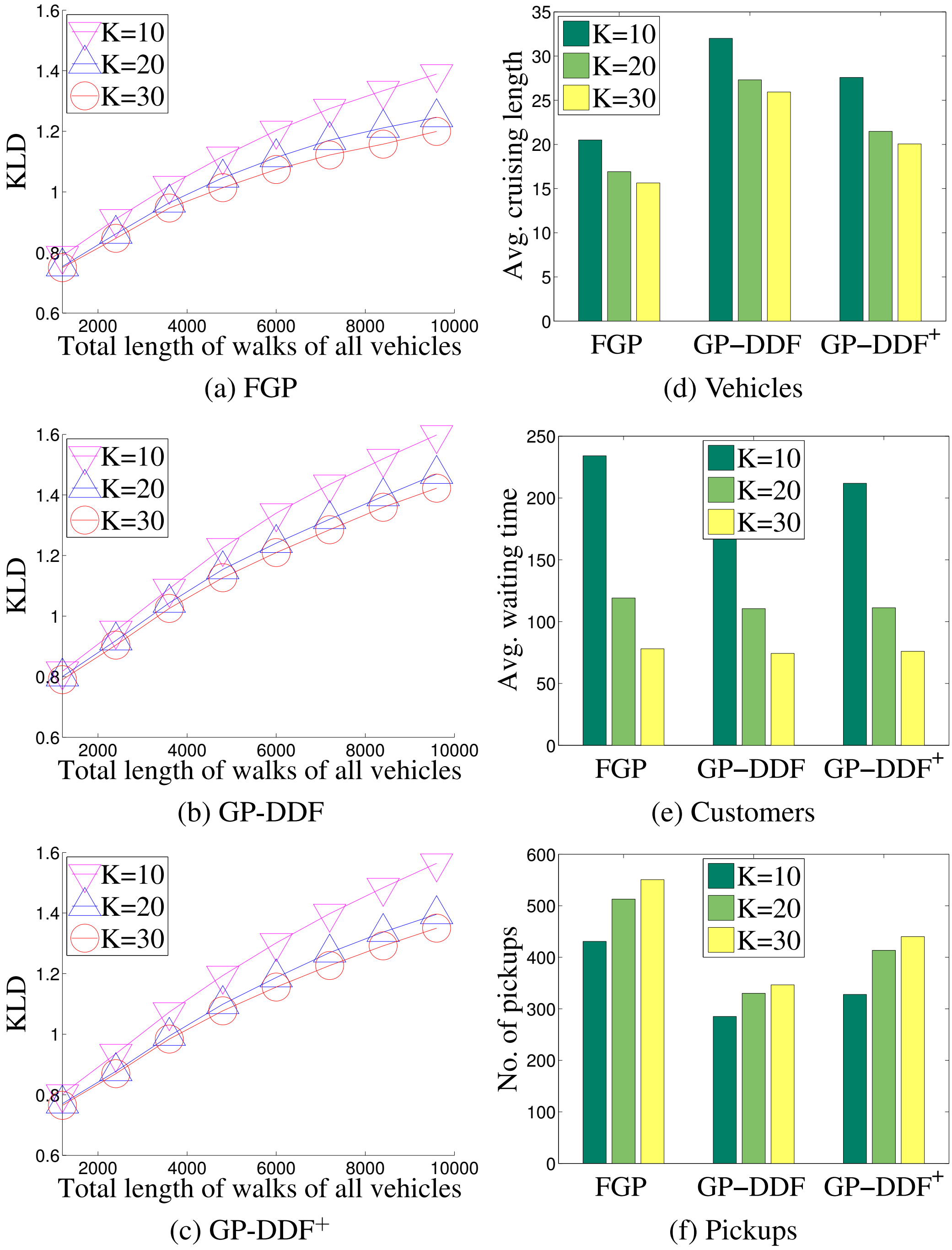
(b) GP-DDF

(e) GP-DDF

(c) GP-DDF$^+$

(f) GP-DDF$^+$

Figure 7.3: Scalability of MoD systems in sensing and predicting mobility demands.



(a) FGP

(d) Vehicles

(b) GP-DDF

(e) Customers

(c) GP-DDF$^+$

(f) Pickups

Figure 7.4: Scalability of MoD systems in servicing mobility demands.

Chapter 8

Conclusion & Future Work

This thesis presents a set of novel techniques based on a class of Bayesian nonparametric models: *Gaussian processes* (GP), which aim to 1) achieve accurate traffic modeling and prediction in real world situation; 2) provide efficient and scalable traffic prediction with a large phenomenon data; 3) and perform decentralized perception of spatiotemporal traffic phenomenon with mobile sensors. The proposed algorithms have been successfully applied in large-scale modeling and prediction of spatiotemporal environmental phenomena (i.e., urban traffic phenomena). In the following, Section 8.1 summarizes the particular contributions of this thesis; Section 8.2 describes the limitations and presents the directions for future research.

8.1 Contributions

First, we propose a novel relational GP to accurately model spatiotemporal traffic phenomena in real world situation (i.e., a traffic condition over road network and an urban mobility demand pattern containing skewness and extremity in measurements).

Second, we present three novel parallel GPs: *parallel partially independent training conditional* (pPITC), *parallel partially independent conditional*(pPIC) and *parallel incomplete Cholesky factorization* (pICF)-based approximations of GP model, which can distribute computational load into parallel/multi-core machines, thereby achieving real-time prediction given a large phenomenon data. The predictive performances of such parallel GPs are theoretically guaranteed to be equivalent to that of some centralized approaches to approximate GP regression. Furthermore, the proposed parallel GPs are implemented using the *message passing interface* (MPI) framework to run in a cluster of 20 computing nodes. Both theoretical and empirical results show that our parallel GPs achieve significantly better time efficiency than that of full GP while achieving comparable accuracy; the parallel GPs also achieve fine speedups to their centralized counterparts.

Third, we propose a decentralized algorithm framework: *Gaussian process-based decentralized data fusion and active sensing* (D^2FAS) which is composed of a *decentralized data fusion* (DDF) component that can cooperatively assimilate the distributed traffic phenomenon data into a globally consistent predictive model and a *decentralized active sensing* (DAS) component that can guide mobile sensors to cooperatively collect the most informative phenomenon data.

DDF component: We propose a novel *Gaussian process-based decentralized data fusion* (GP-DDF) algorithm that can achieve remarkably efficient and scalable prediction of phenomenon and a novel *Gaussian process-based decentralized data fusion with local augmentation* (GP-DDF$^+$) algorithm that can achieve better predictive accuracy while preserving time efficiency of GP-DDF; The predictive performances of both GP-DDF and GP-DDF$^+$ are theoretically guaranteed to be equivalent to that of some sophisticated centralized sparse approximations of exact/full GP.

DAS component: We first propose a novel *partially decentralized active sensing* (PDAS) algorithm which exploits property in correlation structure of GP-DDF to enable mobile sensors cooperatively selecting a joint walk of approximated maximum posterior Gaussian entropy; The performance of PDAS is theoretically guaranteed, and various practical environment conditions can be established to ensure it be comparably well. Then, in certain situation where PDAS algorithm cannot perform or perform poorly, a *fully decentralized active sensing* (FDAS) algorithm is proposed to make each mobile sensor gather phenomenon data along its locally optimal walk.

Lastly, we propose D^2FAS algorithms running with active mobile sensors for monitoring traffic conditions and sensing/servicing urban mobility demands; These algorithms are then simulated on two real-world datasets. The theoretical and empirical results show that the proposed D^2FAS algorithms are significantly more time-efficient, more scalable in the size of data and number of sensors than the state-of-the-art centralized approaches, while achieving comparable predictive accuracy. Therefore, the proposed D^{2}FAS framework is of significant value in practical deployment of active mobile sensors to monitor traffic conditions over road networks and to sense/service urban mobility demands.

8.2 Future Directions

Our D^{2}FAS framework opens many research avenues for future studies:

Firstly, current decentralized data fusion algorithms rely on a sufficiently large common support set to compute accurate enough prediction of the field. This size could be very large when the unknown environment phenomenon is on a large scale and requires more points for summarizing data and predicting unobserved locations. However, a larger size of support set will increase both the time and communication overheads consequently hindering the mobile sensor network from scaling up. However, it is not fully addressed in existing literatures in terms of the optimal size and position of the support set. Therefore, one corresponding direction to tackle is to reduce the impact caused by a large common support set on the performance of DDF algorithms via deciding an optimal size and position of the common support.

Secondly, currently PDAS (Section 5.2.3) algorithm results in time spiking (see Figure 6.3d) in the earlier stage when a large cluster of mobile sensors is formed due to random positioning assumption. This problem is not serious when mobile sensor have a tendency of spreading out in the long term. However, it will incur a large amount of time if mobile sensors tend to walk into large clusters; Consequently, the sensor network cannot scale up stably. By addressing this unstableness, we can further increase the number of mobile sensors performing environmental sensing tasks in D^{2}FAS framework.

Finally, current D^{2}FAS framework assume a fairly good communication condition. However, the communication channels in practice can be occasionally unavailable because of limited communication range and distorted wireless signal; the limited mobility of mobile sensors and large-scale environmental field cause the full communication network to break apart into sub-networks which can split and merge dynamically. It is more challenging to relax the assumption that mobile sensors collected the same environmental measurement given the same inputs. For our future work, the D^{2}FAS framework can be enhanced to work under these practical conditions.

Bibliography

[Agussurja and Lau, 2012] L. Agussurja and H. C. Lau. Toward large-scale agent guidance in an urban taxi service. In *Proc. UAI*, pages 36–43, 2012.

[Bekkerman *et al.*, 2011] R. Bekkerman, M. Bilenko, and J. Langford. *Scaling up Machine Learning: Parallel and Distributed Approaches*. Cambridge Univ. Press, NY, 2011.

[Berbeglia *et al.*, 2010] G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *EJOR*, 202(1):8–15, 2010.

[Borg and Groenen, 2005] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, NY, 2005.

[Bryan *et al.*, 2005] Brent Bryan, Jeff Schneider, Robert C. Nichol, Christopher J. Miller, Christopher R. Genovese, and Larry Wasserman. Active learning for identifying function threshold boundaries. In *Proc. NIPS*, 2005.

[Chang *et al.*, 2007] E. Y. Chang, K. Zhu, H. Wang, H. Bai, J. Li, Z. Qiu, and H. Cui. Parallelizing support vector machines on distributed computers. In *Proc. NIPS*, 2007.

[Chang *et al.*, 2010] H.-W. Chang, Y.-C. Tai, and J. Y.-J. Hsu. Context-aware taxi demand hotspots prediction. *Int. J. Business Intelligence and Data Mining*, 5(1):3–18, 2010.

[Chen *et al.*, 2011] H. Chen, H. A. Rakha, and S. Sadek. Real-time freeway traffic state prediction: A particle filter approach. In *Proc. IEEE ITSC*, pages 626–631, 2011.

[Chen *et al.*, 2012] J. Chen, K. H. Low, C. K.-Y. Tan, A. Oran, P. Jaillet, J. M. Dolan, and G. S. Sukhatme. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, pages 163–173, 2012.

[Chen *et al.*, 2013a] Jie Chen, Nannan Cao, Kian Hsiang Low, Ruofei Ouyang, Colin Keng-Yan Tan, and Patrick Jaillet. Parallel gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, pages 152–161, May 2013.

[Chen *et al.*, 2013b] Jie Chen, Kian Hsiang Low, and Colin Keng-Yan Tan. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*, June 2013.

[Choi and Oh, 2008] Joonho Choi, Jongeun Lee and Songhwai Oh. Biologically-Inspired Navigation Strategies for Swarm Intelligence Using Spatial Gaussian Processes. In *In Proceedings of the 17th IFAC World Congress*, 2008.

[Choi *et al.*, 2007] Jongeun Choi, Songhwai Oh, and R. Horowitz. Cooperatively learning mobile agents for gradient climbing. In *Proc. CDC*, pages 3139 –3144, December 2007.

[Choudhury *et al.*, 2002] A. Choudhury, P. B. Nair, and A. J. Keane. A data parallel approach for large-scale Gaussian process modeling. In *Proc. SDM*, pages 95–111, 2002.

[Chung *et al.*, 2004] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray. On a decentralized active sensing strategy using mobile sensor platforms in a network. In *Proc. CDC*, pages 1914–1919, 2004.

[Coates, 2004] M. Coates. Distributed particle filters for sensor networks. In *Proc. IPSN*, pages 99–107, 2004.

[Cortes, 2009] J. Cortes. Distributed Kriged Kalman filter for spatial estimation. *IEEE Trans. Automat. Contr.*, 54(12):2816–2827, 2009.

[Das and Srivastava, 2010] K. Das and A. N. Srivastava. Block-GP: Scalable Gaussian process regression for multimodal data. In *Proc. ICDM*, pages 791–796, 2010.

[Dolan *et al.*, 2009] John M. Dolan, Gregg Podnar, Stephen B. Stancliff, Kian Hsiang Low, Alberto Elfes, John Higinbotham, Jeffrey Hosler, Tiffany Moisan, and John Moisan. Cooperative aquatic sensing using the telesupervised adaptive ocean sensor fleet. In *Proc. IPSN-09 Workshop on Sensor Networks for Earth and Space Science Applications*, volume 7473, September 2009.

[Frank *et al.*, 2011] Barbara Frank, Cyrill Stachniss, Nichola Abdo, and Wolfram Burgard. Using gaussian process regression for efficient motion planning in environments with deformable objects. In *Proc. AAAI Workshop on Automated Action Planning for Autonomous Mobile Robots*, volume WS-11-09 of *AAAI Workshops*. AAAI, 2011.

[Furrer *et al.*, 2006] R. Furrer, M. G. Genton, and D. Nychka. Covariance tapering for interpolation of large spatial datasets. *JCGS*, 15(3):502–523, 2006.

[Ge *et al.*, 2010] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani. An energy-efficient mobile recommender system. In *Proc. ACM SIGKDD*, pages 899–908, 2010.

[GM, 2012] *GM Shows Chevrolet EN-V 2.0 Mobility Concept Vehicle*. General Motors Co. (http://media.gm.com/media/us/en/gm/news.detail.html/content/Pages/news/us/en/2012/Apr/0423_EN-V_2_Rendering.html), 2012.

[Golub and Van Loan, 1996] G. H. Golub and C.-F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, 3rd edition, 1996.

[Graham and Cortés, 2009] Rishi Graham and Jorge Cortés. Distributed sampling of random fields with unknown covariance. In *Proc. ACC*, ACC'09, pages 4543–4548, 2009.

[Graham and Cortes, 2010] R. Graham and J. Cortes. Spatial statistics and distributed estimation by robotic sensor networks. In *Proc. ACC*, pages 2422 –2427, 2010.

[Graham and Cortes, 2011] Rishi Graham and Jorge Cortes. Cooperative adaptive sampling of random fields with partially known covariance. *International Journal of Robust and Nonlinear Control*, 2011.

[Guestrin *et al.*, 2004] C. Guestrin, P. Bodik, R. Thibaus, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Proc. IPSN*, pages 1–10, 2004.

[Herring *et al.*, 2010] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. Estimating arterial traffic conditions using sparse probe data. In *Proc. IEEE ITSC*, pages 929–936, 2010.

[Hofleitner *et al.*, 2012a] A. Hofleitner, R. Herring, and A. Bayen. Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach. In *Proc. TRB 91st Annual Meeting*, 2012.

[Hofleitner *et al.*, 2012b] A. Hofleitner, R. Herring, A. Bayen, Y. Han, F. Moutarde, and A. de La Fortelle. Large-scale estimation of arterial traffic and structural analysis of traffic patterns from probe vehicles. In *Proc. TRB 91st Annual Meeting*, 2012.

[Hohn, 1998] M. E. Hohn. *Geostatistics and Petroleum Geology*. Springer, 2nd edition, 1998.

[Hollinger *et al.*, 2012] G.A. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme. Uncertainty-driven view planning for underwater inspection. In *Proc. ICRA*, pages 4884–4891, 2012.

[Ingram and Cornford, 2010] B. Ingram and D. Cornford. Parallel geostatistics for sparse and dense datasets. In P. M. Atkinson and C. D. Lloyd, editors, *Proc. geoENV VII*, pages 371–381. Quantitative Geology and Geostatistics Volume 16, Springer, Netherlands, 2010.

[Ipsen and Lee, 2003] I. C. F. Ipsen and D. J. Lee. Determinant approximations. Technical Report CRSC-TR03-30, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, 2003.

[Jelasity *et al.*, 2005] M. Jelasity, A. Montresor, and O. Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM T. Comput. Syst.*, 23(3):219–252, 2005.

[Kamarianakis and Prastacos, 2003] Y. Kamarianakis and P. Prastacos. Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transport. Res. Rec.*, 1857:74–84, 2003.

[Krause *et al.*, 2008a] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *Proc. IPSN*, pages 481–492, 2008.

[Krause *et al.*, 2008b] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

[Lawrence *et al.*, 2003] Neil D. Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, pages 609–616. MIT Press, 2003.

[Leonard *et al.*, 2007] N.E. Leonard, D.A. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, and R.E. Davis. Collective motion, sensor networks, and ocean sampling. *Proc. IEEE*, 95(1):48 –74, January 2007.

[Li *et al.*, 2012] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, and Z. Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Front. Comput. Sci.*, 6(1):111–121, 2012.

[Low *et al.*, 2008a] K. H. Low, J. M. Dolan, and P. Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proc. AAMAS*, pages 23–30, 2008.

[Low *et al.*, 2008b] K. H. Low, J. M. Dolan, and P. Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proc. AAMAS*, pages 23–30, 2008.

[Low *et al.*, 2009a] K. H. Low, J. M. Dolan, and P. Khosla. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proc. ICAPS*, pages 233–240, 2009.

[Low *et al.*, 2009b] K. H. Low, J. M. Dolan, and P. Khosla. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proc. ICAPS*, pages 233–240, 2009.

[Low *et al.*, 2009c] Kian Hsiang Low, Gregg Podnar, Stephen B. Stancliff, John M. Dolan, and Alberto Elfes. Robot boats as a mobile aquatic sensor network. In *2009 International Conference on Information Processing in*

Sensor Networks (IPSN) Workshop on Sensor Networks for Earth and Space Science Applications: ESSA 2009, April 2009.

[Low *et al.*, 2011] K. H. Low, J. M. Dolan, and P. Khosla. Active Markov information-theoretic path planning for robotic environmental sensing. In *Proc. AAMAS*, pages 753–760, 2011.

[Low *et al.*, 2012] Kian Hsiang Low, Jie Chen, John M. Dolan, Steve Chien, and David R. Thompson. Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing. In *Proc. AAMAS*, pages 105–112, 2012.

[Min and Wynter, 2011] W. Min and L. Wynter. Real-time road traffic prediction with spatio-temporal correlations. *Transport. Res. C-Emer.*, 19(4):606–616, 2011.

[Mitchell *et al.*, 2010] W. J. Mitchell, C. E. Borroni-Bird, and L. D. Burns. *Reinventing the Automobile: Personal Urban Mobility for the 21st Century*. MIT Press, Cambridge, MA, 2010.

[Mitchell, 2008] W. J. Mitchell. Mobility on demand: Future of transportation in cities. Technical Report, MIT Media Laboratory, 2008.

[Neumann *et al.*, 2009] M. Neumann, K. Kersting, Z. Xu, and D. Schulz. Stacked Gaussian process learning. In *Proc. ICDM*, pages 387–396, 2009.

[Oh *et al.*, 2010] Songhwai Oh, Yunfei Xu, and Jongeun Choi. Explorative navigation of mobile sensor networks using sparse Gaussian processes. In *Proc. CDC*, pages 3851–3856, dec. 2010.

[Olfati-Saber and Shamma, 2005] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proc. CDC*, pages 6698–6703, 2005.

[Park *et al.*, 2011] C. Park, J. Z. Huang, and Y. Ding. Domain decomposition approach for fast Gaussian process regression of large spatial data sets. *JMLR*, 12:1697–1728, 2011.

[Paskin and Guestrin, 2004] M. A. Paskin and C. Guestrin. Robust probabilistic inference in distributed systems. In *Proc. UAI*, pages 436–445, 2004.

[Pavone *et al.*, 2012] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus. Robotic load balancing for mobility-on-demand systems. *IJRR*, 31(7):839–854, 2012.

[Peeta and Ziliaskopoulos, 2001] S. Peeta and A. K. Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1:233–265, 2001.

[Pjesivac-Grbovic *et al.*, 2007] J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. E. Fagg, E. Gabriel, and J. Dongarra. Performance analysis of MPI collective operations. *Cluster Computing*, 10(2):127–143, 2007.

[Podnar *et al.*, 2010] Gregg Podnar, John M. Dolan, Kian Hsiang Low, and Alberto Elfes. Telesupervised remote surface water quality sensing. In *Proc. IEEE Aerospace Conference*, pages 1–9, March 2010.

[Popa and Lewis, 2008] Dan O. Popa and Frank L. Lewis. Algorithms for robotic deployment of WSN in adaptive sampling applications. In Yingshu Li, My T. Thai, and Weili Wu, editors, *Wireless Sensor Networks and Applications*, Signals and Communication Technology, pages 35–64. Springer US, 2008.

[Powell *et al.*, 2011] J. W. Powell, Y. Huang, F. Bastani, and M. Ji. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *Proc. SSTD*, 2011.

[Quiñonero-Candela and Rasmussen, 2005] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *JMLR*, 6:1939–1959, 2005.

[Rahimi *et al.*, 2005] M. Rahimi, M. Hansen, W.J. Kaiser, G.S. Sukhatme, and D. Estrin. Adaptive sampling for environmental field estimation using robotic sensors. In *Proc. IROS*, pages 3692 – 3698, August 2005.

[Rasmussen and Williams, 2006] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[Rosencrantz *et al.*, 2003] M. Rosencrantz, G. Gordon, and S. Thrun. Decentralized sensor fusion with distributed particle filters. In *Proc. UAI*, pages 493–500, 2003.

[RPT, 2012] *Hong Kong in Figures*. Census and Statistics Department, Hong Kong Special Administrative Region (http://www.censtatd.gov.hk); *Singapore Land Transport: Statistics in Brief*. Land Transport Authority of Singapore (http://www.lta.gov.sg), 2012.

[Schölkopf and Smola, 2002] Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. The MIT Press, 1st edition, 2002.

[Schrank *et al.*, 2011] D. Schrank, T. Lomax, and B. Eisele. *TTI's 2011 Urban Mobility Report*. Texas Transportation Institute, Texas A&M University, 2011.

[Schwaighofer and Tresp, 2002] A. Schwaighofer and V. Tresp. Transductive and inductive methods for approximate Gaussian process regression. In *Proc. NIPS*, pages 953–960, 2002.

[Seeger and Williams, 2003] M. Seeger and C. Williams. Fast forward selection to speed up sparse Gaussian process regression. In *Proc. AISTATS*, 2003.

[Singh *et al.*, 2006] Aarti Singh, Robert Nowak, and Parmesh Ramanathan. Active learning for adaptive mobile sensing networks. In *Proc. IPSN*, IPSN '06, pages 60–68, New York, NY, USA, 2006. ACM.

[Singh *et al.*, 2007] Amarjeet Singh, Andreas Krause, Carlos Guestrin, William Kaiser, and Maxim Batalin. Efficient planning of informative paths for multiple robots. In *Proc. IJCAI*, IJCAI'07, pages 2204–2211, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[Snelson and Ghahramani, 2005] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Proc. NIPS*, 2005.

[Snelson, 2007] E. Snelson. Local and global sparse Gaussian process approximations. In *Proc. AISTATS*, 2007.

[Srebotnjak *et al.*, 2010] Tanja Srebotnjak, Christine Polzin, Stefan Giljum, Sophie Herbert, and Stephan Lutter. Establishing environmental sustainability thresholds and indicators final report. Technical report, Sustainable Europe Research Institute, 2010.

[Srinivasan and Jovanis, 1996] K. K. Srinivasan and P. P. Jovanis. Determination of number of probe vehicle required for reliable travel time measurement in urban network. *Transport. Res. Rec.*, 1537:15–22, 1996.

[Stewart and Sun, 1990] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[Stranders *et al.*, 2009] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. In *Proc. IJCAI*, pages 299–304, 2009.

[Sukkarieh *et al.*, 2003] Salah Sukkarieh, Eric Nettleton, Jong-Hyuk Kim, Matthew Ridley, Ali Goktogan, and Hugh Durrant-Whyte. The ANSER project: Data fusion across multiple uninhabited air vehicles. *The International Journal of Robotics Research*, 22(7-8):505–539, 2003.

[Turner *et al.*, 1998] S. M. Turner, W. L. Eisele, R. J. Benz, and D. J. Holdener. Travel time data collection handbook. Technical Report FHWA-PL-98-035, Federal Highway Administration, Office of Highway Information Management, Washington, DC, 1998.

[UNS, 2010] United nations millennium campaign: Ensure environmental sustainability (http://www.millenniumcampaign.org/goal-7-ensure-environmental-sustainability), 2010.

[Vanhatalo and Vehtari, 2008] J. Vanhatalo and A. Vehtari. Modeling local and global phenomena with sparse Gaussian processes. In *Proc. UAI*, pages 571–578, 2008.

[Vasudevan *et al.*, 2009] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair. Gaussian process modeling of large scale terrain. In *Proc. ICRA*, pages 1047–1053, 2009.

[Vijayakumar *et al.*, 2005] S. Vijayakumar, A. D'Souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Comput.*, 17(12):2602–2634, 2005.

[Wang and Papageorgiou, 2005] Y. Wang and M. Papageorgiou. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transport. Res. B-Meth.*, 39(2):141–167, 2005.

[Webster and Oliver, 2007] R. Webster and M. Oliver. *Geostatistics for Environmental Scientists*. John Wiley & Sons, Inc., NY, 2nd edition, 2007.

[Williams and Seeger, 2000] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Proc. NIPS*, pages 682–688, 2000.

[Work *et al.*, 2010] D. B. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. Bayen. A traffic model for velocity data assimilation. *AMRX*, 2010(1):1–35, 2010.

[Yuan *et al.*, 2012] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. T-Finder: A recommender system for finding passengers and vacant taxis. *IEEE T. Knowl. Data. En*, 2012.

[Zhang and Sukhatme, 2007] Bin Zhang and G.S. Sukhatme. Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network. In *Proc. ICRA*, pages 3673 –3680, April 2007.

Appendices

Appendix A

Proof of Theorem 1

We have to first simplify the $\Gamma_{\mathcal{UD}}\left(\Gamma_{\mathcal{DD}}+\Lambda\right)^{-1}$ term in the expressions of $\mu^{\text{PITC}}_{\mathcal{U}|\mathcal{D}}$ (4.10) and $\Sigma^{\text{PITC}}_{\mathcal{UU}|D}$ (4.11).

$$
\begin{aligned}
&\left(\Gamma_{\mathcal{DD}}+\Lambda\right)^{-1} \\
&=\left(\Sigma_{\mathcal{DS}}\Sigma^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{SD}}+\Lambda\right)^{-1} \\
&=\Lambda^{-1}-\Lambda^{-1}\Sigma_{\mathcal{DS}}\left(\Sigma_{\mathcal{SS}}+\Sigma_{\mathcal{SD}}\Lambda^{-1}\Sigma_{\mathcal{DS}}\right)^{-1}\Sigma_{\mathcal{SD}}\Lambda^{-1} \\
&=\Lambda^{-1}-\Lambda^{-1}\Sigma_{\mathcal{DS}}\ddot{\Sigma}^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{SD}}\Lambda^{-1}\ .
\end{aligned}
\tag{A.1}
$$

The second equality follows from matrix inversion lemma. The last equality is due to

$$
\begin{aligned}
&\Sigma_{\mathcal{SS}}+\Sigma_{\mathcal{SD}}\Lambda^{-1}\Sigma_{\mathcal{DS}} \\
&=\Sigma_{\mathcal{SS}}+\sum_{m=1}^{M}\Sigma_{\mathcal{SD}_m}\Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}\Sigma_{\mathcal{D}_m\mathcal{S}} \\
&=\Sigma_{\mathcal{SS}}+\sum_{m=1}^{M}\dot{\Sigma}^{m}_{\mathcal{SS}}=\ddot{\Sigma}_{\mathcal{SS}}\ .
\end{aligned}
\tag{A.2}
$$

Using (4.12) and (A.1),

$$
\begin{aligned}
&\Gamma_{\mathcal{U}_m\mathcal{D}}\left(\Gamma_{\mathcal{DD}}+\Lambda\right)^{-1} \\
&=\Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{SD}}\left(\Lambda^{-1}-\Lambda^{-1}\Sigma_{\mathcal{DS}}\ddot{\Sigma}^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{SD}}\Lambda^{-1}\right) \\
&=\Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}}\left(\ddot{\Sigma}_{\mathcal{SS}}-\Sigma_{\mathcal{SD}}\Lambda^{-1}\Sigma_{\mathcal{DS}}\right)\ddot{\Sigma}^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{SD}}\Lambda^{-1} \\
&=\Sigma_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{SD}}\Lambda^{-1}
\end{aligned}
\tag{A.3}
$$

The third equality is due to (A.2).

For each machine $m = 1, \ldots, M$, we can now prove that

$$\begin{aligned}
\mu^{\text{PITC}}_{\mathcal{U}_m|\mathcal{D}} &= \mu_{\mathcal{U}_m} + \Gamma_{\mathcal{U}_m\mathcal{D}} \left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1} \left(y_{\mathcal{D}} - \mu_{\mathcal{D}}\right) \\
&= \mu_{\mathcal{U}_m} + \Sigma_{\mathcal{U}_m\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{D}} \Lambda^{-1} \left(y_{\mathcal{D}} - \mu_{\mathcal{D}}\right) \\
&= \mu_{\mathcal{U}_m} + \Sigma_{\mathcal{U}_m\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}} \ddot{y}_{\mathcal{S}} \\
&= \widehat{\mu}_{\mathcal{U}_m} \ .
\end{aligned}$$

The first equality is by definition (4.10). The second equality is due to (A.3). The third equality follows from $\Sigma_{\mathcal{S}\mathcal{D}}\Lambda^{-1}\left(y_{\mathcal{D}} - \mu_{\mathcal{D}}\right) = \sum_{m=1}^{M} \Sigma_{\mathcal{S}\mathcal{D}_m} \Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}} \left(y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}\right) = \sum_{m=1}^{M} \dot{y}^m_{\mathcal{S}} = \ddot{y}_{\mathcal{S}}$. Also,

$$\begin{aligned}
&\Sigma^{\text{PITC}}_{\mathcal{U}_m\mathcal{U}_m|\mathcal{D}} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \Gamma_{\mathcal{U}_m\mathcal{D}} \left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1} \Gamma_{\mathcal{D}\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \Sigma_{\mathcal{U}_m\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{D}} \Lambda^{-1} \Sigma_{\mathcal{D}\mathcal{S}} \Sigma^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \Big(\Sigma_{\mathcal{U}_m\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{D}} \Lambda^{-1} \Sigma_{\mathcal{D}\mathcal{S}} \Sigma^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m} \\
&\quad - \Sigma_{\mathcal{U}_m\mathcal{S}} \Sigma^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m}\Big) - \Sigma_{\mathcal{U}_m\mathcal{S}} \Sigma^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \Sigma_{\mathcal{U}_m\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}} \left(\Sigma_{\mathcal{S}\mathcal{D}} \Lambda^{-1} \Sigma_{\mathcal{D}\mathcal{S}} - \ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\right) \Sigma^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m} \\
&\quad - \Sigma_{\mathcal{U}_m\mathcal{S}} \Sigma^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \left(\Sigma_{\mathcal{U}_m\mathcal{S}} \Sigma^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m} - \Sigma_{\mathcal{U}_m\mathcal{S}} \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{U}_m}\right) \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \Sigma_{\mathcal{U}_m\mathcal{S}} \left(\Sigma^{-1}_{\mathcal{S}\mathcal{S}} - \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\right) \Sigma_{\mathcal{S}\mathcal{U}_m} \\
&= \widehat{\Sigma}_{\mathcal{U}_m\mathcal{U}_m} \ .
\end{aligned} \tag{A.4}$$

The first equality is by definition (4.11). The second equality follows from (4.12) and (A.3). The fifth equality is due to (A.2).

Since our primary interest in the work of this paper is to provide the predictive means and their corresponding predictive variances, the above equivalence results suffice. However, if the entire predictive covariance matrix $\widehat{\Sigma}_{\mathcal{U}\mathcal{U}}$ for any set $\mathcal{U}$ of inputs is desired (say, to calculate the joint entropy), then it is necessary to compute $\widehat{\Sigma}_{\mathcal{U}_i\mathcal{U}_j}$ for $i, j = 1, \ldots, M$ such that $i \neq j$. Define

$$\widehat{\Sigma}_{\mathcal{U}_i\mathcal{U}_j} \triangleq \Sigma_{\mathcal{U}_i\mathcal{U}_j} - \Sigma_{\mathcal{U}_i\mathcal{S}} \left(\Sigma^{-1}_{\mathcal{S}\mathcal{S}} - \ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\right) \Sigma_{\mathcal{S}\mathcal{U}_j} \tag{A.5}$$

for $i, j = 1, \ldots, M$ such that $i \neq j$. So, for a machine i to compute $\widehat{\Sigma}_{\mathcal{U}_i\mathcal{U}_j}$, it has to receive $\mathcal{U}_j$ from machine j.

Similar to (A.4), we can prove the equivalence result $\widehat{\Sigma}_{\mathcal{U}_i\mathcal{U}_j} = \Sigma^{\text{PITC}}_{\mathcal{U}_i\mathcal{U}_j|\mathcal{D}}$ for any two machines $i, j = 1, \ldots, M$ such that $i \neq j$.

Appendix B

Proof of Theorem 2

We will first derive the expressions of four components useful for completing the proof later. For each machine $m = 1, \ldots, M$,

$$\begin{aligned}
&\widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) \\
&= \sum_{i \neq m} \Gamma_{\mathcal{U}_m\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}(y_{\mathcal{D}_i} - \mu_{\mathcal{D}_i}) \\
&\quad + \Sigma_{\mathcal{U}_m\mathcal{D}_m}\Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}(y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}) \\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}} \sum_{i \neq m} \left(\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}(y_{\mathcal{D}_i} - \mu_{\mathcal{D}_i})\right) + \dot{y}^m_{\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}} \sum_{i \neq m} \dot{y}^i_{\mathcal{S}} + \dot{y}^m_{\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}(\ddot{y}_{\mathcal{S}} - \dot{y}^m_{\mathcal{S}}) + \dot{y}^m_{\mathcal{U}_m}\ .
\end{aligned} \tag{B.1}$$

The first two equalities expand the first component using the definition of Λ (Theorem 1), (4.1), (4.12), (4.15), and (4.16). The last two equalities exploit (4.1) and (4.3).

$$\begin{aligned}
&\widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\Sigma_{\mathcal{D}\mathcal{S}} \\
&= \sum_{i \neq m} \Gamma_{\mathcal{U}_m\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Sigma_{\mathcal{D}_i\mathcal{S}} + \Sigma_{\mathcal{U}_m\mathcal{D}_m}\Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}\Sigma_{\mathcal{D}_m\mathcal{S}} \\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}} \sum_{i \neq m} \left(\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Sigma_{\mathcal{D}_i S}\right) \\
&\quad + \Sigma_{\mathcal{U}_m\mathcal{D}_m}\Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}\Sigma_{\mathcal{D}_m\mathcal{S}} \\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}} \sum_{i \neq m} \dot{\Sigma}^i_{\mathcal{S}\mathcal{S}} + \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{S}} \\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\left(\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} - \dot{\Sigma}^m_{\mathcal{S}\mathcal{S}} - \Sigma_{\mathcal{S}\mathcal{S}}\right) + \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{S}} \\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} - \Phi^m_{\mathcal{U}_m\mathcal{S}}\ .
\end{aligned} \tag{B.2}$$

The first two equalities expand the second component by the same trick as that in (B.1). The third and fourth equalities exploit (4.2) and (4.4), respectively. The last equality is due to (4.9).

Let $\alpha_{\mathcal{U}_m\mathcal{S}} \triangleq \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}$ and its transpose is $\alpha_{\mathcal{S}\mathcal{U}_m}$. By using similar tricks in (B.1) and (B.2), we can derive the expressions of the remaining two components.

$$\begin{aligned}
&\widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}_m}\\
&= \sum_{i\neq m} \Gamma_{\mathcal{U}_m\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Gamma_{\mathcal{D}_i\mathcal{U}_m} + \Sigma_{\mathcal{U}_m\mathcal{D}_m}\Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}\Sigma_{\mathcal{D}_m\mathcal{U}_m}\\
&= \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\sum_{i\neq m}\left(\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Sigma_{\mathcal{D}_i\mathcal{S}}\right)\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{U}_m} + \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{U}_m}\\
&= \alpha_{\mathcal{U}_m\mathcal{S}}\sum_{i\neq m}\left(\dot{\Sigma}^i_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_m} + \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{U}_m}\\
&= \alpha_{\mathcal{U}_m\mathcal{S}}\left(\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} - \dot{\Sigma}^m_{\mathcal{S}\mathcal{S}} - \Sigma_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_m} + \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{U}_m}\\
&= \alpha_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_m} - \alpha_{\mathcal{U}_m\mathcal{S}}\Phi^m_{\mathcal{S}\mathcal{U}_m} - \alpha_{\mathcal{U}_m\mathcal{S}}\dot{\Sigma}^m_{\mathcal{S}\mathcal{U}_m} + \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{U}_m}\ .
\end{aligned} \tag{B.3}$$

For any two machines $i, j = 1, \ldots, M$ such that $i \neq j$,

$$\begin{aligned}
&\widetilde{\Gamma}_{\mathcal{U}_i\mathcal{D}}\Lambda^{-1}\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}_j}\\
&= \sum_{m\neq i,j}\Gamma_{\mathcal{U}_i\mathcal{D}_m}\Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}\Gamma_{\mathcal{D}_m\mathcal{U}_j}\\
&\quad + \Sigma_{\mathcal{U}_i\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Gamma_{\mathcal{D}_i\mathcal{U}_j} + \Gamma_{\mathcal{U}_i\mathcal{D}_j}\Sigma^{-1}_{\mathcal{D}_j\mathcal{D}_j|\mathcal{S}}\Sigma_{\mathcal{D}_j\mathcal{U}_j}\\
&= \Sigma_{\mathcal{U}_i\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\sum_{m\neq i,j}\left(\Sigma_{\mathcal{S}\mathcal{D}_m}\Sigma^{-1}_{\mathcal{D}_m\mathcal{D}_m|\mathcal{S}}\Sigma_{\mathcal{D}_m\mathcal{S}}\right)\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{U}_j}\\
&\quad + \Sigma_{\mathcal{U}_i\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{U}_j}\\
&\quad + \Sigma_{\mathcal{U}_i\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_j}\Sigma^{-1}_{\mathcal{D}_j\mathcal{D}_j|\mathcal{S}}\Sigma_{\mathcal{D}_j\mathcal{U}_j}\\
&= \alpha_{\mathcal{U}_i\mathcal{S}}\left(\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} - \dot{\Sigma}^i_{\mathcal{S}\mathcal{S}} - \dot{\Sigma}^j_{\mathcal{S}\mathcal{S}} - \Sigma_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_j}\\
&\quad + \dot{\Sigma}^i_{\mathcal{U}_i\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} + \alpha_{\mathcal{U}_i\mathcal{S}}\dot{\Sigma}^j_{\mathcal{S}\mathcal{U}_j}\\
&= \alpha_{\mathcal{U}_i\mathcal{S}}\left(\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} + \Sigma_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_j} - \alpha_{\mathcal{U}_i\mathcal{S}}\left(\dot{\Sigma}^i_{\mathcal{S}\mathcal{S}} + \Sigma_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_j}\\
&\quad - \alpha_{\mathcal{U}_i\mathcal{S}}\left(\dot{\Sigma}^j_{\mathcal{S}\mathcal{S}} + \Sigma_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_j} + \dot{\Sigma}^i_{\mathcal{U}_i\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} + \alpha_{\mathcal{U}_i\mathcal{S}}\dot{\Sigma}^j_{\mathcal{S}\mathcal{U}_j}\\
&= \alpha_{\mathcal{U}_i\mathcal{S}}\left(\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} + \Sigma_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_j} - \left(\alpha_{\mathcal{U}_i\mathcal{S}}\dot{\Sigma}^i_{\mathcal{S}\mathcal{S}} + \alpha_{\mathcal{U}_i\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}\right.\\
&\quad \left. - \dot{\Sigma}^i_{\mathcal{U}_i\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_j} - \alpha_{\mathcal{U}_i\mathcal{S}}\left(\dot{\Sigma}^j_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} + \Sigma_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} - \dot{\Sigma}^j_{\mathcal{S}\mathcal{U}_j}\right)\\
&= \alpha_{\mathcal{U}_i\mathcal{S}}\left(\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} + \Sigma_{\mathcal{S}\mathcal{S}}\right)\alpha_{\mathcal{S}\mathcal{U}_j} - \Phi^i_{\mathcal{U}_i\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} - \alpha_{\mathcal{U}_i\mathcal{S}}\Phi^j_{\mathcal{S}\mathcal{U}_j}\\
&= \alpha_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} + \Sigma_{\mathcal{U}_i\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{U}_j} - \Phi^i_{\mathcal{U}_i\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} - \alpha_{\mathcal{U}_i\mathcal{S}}\Phi^j_{\mathcal{S}\mathcal{U}_j}
\end{aligned} \tag{B.4}$$

For each machine $m = 1, \ldots, M$, we can now prove that

$$\begin{aligned}
&\mu^{\text{PIC}}_{\mathcal{U}_m|\mathcal{D}} \\
&= \mu_{\mathcal{U}_m} + \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}} \left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1} (y_{\mathcal{D}} - \mu_{\mathcal{D}}) \\
&= \mu_{\mathcal{U}_m} + \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) \\
&\quad - \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\Sigma_{\mathcal{D}\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{D}}\Lambda^{-1}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) \\
&= \mu_{\mathcal{U}_m} + \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) - \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\Sigma_{\mathcal{D}\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\ddot{y}_{\mathcal{S}} \\
&= \mu_{\mathcal{U}_m} + \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}(\ddot{y}_{\mathcal{S}} - \dot{y}^m_{\mathcal{S}}) + \dot{y}^m_{\mathcal{U}_m} \\
&\quad - \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\Sigma_{\mathcal{D}\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\ddot{y}_{\mathcal{S}} \\
&= \mu_{\mathcal{U}_m} + \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}(\ddot{y}_{\mathcal{S}} - \dot{y}^m_{\mathcal{S}}) + \dot{y}^m_{\mathcal{U}_m} \\
&\quad - \left(\Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} - \Phi^m_{\mathcal{U}_m\mathcal{S}}\right)\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\ddot{y}_{\mathcal{S}} \\
&= \mu_{\mathcal{U}_m} + \left(\Phi^m_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\ddot{y}_{\mathcal{S}} - \Sigma_{\mathcal{U}_m\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\dot{y}^m_{\mathcal{S}}\right) + \dot{y}^m_{\mathcal{U}_m} \\
&= \widehat{\mu}^+_{\mathcal{U}_m} \,.
\end{aligned}$$

The first equality is by definition (4.13). The second equality is due to (A.1). The third equality is due to the definition of global summary (4.3). The fourth and fifth equalities are due to (B.1) and (B.2), respectively. Also,

$$\begin{aligned}
&\Sigma^{\text{PIC}}_{\mathcal{U}_m\mathcal{U}_m|\mathcal{D}} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\left(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda\right)^{-1}\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}_m} + \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\Sigma_{\mathcal{D}\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{D}}\Lambda^{-1}\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \widetilde{\Gamma}_{\mathcal{U}_m\mathcal{D}}\Lambda^{-1}\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}_m} \\
&\quad + \left(\alpha_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}} - \Phi^m_{\mathcal{U}_m\mathcal{S}}\right)\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\left(\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_m} - \Phi^m_{\mathcal{S}\mathcal{U}_m}\right) \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \alpha_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_m} + \alpha_{\mathcal{U}_m\mathcal{S}}\Phi^m_{\mathcal{S}\mathcal{U}_m} + \alpha_{\mathcal{U}_m\mathcal{S}}\dot{\Sigma}^m_{\mathcal{S}\mathcal{U}_m} \\
&\quad - \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{U}_m} + \alpha_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_m} - \alpha_{\mathcal{U}_m\mathcal{S}}\Phi^m_{\mathcal{S}\mathcal{U}_m} - \Phi^m_{\mathcal{U}_m\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_m} \\
&\quad + \Phi^m_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^m_{\mathcal{S}\mathcal{U}_m} \\
&= \Sigma_{\mathcal{U}_m\mathcal{U}_m} - \left(\Phi^m_{\mathcal{U}_m\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_m} - \alpha_{\mathcal{U}_m\mathcal{S}}\dot{\Sigma}^m_{\mathcal{S}\mathcal{U}_m} - \Phi^m_{\mathcal{U}_m\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^m_{\mathcal{S}\mathcal{U}_m}\right) \\
&\quad - \dot{\Sigma}^m_{\mathcal{U}_m\mathcal{U}_m} \\
&= \widehat{\Sigma}^+_{\mathcal{U}_m\mathcal{U}_m} \,.
\end{aligned}$$

The first equality is by definition (4.14). The second equality is due to (A.1). The third equality is due to (B.2). The fourth equality is due to (B.3). The last two equalities are by definition (4.8).

Since our primary interest in the work of this paper is to provide the predictive means and their corresponding predictive variances, the above equivalence results suffice. However, if the entire predictive covariance matrix $\widehat{\Sigma}^{+}_{\mathcal{U}\mathcal{U}}$ for any set $\mathcal{U}$ of inputs is desired (say, to calculate the joint entropy), then it is necessary to compute $\widehat{\Sigma}^{+}_{\mathcal{U}_i\mathcal{U}_j}$ for $i, j = 1, \ldots, M$ such that $i \neq j$. Define

$$\widehat{\Sigma}^{+}_{\mathcal{U}_i\mathcal{U}_j} \triangleq \Sigma_{\mathcal{U}_i\mathcal{U}_j|\mathcal{S}} + \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j} \tag{B.5}$$

for $i, j = 1, \ldots, M$ such that $i \neq j$. So, for a machine i to compute $\widehat{\Sigma}^{+}_{\mathcal{U}_i\mathcal{U}_j}$, it has to receive $\mathcal{U}_j$ and $\Phi^{j}_{\mathcal{S}\mathcal{U}_j}$ from machine j.

We can now prove the equivalence result $\widehat{\Sigma}^{+}_{\mathcal{U}_i\mathcal{U}_j} = \Sigma^{\text{PIC}}_{\mathcal{U}_i\mathcal{U}_j|\mathcal{D}}$ for any two machines $i, j = 1, \ldots, M$ such that $i \neq j$:

$$\begin{aligned}
&\Sigma^{\text{PIC}}_{\mathcal{U}_i\mathcal{U}_j|\mathcal{D}} \\
&= \Sigma_{\mathcal{U}_i\mathcal{U}_j} - \widetilde{\Gamma}_{\mathcal{U}_i\mathcal{D}}\Lambda^{-1}\widetilde{\Gamma}_{\mathcal{D}\mathcal{U}_j} + \alpha_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} \\
&\quad -\alpha_{\mathcal{U}_i\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j} - \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} + \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j} \\
&= \Sigma_{\mathcal{U}_i\mathcal{U}_j} - \big(\alpha_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} + \Sigma_{\mathcal{U}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{U}_j} - \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} \\
&\quad - \alpha_{\mathcal{U}_i\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j}\big) + \alpha_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}_{\mathcal{S}\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} - \alpha_{\mathcal{U}_i\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j} - \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\alpha_{\mathcal{S}\mathcal{U}_j} \\
&\quad + \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j} \\
&= \Sigma_{\mathcal{U}_i\mathcal{U}_j} - \Sigma_{\mathcal{U}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{U}_j} + \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j} \\
&= \Sigma_{\mathcal{U}_i\mathcal{U}_j|\mathcal{S}} + \Phi^{i}_{\mathcal{U}_i\mathcal{S}}\ddot{\Sigma}^{-1}_{\mathcal{S}\mathcal{S}}\Phi^{j}_{\mathcal{S}\mathcal{U}_j} \\
&= \widehat{\Sigma}^{+}_{\mathcal{U}_i\mathcal{U}_j}\ .
\end{aligned}$$

The first equality is obtained using a similar trick as the previous derivation. The second equality is due to (B.4). The second last equality is by the definition of posterior covariance in GP model (3.2). The last equality is by definition (B.5).

Appendix C

Proof of Theorem 3

$$
\begin{aligned}
&\mu^{\text{ICF}}_{\mathcal{U}|\mathcal{D}} \\
&= \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{D}}(F^{\top}F + \sigma_n^2 I)^{-1}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) \\
&= \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{D}}\Big(\sigma_n^{-2}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) \\
&\quad -\sigma_n^{-4}F^{\top}(I + \sigma_n^{-2}FF^{\top})^{-1}F(y_{\mathcal{D}} - \mu_{\mathcal{D}})\Big) \\
&= \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{D}}\left(\sigma_n^{-2}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) - \sigma_n^{-4}F^{\top}\Phi^{-1}\sum_{m=1}^{M}\dot{y}_m\right) \\
&= \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{D}}\left(\sigma_n^{-2}(y_{\mathcal{D}} - \mu_{\mathcal{D}}) - \sigma_n^{-4}F^{\top}\ddot{y}\right) \\
&= \mu_{\mathcal{U}} + \sum_{m=1}^{M}\Sigma_{\mathcal{U}\mathcal{D}_m}\left(\sigma_n^{-2}(y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}) - \sigma_n^{-4}F_m^{\top}\ddot{y}\right) \\
&= \mu_{\mathcal{U}} + \sum_{m=1}^{M}\sigma_n^{-2}\Sigma_{\mathcal{U}\mathcal{D}_m}(y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}) - \sigma_n^{-4}\dot{\Sigma}_m^{\top}\ddot{y} \\
&= \mu_{\mathcal{U}} + \sum_{m=1}^{M}\widetilde{\mu}_{\mathcal{U}}^{m} \\
&= \widetilde{\mu}_{\mathcal{U}}\ .
\end{aligned}
$$

The first equality is by definition (4.27). The second equality is due to matrix inversion lemma. The third equality follows from $I + \sigma_n^{-2}FF^{\top} = I + \sigma_n^{-2}\sum_{m=1}^{M}F_mF_m^{\top} = I + \sigma_n^{-2}\sum_{m=1}^{M}\Phi_m = \Phi$ and $F(y_{\mathcal{D}} - \mu_{\mathcal{D}}) = \sum_{m=1}^{M}F_m(y_{\mathcal{D}_m} - \mu_{\mathcal{D}_m}) = \sum_{m=1}^{M}\dot{y}_m$. The fourth equality is due to (4.21). The second last equality follows from (4.23). The last equality is by definition (4.25).

Similarly,

$$
\begin{aligned}
&\Sigma^{\text{ICF}}_{\mathcal{U}\mathcal{U}|\mathcal{D}} \\
&= \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{D}}(F^\top F + \sigma_n^2 I)^{-1}\Sigma_{\mathcal{D}\mathcal{U}} \\
&= \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{D}}\Big(\sigma_n^{-2}\Sigma_{\mathcal{D}\mathcal{U}} \\
&\quad -\sigma_n^{-4}F^\top(I + \sigma_n^{-2}FF^\top)^{-1}F\Sigma_{\mathcal{D}\mathcal{U}}\Big) \\
&= \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{D}}\left(\sigma_n^{-2}\Sigma_{\mathcal{D}\mathcal{U}} - \sigma_n^{-4}F^\top\Phi^{-1}\sum_{m=1}^{M}\dot{\Sigma}_m\right) \\
&= \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{D}}\left(\sigma_n^{-2}\Sigma_{\mathcal{D}\mathcal{U}} - \sigma_n^{-4}F^\top\ddot{\Sigma}\right) \\
&= \Sigma_{\mathcal{U}\mathcal{U}} - \sum_{m=1}^{M}\Sigma_{\mathcal{U}\mathcal{D}_m}\left(\sigma_n^{-2}\Sigma_{\mathcal{D}_m\mathcal{U}} - \sigma_n^{-4}F_m^\top\ddot{\Sigma}\right) \\
&= \Sigma_{\mathcal{U}\mathcal{U}} - \sum_{m=1}^{M}\sigma_n^{-2}\Sigma_{\mathcal{U}\mathcal{D}_m}\Sigma_{\mathcal{D}_m\mathcal{U}} - \sigma_n^{-4}\dot{\Sigma}_m^\top\ddot{\Sigma} \\
&= \Sigma_{\mathcal{U}\mathcal{U}} - \sum_{m=1}^{M}\widetilde{\Sigma}^m_{\mathcal{U}\mathcal{U}} \\
&= \widetilde{\Sigma}_{\mathcal{U}\mathcal{U}}\ .
\end{aligned}
$$

The first equality is by definition (4.28). The second equality is due to matrix inversion lemma. The third equality follows from $I + \sigma_n^{-2}FF^\top = \Phi$ and $F\Sigma_{\mathcal{D}\mathcal{U}} = \sum_{m=1}^{M}F_m\Sigma_{\mathcal{D}_m\mathcal{U}} = \sum_{m=1}^{M}\dot{\Sigma}_m$. The fourth equality is due to (4.22). The second last equality follows from (4.24). The last equality is by definition (4.26).

Appendix D

Proof of Theorem 4

Let $\underline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w} \triangleq \widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w} - \overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ and ρ_w be the spectral radius of $\left(\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right)^{-1}\underline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$. We have to first bound ρ_w from above.

For any joint walk w, $\left(\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right)^{-1}\underline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ comprises diagonal blocks of size $\left|\mathcal{U}_{w\mathcal{V}_n}\right| \times \left|\mathcal{U}_{w\mathcal{V}_n}\right|$ with components of value 0 for $n = 1,\ldots,\mathcal{K}$ and off-diagonal blocks of the form $\left(\widehat{\Sigma}_{\mathcal{U}_{w\mathcal{V}_n}\mathcal{U}_{w\mathcal{V}_n}}\right)^{-1}\widehat{\Sigma}_{\mathcal{U}_{w\mathcal{V}_n}\mathcal{U}_{w\mathcal{V}_{n'}}}$ for $n, n' = 1,\ldots,\mathcal{K}$ and $n \neq n'$. We know that any pair of sensors $k \in \mathcal{V}_n$ and $k' \in \mathcal{V}_{n'}$ reside in different connected components of coordination graph $\mathcal{G}$ and are therefore not adjacent. So, by Definition 18,

$$\max_{i,i'}\left|\left[\widehat{\Sigma}_{\mathcal{U}_{w\mathcal{V}_n}\mathcal{U}_{w\mathcal{V}_{n'}}}\right]_{ii'}\right| \leq \varepsilon \tag{D.1}$$

for $n, n' = 1,\ldots,\mathcal{K}$ and $n \neq n'$. Using (5.20) and (D.1), each component in any off-diagonal block of $\left(\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right)^{-1}\underline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}$ can be bounded as follows:

$$\max_{i,i'}\left|\left[\left(\widehat{\Sigma}_{\mathcal{U}_{w\mathcal{V}_n}\mathcal{U}_{w\mathcal{V}_n}}\right)^{-1}\widehat{\Sigma}_{\mathcal{U}_{w\mathcal{V}_n}\mathcal{U}_{w\mathcal{V}_{n'}}}\right]_{ii'}\right| \leq \left|\mathcal{U}_{w\mathcal{V}_n}\right|\xi\varepsilon \tag{D.2}$$

for $n, n' = 1,\ldots,\mathcal{K}$ and $n \neq n'$. It follows from (D.2) that

$$\max_{i,i'}\left|\left[\left(\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right)^{-1}\underline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right]_{ii'}\right| \leq \max_n\left|\mathcal{U}_{w\mathcal{V}_n}\right|\xi\varepsilon \leq H\kappa\xi\varepsilon\ . \tag{D.3}$$

The last inequality is due to $\max_n\left|\mathcal{U}_{w\mathcal{V}_n}\right| \leq H\max_n|\mathcal{V}_n| \leq H\kappa$. Then,

$$\begin{aligned}\rho_w &\leq \left\|\left(\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right)^{-1}\underline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right\|_2\\ &\leq |\mathcal{U}_w|\max_{i,i'}\left|\left[\left(\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right)^{-1}\underline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right]_{ii'}\right|\\ &\leq KH^2\kappa\xi\varepsilon\ .\end{aligned} \tag{D.4}$$

The first two inequalities follow from standard properties of matrix norm [Golub and Van Loan, 1996; Stewart and Sun, 1990]. The last inequality is due to (D.3).

The rest of this proof utilizes the following result of [Ipsen and Lee, 2003] that is revised to reflect our notations:

Theorem 5. *If* $|\mathcal{U}_w|\rho_w^2 < 1$, *then* $\log\left|\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right| \leq \log\left|\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right| \leq \log\left|\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right| - \log(1 - |\mathcal{U}_w|\rho_w^2)$ *for any joint walk* w.

Using Theorem 5 followed by (D.4),

$$\begin{aligned}\log\left|\overline{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right| - \log\left|\widehat{\Sigma}_{\mathcal{U}_w\mathcal{U}_w}\right| &\leq \log\frac{1}{1-|\mathcal{U}_w|\rho_w^2} \\ &\leq \log\frac{1}{1-(K^{1.5}H^{2.5}\kappa\xi\varepsilon)^2}\end{aligned} \tag{D.5}$$

for any joint walk w.

$$\begin{aligned}&\widehat{\mathbb{H}}[Z_{\mathcal{U}_{w^*}}] - \widehat{\mathbb{H}}\left[Z_{\mathcal{U}_{\widehat{w}}}\right] \\ &= \frac{1}{2}\left((|\mathcal{U}_{w^*}| - |\mathcal{U}_{\widehat{w}}|)\log(2\pi e) + \log\left|\widehat{\Sigma}_{\mathcal{U}_{w^*}\mathcal{U}_{w^*}}\right| - \log\left|\widehat{\Sigma}_{\mathcal{U}_{\widehat{w}}\mathcal{U}_{\widehat{w}}}\right|\right) \\ &\leq \frac{1}{2}\left((|\mathcal{U}_{w^*}| - |\mathcal{U}_{\widehat{w}}|)\log(2\pi e) + \log\left|\overline{\Sigma}_{\mathcal{U}_{w^*}\mathcal{U}_{w^*}}\right| - \log\left|\widehat{\Sigma}_{\mathcal{U}_{\widehat{w}}\mathcal{U}_{\widehat{w}}}\right|\right) \\ &\leq \frac{1}{2}\left((|\mathcal{U}_{\widehat{w}}| - |\mathcal{U}_{\widehat{w}}|)\log(2\pi e) + \log\left|\overline{\Sigma}_{\mathcal{U}_{\widehat{w}}\mathcal{U}_{\widehat{w}}}\right| - \log\left|\widehat{\Sigma}_{\mathcal{U}_{\widehat{w}}\mathcal{U}_{\widehat{w}}}\right|\right) \\ &\leq \frac{1}{2}\log\frac{1}{1-(K^{1.5}H^{2.5}\kappa\xi\varepsilon)^2}\ .\end{aligned}$$

The first equality is due to (5.11). The first, second, and last inequalities follow from Theorem 5, (5.18), and (D.5), respectively.