EXPLOITING DECENTRALIZED MULTI-AGENT COORDINATION FOR LARGE-SCALE MACHINE LEARNING PROBLEMS

OUYANG RUOFEI

(B.Sc. ECNU)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2016

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.

OUYANG RUOFEI

2016

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Low Kian Hsiang, for providing timely advice, valuable guidance, and considerable encouragement during my Ph.D. studies on conducting sound research and being a good person.

I would also like to thank my thesis committee members, Prof. David Hsu and Prof. Lee Wee Sun, for devoting time and effort to read this thesis and providing constructive comments for my GRP and TP.

Many thanks to the members in MapleCG research group. Especially Trong Nghia Hoang for helping me in problem formulation, Jaemin Son for helping me in implementing ANOVA-DCOP, Keng Kiat Lim for helping me in collecting the data for robot experiment, Xu Nuo and Zhang Yehong for helping me in proofreading the thesis.

Many thanks to my buddies in GoInvest, Eugene Chua, Tan See Youu, Lee Chun Hoe and Tessa Voon for conducting the experiment on the financial data.

Last but not least, I would like to express my indebtedness to my parents and my girlfriend Xiaojun for their understanding and support along the way.

Table of contents

Summary

Nowadays, the scale of machine learning problems becomes much larger than before. It raises a huge demand in distributed perception and distributed computation. A multi-agent system provides exceptional scalability for problems like active sensing and data fusion. However, many rich characteristics of large-scale machine learning problems have not been addressed yet such as large input domain, nonstationarity, and high dimensionality. This thesis identifies the challenges related to these characteristics from multi-agent perspective. By exploiting the correlation structure of data in large-scale problems, we propose multi-agent coordination schemes that can improve the scalability of the machine learning models while preserving the computation accuracy. To elaborate, the machine learning problems we are solving with multi-agent coordination techniques are:

(a) Gaussian process regression. To perform distributed regression on a large-scale environmental phenomenon, data compression is often required due to the communication costs. Currently, decentralized data fusion methods encapsulate the data into local summaries based on a fixed support set. However in a large-scale field, this fixed support set, acting as a centralized component in the decentralized system, cannot approximate the correlation structure of the entire phenomenon well. It leads to evident losses in data summarization. Consequently, the regression performance will be significantly reduced.

In order to approximate the correlation structure accurately, we propose an agent-centric support set to allow every agent in the data fusion system to choose a possibly different support set and dynamically switch to another one during execution for encapsulating its own data into a local summary which, perhaps surprisingly, can still be assimilated with the other agents' local summaries into a globally consistent summary. Together with an information sharing mechanism we designed, the new decentralized data fusion methods with agent-centric support set can be applied to regression problems on a much larger environmental phenomenon with high performance.

(b) Active learning. In the context of environmental sensing, active learning/active sensing is a process of taking observations to minimize the uncertainty in an environmental field. The uncertainty is quantified based on the correlation structure of the phenomenon

which is traditionally assumed to be stationary for computational sake. In a large-scale environmental field, this stationary assumption is often violated. Therefore, existing active sensing algorithms perform sub-optimally for a nonstationary environmental phenomenon.

To the best of our knowledge, our decentralized multi-robot active sensing (DEC-MAS) algorithm is the first work to address nonstationarity issue in the context of active sensing. The uncertainty in the phenomenon is quantified based on the nonstationary correlation structure estimated by Dirichlet process mixture of Gaussian processes (DPM-GPs). Further, our DEC-MAS algorithm can efficiently coordinate the exploration of multiple robots to automatically trade-off between learning the unknown, nonstationary correlation structure and minimizing the uncertainty of the environmental phenomenon. It enables multi-agent active sensing techniques to be applied to a large-scale nonstationary environmental phenomenon.

(c) Bayesian optimization. Optimizing an unknown objective function is challenging for traditional optimization methods. Alternatively, in this situation, people use Bayesian optimization which is a modern optimization technique that can optimize a function by only utilizing the observation information (input and output values) collected through simulations. When the input dimension of the function is low, a few simulated observations can generate good result already. However, for high dimensional function, a huge number of observations are required which is impractical when the simulation consumes lots of time and resources.

Fortunately, many high dimensional problems have sparse correlation structure. Our ANOVA-DCOP work can decompose the correlation structure in the original high-dimensional problem into many correlation structures of subsets of dimensions based on ANOVA kernel function. It significantly reduces the size of input space into a collection of lower-dimensional subspaces. Additionally, we reformulate the Bayesian optimization problem as a decentralized constrained optimization problem (DCOP) that can be efficiently solved by multi-agent coordination techniques so that it can scale up to problems with hundreds of dimensions.

List of figures

List of tables

List of symbols

Abbreviations

MAS	multi-agent system
DEC-MAS	decentralized Multi-agent active sensing
GP	Gaussian process
GPM-GPs	Dirichlet process mixture of Gaussian processes
GP-DDF	Gaussian process decentralized data fusion
BO	Bayesian optimization
DCOP	decentralized constrained optimization problem
PITC	partially independent training conditional approximation of GP
PIC	partially independent conditional approximation of GP
RMSE	root mean square error

Symbols

$\mathcal{X}$	input domain
Ω	discretized input domain
D	a set of observed inputs

X	a set of unobserved inputs
Y_D	observed measurements on D
z_D	component label on D
S	support set
θ	hyperparameters of a kernel function
ε	noise of observation
G	coordination graph
V	vertex of graph G
E	edges of graph G
$\mathcal{G}$	DCOP coordination graph
$\mathcal{V}$	variable vertex of graph $\mathcal{G}$
$\mathcal{F}$	function vertex of graph $\mathcal{G}$
$\mathcal{E}$	edges of graph $\mathcal{G}$
Σ	covariance matrix
μ	mean value of measurement
σ_s	signal variance
σ_n	noise variance
ℓ_i	length-scale on i-th dimension
$(\nu_{\mathcal{S}\|\mathcal{D}}, \Psi_{\mathcal{S}\mathcal{S}\|\mathcal{D}})$	local summary
$(\nu_{\mathcal{S}\|\mathcal{D}}, \Psi_{\mathcal{S}\mathcal{S}\|\mathcal{D}})$	global summary

$(\omega_{\mathcal{S}\|\mathcal{D}}, \Phi_{\mathcal{S}\mathcal{S}\|\mathcal{D}})$	prior summary

Functions

$\kappa(\cdot,\cdot)$	kernek function
$\delta_{xx'}$	Kronecker delta
log	logarithm to base e
$\mathbb{H}$	entropy of a probabilistic distribution
$\mathbb{E}$	expectation of a random variable
max	maximum value of a function
arg max	argument of the maximum of a function
inf	infimum of a set

Chapter 1

Introduction

1.1 Motivation

Nowadays, the scale of machine learning problems that we are interested in becomes much larger than before. For example, monitoring the traffic condition in the road network of a big city requires collecting speed data from quantities of road segments and reconstructing the traffic speed distribution of the entire urban road network by a regression model (Chen et al., 2015, Kamarianakis and Prastacos, 2003, Min and Wynter, 2011, Wang and Papageorgiou, 2005, Work et al., 2010). Machine learning problems of such a scale motivates the need to design and develop distributed algorithms for solving them efficiently and scalably.

To develop distributed algorithms, people seek techniques from multi-agent community. Current research works on multi-agent system (Cao et al., 2013, Chen et al., 2012, 2013b, Leonard et al., 2007, Low et al., 2012, Rogers et al., 2011, Singh et al., 2009) have developed various decentralized multi-agent coordination schemes to scale up machine learning problems such as environmental sensing and data fusion. A typical multi-agent system decomposes a large-scale problem into a list of small-scale sub-problems and then assigns an agent to solve each small-scale problem separately. After the sub-problems are solved, the original problem is solved by combining the sub-problems' solutions through decentralized multi-agent coordination.

Existing multi-agent systems can resolve the scalability issues with respect to the size of the data (Chen et al., 2011, Guestrin et al., 2004, Low et al., 2015, Sun et al., 2015). However, besides the size of the data, large-scale machine learning problems have many complex characteristics that have not been addressed yet:

- Large input domain in large-scale regression
- Nonstationarity in large-scale active learning
- High dimensionality in optimizing a complex unknown function

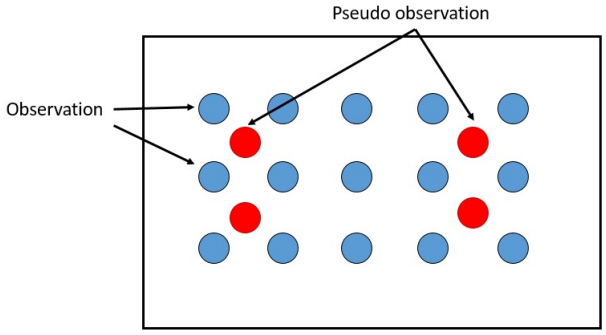
In the next few subsections, we will discuss three challenges in large-scale machine learning problems specific to those characteristics that are critical to the performance of many applications.

1.1.1 Regression at Scale

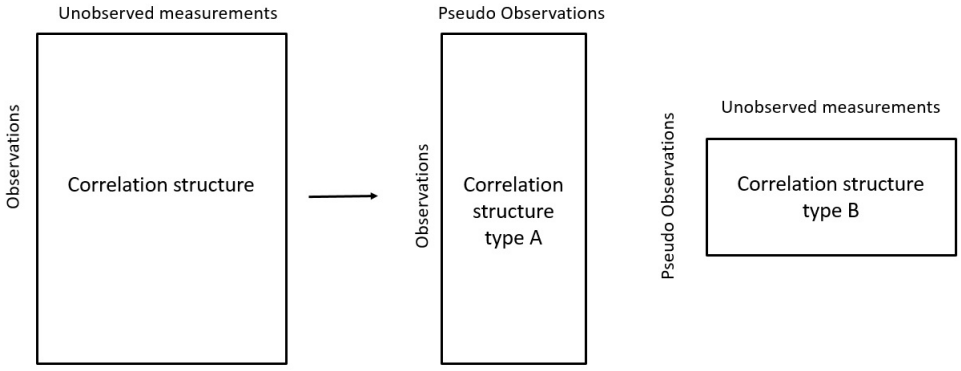
Regression is one of the fundamental machine learning problems. Usually, a centralized model is ill-suited for regression over massive volume of data because it suffers from poor scalability in the data size and a single point of failure. Therefore, some decentralized data fusion methods have been developed to improve the scalability and robustness.

Decentralized Data Fusion is a process of integrating data from multiple sources to form a consistent representation of a target environmental phenomenon (Chen et al., 2012, 2013b, Cortes, 2009, Guestrin et al., 2004). In decentralized data fusion, each agent takes observations from its allocated area and performs the local regression within this area. To achieve high accuracy in data fusion, the agent needs to retrieve information from its neighboring agents.

Sharing a large amount of data/observations consumes lots of time and resources. In practice, instead of directly sharing the original observations, it is better for each agent to "compress" the observations into small-sized local summaries by certain approximation method, and then share the small-sized local summaries with other neighboring agents.



(a) A support set is a set of locations of pseudo observations.



(b) Approximation of correlation structure.

Fig. 1.1 Illustration of support set.

To reduce the information loss in the summarization process, recent works (Chen et al., 2012, 2013b) on Gaussian process decentralized data fusion (GP-DDF) methods approximate the correlation structure of the environmental phenomenon. A correlation structure is a set of the pairwise correlations between all the observed and unobserved measurements. The approximation process in GP-DDF relies on a notion of a fixed support set which is a small-sized set of locations in the environmental field(see Fig. 1.1a). This fixed support set decomposes the original correlation structure in the measurements by two types of low-rank correlation structures (see Fig. 1.1b): A) correlation structure between the observations and the pseudo observations at locations in the support set; B) correlation structure between the pseudo observations at locations in the support set and the unobserved measurements. With the fixed support set, GP-DDF methods compute the values of pseudo observations from the actual observations based on A-type correlation structure and predict the unobserved

measurements with these values based on the B-type correlation structure. The accuracy of the approximation depends on the spatial correlation between the observations and the pseudo observations. If the observations and the pseudo observations are far from each other, their correlations are weak. Consequently, the pseudo observations can not summarize the observations accurately.

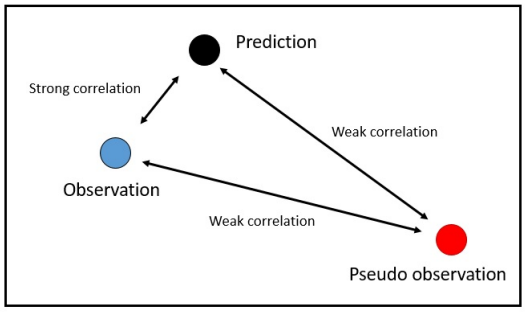


Fig. 1.2 Illustration of challenge of large input domain.

Now, we can introduce the first challenge in large-scale regression problem on spatial data: due to the large input domain, a fixed support set cannot accurately approximate the correlation structure of large-scale phenomenon. In decentralized data fusion applications, the size of the target environmental field we are interested in can be extremely large. For example, we want to reconstruct the traffic speed distribution for each road segment over entire Singapore. In this scenario, the fixed support set is a critical issue. Let us illustrate with Fig. 1.2. In the figure, the observation has a strong correlation with the unobserved measurement so that it can predict it well. However, if we deploy a fixed support set far from the observation, the pseudo observation at the location in the support set is weakly correlated with the observation. It is not able to deliver the information from observation to the unobserved measurement for prediction. The support set is restricted by its size to limit the computational overheads and can only sparsely cover the large-scale phenomenon. In environmental field with large input domain, a huge number of observations will be far from the fixed support set. As a result, the fixed support set cannot accurately approximate the correlation structure between all the observed and unobserved measurements. Consequently, huge information loss is expected.

Moreover, besides the major challenge, there are two more limitations from practical consideration:

1. When the domain of the phenomenon of interest expands, the size of the support set must also be increased proportionally to cover and predict the phenomenon well at the expense of greater time, space, and communication overheads, which grows prohibitively expensive.

2. If the current support set needs to be replaced by a new support set of different size and input locations (e.g., due to change in domain size, time, space, and communication requirements, using an improved active learning criterion to select a support set that better covers and predicts the phenomenon), then all the previously gathered observations (if not discarded after summarization using the old support set) have to be re-summarized into local summaries based on the new support set, which is not scalable.

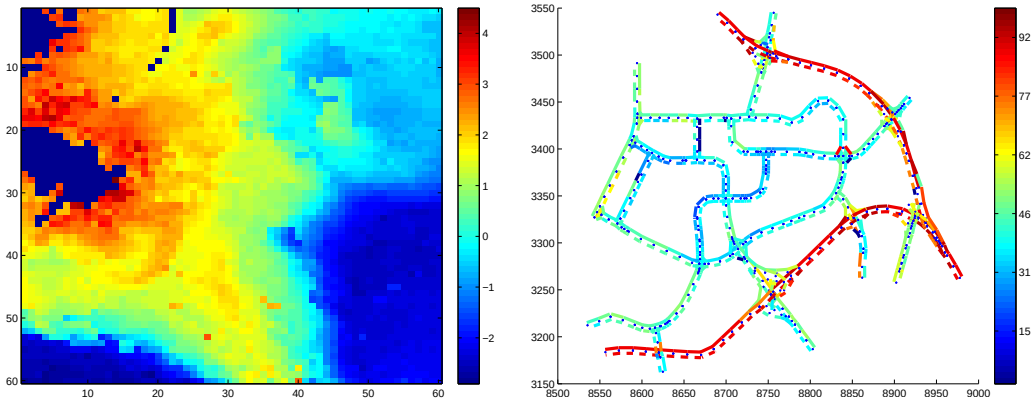
The fixed support set is a centralized component in decentralized data fusion methods which in nature contradicts with the original intention of decentralization. To address the challenge related to large input domain, it is essential to remove this centralized component so that the data fusion model can be truly decentralized.

1.1.2 Active Learning at Scale

In order to generate an accurate regression result with a limited number of observations, taking the most informative observations/samples is a key step. **Active learning** is a fundamental machine learning problem to choose the most informative observations by minimizing the uncertainty quantified in the original regression problem. In multi-agent community, multi-agent active sensing (MAS) is an active learning method for exploring large-scale environmental phenomenon. Its objective is to coordinate a team of mobile agents to actively gather the most informative observations for predicting a spatially varying phenomenon of interest while being subject to resource cost constraints (e.g., number of deployed agents, energy consumption, mission time).

To quantify the uncertainty, a number of MAS algorithms characterize the correlation structure in the environmental phenomenon so that the predictive uncertainty can be formally computed (e.g., mean square error, entropy, mutual information). Subsequently, multiple agents are directed to explore the highly uncertain areas of the environmental phenomenon. In order not to incur a high computational expense, these algorithms have assumed the spatial correlation structure to be known (or estimated crudely using sparse prior data) and stationary.

Stationarity is a statistical term to describe a special type of environmental phenomenon in which the degree of smoothness of the spatial variation of the measurements is the same across the entire phenomenon. Many small-scale environmental phenomena are stationary which indeed result in a stationary correlation structure in the measurements. However, when the scale of the phenomena becomes larger, this stationary assumption will be violated, and the underlying correlation structure in the measurements is actually nonstationary.



(a) (b)

Fig. 1.3 Demonstration of real-world nonstationary environmental phenomena: (a) Plankton density (chl-a) phenomenon (measured in mg/m^3) in log-scale in Gulf of Mexico, and (b) traffic (road speeds) phenomenon (measured in km/h) over an urban road network.

For example (see Fig. 1.3), in some ocean phenomena (e.g., temperature, salinity, sea surface height), their measurements far offshore are more smoothly varying (i.e., more spatially correlated) in the cross-shore direction than nearshore (Li et al., 2008). Urban traffic network is a combination of highways and small roads. It also displays nonstationary phenomena (e.g., traffic speeds, taxi demands) which pose important considerations to traffic routing and signal control.

Here, we introduce the second challenge of large-scale machine learning related to active learning: most large-scale phenomena are nonstationary in nature which leads to wrongly quantified uncertainty based on current MAS algorithms. Although existing MAS algorithms can still be used for sampling a non-stationary phenomenon by assuming, albeit incorrectly, its spatial correlation structure to be known and stationary in order to preserve time efficiency. They can gather the most informative observations under an assumed stationary correlation structure, but they will perform sub-optimally with respect to the true nonstationary correlation structure.

A more desirable MAS algorithm should instead be designed to consider the informativeness of its selected observations based on the true nonstationary correlation structure. Furthermore, unlike the stationary structure which can be pre-determined with a small number of observations in the early stage of active sensing, the nonstationary correlation structure has to assume to be unknown before the target phenomenon is well explored due to its complexity. It needs to be updated using the newly taken observations in the whole active sensing process, which raises a fundamental issue faced by active learning. How can a MAS algorithm trade off between these two possibly conflicting criteria? Should the next observation to be taken to a) estimate the unknown nonstationary correlation structure or b) minimize the uncertainty of the phenomenon based on the estimated nonstationary correlation structure?

1.1.3 Optimization at Scale

Most of the machine learning tasks or the training procedure of these tasks can be formulated as optimization problems. For instance, active learning is to minimize the uncertainty in the original task; fitting a regression model is to minimize the likelihood function using the training data. Usually, in order to solve an optimization problem, we need to know the expression of the objective function (if possible, the expression of the derivative or the second derivative) to search the maximum/minimum value of the function.

In practice, some problem in nature is so complex that it may not have an analytical expression that can be solved by traditional optimization methods. Alternatively, it is more convenient to analyze the problem through simulation. For example, *El Niño* is a complex

meteorological phenomenon (Larkin and Harrison, 2005) so that researchers analyse its behavior through simulation. The financial market is highly unpredictable so that many trading strategies (Wang et al., 2012) can only be evaluated through backtesting (simulate the trading strategy using historical data). In these scenarios, the objective function is a black box. We can only optimize the function based on the knowledge of the observed input-output value pairs.

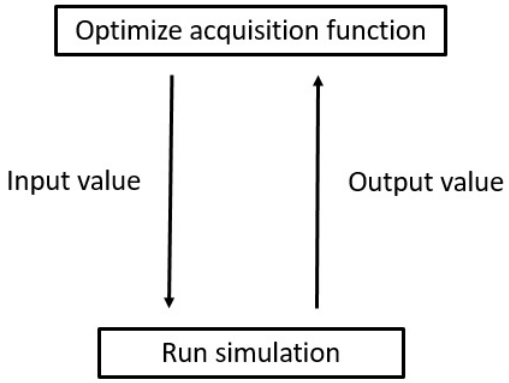


Fig. 1.4 Illustration of the procedure of Bayesian optimization.

Bayesian optimization (BO) is a modern optimization technique that can find the global optimum of an unknown functions with limited function evaluations. There are two ingredients in BO: The first ingredient is a prior distribution that captures the belief over the unknown objective function based on the observations (the input-output pairs); The second ingredient is a risk function that describes the deviation of current optimum from the true global optimum. Existing BO works integrated these two ingredients by an acquisition function. This acquisition function has an analytical expression that can be evaluated. The most interesting thing is that BO transforms the problem of optimizing an unknown function into two relatively easier problems: a) estimating the acquisition function and b) optimizing the acquisition function. The acquisition function is estimated by taking observations from simulations. By performing optimization on the acquisition function, we can know which input value should be set for the next round of simulation. After iteratively conducting the

simulation (see Fig. 1.4) with the guidance of BO, eventually, the optimum value of the original function can be found.

However, learning and optimizing an acquisition function may not be an easy job in the real-world situation, especially when the function has high input dimension, which is the third challenge in large-scale machine learning. In many optimization problems conducted through simulation, there are a large amount of parameters (each parameter is a dimension of the input) that need to be tuned. In BO framework, estimating and optimizing the acquisition function require learning the correlation structure between all the observations. The correlation between two observations depends on all the input dimensions. Thus, to capture the true latent correlation structure, the observations need to form a sufficient coverage of the input space in every dimension. Otherwise, the acquisition function may be not able to correctly describe the black-box function and the risk function. This is not a serious problem when the objective function's input dimension is low. However, when the function contains a huge number of parameters, a large volume of observations are required to cover the high dimensional input space. Unfortunately, in most simulations, this is time- and resource-consuming. With a limited budget, we can only run a limited number of simulations so that the observations may not be sufficient to form a good coverage of the input space. It will lead to a poor estimation of the correlation structure in the observations. Consequently, the optimization result will be far from the true optimum in the end.

1.2 Objectives

In the above subsections, we have discussed three characteristics (large input domain (section 2.1), nonstationarity (section 1.1.2) and high dimensionality (section 1.1.3)) in the large-scale machine learning problems and their related challenges. An interesting thing is that the three challenges share a common critical component: the underlying correlation structure is inappropriately estimated or approximated in the context of large-scale machine learning problem by ignoring those characteristics.

- A fixed support set is not suitable for approximating the correlation structure of the measurements in large input domain.
- Active sensing algorithm based on the stationary assumption cannot learn the actually nonstationary correlation structure of the large-scale environmental phenomenon.
- The observations within limited budget are insufficient for learning the correlation structure in high dimensional input space.

In order to address the three challenges, we ask the following research question:

In the context of large-scale machine learning, how can the correlation structure of the data be exploited for constructing multi-agent coordination schemes that can improve the scalability of the machine learning models while preserving the computation accuracy?

If the true underlying correlation structure can be captured correctly, multi-agent coordination will provide great scalability to the solutions of the machine learning problems. In a large environmental field, one support set cannot approximate the correlation structure of observations in the entire domain but it can approximate the correlation structure within a certain range. So it may be possible to use multiple local support sets to approximate the local correlation structures with high accuracy and then use multi-agent coordination to share the knowledge of each approximated local correlation structure.

In a nonstationary environmental field, instead of minimizing the uncertainty in the field based on a stationary correlation structure, the active sensing algorithm should learn the nonstationary structure and minimize the uncertainty based on the estimated correlation structure.

Even though a huge number of parameters exist in high-dimensional unknown function optimization problem, not all of them are correlated. We can exploit the sparsity of the correlation structure among inputs so that a high dimensional input space can be decomposed into small subspaces which can be densely covered by a few observations.

1.3 Contributions

By constructing specific multi-agent coordination scheme according to the true underlying correlation structure in large-scale machine learning problems, this thesis is trying to answer the research question with the following contributions.

To address the challenge with large input domain:

- We present novel *Gaussian process decentralized data fusion algorithms* with *agent-centric support sets* for distributed cooperative perception of large-scale environmental phenomenon (section 3.2). In contrast with GP-DDF methods using fixed support set, our proposed algorithms allow every sensing agent to choose a possibly different support set and dynamically switch to another one during execution for encapsulating its own data into a local summary that, perhaps surprisingly, can still be assimilated with the other agents' local summaries (i.e., based on their current choices of support sets) into a globally consistent summary to be used for predicting the phenomenon.

- We propose a new transfer learning mechanism (section 3.2) for a team of mobile sensing agents capable of sharing and transferring information encapsulated in a summary based on a support set to that utilizing a different support set with some loss that can be theoretically bounded and analyzed. To alleviate the issue of information loss accumulating over multiple instances of transfer learning, we propose an information sharing mechanism to be incorporated into our GP-DDF algorithms.

- Our proposed algorithms can overcome the following three limitations of GP-DDF methods (Chen et al., 2012, 2013b, 2015):

    1. For any unobserved input location, an agent can choose a small, constant-sized (i.e., independent of domain size of the phenomenon) but sufficiently dense support set surrounding it to predict its measurement accurately while preserving time, space, and communication efficiencies;

    2. The agents can reduce the information loss due to summarization by choosing or dynamically switching to a support set "close" to their local data;

3. Without needing to retain previously gathered data, an agent can choose or dynamically switch to a new support set whose summary can be constructed using information transferred from the summary based on its current support set, thus preserving scalability to big data.

- Finally, we empirically evaluate the performance of our proposed algorithms using three real-world datasets, one of which is millions in size (section 3.3).

To address the challenge with nonstationarity:

- We present a *decentralized multi-robot active sensing* (DEC-MAS) algorithm that can efficiently coordinate the exploration of multiple robots to automatically trade-off between learning the unknown, nonstationary correlation structure and minimizing the uncertainty of the environmental phenomena. Further, our DEC-MAS algorithm models a nonstationary phenomenon as a *Dirichlet process mixture of Gaussian processes* (DPM-GPs) (Section 4.1): Using the gathered observations, DPM-GPs can learn to automatically partition the phenomenon into separate local areas, each of which comprises measurements that vary according to a stationary spatial correlation structure and can thus be modeled by a locally stationary Gaussian process.

- We demonstrate how DPM-GPs and its structural properties can be exploited to (a) formalize an active sensing criterion that trades off between gathering the most informative observations for estimating the unknown partition (i.e., a key component of the nonstationary correlation structure) vs. that for predicting the phenomenon given the current, possibly imprecise estimate of the partition (Section 4.2), and (b) support effective and efficient decentralized coordination (Section 4.3).

- We also provide a theoretical performance guarantee for DEC-MAS and analyze its time complexity (section 4.3).

- Finally, we empirically demonstrate using two real-world datasets that DEC-MAS outperforms the state-of-the-art MAS algorithms (Section 4.4).

To address the challenge with high dimensionality:

- We present a Bayesian optimization method using Gaussian process prior with ANOVA kernel function (section 5.2) that can decompose the correlation structure in high dimensions into a list of correlation structures of subsets of dimensions. Correspondingly, the high dimensional input space is decomposed into small subspaces so that a few observations can densely cover each subspace to learn and optimize the acquisition function in BO accurately.

- To the best of our knowledge, ANOVA-DCOP is the first work to introduce multi-agent coordination into high dimensional Bayesian optimization problem (section 5.2.3) by exploiting the sparse correlation structure using ANOVA kernel. We formulate the optimization of acquisition function as a decentralized constraint optimization problem (DCOP) which can be solved efficiently using multi-agent coordination. We theoretically bound the regret of the proposed algorithm and analyze its time complexity (section 5.3).

- Finally, we empirically evaluate the performance using two high dimensional functions with known optimum value and one real financial problem. The results show that our method outperforms the existing high dimensional BO methods when the problem has sparse correlation structure among the inputs (section 5.4).

Chapter 2

Related Works

This chapter reviews three large-scale machine learning problems (regression in section 2.1, active learning in section 2.2, optimization in section 2.3) and their characteristics. We identify the challenges due to these characteristics in the existing works and position our work in the literature to highlight our contributions in addressing these challenges.

2.1 Regression and Low Rank Approximation

Regression is one of the fundamental machine learning problems. In the last decades, kernel method (Schölkopf and Smola, 2002) has demonstrated great performance in solving regression problems. Methods such as kernel regression (Jaakkola and Haussler, 1999), support vector regression (Smola and Schölkopf, 2004) and Gaussian process (Rasmussen and Williams, 2006) are widely used in data analytical applications. Within those methods, Gaussian process or so-called Kriging (Stein, 2012) in the geostatistics community is a Bayesian nonparametric model which shows significant robustness in analyzing spatially varying possibly noisy environmental phenomenon. It has integrated with many multi-agent techniques in environmental sensing and data fusion tasks (Krause and Golovin, 2014, Krause et al., 2008a, Meliou et al., 2007b, Osborne et al., 2008, Snelson, 2007).

Although the kernel methods can effectively capture the correlation structure in the problem, the computation usually involves cubic time complexity due to inverting a kernel

matrix. It suffers from serious scalability issue in the size of the data. Many works (Hsieh et al., 2014, Le et al., 2013, Yang et al., 2012) have explored the sparse correlation structure to learn the kernel in a more efficient way and preserve the model accuracy. Specifically for the kernel matrix in Gaussian process model, there are many low-rank approximation methods (Quiñonero-Candela and Rasmussen, 2005, Snelson and Ghahramani, 2007, Snelson and Gharahmani, 2005) that have been developed based on a notion called support set. The support set introduces the conditional independence in the measurements in order to form a sparse representation of original correlation structure.

To improve the scalability of large-scale regression problems, people often seek help from online learning methods and decentralized data fusion algorithms. Many online regression methods (Ngu, Csató and Opper, 2002, Seeger and Williams, 2003, Xu et al., 2014) use a fixed support set for computational sake. Since the correlation structure within the fixed support set is consistent, many time consuming computations only need to be done once. Decentralized data fusion algorithms such as GP-DDF (Chen et al., 2012) and GP-DDF+ (Chen et al., 2013b) also utilize a fixed support set to allow multiple agents agents to perform data fusion in a decentralized manner. Each agent is able to take observations and encapsulate them into a local summary based on the fixed support set. Then, multiple agents can share their local summaries and reconstruct a consistent global summary for prediction.

As can be seen that, GP-DDF and GP-DDF+ are decentralized algorithms but rely on this single fixed support set which is a centralized component in a decentralized system. In a large-scale environmental field, a large volume of observations taken by the agents are far from this fixed support set so that the computed local summaries are inaccurate. Our proposed method utilize an agent-centric support set so that multiple support sets can densely cover the environmental field. No matter where the agents are, the observations they take can always find a close support set to compute an accurate local summary. Later on, the local summaries can be shared between agents using a transfer learning mechanism we proposed. In this way, we remove the centralized component in the original algorithm. It not only decentralized the agents' actions but also the model itself.

In literature, there are some works that share the similar idea of agent-centric support sets. For example, Deisenroth and Ng (2015) construct a tree structure of Gaussian process. It splits the environmental field into many sub-fields, and each sub-field is modeled by a Gaussian process. Hence, they are not able to share information between the sub-fields. The work of Bui and Turner (2014) on the other hand, constructs a treed support set that can allow the computation scales linearly with the size of the support set. However, their method can not be learned in an incremental way so that it is not suitable for decentralized data fusion. Our agent-centric support set can share local summaries between the sub-fields using the proposed transfer learning mechanism and it allows the agents to add new observations incrementally into the local summaries, which is more applicable for large-scale data fusion.

2.2 Active Learning and Nonstationarity

Active learning algorithms allow the agent to actively choose the data from the domain it learns. It can achieve better performance with less training data than traditional passive learning algorithms. Multi-agent active sensing is a particular type of active learning in the multi-agent system which involves multiple agents to take the most informative observations from the target environmental phenomenon by minimizing the uncertainty in that phenomenon. If we assume the phenomenon follows a Gaussian process prior, the uncertainty of the phenomenon can be quantified based on the correlation structure as mean square error (Low et al., 2008), entropy (Low et al., 2009) or mutual information (Meliou et al., 2007a).

The uncertainty quantified in the existing literature is based on the assumption that the phenomenon is stationary. However, most of the large-scale phenomena are nonstationary in nature. Therefore, quantifying the uncertainty using the stationary assumption will be incorrect, which will result in sub-optimal active sensing performance. Out of the active sensing topic, there are some existing works that have discussed nonstationary data modelling. The methods can be categorized into two main types: a single model with nonstationary kernel function (S) and a mixture of stationary models (M) as shown in table 2.1.

Type	Works	Citation
S	Dot product kernel	Schölkopf and Smola (2002), sec. 7.8
	Neural network kernel	Neal (1996)
	Non-linear warping	Sampson and Guttorp (1992)
	Kernel averaging	Paciorek and Schervish (2003)
M	Hotspots	Low et al. (2009)
	Voronoi tessellation	Cortes et al. (2004)
	Mixture of GPs	Tresp (2001)
	Infinite maxture of GPs	Rasmussen and Ghahramani (2002)

Table 2.1 Modeling nonstationary data

In single model methods, dot product kernel (Schölkopf and Smola, 2002) is too simple to model the real problem. The other two kernels (Neal, 1996, Sampson and Guttorp, 1992) require specific domain knowledge to design the actual kernel. Kernel averaging (Paciorek and Schervish, 2003) is a more general method but it contains a huge number of parameters need to be learned from the data, which is too time-consuming for large-scale environmental sensing applications. Mixture model, on the other hand, is more practical for real-world problems. The works (Cortes et al., 2004, Low et al., 2009, Rasmussen and Ghahramani, 2002, Tresp, 2001) share the similar ideas to split the environmental field into several subfields. The difference is how they split the field. Low et al. (2009) separate the field as highly varying hotspots [1] and smooth background. Cortes et al. (2004) separate the field as a Voronoi graph. These two works require the domain knowledge in the environmental field. The works on mixture of Gaussian processes (Rasmussen and Ghahramani, 2002, Tresp, 2001) are more general modeling techniques that can detect the stationary subfield during the active sensing. A Dirichlet process mixture of Gaussian processes (DPM-GPs) (Rasmussen and Ghahramani, 2002) can dynamically change the number of sub-models in the mixture to fit the data. It is highly practical for the large-scale environmental phenomenon because a large-scale environmental phenomenon usually has globally nonstationary but locally stationary behavior that naturally fits the mixture model.

The methods we mentioned above focus on the modeling of the phenomenon. None of them is specifically designed to direct the agents to actively taking observations from the

[1] hotspots exhibiting extreme measurements and much higher spatial variability than the rest of the field

environmental phenomenon. To the best of my knowledge, there is only one work (Krause and Guestrin, 2007) has tried to learn the correlation structure during the active sensing process but it still has a stationary assumption which is not applicable for a large-scale nonstationary phenomenon.

Our proposed DEC-MES algorithm is the first work that addresses the nonstationarity challenge in the context of active sensing. We use DPM-GPs to fit the nonstationary phenomenon. We derive the formulation of uncertainty based on the mixture model and propose an active sensing criterion to direct multiple agents to take observations that can minimize the uncertainty based on the nonstationary correlation structure learned from the model.

2.3 Optimization and High Dimensionality

Optimization is to search the maximum/minimum value of an objective function under certain constraints. With different structures and constraints over the objective function, the optimization methods can be quite different. In the simplest case, when the objective function is convex, it is easy to use convex optimization methods (Boyd and Vandenberghe, 2004) to get the global optimum with guarantees. When the objective function is not convex, it is still possible to use gradient decent methods (Qian, 1999) to search the local optimum.

However, the objective function may not have analytical expression in complex problems. The method to optimize an unknown function is quite different from the traditional optimization method. Usually, it requires many trials of simulations. The only information we have is the observed input-output pairs from the simulations. In this scenario, some heuristic search methods like genetic algorithms (Akbari and Ziarati, 2011), Monte Carlo methods (Rubinstein and Kroese, 2011), swarm intelligence (Parsopoulos and Vrahatis, 2002) are applied to search the optimum via a huge number of simulations. Those methods require heuristics with domain knowledge in order to generate good results (Tomoiagă et al., 2013). Additionally, those methods require a huge number of simulations, which is impractical for real complex problem since they cost lots of time and resources.

Bayesian optimization (Snoek et al., 2012) is a modern optimization technique that is suitable for optimizing unknown objective functions. Another name of BO is efficient global optimization in the experimental design literature (Jones et al., 1998). It assumes that the unknown function is distributed as a Gaussian process. The belief over the function is updated by the simulated input and output pairs. Bayesian optimization utilizes an acquisition function to capture the shape of unknown function and evaluate the risk on the deviation from the true optimum. This acquisition function automatically balances the exploration and exploitation in choosing the input values for new simulations, which result in relatively fewer trials of simulations in searching the optimum. In the literature, there are three commonly used acquisition functions:

- Probability of improvement (Kushner, 1964). Intuitively, it is to maximize the probability of improving the best current value.

- Expected improvement (Močkus, 1975). Alternatively, one could choose to maximize the expected improvement over the current best value. It demonstrates better performance than the probability of improvement.

- Gaussian process upper confidence bound (GP-UCB) (Srinivas et al., 2009). A more recent method is to exploit the upper/lower confidence bounds (for maximization/minimization) to construct a parametric form of the acquisition function to minimize the regret in searching the optimum. The regret bound of GP-UCB can be derived analytically so that we use it as the acquisition function for our ANOVA-DCOP method.

Many large-scale optimization problems require methods that can deal with high dimensional input. Although BO is successful in solving some problems, especially in learning parameters of machine learning models, it is restricted to problems with less than ten dimensions (Wang et al., 2016). It is challenging to scale BO to high dimensions. To the best of our knowledge, there are three works that specifically addressed high dimensional BO problems: Subspace learning (Djolonga et al., 2013), random embedding (Wang et al.,

2016) and additive model (Kirthevasan et al., 2015). The first two methods (Djolonga et al., 2013, Wang et al., 2016) explore the intrinsic low-rank dimensions in the original high dimensional space. They require specifying the number of the intrinsic dimensions. When the specified number is far from the ground truth, their methods will generate poor results as demonstrated in our experiments in section 5.4. On the other hand, the additive model has a strong assumption that the dimensions are mutually independent. It totally ignores the correlation structure in the dimensions. They ease the limitation by brutally grouping the dimensions and assuming the groups are mutually independent.

The work of additive model (Kirthevasan et al., 2015) motivated our ANOVA-DCOP method. Instead of assuming all the dimensions are independent, it is more reasonable to explore the sparse correlation structure in the dimensions. In contrast with brutally grouping the dimensions, we introduce multi-agent coordination techniques to learn the correlation structure in the process of BO. Unlike the other two methods which require specifying a fixed number of intrinsic dimensions, our work has the flexibility to adjust the correlation structure dynamically.

In multi-agent community, there are many existing works on distributed optimization problems. For example, Dantzig-Wolfe decomposition (Chung, 2011, Frangioni and Gendron, 2013) is one of the distributed optimization methods in linear programming. When the input space of the problem has linear consistent decomposable structure for both objective function and constraints, Dantzig-Wolfe method can decompose the optimization problem into many subproblems. Other problems that beyond linear programming are often cast as decentralized constrained optimization problem (DCOP) (Shoham and Leyton-Brown, 2008). DCOP requires the objective function have a linear summation structure so that the problem can be solved by sharing information between multiple agents. Many notable problems can be formulated as DCOP such as distributed graph coloring and distributed multiple knapsack problem (Frangioni and Gendron, 2013). Researchers have designed many multi-agent coordination methods to solve this problem. Adopt (Modi et al., 2005) and DPOP (Petcu and Faltings, 2005) are two wildly used methods. They can guarantee the optimal solution for an optimization problem but they suffer from exponentially growing coordination overhead. A

breakthrough in the literature is bounded max-sum (Farinelli et al., 2009, Kim and Lesser, 2013). It significantly reduces the time complexity in agent coordination and still maintains the near-optimal solution of the problem. Our work of ANOVA-DCOP introduces DCOP to Bayesian optimization so that multi-agent coordination can be used to optimize an unknown function which is previously not feasible in the literature. Moreover, the bounded max-sum method we are using together with ANOVA kernel can scale up to optimizing an unknown function with hundreds of dimensions.

To sum up, we have reviewed the current literature related to the three challenges in large-scale machine learning problems and positioned our work in the literature to demonstrate the contributions to filling the gap with the scalability challenges due to the characteristics of the large-scale problem. The remaining chapters of the thesis are organized as follows: Chapter 3 explains our GP-DDF method with agent-centric support set to address the challenge of large input domain; Chapter 4 explains our DEC-MES method to address the challenge of nonstationarity; Chapter 5 explains our ANOVA-DCOP method to address the challenge of high dimensionality. Finally, chapter 6 concludes our works and discusses the potentials of utilizing multi-agent techniques in large-scale machine learning problems.

Chapter 3

Gaussian Process Decentralized Data Fusion with Agent-Centric Support Sets for Large-Scale Distributed Cooperative Perception

Motivated by the challenge of large input domain in large-scale regression problem, this chapter presents a novel decentralized data fusion method using agent-centric support set. In section 3.2 and section 3.2, we introduce the agent-centric support set and the transfer learning mechanism to share the local summaries between two agents. Further, in section 3.3, we design a multi-agent coordination algorithm in order to apply the method in the large-scale environmental field. Lastly, we evaluate our GP-DDF method with three real-world datasets.

3.1 Background and Notations

Modeling Spatially Varying Environmental Phenomena with Gaussian Processes (GPs).

A Gaussian process (GP) can model a spatially varying environmental phenomenon as follows: The phenomenon is defined to vary as a realization of a GP. Let $\mathcal{X}$ be a set representing the domain of the phenomenon such that each location $x \in \mathcal{X}$ is associated with a realized (random) measurement y_x (Y_x) if it is observed (unobserved). Let $\{Y_x\}_{x\in\mathcal{X}}$ denote a GP, that is, any 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 $\sigma_{xx'} \triangleq \mathrm{cov}[Y_x, Y_{x'}]$ for all $x, x' \in \mathcal{X}$, the latter of which characterizes the spatial correlation structure of the phenomenon and can be defined, for example, by the widely-used squared exponential covariance function

$$\sigma_{xx'} \triangleq \sigma_s^2 \exp(-0.5\|\Lambda^{-1}(x-x')\|^2) + \sigma_n^2 \delta_{xx'} \tag{3.1}$$

where σ_s^2 and σ_n^2 are, respectively, its signal and noise variance hyperparameters controlling the intensity and the noise of the measurements, Λ is a diagonal matrix with length-scale hyperparameters ℓ_1 and ℓ_2 controlling, respectively, the degree of spatial correlation or "similarity" between measurements in the horizontal and vertical directions of the phenomenon, and $\delta_{xx'}$ is a Kronecker delta that is 1 if $x = x'$, and 0 otherwise.

Supposing a column vector $y_\mathcal{D} \triangleq (y_{x'})^\top_{x'\in\mathcal{D}}$ of realized measurements is observed for some set $\mathcal{D} \subset \mathcal{X}$ of locations, a GP model can exploit these observations/data to perform probabilistic regression by providing a Gaussian *posterior*/predictive distribution

$$\mathcal{N}\big(\mu_x + \Sigma_{x\mathcal{D}}\Sigma_{\mathcal{D}\mathcal{D}}^{-1}(y_\mathcal{D} - \mu_\mathcal{D}), \sigma_{xx} - \Sigma_{x\mathcal{D}}\Sigma_{\mathcal{D}\mathcal{D}}^{-1}\Sigma_{\mathcal{D}x}\big) \tag{3.2}$$

of the measurement for any unobserved location $x \in \mathcal{X} \setminus \mathcal{D}$ where $\mu_\mathcal{D} \triangleq (\mu_{x'})^\top_{x'\in\mathcal{D}}$, $\Sigma_{x\mathcal{D}} \triangleq (\sigma_{xx'})_{x'\in\mathcal{D}}$, $\Sigma_{\mathcal{D}\mathcal{D}} \triangleq (\sigma_{x'x''})_{x',x''\in\mathcal{D}}$, and $\Sigma_{\mathcal{D}x} \triangleq \Sigma_{x\mathcal{D}}^\top$. To predict the phenomenon, a naive approach to data fusion is to fully communicate all the data to every mobile sensing agent,

each of which then predicts the phenomenon separately using the Gaussian predictive distribution in (3.2). Such an approach, however, scales poorly in the data size $|\mathcal{D}|$ due to the need to invert $\Sigma_{\mathcal{D}\mathcal{D}}$ which incurs $\mathcal{O}(|\mathcal{D}|^3)$ time.

GP Decentralized Data Fusion (GP-DDF).

To improve the scalability of the GP model for practical use in data fusion, the work of (Chen et al., 2015) has proposed efficient and scalable GP decentralized data fusion algorithms for cooperative perception of environmental phenomena that can distribute the computational load among the mobile sensing agents. The intuition of the GP-DDF algorithm of (Chen et al., 2015) is as follows: Each of the N mobile sensing agents constructs a local summary of the data/observations taken along its own path based on a common support set $\mathcal{S} \subset \mathcal{X}$ known to all the other agents and communicates its local summary to them. Then, it assimilates the local summaries received from the other agents into a globally consistent summary which is used to compute a Gaussian predictive distribution for predicting the phenomenon.

Formally, the local and global summaries and the Gaussian predictive distribution induced by GP-DDF are defined as follows:

Definition 1 (Local Summary)**.** *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all N mobile sensing agents, each agent i encapsulates a column vector $y_{\mathcal{D}_i}$ of realized measurements for its observed locations $\mathcal{D}_i$ into a local summary $(\nu_{\mathcal{S}|\mathcal{D}_i}, \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ where*

$$\begin{aligned} \nu_{\mathcal{B}|\mathcal{D}_i} &\triangleq \Sigma_{\mathcal{B}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}(y_{\mathcal{D}_i} - \mu_{\mathcal{D}_i}) \ , \\ \Psi_{\mathcal{B}\mathcal{B}'|\mathcal{D}_i} &\triangleq \Sigma_{\mathcal{B}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Sigma_{\mathcal{D}_i\mathcal{B}'} \end{aligned} \tag{3.3}$$

for all $\mathcal{B}, \mathcal{B}' \subset \mathcal{X}$ and $\Sigma_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}} \triangleq \Sigma_{\mathcal{D}_i\mathcal{D}_i} - \Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{D}_i}$.

Definition 2 (Global Summary)**.** *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all N mobile sensing agents and the local summary $(\nu_{\mathcal{S}|\mathcal{D}_i}, \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ of every agent $i = 1, \ldots, N$, a global summary is defined as a tuple $(\dot{\nu}_{\mathcal{S}}, \dot{\Psi}_{\mathcal{S}\mathcal{S}})$ where*

$$\dot{\nu}_{\mathcal{S}} \triangleq \sum_{i=1}^{N} \nu_{\mathcal{S}|\mathcal{D}_i} \ \text{ and } \ \dot{\Psi}_{\mathcal{S}\mathcal{S}} \triangleq \sum_{i=1}^{N} \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i} + \Sigma_{\mathcal{S}\mathcal{S}} \ . \tag{3.4}$$

Definition 3 (GP-DDF)**.** *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all N agents and the global summary $(\dot{\nu}_{\mathcal{S}}, \dot{\Psi}_{\mathcal{S}\mathcal{S}})$, the GP-DDF algorithm run by each agent computes a Gaussian predictive distribution $\mathcal{N}(\overline{\mu}_x, \overline{\sigma}_x^2)$ of the measurement for any unobserved location $x \in \mathcal{X} \setminus \mathcal{D}$ where*

$$\overline{\mu}_x \triangleq \mu_x + \Sigma_{x\mathcal{S}} \dot{\Psi}_{\mathcal{S}\mathcal{S}}^{-1} \dot{\nu}_{\mathcal{S}} \ , \ \overline{\sigma}_x^2 \triangleq \sigma_{xx} - \Sigma_{x\mathcal{S}} (\Sigma_{\mathcal{S}\mathcal{S}}^{-1} - \dot{\Psi}_{\mathcal{S}\mathcal{S}}^{-1}) \Sigma_{\mathcal{S}x} \ . \tag{3.5}$$

The Gaussian predictive distribution (3.5) computed by the GP-DDF algorithm is theoretically guaranteed by Chen et al. (2015) to be equivalent to that induced by the centralized *partially independent training conditional* (PITC) approximation (Quiñonero-Candela and Rasmussen, 2005) of the GP model. Running GP-DDF on each of the N agents can, however, reduce the $\mathcal{O}(|\mathcal{D}|((|\mathcal{D}|/N)^2 + |\mathcal{S}|^2))$ time incurred by PITC to only $\mathcal{O}((|\mathcal{D}|/N)^3 + |\mathcal{S}|^3 + |\mathcal{S}|^2 N)$ time, hence scaling considerably better with increasing data size $|\mathcal{D}|$.

Though GP-DDF scales well with big data, it can predict poorly due to information loss caused by summarizing the measurements and correlation structure of the data/observations and sparse coverage of the areas with highly varying measurements by the support set. To address its shortcoming, the GP-DDF$^+$ algorithm of Chen et al. (2015) exploits the data local to an agent to improve the predictions for unobserved locations "close" to its data (in the correlation sense) while preserving the efficiency of GP-DDF by adopting its idea of summarizing information into local and global summaries (Definitions 1 and 2).

Definition 4 (GP-DDF$^+$)**.** *Given a common support set $\mathcal{S} \subset \mathcal{X}$ known to all N agents, global summary $(\dot{\nu}_{\mathcal{S}}, \dot{\Psi}_{\mathcal{S}\mathcal{S}})$, local summary $(\nu_{\mathcal{S}|\mathcal{D}_i}, \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$, and a column vector $y_{\mathcal{D}_i}$ of realized measurements for observed locations $\mathcal{D}_i$, the GP-DDF$^+$ algorithm run by each agent i computes a Gaussian predictive distribution $\mathcal{N}(\overline{\mu}_x, \overline{\sigma}_x^2)$ of the measurement for any unobserved location $x \in \mathcal{X} \setminus \mathcal{D}$ where*

$$\begin{aligned} \overline{\mu}_x &\triangleq \mu_x + \left(\gamma_{x\mathcal{S}}^i \dot{\Psi}_{\mathcal{S}\mathcal{S}}^{-1} \dot{\nu}_{\mathcal{S}} - \Sigma_{x\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{S}}^{-1} \nu_{\mathcal{S}|\mathcal{D}_i}\right) + \nu_{x|\mathcal{D}_i} \ , \\ \overline{\sigma}_x^2 &\triangleq \sigma_{xx} - \Big(\gamma_{x\mathcal{S}}^i \Sigma_{\mathcal{S}\mathcal{S}}^{-1} \Sigma_{\mathcal{S}x} - \Sigma_{x\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{S}}^{-1} \Psi_{\mathcal{S}x|\mathcal{D}_i} \\ &\qquad - \gamma_{x\mathcal{S}}^i \dot{\Psi}_{\mathcal{S}\mathcal{S}}^{-1} \gamma_{\mathcal{S}x}^i\Big) - \Psi_{xx|\mathcal{D}_i} \end{aligned} \tag{3.6}$$

such that $\gamma^i_{x\mathcal{S}} \triangleq \Sigma_{x\mathcal{S}} + \Sigma_{x\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i} - \Psi_{x\mathcal{S}|\mathcal{D}_i}$ *and* $\gamma^i_{\mathcal{S}x} \triangleq \gamma^{i\top}_{x\mathcal{S}}$.

The Gaussian predictive distribution (3.6) computed by the GP-DDF$^+$ algorithm is observed to exploit local and global summaries (i.e., terms within brackets) as well as data local to agent i (i.e., $\nu_{x|\mathcal{D}_i}$ and $\Psi_{xx|\mathcal{D}_i}$ terms) and theoretically guaranteed by Chen et al. (2015) to be equivalent to that induced by the centralized *partially independent conditional* (PIC) approximation (Snelson and Ghahramani, 2007) of the GP model. In terms of time complexity, GP-DDF$^+$ shares the same improvement in scalability over PIC as that of GP-DDF over PITC.

3.2 GP-DDF with Agent-Centric Support Sets

Transfer Learning.

It can be observed from (3.5) and (3.6) that the GP-DDF and GP-DDF$^+$ algorithms depend on a common support set $\mathcal{S}$ known to all N mobile sensing agents, which raises critical limitations due to the large input domain: (a) Their cubic time cost in $|\mathcal{S}|$ prohibits increasing the size of $\mathcal{S}$ too much to preserve their efficiency, which consequently limits the expansion of the domain of the phenomenon for which it can still be covered and predicted well; (b) if $\mathcal{S}$ sparsely covers the large-scale phenomenon due to its restricted size and is thus "far" from the data and unobserved locations to be predicted, then the values of the components in terms like $\Sigma_{\mathcal{S}\mathcal{D}_i}$ and $\Sigma_{x\mathcal{S}}$ tend to zero, which degrade their predictive performance; and (c) when switching to a new support set, they have to wastefully discard all previous summaries based on the old support set.

To address the above limitations, a straightforward approach inspired by the local GPs method (Choudhury et al., 2002, Das and Srivastava, 2010) is to partition the domain of the phenomenon into local areas and run GP-DDF or GP-DDF$^+$ with a different, sufficiently dense support set for each local area. But, such an approach often suffers from discontinuities in predictions on the boundaries between local areas [1] and only utilizes the data within a

[1] An exception is the work of Park et al. (2011) that overcomes this boundary effect by imposing continuity constraints along the boundaries in a centralized manner.

local area for its predictions, thereby performing poorly in local areas with little/no data. These drawbacks motivate the need to design and develop a transfer learning mechanism for a team of mobile sensing agents capable of sharing and transferring information encapsulated in a summary based on a support set for a local area to that utilizing a different support set for another area. In this section, we will describe our transfer learning mechanism and its use in our GP-DDF or GP-DDF$^+$ algorithm with agent-centric support sets and theoretically bound and analyze its resulting loss of information.

Specifically, supposing a mobile sensing agent i moves from a local area with support set $\mathcal{S}$ to another local area with a different support set $\mathcal{S}'$ (i.e., $\mathcal{S}\bigcap\mathcal{S}'=\emptyset$), the local summary $(\nu_{\mathcal{S}'|\mathcal{D}_i},\Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ based on the new support set $\mathcal{S}'$ can be derived *exactly* from the local summary $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ utilizing the old support set $\mathcal{S}$ only when the data $(\mathcal{D}_i,y_{\mathcal{D}_i})$ gathered by agent i (i.e., discarded after encapsulating into $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$) in the local area with support set $\mathcal{S}$ can be *fully* recovered from $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$, which is unfortunately not possible. Our key idea is thus to derive the local summary $(\nu_{\mathcal{S}'|\mathcal{D}_i},\Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ *approximately* from $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ in an efficient and scalable manner by exploiting the following important definition:

Definition 5 (Prior Summary)**.** *Given a support set $\mathcal{S}\subset\mathcal{X}$ for a local area, each mobile sensing agent i encapsulates a column vector $y_{\mathcal{D}_i}$ of realized measurements for its observed locations $\mathcal{D}_i$ into a prior summary $(\omega_{\mathcal{S}|\mathcal{D}_i},\Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ where*

$$\omega_{\mathcal{S}|\mathcal{D}_i}\triangleq\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}(y_{\mathcal{D}_i}-\mu_{\mathcal{D}_i})\ ,\ \Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}\triangleq\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}. \tag{3.7}$$

The prior summary $(\omega_{\mathcal{S}|\mathcal{D}_i},\Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ (3.7) is defined in a similar manner to the local summary $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ (3.3) except for the $\Sigma_{\mathcal{D}_i\mathcal{D}_i}$ term in the former replacing the $\Sigma_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}$ term in the latter and is the main ingredient for making our proposed transfer learning mechanism efficient and scalable. Interestingly, the prior summary based on the new support set $\mathcal{S}'$ can be approximated from the prior summary utilizing the old support set S as follows:

Proposition 1. *If $Y_{\mathcal{S}'}$ and $Y_{\mathcal{D}_i}$ are conditionally independent given $Y_{\mathcal{S}}$ (i.e., $\Sigma_{\mathcal{S}'\mathcal{D}_i|\mathcal{S}}=\Sigma_{\mathcal{S}'\mathcal{D}_i}-\Sigma_{\mathcal{S}'\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{D}_i}=\underline{0}$) for $i=1,\ldots,N$, then*

$$\omega_{\mathcal{S}'|\mathcal{D}_i}=\Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\omega_{\mathcal{S}|\mathcal{D}_i},\ \Phi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i}=\Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{S}'}. \tag{3.8}$$

Its proof is in Appendix A.1.

Remark. The conditional independence assumption in Proposition 1 extends that on the training conditionals of PITC and PIC (Section 3.1) which have already assumed conditional independence of $Y_{\mathcal{D}_1},\ldots,Y_{\mathcal{D}_N}$ given $Y_{\mathcal{S}}$. Alternatively, it can be interpreted as a low-rank covariance matrix approximation $\Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}$ of $\Sigma_{\mathcal{S}'\mathcal{D}_i}$. The quality of this approximation will be theoretically guaranteed later.

To efficiently and scalably derive the local summary $(\nu_{\mathcal{S}'|\mathcal{D}_i},\Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ approximately from $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$, our transfer learning mechanism will first have to transform the local summary $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ to the prior summary $(\omega_{\mathcal{S}|\mathcal{D}_i},\Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ based on the old support set $\mathcal{S}$, then use the latter to approximate the prior summary $(\omega_{\mathcal{S}'|\mathcal{D}_i},\Phi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ based on the new support set $\mathcal{S}'$ by exploiting Proposition 1, and finally transform the approximated prior summary back to approximate the local summary $(\nu_{\mathcal{S}'|\mathcal{D}_i},\Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$, as detailed in Algorithm 1 below. The above two transformations can be achieved by establishing the following relationship between the local summary and prior summary:

Proposition 2. *Given a support set $\mathcal{S}\subset\mathcal{X}$ for a local area, the local summary* $(\nu_{\mathcal{S}|\mathcal{D}_i},\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ (3.3) *and the prior summary* $(\omega_{\mathcal{S}|\mathcal{D}_i},\Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ (3.7) *of agent i are related by*

$$\Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}^{-1}\omega_{\mathcal{S}|\mathcal{D}_i}=\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}^{-1}\nu_{\mathcal{S}|\mathcal{D}_i}\ ,\ \Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}^{-1}=\Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}^{-1}+\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\ . \tag{3.9}$$

Its proof is in Appendix A.2.

Supposing agent i has gathered additional data $(\mathcal{D}_i',y_{\mathcal{D}_i'})$ from the local area with the new support set $\mathcal{S}'$, it can be encapsulated into a local summary $(\nu_{\mathcal{S}'|\mathcal{D}_i'},\Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i'})$ that is assimilated with the approximated local summary $(\nu_{\mathcal{S}'|\mathcal{D}_i},\Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ by simply summing them up:

$$\begin{aligned}\nu_{\mathcal{S}'|\mathcal{D}_i\cup\mathcal{D}_i'} &= \nu_{\mathcal{S}'|\mathcal{D}_i}+\nu_{\mathcal{S}'|\mathcal{D}_i'}\ ,\\ \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i\cup\mathcal{D}_i'} &= \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i}+\Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i'}\ ,\end{aligned} \tag{3.10}$$

which require making a further assumption of conditional independence between $\mathcal{D}_i'$ and $\mathcal{D}_j$ given the support set $\mathcal{S}'$ for $j=1,\ldots,N$.

Finally, to assimilate the local summary of agent i with the other agents' local summaries (i.e., based on their current choices of support sets) into a global summary to be used for predicting the phenomenon, the local summary $(\nu_{\mathcal{S}'|\mathcal{D}_j}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_j})$ of every other agent $j \neq i$ based on agent i's support set $\mathcal{S}'$ can be derived approximately from the received local summary $(\nu_{\mathcal{S}''|\mathcal{D}_j}, \Psi_{\mathcal{S}''\mathcal{S}''|\mathcal{D}_j})$ based on agent j's support set $\mathcal{S}'' \neq \mathcal{S}'$ using exactly the same transfer learning mechanism described above. Then, the global summary $(\dot{\nu}_{\mathcal{S}'}, \dot{\Psi}_{\mathcal{S}'\mathcal{S}'})$ can be computed via (3.4) and used by the GP-DDF or GP-DDF$^+$ algorithm (Section 3.1).

Algorithm 1: GP-DDF/GP-DDF$^+$ with agent-centric support sets based on transfer learning for agent i

if agent i transits from local area with support set $\mathcal{S}$ to local area with support set $\mathcal{S}'$ **then**

- `/* Transfer learning mechanism */`
- Construct local summary $(\nu_{\mathcal{S}|\mathcal{D}_i}, \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ and transform it to prior summary $(\omega_{\mathcal{S}|\mathcal{D}_i}, \Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ by (3.9);
- Derive prior summary $(\omega_{\mathcal{S}'|\mathcal{D}_i}, \Phi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ based on $\mathcal{S}'$ approximately from $(\omega_{\mathcal{S}|\mathcal{D}_i}, \Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ by (3.8);
- Transform prior summary $(\omega_{\mathcal{S}'|\mathcal{D}_i}, \Phi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ to local summary $(\nu_{\mathcal{S}'|\mathcal{D}_i}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ by (3.9);

if agent i has to predict the phenomenon **then**

- **if** data $(\mathcal{D}'_i, y_{\mathcal{D}'_i})$ is available from local area with support set $\mathcal{S}'$ **then**
  - Assimilate local summaries $(\nu_{\mathcal{S}'|\mathcal{D}_i}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i})$ with $(\nu_{\mathcal{S}'|\mathcal{D}'_i}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}'_i})$ to yield $(\nu_{\mathcal{S}'|\mathcal{D}_i\cup\mathcal{D}'_i}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i\cup\mathcal{D}'_i})$ by (3.10);
- Exchange local summary with every agent $j \neq i$;
- **foreach** agent $j \neq i$ in local area with support set $\mathcal{S}'' \neq \mathcal{S}'$ **do**
  - Derive local summary $(\nu_{\mathcal{S}'|\mathcal{D}_j}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_j})$ based on $\mathcal{S}'$ approximately from received local summary $(\nu_{\mathcal{S}''|\mathcal{D}_j}, \Psi_{\mathcal{S}''\mathcal{S}''|\mathcal{D}_j})$ based on $\mathcal{S}''$ using the above transfer learning mechanism;
- Compute global summary $(\dot{\nu}_{\mathcal{S}'}, \dot{\Psi}_{\mathcal{S}'\mathcal{S}'})$ by (3.4) using local summaries $(\nu_{\mathcal{S}'|\mathcal{D}_i\cup\mathcal{D}'_i}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i\cup\mathcal{D}'_i})$ and $(\nu_{\mathcal{S}'|\mathcal{D}_j}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_j})$ of every agent $j \neq i$;
- Run GP-DDF (3.5) or GP-DDF$^+$ (3.6);

Supposing $|\mathcal{S}| = |\mathcal{S}'| = |\mathcal{S}''|$ for simplicity, our transfer learning mechanism in Algorithm 1 incurs only $\mathcal{O}(|\mathcal{S}|^3)$ time (i.e., independent of data size $|\mathcal{D}|$) due to multiplication and

inversion of matrices of size $|\mathcal{S}|$ by $|\mathcal{S}|$. Since the support set for every local area is expected to be small, our transfer learning mechanism is efficient and scalable.

Recall from the remark after Proposition 1 that our transfer learning mechanism has utilized a low-rank covariance matrix approximation $\Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}$ of $\Sigma_{\mathcal{S}'\mathcal{D}_i}$. To theoretically bound the information loss resulting from such an approximation, we first observe that it resembles the Nyström low-rank approximation except that the latter typically involves approximating a symmetric positive semi-definite matrix like $\Sigma_{\mathcal{S}'\mathcal{S}'}$ or $\Sigma_{\mathcal{D}_i\mathcal{D}_i}$ instead of $\Sigma_{\mathcal{S}'\mathcal{D}_i}$, which precludes a direct application of existing results on Nyström approximation (Sun et al., 2015) to our theoretical analysis. Fortunately, we can exploit the idea of clustering with respect to $\mathcal{S}$ for our theoretical analysis which is inspired by that of the Nyström approximation of Zhang et al. (2008) but results in a different loss bound depending on the GP hyperparameters (Section 3.1) and the "closeness" of $\mathcal{S}'$ and $\mathcal{D}_i$ to $\mathcal{S}$ in the correlation sense.

Define $c(x)$ as a function mapping each $x \in \mathcal{D}_i \bigcup \mathcal{S}'$ to the "closest" $c(x) \in \mathcal{S}$, that is, $c : \mathcal{D}_i \bigcup \mathcal{S}' \rightarrow \mathcal{S}$ where $c(x) \triangleq \arg\min_{s\in\mathcal{S}} ||\Lambda^{-1}(x-s)||$. Then, partition $\mathcal{D}_i$ ($\mathcal{S}'$) into $|\mathcal{S}|$ disjoint subsets $\mathcal{D}_{is} \triangleq \{x \in \mathcal{D}_i \mid c(x) = s\}$ ($\mathcal{S}'_s \triangleq \{x \in \mathcal{S}' \mid c(x) = s\}$) for $s \in \mathcal{S}$. Intuitively, $\mathcal{D}_{is}$ ($\mathcal{S}'_s$) is a cluster of locations in $\mathcal{D}_i$ ($\mathcal{S}'$) that are closest to location s in the support set $\mathcal{S}$.

Our main result below theoretically bounds the information loss $||\Sigma_{\mathcal{S}'\mathcal{D}_i} - \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}||_F$ resulting from the low-rank approximation $\Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}$ of $\Sigma_{\mathcal{S}'\mathcal{D}_i}$ with respect to the Frobenius norm:

Theorem 1. *Let $\sigma_{xx'}$ be defined by a squared exponential covariance function* (3.1)*,$T \triangleq \arg\max_{s\in\mathcal{S}} |\mathcal{D}_{is}|$, $T' \triangleq \arg\max_{s\in\mathcal{S}} |\mathcal{S}'_s|$, $\varepsilon_{\mathcal{S}'} \triangleq |\mathcal{S}'|^{-1}\sum_{x\in\mathcal{S}'} ||\Lambda^{-1}(x-c(x))||^2$, and $\varepsilon_{\mathcal{D}_i} \triangleq |\mathcal{D}_i|^{-1}\sum_{x\in\mathcal{D}_i} ||\Lambda^{-1}(x-c(x))||^2$. Then, $||\Sigma_{\mathcal{S}'\mathcal{D}_i} - \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}||_F$ has a upper bound:*

$$\sqrt{3/e}\sigma_s^2|\mathcal{S}|TT'\left(\sqrt{\varepsilon_{\mathcal{S}'}+\varepsilon_{\mathcal{D}_i}}+\sqrt{\varepsilon_{\mathcal{S}'}}+\sqrt{\varepsilon_{\mathcal{D}_i}}\ +\sigma_s^2||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F|\mathcal{S}|\sqrt{3\varepsilon_{\mathcal{S}'}\varepsilon_{\mathcal{D}_i}/e}\right). \quad (3.11)$$

Its proof is in Appendix A.3. Note that a similar result to Theorem 1 can be derived for other commonly-used covariance functions such as those presented in the work of Zhang et al. (2008).

It can be observed from Theorem 1 that the information loss $||\Sigma_{\mathcal{S}'\mathcal{D}_i} - \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}||_F$ can be reduced when the signal variance σ_s^2 is small, the length-scales ℓ_1 and/or ℓ_2 are large, the mobile sensing agent i utilizes a support set $\mathcal{S}$ "close" to its observed locations $\mathcal{D}_i$ in a local area (i.e., smaller $\varepsilon_{\mathcal{D}_i}$) and moves to another local area with a support set $\mathcal{S}'$ "close" to $\mathcal{S}$ (i.e., smaller $\varepsilon_{\mathcal{S}'}$).

Lazy Transfer Learning.

Theorem 1 above further reveals that every instance of transfer learning in Algorithm 1 incurs some information loss which accumulates over multiple instances when the agent transits between many local areas and consequently degrades its resulting predictive performance. This motivates the need to be frugal in the number of instances of transfer learning to be performed.

To achieve this, our key idea is to delay transfer learning till prediction time but in a memory-efficient manner[2].

Specifically, we propose the following information sharing mechanism to reduce memory requirements for a team of mobile sensing agents: When agent i leaves a local area, its local summary is communicated to another agent in the same area who assimilates it with its own local summary using (3.4). However, if no other agent is in the same area, then agent i stores a backup of its local summary.

On the other hand, when agent i enters a local area containing other agents, it simply obtains its corresponding support set to encapsulate its new data gathered in this area. But, if no other agent is in this area, then agent i retrieves (and removes) the backup of its corresponding local summary from an agent who has previously visited this area[3]. If no agent has such a backup, then agent i is the first to visit this area and constructs a new support set for it. Algorithm 4 (Appendix A.4) details the GP-DDF/GP-DDF$^+$ algorithm

[2]Naively, an agent can delay transfer learning by simply storing a separate local summary based on the support set for every previously visited local area, which is not memory-efficient.

[3]Multiple backups of the local summary for the same local area may exist if agents leave this area at the same time, which rarely happens. In this case, agent i should retrieve (and remove) all these backups from the agents storing them.

with agent-centric support sets by incorporating the above information sharing mechanism in order to achieve memory-efficient lazy transfer learning.

To analyze the memory requirements of our information sharing mechanism in Algorithm 4 (Appendix A.4), let the domain of the phenomenon be partitioned into K local areas. Then, the team of N mobile sensing agents incurs a total of $\mathcal{O}((K+N)|\mathcal{S}|^2)$ memory in the worst case when all the agents reside in the same local area and the last agent entering this area stores the backups of the local summaries for the other $K-1$ local areas.

However, the agents are usually well-distributed over the entire phenomenon in practice: In the case of evenly distributed agents, the team incurs a total of $\mathcal{O}(\max(K,N)|\mathcal{S}|^2)$ memory. So, each agent incurs an amortized memory cost of $\mathcal{O}(\max(K,N)|\mathcal{S}|^2/N)$.

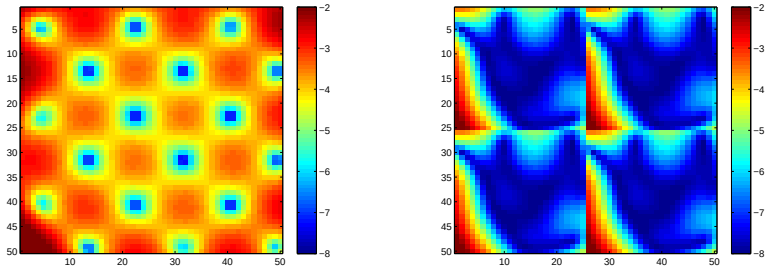
A limitation of the information sharing mechanism in Algorithm 4 (Appendix A.4) is its susceptibility to agent failure: If an agent stores the backups of the local summaries for many local areas and breaks down, then all the information on these local areas will be lost. Its robustness to agent failure can be improved by distributing multiple agents to every local area to reduce its risk of being empty and hence its likelihood of inducing a backup.

3.3 Experiments and Discussion

In the experiments, we first test the agent-centric support set and transfer learning mechanism on a simulated spatial phenomenon, and then the entire performance of our algorithm will be evaluated with two real world datasets and one of them is millions in size.

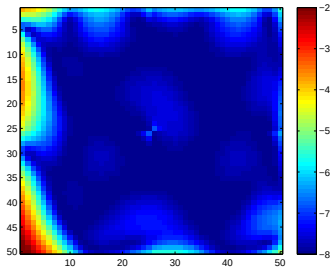
3.3.1 Simulated Spatial Phenomena

The toy experiment here is set up to demonstrate the effectiveness of our proposed lazy transfer learning mechanism (Section 3.2) that is driving our GP-DDF/GP-DDF$^+$ algorithms with agent-centric support sets (Appendix A.4): A number of 2-dimensional spatial phenomena of size 50 by 50 are generated using signal variance $\sigma_s^2 = 1$, noise variance $\sigma_n^2 = 0.01$, and by varying the length-scale $\ell_1 = \ell_2$ from 1 to 20. The domain of the spatial phenomenon is partitioned into 4 disjoint local areas of size 25 by 25 (Fig. 3.1), each of which contains

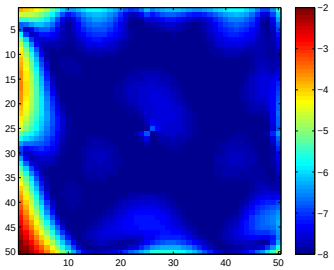


(a) GP-DDF

(b) Local PITCs



(c) Full PITCs



(d) GP-DDF-ASS

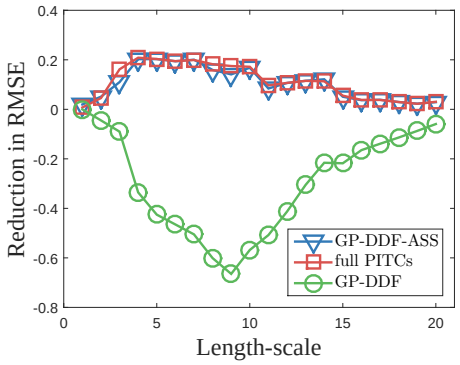
Fig. 3.1 (a-d) Maps of log-predictive variance (i.e., $\log \overline{\sigma}_x^2$ for all $x \in \mathcal{X}$) over a spatial phenomenon with length-scale of 10 achieved by the tested decentralized data fusion algorithms.

an agent moving randomly within to gather 25 local data/observations. We compare the predictive performance of the following decentralized data fusion algorithms: (a) Original GP-DDF (Chen et al., 2012, 2015) with a common support set of size 18 distributed over the entire phenomenon and known to all 4 agents, (b) *PITCs utilizing local information* (local PITCs) with agent-centric support sets assign a different PITC to each agent summarizing its gathered local data based on a support set of size 18 distributed over its residing local area, (c) *PITCs utilizing full information* (full PITCs) with agent-centric support sets assign a different PITC to each agent summarizing its gathered local data as well as those communicated by the other agents (i.e., full data gathered by all agents) based on a support set of size 18 distributed over its residing local area, (d) *GP-DDF with agent-centric support sets* (GP-DDF-ASS) each of size 18 and distributed over a different local area (Algorithm 4 in Appendix A.4). Note that if our proposed lazy transfer learning mechanism in GP-DDF-ASS incurs minimal (total) information loss, then its predictive performance will be similar to that of full PITCs (local PITCs).

Fig. 3.1 shows results of the maps of log-predictive variance (i.e., $\log \overline{\sigma}_x^2$ for all $x \in \mathcal{X}$) over a spatial phenomenon with length-scale of 10 achieved by the tested decentralized data fusion algorithms. It can be observed from Fig. 3.1a that GP-DDF achieves the worst predictive performance since the size of its common support set is only a quarter of that used by the other tested algorithms. From Fig. 3.1b, though local PITCs can predict better than GP-DDF, the predictive uncertainty at the boundaries between local areas remains very high, which is previously explained in Section 3.2. Fig. 3.1c shows the most ideal predictive performance achieved by full PITCs because each agent exploits the full data gathered by and exchanged with all agents for encapsulating into a global summary based on the support set distributed over its residing local area. Fig. 3.1d reveals that GP-DDF-ASS can achieve predictive performance comparable to that of full PITCs without needing to exchange the full data between all agents due to minimal information loss by our proposed lazy transfer learning mechanism.

Recall from Theorem 1 (Section 3.2) that the information loss incurred by our proposed transfer learning mechanism depends on the closeness between the support sets distributed over different local areas as well as the closeness (i.e., in the correlation sense) between the support sets and the data/observations. The effect of varying such closeness on the performance of our transfer learning mechanism can be empirically investigated by alternatively changing the length-scale to control the degree of spatial correlation between the measurements of the phenomenon.

Fig. 3.2 shows results of the reduction in RMSE of GP-DDF, full PITCs, and GP-DDF-ASS over local PITCs with varying lengthscales from 1 to 20. It can be observed that only GP-DDF performs worse than local PITCs while both GP-DDF-ASS and full PITCs perform significantly better than local PITCs, all of which are explained previously. Interestingly, the reduction in RMSEs varies for different length-scales and tends to zero when the length-scale is either too small or large. With a very small length-scale, the correlations between the support sets distributed over different local areas and between the support sets and the data/observations become near-zero, hence resulting in poor transfer learning for GP-DDF-ASS. This agrees with the observation in our theoretical analysis for Theorem 1 (Section 3.2).



(e) Reduction in RMSE

Fig. 3.2 Graphs of reduction in RMSE of GP-DDF, full PITCs, and GP-DDF-ASS over local PITCs vs. varying length-scales.

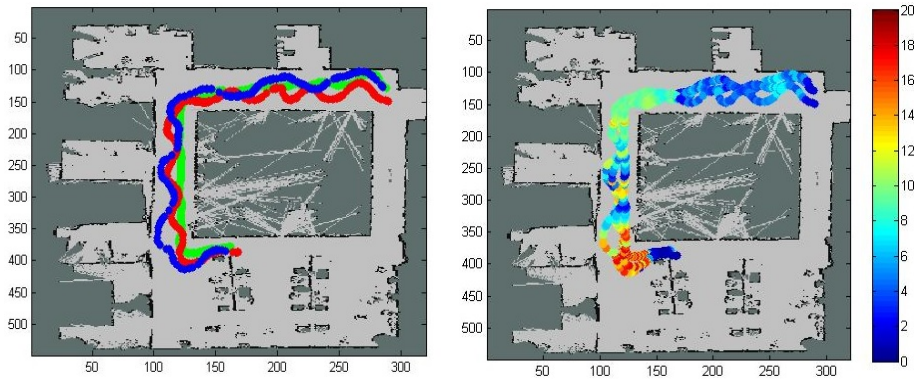
With a very large length-scale, though their correlations are strong, the local observations/data can be used by local PITCs to predict very well, hence making transfer learning redundant. Our transfer learning mechanism performs best with intermediate length-scales where the correlations between the support sets distributed over different local areas and between the support sets and the data are sufficiently strong but not to the extent of achieving good predictions with simply local data.

3.3.2 Experiments on Real-World data

The performance of our GP-DDF and GP-DDF$^+$ algorithms with agent-centric support sets are empirically evaluated using the following two real-world datasets:

(a) The indoor lighting quality dataset contains 1200 observations of relative lighting level gathered simultaneously by three real Pioneer 3-DX mobile robots mounted with SICK LMS200 laser rangefinders and weather boards while patrolling an office environment, as shown in Fig. 3.3. The domain of interest is partitioned into $K = 8$ consecutive local areas and the robots patrol to and fro across them such that they visit all $K = 8$ local areas exactly twice to gather observations of relative lighting level.

(b) the monthly sea surface temperature (°C) dataset (Fig. 3.4) is bounded within lat. 35.75-14.25S and lon. 80.25-104.25E (i.e., in the Indian ocean) and gathered from Dec. 2002 to Dec. 2015 with a data size of $1,083,608$. The huge spatiotemporal domain of



(a) (b)

Fig. 3.3 (a) Red, green, and blue trajectories of three Pioneer 3-DX mobile robots in an office environment generated by AMCL package in ROS, along which (b) 1200 observations of relative lighting level are gathered simultaneously by the robots at locations denoted by small colored circles.

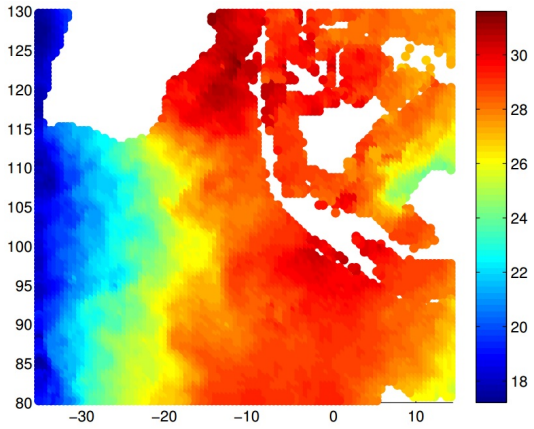


Fig. 3.4 Temperature phenomenon bounded within lat. 35.75-14.25S and lon. 80.25-104.25E in Dec. 2015.

this phenomenon comprises 5-dimensional input feature vectors of latitude, longitude, year, month, and season, and is spatially partitioned into 32 disjoint local areas, each of which is further temporally partitioned into 64 disjoint intervals (hence, $K = 2048$) and assigned 2 agents moving randomly within to gather local observations; the results are averaged over 10 runs. We have also investigated the effect of varying degrees of spatial correlation (specifically, length-scale) between measurements of simulated spatial phenomena on the effectiveness of our proposed lazy transfer learning mechanism (Section 3.2) due to varying

extents on its resulting information loss (Theorem 1) and reported the empirical results in Appendix 3.3.1.

Two performance metrics are used in our experiments: (a) *Root-mean-square error* (RMSE) $\sqrt{|\mathcal{X}|^{-1}\sum_{x\in\mathcal{X}}(\overline{\mu}_x - y_x)^2}$ measures the predictive performance of the tested algorithms while (b) incurred time measures their efficiency and scalability. The performance of our *GP-DDF and GP-DDF$^+$ algorithms with agent-centric support sets* (respectively, GP-DDF-ASS and GP-DDF$^+$-ASS), each of which is of size 30 (200) and randomly distributed over a different local area of the office environment (temperature phenomenon), are compared against that of the local GPs method (Choudhury et al., 2002, Das and Srivastava, 2010) and state-of-the-art GP-DDF and GP-DDF$^+$ (Chen et al., 2015) with a common support set of size 30 (200) randomly distributed over the entire office environment (temperature phenomenon) and known to all agents; consequently, the latter construct local summaries of the same size. The hyperparameters of GP-DDF-ASS and GP-DDF$^+$-ASS are learned using maximum likelihood estimation, as detailed in Appendix A.5.

Predictive Performance.

Figs. 3.5a and 3.5c show results of decreasing RMSE achieved by tested algorithms with an increasing total number of observations, which is expected. It can be observed that GP-DDF-ASS and GP-DDF$^+$-ASS, respectively, outperform GP-DDF and GP-DDF$^+$, as explained previously in the introduction. Furthermore, the performance improvement of GP-DDF-ASS over GP-DDF is larger than that of GP-DDF$^+$-ASS over GP-DDF$^+$, which demonstrates the effectiveness of our lazy transfer learning mechanism, especially when some local areas lack data/observations. This also explains the better predictive performance of GP-DDF$^+$-ASS over local GPs, even though they both exploit local data.

Time Efficiency.

In this experiment, we specifically evaluate the time efficiency of our transfer learning mechanism (Section 3.2) in GP-DDF-ASS and GP-DDF$^+$-ASS with respect to the number of observations; to do this, we have intentionally ignored the time incurred by their information

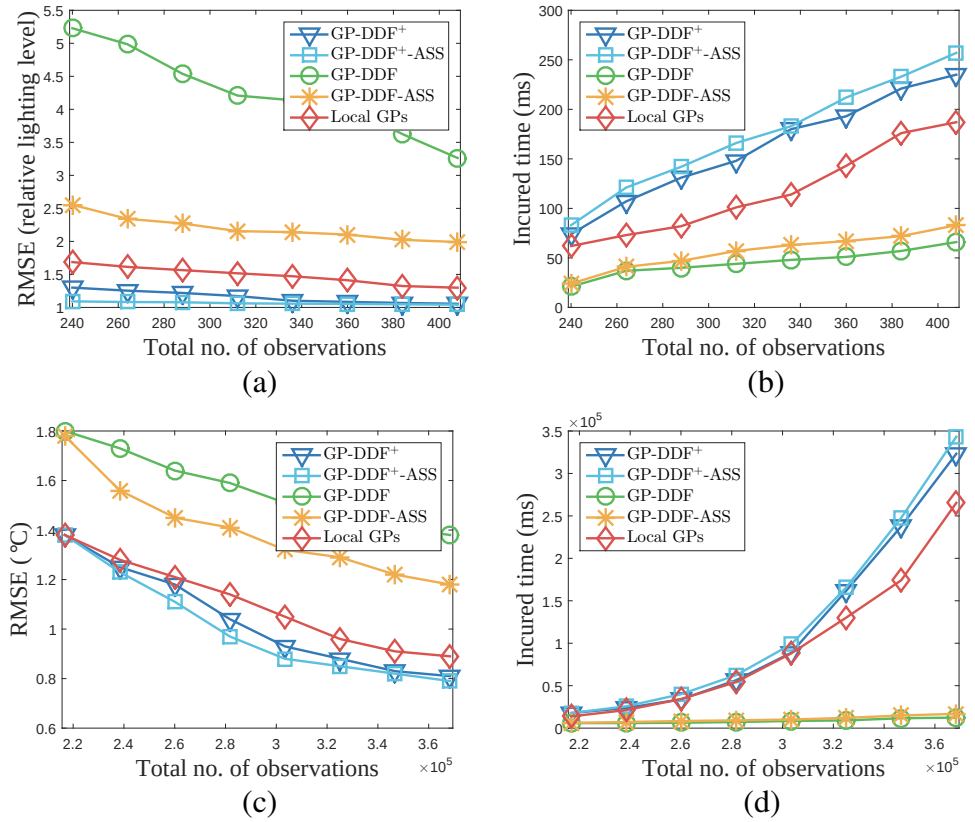


Fig. 3.5 Graphs of RMSE and total time incurred by tested algorithms vs. total no. of observations for (a-b) indoor lighting quality and (c-d) temperature phenomenon.

sharing mechanism (i.e., first if-then construct in Algorithm 4 in Appendix A.4) and compared their resulting incurred time with that of GP-DDF and GP-DDF$^+$ (i.e., without transfer learning). Figs. 3.5b and 3.5d show results of increasing total time incurred by tested algorithms when the total number of observations increases, which is expected (Section 3.1). It can be observed that GP-DDF-ASS and GP-DDF$^+$-ASS, respectively, incur only slightly more time than GP-DDF and GP-DDF$^+$ (i.e., due to an extra small fixed cost of $\mathcal{O}(|\mathcal{S}|^3)$ time for transfer learning (Section 3.2)) to achieve more superior predictive performance, especially for GP-DDF-ASS. GP-DDF$^+$-ASS incurs more time than GP-DDF-ASS (local GPs) to further exploit local data (support set and transfer learning) for improving its predictive performance. For time-critical applications, we recommend using GP-DDF-ASS over GP-DDF$^+$-ASS since its incurred time is small and increases very gradually with more observations. For big data applications, GP-DDF$^+$-ASS is instead preferred since a large amount of local data is often available in nearly every local area for prediction.

We have also empirically evaluated the scalability of GP-DDF-ASS and GP-DDF$^+$-ASS in the number of agents and observed that their total incurred time decrease with more agents and they, respectively, incur only slightly more time than GP-DDF and GP-DDF$^+$ due to their information sharing mechanism described in Section 3.2 (i.e., first if-then construct in Algorithm 4 in Appendix A.4). The empirical results are are reported in detail in Appendix A.6.

3.4 Conclusion

This chapter describes novel GP-DDF-ASS and GP-DDF$^+$-ASS algorithms for distributed cooperative perception of large-scale environmental phenomena. To overcome the limitations of GP-DDF and GP-DDF$^+$ (Chen et al., 2012, 2013b, 2015), our proposed GP-DDF-ASS and GP-DDF$^+$-ASS algorithms employ a new transfer learning mechanism between agents which is capable of sharing and transferring information encapsulated in a summary based on a support set to that utilizing a different support set with some loss that can be theoretically bounded and analyzed. To alleviate the issue of information loss accumulating over multiple instances of transfer learning, GP-DDF-ASS and GP-DDF$^+$-ASS exploit an information sharing mechanism to achieve memory-efficient lazy transfer learning. Empirical evaluation on two real-world datasets show that our transfer learning and information sharing mechanisms make GP-DDF-ASS and GP-DDF$^+$-ASS incur only slightly more time than GP-DDF and GP-DDF$^+$ (i.e., without transfer learning) to achieve more superior predictive performance.

Chapter 4

Multi-Robot Active Sensing of Non-Stationary Gaussian Process-Based Environmental Phenomena

Motivated by the challenge of nonstationarity in large-scale active learning problem, this chapter explains the details of our proposed DEC-MAS algorithm to address nonstationarity issue in active sensing. Section 4.1 describes the Gaussian process model and mixture model to quantify the uncertainty in a nonstationary environmental field. Section 4.3 illustrates the DEC-MAS algorithm with the proposed active sensing criterion. Lastly, section 4.4 demonstrates the empirical performance of the DEC-MAS on two real-world datasets.

4.1 Modeling a Phenomenon

4.1.1 Gaussian Process (GP)

A nonstationary environmental phenomenon can be viewed as a mixture of stationary phenomena. We will first introduce the modeling of a stationary field by a Gaussian process with additional notations that captures the specific local smoothness, and then introduce the mixture model that combines multiple Gaussian processes.

A Gaussian process (GP) (Rasmussen and Williams, 2006) can be used to model a spatially varying phenomenon as follows: The phenomenon is defined to vary as a realization of a GP. Let V denote a set of sampling units representing the domain of the phenomenon such that each sampling unit $x \in V$ is specified by a d-dimensional feature vector and is also associated with a realized (random) measurement y_x (Y_x) if x is sampled/observed (unobserved). Let $\{Y_x\}_{x \in V}$ denote a GP, that is, every finite subset of $\{Y_x\}_{x \in V}$ has a multivariate Gaussian distribution (Chen et al., 2013a, Rasmussen and Williams, 2006). The GP is fully specified by its *prior* mean $\mu_x \triangleq \mathbb{E}[Y_x]$ and covariance $\sigma_{xx'|\theta} \triangleq \text{cov}[Y_x, Y_{x'}|\theta]$ for all $x, x' \in V$, the latter of which characterizes the spatial correlation structure of the phenomenon and can be defined using a covariance function parameterized by θ, as described later.

Supposing a column vector y_D of realized measurements is available for some set $D \subset V$ of observed sampling units, the GP can exploit these observations to predict the measurements for any set $X \subseteq V \setminus D$ of unobserved sampling units as well as provide their corresponding predictive uncertainties using the following Gaussian *posterior* mean vector and covariance matrix, respectively:

$$\mu_{X|D,\theta} \triangleq \mu_X + \Sigma_{XD|\theta}\Sigma_{DD|\theta}^{-1}(y_D - \mu_D) \tag{4.1}$$

$$\Sigma_{XX|D,\theta} \triangleq \Sigma_{XX|\theta} - \Sigma_{XD|\theta}\Sigma_{DD|\theta}^{-1}\Sigma_{DX|\theta} \tag{4.2}$$

where μ_X (μ_D) is a column vector with mean components μ_x for all $x \in X$ ($x \in D$), $\Sigma_{XX|\theta}$ ($\Sigma_{DD|\theta}$) is a covariance matrix with covariance components $\sigma_{xx'|\theta}$ for all $x, x' \in X$ ($x, x' \in D$), $\Sigma_{XD|\theta}$ is a covariance matrix with covariance components $\sigma_{xx'|\theta}$ for all $x \in X, x' \in D$, and $\Sigma_{DX|\theta}$ is the transpose of $\Sigma_{XD|\theta}$. The posterior covariance matrix $\Sigma_{XX|D,\theta}$ (4.2), which is independent of the measurements y_D, can be used to quantify the uncertainty of the predictions through, for example, the Gaussian posterior joint entropy:

$$\mathbb{H}[Y_X|y_D, \theta] \triangleq \frac{1}{2}\log(2\pi e)^{|X|}\left|\Sigma_{XX|D,\theta}\right| . \tag{4.3}$$

We will focus on using this entropy-based measure of predictive uncertainty in this work.

A GP can model a stationary phenomenon by defining its prior covariance $\sigma_{xx'|\theta}$ using a stationary covariance function (Rasmussen and Williams, 2006), that is, it is a function of

$x - x'$. Hence, it is invariant to translations in the domain V. A common choice is the squared exponential covariance function:

$$\sigma_{xx'|\theta} \triangleq \sigma_s^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i - x_i'}{\ell_i}\right)^2\right) + \sigma_n^2 \delta_{xx'} \tag{4.4}$$

where $x_i\,(x_i')$ is the i-th component of the d-dimensional feature vector $x\,(x')$, the set of hyperparameters $\theta \triangleq \{\sigma_s^2, \sigma_n^2, \ell_1, \ldots, \ell_d\}$ are, respectively, signal and noise variances and length-scales, and $\delta_{xx'}$ is a Kronecker delta that is 1 if $x = x'$ and 0 otherwise. Intuitively, the signal and noise variances describe, respectively, the intensity and noise of the measurements while each length-scale ℓ_i controls the degree of smoothness in the spatial variation of the measurements (i.e., spatial correlation or "similarity" between measurements) with respect to the i-th feature component. If the hyperparameters are not known, they can be trained using the available observations via *maximum likelihood estimation* (MLE) (Rasmussen and Williams, 2006), that is, by choosing θ that maximizes the log marginal likelihood $\log p(y_D|\theta) =$

$$-\frac{1}{2}(y_D - \mu_D)^\top \Sigma_{DD|\theta}^{-1}(y_D - \mu_D) - \frac{1}{2}\log(2\pi)^{|D|}\left|\Sigma_{DD|\theta}\right| \ . \tag{4.5}$$

Similarly, a GP can model a non-stationary phenomenon by specifying its prior covariance with a non-stationary covariance function, the choice of which involves a trade-off between the richness of the resulting GP model vs. computational efficiency. For example, the simple non-stationary polynomial and neural network covariance functions (Rasmussen and Williams, 2006) only need a few hyperparameters to be determined. But, they do not exhibit a desirable *locality property*[1] that holds for many stationary covariance functions (e.g., (4.4)) and, more importantly, has been widely exploited by existing MAS algorithms mentioned in the literature review to achieve time efficiency. On the other hand, the complex non-stationary version of Matérn covariance function (Paciorek and Schervish, 2003) requires a large number of hyperparameters to be specified. Though it can capture the locality property, the training

[1]The locality property (Krause et al., 2008b) states that the spatial correlation of measurements between sampling units decreases to zero with increasing distance between them.

of its hyperparameters, when unknown, is computationally expensive. An alternative to using a single GP is to consider modeling the non-stationary phenomenon with a mixture of GPs that can provide a fine balance between richness and efficiency as well as a useful structural property to be exploited by our DEC-MAS algorithm, as described next.

4.1.2 Dirichlet Process Mixture of Gaussian Processes (DPM-GPs)

It is often observed (e.g., see Fig. 1.3) that the measurements in separate areas of a non-stationary phenomenon vary according to different locally stationary spatial correlation structures (Sampson et al., 2001). Such a phenomenon can be modeled with high fidelity by a Dirichlet process mixture of locally stationary GPs (Rasmussen and Ghahramani, 2002), which offers the following representational and computational advantages over a single non-stationary GP (Section 4.1.1): (a) It preserves the use of the well-studied and widely-applied stationary covariance functions, many of which exhibit the locality property (Section 4.1.1) and are computationally friendly with only a few (unknown) hyperparameters to be trained, (b) the required number of locally stationary GPs can automatically grow with the increasing complexity of the phenomenon, and (c) each locally stationary GP only incurs cubic time in the size of the observations that are local to its corresponding area of prediction instead of over the entire phenomenon.

A DPM-GPs can model a non-stationary phenomenon as follows: The phenomenon is defined to vary as a realization of a DPM-GPs. Let its number of locally stationary GP components be denoted by K. For each GP component $k = 1, \ldots, K$, its prior covariance characterizes a locally stationary spatial correlation structure and is defined using a stationary covariance function parameterized by θ_k. In order to estimate the unknown θ_k using MLE (4.5), the measurements y_{D_k} (where $D_k \subseteq D$) that are induced by GP component k have to be identified first. That is, every observed sampling unit $x \in D$ has to be associated with a realized component label denoted by z_x and $D_k \triangleq \{x \in D | z_x = k\}$. To realize these component labels $z_D \triangleq \{z_x\}_{x \in D}$, we use Gibbs sampling, as detailed next.

Each random component label, denoted by Z_x, for all $x \in D$ follows a sampling unit-dependent Dirichlet process prior:

$$p(Z_x = k|z_{D\setminus\{x\}}, \theta_k) = \begin{cases} \dfrac{n_{xk}}{|D|-1+\alpha} & \text{if } k \leq K, \\ \dfrac{\alpha}{|D|-1+\alpha} & \text{if } k = K+1, \end{cases} \tag{4.6}$$

where $n_{xk} \triangleq (|D|-1)(\sum_{x'\in N_x} \sigma_{xx'|\theta_k} \delta_{z_{x'}k})/(\sum_{x'\in N_x} \sigma_{xx'|\theta_k})$, $N_x \triangleq \{x' \in D \setminus \{x\} | d_G(x,x') \leq \gamma\}$ for some $\gamma > 0$, $d_G(x,x')$ is the shortest path length between sampling units x and x' with respect to the topology of a graph G induced from V to be traversed by the robots (Section 4.2), $\sigma_{xx'|\theta_k}$ is previously defined in (4.4), and α denotes a concentration parameter. The Dirichlet process prior (4.6) can be understood as follows: When $k = 1, \ldots, K$, the probability of the observation at x being induced by GP component k is proportional to the number of neighboring observed sampling units with the same component label k weighted by their proximity to x. Its probability of being induced by a new GP component $K+1$ is proportional to α. Hence, α controls the addition of new GP components. For the new GP component $K+1$, its θ_{K+1} is sampled from a pre-defined uniform distribution.

Given the realized measurements y_D for the set D of observed sampling units, the Dirichlet process prior can be updated using Bayes' rule to the following posterior:

$$\begin{aligned} &p(Z_x = k|z_{D\setminus\{x\}}, y_D, \theta_k) \propto \\ &\begin{cases} p(y_x|Z_x = k, y_{D_k\setminus\{x\}}, \theta_k) p(Z_x = k|z_{D\setminus\{x\}}, \theta_k) & \text{if } k \leq K, \\ p(y_x|Z_x = k, \theta_k) p(Z_x = k|z_{D\setminus\{x\}}, \theta_k) & \text{if } k = K+1, \end{cases} \end{aligned} \tag{4.7}$$

where

$$p(y_x|Z_x = k, y_{D_k\setminus\{x\}}, \theta_k) \sim \mathcal{N}(\mu_{x|D_k\setminus\{x\},\theta_k}, \Sigma_{xx|D_k\setminus\{x\},\theta_k})$$

for $k \leq K$ and $p(y_x|Z_x = K+1, \theta_{K+1}) \sim \mathcal{N}(\mu_x, \sigma_{xx|\theta_{K+1}})$. It can be seen from $p(y_x|Z_x = k, y_{D_k\setminus\{x\}}, \theta_k)$ that an observation induced by a GP component is conditionally independent of the observations induced by the other GP components, a structural property of which will be exploited by our DEC-MAS algorithm (Sections 4.2 and 4.3).

Using the posterior (4.7), Gibbs sampling (Gilks et al., 1996) is performed (starting with $K = 1$) to realize the component labels z_D. In Appendix, we propose two heuristics to speed up the convergence of this sampling method. Given z_D, θ_k can now be trained using MLE

(4.5). Such a process of Gibbs sampling followed by MLE is iterated until the values of z_D stabilize or a user-defined limit is reached.

Given the realized measurements y_D and component labels z_D for observed sampling units D, the DPM-GPs can exploit them to predict the measurement for an unobserved sampling unit x by aggregating the predictions of the K GP components weighted by their probability of inducing it:

$$\mu_{x|D,\theta} = \sum_{k=1}^{K} \mu_{x|D_k,\theta_k}\ p(Z_x = k|z_D,\theta_k) \tag{4.8}$$

where $\theta \triangleq \{\theta_1,\ldots,\theta_K\}$ and $p(Z_x = k|z_D,\theta_k)$, which is defined in a similar way to (4.6), can be used to estimate the unknown partition of the phenomenon.

4.2 Multi-Robot Active Sensing (MAS)

Define a directed graph $G \triangleq (V,E)$ where the domain V of a phenomenon is connected by a set $E \subseteq V \times V$ of edges such that there is an edge (x,x') if and only if a robot can traverse from $x \in V$ to $x' \in V$ within some user-defined cost constraint (e.g., time interval, traveling distance). The MAS problem is then formulated as follows: Supposing the robots have previously observed the measurements y_D from a set $D \subset V$ of sampling units and used these observations to estimate their corresponding component labels z_D by Gibbs sampling and the hyperparameters θ of the DPM-GPs by MLE (Section 4.1.2), they have to coordinate to jointly select the next most informative set X^* of sampling units (i.e., with corresponding measurements and component labels of maximum joint entropy) to be observed:

$$X^* = \arg\max_{X} \mathbb{H}[Y_X, Z_X|y_D, z_D, \theta]\ . \tag{4.9}$$

The next possible sampling unit to be observed by each robot is constrained to be selected from one that is adjacent to the robot's current residing sampling unit in G. Using chain rule for entropy, it can be shown that these max-entropy sampling units X^* minimize the posterior joint entropy (i.e., $\mathbb{H}[Y_{V\setminus(D\bigcup X^*)}, Z_{V\setminus(D\bigcup X^*)}|Y_{X^*}, Z_{X^*}, y_D, z_D, \theta]$) of

the measurements and component labels for the remaining unobserved sampling units (i.e., $V \setminus (D \bigcup X^*)$) in the phenomenon. $\mathbb{H}[Y_{V\setminus(D\bigcup X^*)}, Z_{V\setminus(D\bigcup X^*)}|Y_{X^*}, Z_{X^*}, y_D, z_D, \theta] = \mathbb{H}[Z_{V\setminus(D\bigcup X^*)}|Z_{X^*}, z_D, \theta] + \mathbb{H}[Y_{V\setminus(D\bigcup X^*)}|Y_{X^*}, Z_{V\setminus D}, y_D, z_D, \theta]$ by chain rule for entropy. So, the choice of X^* (4.9) jointly optimizes a trade-off between gathering the most informative observations for estimating the unknown partition (i.e., component labels $Z_{V\setminus(D\bigcup X^*)}$ in unobserved areas) vs. that for predicting the phenomenon (i.e., measurements $Y_{V\setminus(D\bigcup X^*)}$ in unobserved areas) given the current, imprecise estimate of the partition (i.e., component labels $Z_{V\setminus D}$ and z_D).

Unfortunately, evaluating $\mathbb{H}[Y_X, Z_X|y_D, z_D, \theta]$ in (4.9) is prohibitively expensive with a large number $|X|$ of robots, as explained below. We will therefore derive a tractable approximation to $\mathbb{H}[Y_X, Z_X|y_D, z_D, \theta]$:

$$\begin{aligned}
&\mathbb{H}[Y_X, Z_X|y_D, z_D, \theta] \\
&= \mathbb{H}[Z_X|z_D, \theta] + \mathbb{H}[Y_X|Z_X, y_D, z_D, \theta] \\
&\approx \sum_{x\in X} \mathbb{H}[Z_x|z_D, \theta] + \mathbb{H}[Y_X|\widehat{z}_X, y_D, z_D, \theta] \\
&= \sum_{x\in X} \mathbb{H}[Z_x|z_D, \theta] + \sum_{k=1}^{K} \mathbb{H}[Y_{X_k}|\widehat{z}_{X_k}, y_{D_k}, \theta_k]
\end{aligned} \tag{4.10}$$

where $\mathbb{H}[Z_x|z_D, \theta] \triangleq -\sum_{k=1}^{K} p(Z_x = k|z_D, \theta_k) \log p(Z_x = k|z_D, \theta_k)$, $p(Z_x = k|z_D, \theta_k)$ is defined in a similar way to (4.6), $X_k \triangleq \{x \in X|\widehat{z}_x = k\}$, and $\mathbb{H}[Y_{X_k}|\widehat{z}_{X_k}, y_{D_k}, \theta_k]$ can be evaluated in closed form using (4.14). The first equality follows from the chain rule for entropy and can then be expanded to $\sum_{z_X}(-\log p(z_X|z_D, \theta) + \mathbb{H}[Y_X|z_X, y_D, z_D, \theta])p(z_X|z_D, \theta)$, which requires enumerating an exponential (i.e., in the number $|X|$ of robots) number of possible assignments z_X to evaluate the summation. This computational burden is eased by the approximation in (4.10): Its first summation term is obtained using chain rule for entropy followed by assuming conditional independence of Z_x for all $x \in X$ given z_D and θ. Its second term is due to the same conditional independence assumption to yield $p(z_X|z_D, \theta_k) = \prod_{x\in X} p(z_x|z_D, \theta_k)$ followed by plugging the maximum likelihood estimate $Z_X = \widehat{z}_X$ into $\mathbb{H}[Y_X|Z_X, y_D, z_D, \theta]$ where $\widehat{z}_x = \arg\max_{z_x} p(z_x|z_D, \theta_k)$ for all $x \in X$. We conjecture that, in

practice, the assumption becomes less restrictive when the number $|D|$ of observations increases to potentially reduce the degree of violation of conditional independence, the spatial correlation between measurements decreases, and the robots are sufficiently far apart. The last equality in (4.10) arises from the chain rule for entropy and the structural property of DPM-GPs that observations between GP components are conditionally independent (Section 4.1.2).

If the approximation in (4.10) is used as the active sensing criterion instead, then the MAS problem becomes

$$\widetilde{X} = \arg\max_{X} \widetilde{\mathbb{H}}[Y_X, Z_X|y_D, z_D, \theta] \ , \tag{4.11}$$

$$\widetilde{\mathbb{H}}[Y_X, Z_X|y_D, z_D, \theta] \triangleq \sum_{x\in X} \mathbb{H}[Z_x|z_D, \theta] + \sum_{k=1}^{K} \mathbb{H}[Y_{X_k}|\widehat{z}_{X_k}, y_{D_k}, \theta_k]. \tag{4.12}$$

Note that the choice of $\widetilde{X}$ jointly optimizes a trade-off between observing sampling units with most uncertain component labels (i.e., first summation term) vs. that with most uncertain measurements (i.e., second summation term) given the current, imprecise estimate of their labels and z_D.

4.3 Decentralized Multi-Robot Active Sensing (DEC-MAS)

In the previous section, we have presented a *centralized MAS* (CEN-MAS) algorithm (4.11) that coordinates the exploration of multiple robots to jointly optimize a trade-off between observing sampling units with most uncertain component labels vs. that with most uncertain measurements given the current, imprecise estimate of their labels. However, solving (4.11) is computationally costly due to the space of possible X that grows exponentially in the number $|X|$ of robots. To alleviate this computational difficulty, we propose a DEC-MAS algorithm that exploits the structural property of DPM-GPs (Section 4.1.2) and the locality property of stationary covariance functions used by each GP component (Section 4.1.1) for efficient decentralized coordination.

The key idea underlying the need to coordinate any two robots in a team is as follows: Based on (a) the structural property of DPM-GPs that observations between GP components

are conditionally independent (Section 4.1.2), and (b) the locality property of each stationary GP component that the spatial correlation of measurements between sampling units decreases to zero with increasing distance between them (Section 4.1.1), two robots have to coordinate their active sensing only when (a) some pair of their next possible sampling units to be observed are associated with the same GP component (i.e., same estimated component labels), and (b) the correlation of the measurements for such a pair is high enough due to their spatial proximity, respectively. We formalize this idea using the notion of a coordination graph, as defined next.

A *coordination graph* is defined to be an undirected graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ that consists of a set $\mathcal{V}$ of vertices denoting the robots, and a set $\mathcal{E}$ of edges representing coordination dependencies between robots such that there exists an edge $\{r, r'\}$ incident with robots $r \in \mathcal{V}$ and $r' \in \mathcal{V} \setminus \{r\}$ iff

$$\max_{x \in N_r, x' \in N_{r'}} |\varsigma_{xx'}| > \varepsilon$$
$$\varsigma_{xx'} = \begin{cases} \Sigma_{xx'|D_k,\theta_k} & \text{if } \widehat{z}_x = \widehat{z}_{x'} = k, \\ 0 & \text{otherwise,} \end{cases} \tag{4.13}$$

for some $\varepsilon > 0$ where N_r ($N_{r'}$) denotes the set of sampling units adjacent to robot r's (r''s) current residing sampling unit in G. Using (4.13), each robot can determine its adjacency to all the other robots in a decentralized manner and exchange this adjacency information with them so as to construct a consistent adjacency matrix for representing $\mathcal{G}$.

The next step is to determine the connected components of $\mathcal{G}$ whose resulting vertex sets partition the set $\mathcal{V}$ of robots into, say, N disjoint subsets $\mathcal{V}_1, \ldots, \mathcal{V}_N$ such that the robots within each subset have to coordinate their active sensing. Each robot 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, define

$$\widehat{\mathbb{H}}[Y_{X_k}|\widehat{z}_{X_k}, y_{D_k}, \theta_k] \triangleq \frac{1}{2}\log(2\pi e)^{|X_k|}\left|\widehat{\Sigma}_{X_kX_k|D_k,\theta_k}\right| \tag{4.14}$$

where $\widehat{\Sigma}_{X_kX_k|D_k,\theta_k}$ is a block-diagonal matrix comprising diagonal blocks of the form $\Sigma_{X_{kn}X_{kn}|D_k,\theta_k}$ for $n = 1, \ldots, N$ where $X_{kn} \triangleq \{x \in X_n | \widehat{z}_x = k\}$ and X_n denotes a set of next possible sam-

pling units to be observed by the set $\mathcal{V}_n$ of robots for $n = 1,\ldots,N$. So, $X_n = \bigcup_{k=1}^{K} X_{kn}$ and $X = \bigcup_{n=1}^{N} X_n$. Then, it can be derived from (4.14) that

$$\widehat{\mathbb{H}}[Y_{X_k}|\widehat{z}_{X_k}, y_{D_k}, \theta_k] = \sum_{n=1}^{N} \mathbb{H}[Y_{X_{kn}}|\widehat{z}_{X_{kn}}, y_{D_k}, \theta_k] \tag{4.15}$$

by exploiting the property that the log-determinant of a block-diagonal matrix is equal to the sum of log-determinants of its diagonal blocks. The MAS problem (4.11) is consequently approximated by

$$\begin{aligned}
&\max_{X} \sum_{x\in X} \mathbb{H}[Z_x|z_D,\theta] + \sum_{k=1}^{K} \widehat{\mathbb{H}}[Y_{X_k}|\widehat{z}_{X_k}, y_{D_k}, \theta_k] \\
&= \max_{\bigcup_{n=1}^{N} X_n} \sum_{n=1}^{N} \sum_{x\in X_n} \mathbb{H}[Z_x|z_D,\theta] + \sum_{k=1}^{K} \mathbb{H}[Y_{X_{kn}}|\widehat{z}_{X_{kn}}, y_{D_k}, \theta_k] \\
&= \sum_{n=1}^{N} \max_{X_n} \sum_{x\in X_n} \mathbb{H}[Z_x|z_D,\theta] + \sum_{k=1}^{K} \mathbb{H}[Y_{X_{kn}}|\widehat{z}_{X_{kn}}, y_{D_k}, \theta_k]
\end{aligned} \tag{4.16}$$

where the first equality is due to (4.15). More importantly, the last equality can be solved in a partially decentralized manner by each disjoint subset $\mathcal{V}_n$ of robots for $n = 1,\ldots,N$:

$$\widehat{X}_n = \arg\max_{X_n} \sum_{x\in X_n} \mathbb{H}[Z_x|z_D,\theta] + \sum_{k=1}^{K} \mathbb{H}[Y_{X_{kn}}|\widehat{z}_{X_{kn}}, y_{D_k}, \theta_k] \ . \tag{4.17}$$

The degree of decentralization for our DEC-MAS algorithm (4.17) can be varied by controlling ε: Increasing ε causes more robots to become isolated vertices in $\mathcal{G}$, thus decreasing the size $\eta \triangleq \max_n |\mathcal{V}_n|$ of its largest connected component and entailing higher degree of decentralization.

Let

$$\xi \triangleq \max_{k,n,X_{kn},i,i'} \left| \left[\Sigma^{-1}_{X_{kn}X_{kn}|D_k,\theta_k} \right]_{ii'} \right| \tag{4.18}$$

and $\varepsilon \triangleq 0.5K \log 1 \Big/ \left(1 - \left(|\mathcal{V}|^{1.5}\eta\xi\varepsilon\right)^2\right)$. We prove in the theoretical result below that $\widehat{X} = \bigcup_{n=1}^{N} \widehat{X}_n$ is guaranteed to achieve an entropy $\widetilde{\mathbb{H}}[Y_{\widehat{X}}, Z_{\widehat{X}}|y_D, z_D, \theta]$ (i.e., by plugging $\widehat{X}$ into (4.12)) that is at most ε less than the maximum entropy $\widetilde{\mathbb{H}}[Y_{\widetilde{X}}, Z_{\widetilde{X}}|y_D, z_D, \theta]$ achieved by $\widetilde{X}$ (4.11):

Theorem 2 (Performance Guarantee)**.** *If* $|\mathcal{V}|^{1.5}\eta\xi\varepsilon < 1$, *then* $\widetilde{\mathbb{H}}[Y_{\widetilde{X}}, Z_{\widetilde{X}}|y_D, z_D, \theta] - \widetilde{\mathbb{H}}[Y_{\widehat{X}}, Z_{\widehat{X}}|y_D, z_D, \theta] \leq \varepsilon$.

The proof of Theorem 2 is given in Appendix. The implication of Theorem 2 is that our DEC-MAS algorithm (4.17) is competitive (i.e., small ε) as compared to the CEN-MAS algorithm (4.11) when (a) the number $|\mathcal{V}|$ of robots is not too large, (b) the largest connected component of η robots being formed in $\mathcal{G}$ is reasonably small, (c) the minimum required correlation ε between the next possible sampling units to be observed by adjacent robots is kept low, and (d) the number K of GP components is small.

4.3.1 Time and Communication Complexity

In this subsection, we will analyze the time and communication complexity of our DEC-MAS algorithm. Suppose that the observations are distributed evenly among the K GP components and denote the maximum out-degree and in-degree of G by δ and δ', respectively. Then, $|N_x| \leq \Delta \triangleq (\delta + \delta')^\gamma$ for all $x \in D$. Gibbs sampling for estimating the component labels z_D followed by MLE for estimating the hyperparameters θ (Section 4.1.2) incur $\mathcal{O}(M|D|K((|D|/K)^3 + \Delta))$ time over M iterations. Our DEC-MAS algorithm (4.17) incurs $\mathcal{O}(K(|D|/K)^3 + \eta\delta^\eta(K\Delta + (|D|/K)^2 + \eta^2))$ time. By setting $\eta = |\mathcal{V}|$, it yields the time complexity of the CEN-MAS algorithm (4.11) for comparison.

Central to the efficiency of our DEC-MAS algorithm is the requirement of a small η (i.e., size of largest connected component of robots being formed in $\mathcal{G}$ to coordinate their active sensing), which is in fact achieved in practice, as explained by the following observations: For a GP component with small spatial correlation, the posterior entropy of the measurements in the unobserved part of its local area of prediction remains high after sampling, hence attracting more robots to explore it. But, its small spatial correlation entails high degree of decentralization (4.13), thus resulting in a small η. On the other hand, for a GP component with large spatial correlation, the posterior entropy of the measurements in the unobserved part of its local area of prediction becomes low after sampling, hence attracting fewer robots to explore it. So, a small η is also maintained.

For our DEC-MAS algorithm, each robot broadcasts $\mathcal{O}(|\mathcal{V}|)$-sized and $\mathcal{O}(1)$-sized messages on its adjacency information and new observation, respectively.

4.4 Experiments and Discussion

4.4.1 Experimental Setup

This section evaluates the active sensing performance and time efficiency of DEC-MAS empirically on two real-world datasets featuring non-stationary phenomena: (a) June 2012 MODIS plankton density (chlorophyll-a) data of Gulf of Mexico (Fig. 1.3a) discretized into a 60×60 grid of sampling locations/units and bounded within lat. $28.175 - 29.975$N and lon. $87.675 - 89.475$W. The mean density is 4.5 mg/m^3 and standard deviation is 9.8 mg/m^3; (b) Traffic speeds data along 775 road segments (including highways, arterials, slip roads, etc.) of an urban road network (Fig. 1.3b) during the evening peak hours on April 20, 2011. The mean speed is 52.8 km/h and standard deviation is 21.1 km/h. Each sampling unit (i.e., road segment) is specified by a 4-dimensional feature vector: length, number of lanes, speed limit, and direction. This non-stationary traffic phenomenon is modeled using a Dirichlet process mixture of stationary relational GPs; the relational GP is previously developed in (Chen et al., 2012) and its stationary correlation structure can exploit both the road segment features and road network topology information.

For each dataset, 5% of the data are randomly selected as prior observations to estimate their corresponding prior component labels z_D by Gibbs sampling and the prior hyperparameters θ of the DPM-GPs by MLE (Section 4.1.2). Subsequently, they are constantly updated using the new observations gathered by running DEC-MAS repeatedly. For DEC-MAS, ε (4.13) is set to 0.1. The experiments are run on a PC with Intel® Core™2 Quad CPU Q9550 at 2.83 GHz. The results shown below are averaged over 40 trials of randomly selected initial robots' residing sampling units.

Performance metrics. The first metric evaluates active sensing performance of a tested MAS algorithm: It measures *root mean squared error* (RMSE) $\sqrt{\sum_{x \in V} (\mu_{x|D,\theta} - y_x)^2 / |V|}$ over domain V of the phenomenon that results from using the posterior mean $\mu_{x|D,\theta}$ of the

algorithm's utilized model (i.e., (4.1) of GP or (4.8) of DPM-GPs with stationary covariance function (4.4)) to predict the measurements for the remaining unobserved sampling units $V \setminus D$ given the gathered observations. The second metric evaluates the time efficiency and scalability of a tested MAS algorithm by measuring its incurred time.

Comparison of MAS algorithms. The performance of our DEC-MAS algorithm is compared to that of the state-of-the-art MAS algorithms, as listed in Table 4.1 and briefly described next: The *centralized maximum entropy sampling* (CEN-MES) algorithm (Low et al., 2009) repeatedly selects the next set X of sampling units to be observed that maximizes (4.3) based on a stationary GP model. After gathering the observations, CEN-MES can alternatively use DPM-GPs (instead of GP) for prediction (i.e., (4.8)) and we call this CEN-MES+D. The partially *decentralized maximum entropy sampling* (DEC-MES) algorithm (Chen et al., 2012) exploits a similar notion of the coordination graph to split a robot team into disjoint sub-teams, each of which runs CEN-MAS separately without coordinating with other sub-teams. The MAX-SUM algorithm (Rogers et al., 2011) is a general-purpose iterative solver for distributed constraint optimization problems. In (Rogers et al., 2011), MAX-SUM is only used to optimize (4.3) based on the GP model; it does not utilize DPM-GPs nor optimize our novel MAS criterion (4.11), which are done here. Unlike DEC-MAS, the performance guarantee of MAX-SUM offers a non-informative, loose worst-case approximation ratio that only holds for tree-like coordination structures. Lastly, to show the importance of observing sampling units with highly uncertain component labels, the first summation term in (4.17) is removed to yield

$$\max_{X_n} \sum_{k=1}^{K} \mathbb{H}[Y_{X_{kn}} | \widehat{z}_{X_{kn}}, y_{D_k}, \theta_k] \ , \tag{4.19}$$

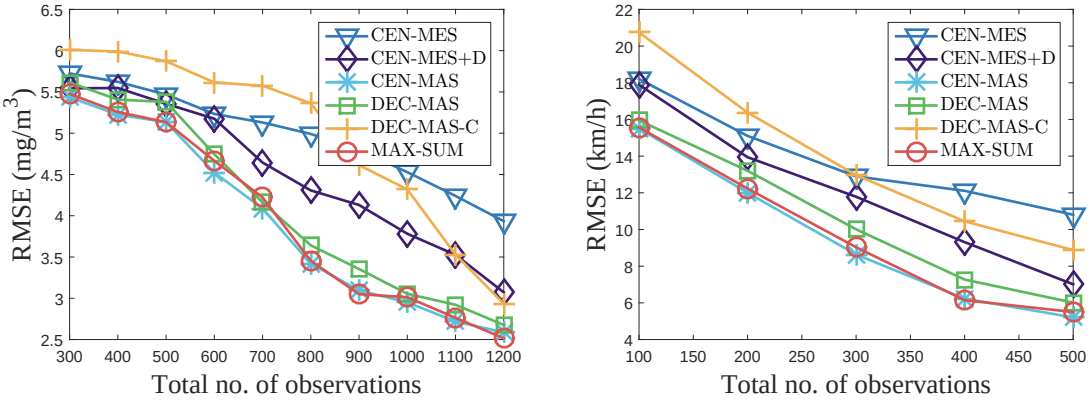
which we call DEC-MAS-C. Note that it is prohibitively expensive to compare with the maximum mutual information-based algorithm of (Krause et al., 2008b), which scales poorly with increasing domain size $|V|$ and is hence not practical for real-time active sensing. For example, it incurred > 62 hours to generate paths for 3 robots to sample a total of 267 observations in a grid of $|V| = 1424$ sampling units, as reported in (Low et al., 2011).

Algorithm	Model	Criterion
CEN-MES (Low et al., 2009)	GP	(4.3)
DEC-MES (Chen et al., 2012)	GP	(4.3)
CEN-MES+D	GP (active sensing) DPM-GPs (prediction)	(4.3)
MAX-SUM (Rogers et al., 2011)	DPM-GPs	(4.11)
CEN-MAS	DPM-GPs	(4.11)
DEC-MAS	DPM-GPs	(4.17)
DEC-MAS-C	DPM-GPs	(4.19)

Table 4.1 Comparison of MAS algorithms (Each algorithm exploits a single model for both active sensing and prediction, except for CEN-MES+D).

4.4.2 Results and Analysis

A. Effect of criterion on predictive performance.



(a) Plankton density (b) Traffic speeds

Fig. 4.1 Graphs of predictive performance vs. total no. $|D|$ of observations gathered by $|\mathcal{V}| = 4$ robots.

Fig. 4.1 shows results of the predictive performance using varying number $|D|$ of observations gathered by $|\mathcal{V}| = 4$ robots running the tested algorithms. The observations are as follows:

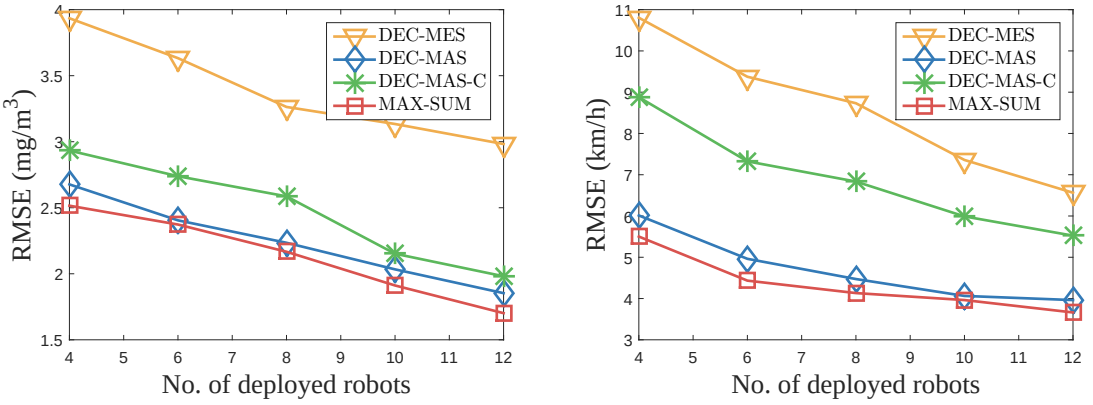
(I) The algorithms optimizing active sensing criterion (4.11) or (4.17) based on DPM-GPs (i.e., CEN-MAS, DEC-MAS, and MAX-SUM) can achieve the best predictive performance (i.e., lowest RMSE) due to the following reasons: (a) DPM-GPs can model and predict the non-stationary phenomena better than a stationary GP, as observed in the performance

improvement of CEN-MES+D over CEN-MES by using DPM-GPs (instead of GP) for prediction, and (b) algorithms optimizing the criteria (4.11) or (4.17) can gather more informative observations than algorithms using criterion (4.3), as observed in the performance improvement of CEN-MAS, DEC-MAS, and MAX-SUM over CEN-MES+D while using DPM-GPs for prediction.

(II) More superior predictive performance can be achieved by jointly optimizing the trade-off between observing sampling units with most uncertain component labels vs. that with most uncertain measurements given the current, imprecise estimate of their labels than by solely addressing the latter criterion; this is observed in the more superior performance of CEN-MAS, DEC-MAS, and MAX-SUM over DEC-MAS-C, the latter of which neglects observing sampling units with highly uncertain labels (4.19).

(III) DEC-MAS optimizing criterion (4.17) can achieve predictive performance close to that of CEN-MAS and MAX-SUM using criterion (4.11). It is prohibitively expensive to obtain results for CEN-MAS with $|\mathcal{V}| > 4$ robots. So, we will only present results for decentralized algorithms from now on.

B. Effect of decentralization on predictive performance.



(a) Plankton density (b) Traffic speeds

Fig. 4.2 Graphs of predictive performance vs. no. $|\mathcal{V}|$ of deployed robots gathering a total of (a) $|D| = 1200$ and (b) $|D| = 500$ observations from plankton density and traffic phenomena, respectively.

Fig. 4.2 shows results of the predictive performance using a total of $|D| = 1200$ and $|D| = 500$ observations gathered from plankton density and traffic phenomena, respectively, by varying number $|\mathcal{V}|$ of robots running the tested decentralized algorithms. The observations are as follows:

(I) The predictive performance of all decentralized algorithms improve with increasing number of robots because every robot is tasked to gather less observations and their performance are thus less adversely affected by their greedy selection of maximum-entropy sampling units. Consequently, more informative unobserved sampling units are explored.

(II) DEC-MAS performs significantly better than DEC-MES and DEC-MAS-C due to the same reasons as that given in the previous observations A(I) and A(II), respectively.

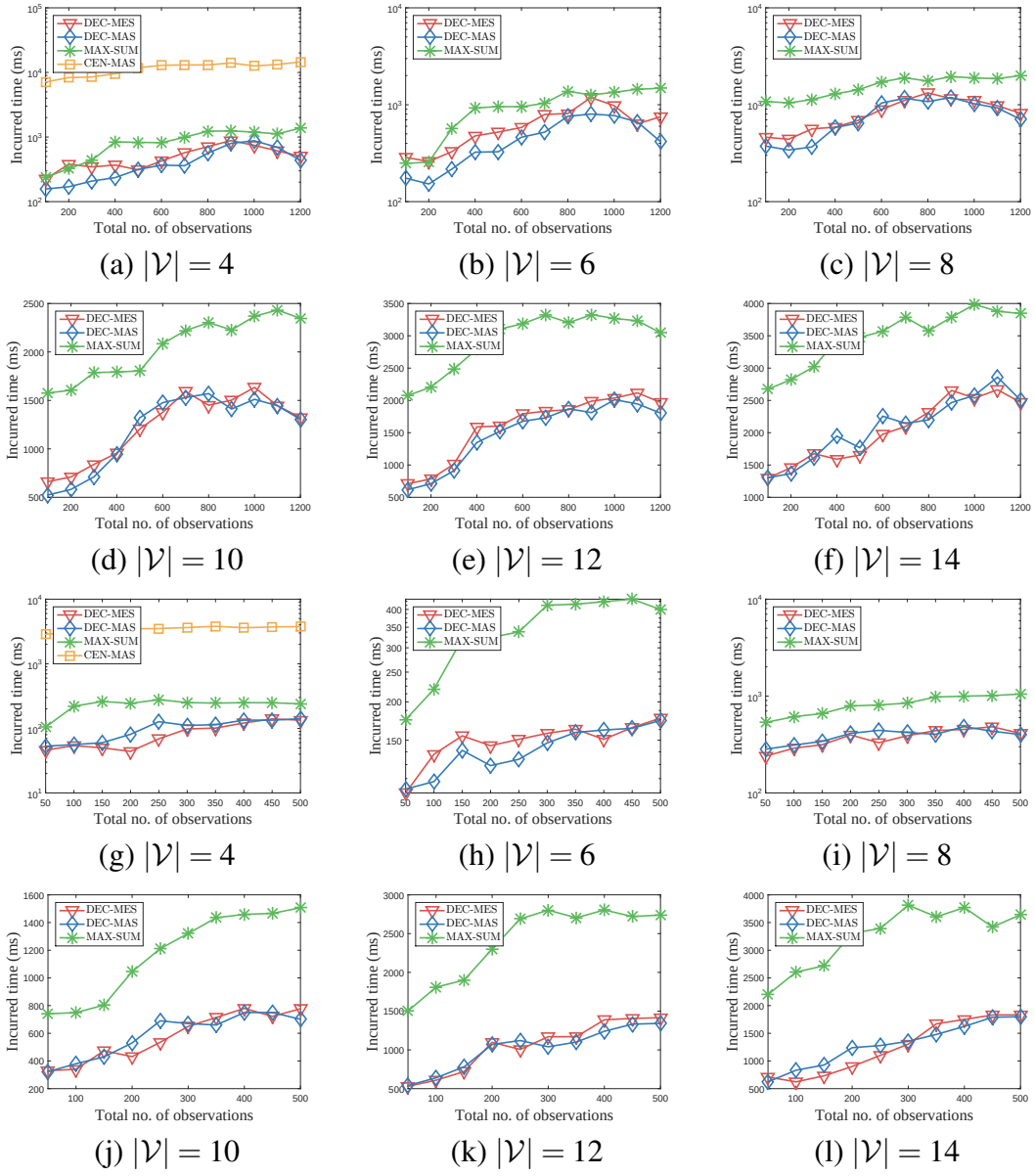
(III) DEC-MAS can achieve predictive performance comparable to that of MAX-SUM. Intuitively, MAX-SUM exploits and exchanges additional coordination information between robots in different connected components formed by DEC-MAS, but this results in little performance improvement of MAX-SUM over DEC-MAS. We will also see later that MAX-SUM is less computationally efficient and significantly less scalable than DEC-MAS in the number of robots.

C. Effect of decentralization on time efficiency and scalability.

Fig. 4.3 shows results of the incurred time of the tested algorithms with varying number of observations and robots. The observations are as follows:

(I) CEN-MAS incurs at least 1 order of magnitude more time than the decentralized algorithms for $|\mathcal{V}| = 4$ robots.

(II) DEC-MAS incurs computational time less than or comparable to that of DEC-MES: Even though DEC-MAS incurs additional time needed to estimate the component labels and compute the entropy of labels in (4.17), it saves time in the following aspects: (a) As mentioned previously in Section 4.1.2, DPM-GPs (i.e., used by DEC-MAS) offers the computational advantage over a single GP (i.e., used by DEC-MES) that each GP component only incurs cubic time in the size of the observations that are local to its corresponding



(a) $|\mathcal{V}| = 4$ (b) $|\mathcal{V}| = 6$ (c) $|\mathcal{V}| = 8$

(d) $|\mathcal{V}| = 10$ (e) $|\mathcal{V}| = 12$ (f) $|\mathcal{V}| = 14$

(g) $|\mathcal{V}| = 4$ (h) $|\mathcal{V}| = 6$ (i) $|\mathcal{V}| = 8$

(j) $|\mathcal{V}| = 10$ (k) $|\mathcal{V}| = 12$ (l) $|\mathcal{V}| = 14$

Fig. 4.3 Graphs of incurred time vs. total no. $|D|$ of observations gathered from (a-f) plankton density and (g-l) traffic phenomena by varying no. $|\mathcal{V}|$ of robots.

area of prediction instead of over the entire phenomenon; and (b) DEC-MAS tends to form smaller connected components than DEC-MES due to the structural property of DPM-GPs that requires two robots to coordinate their active sensing only when some pair of their next possible sampling units to be observed are associated with the same GP component (Section 4.3), and also due to its behavior of keeping the size η of the largest connected component small, as explained in Section 4.3.1.

(III) DEC-MAS is more time-efficient and significantly more scalable than MAX-SUM in the number of robots (Fig. 4.3) while achieving comparable predictive performance (Fig. 4.2). MAX-SUM is computationally more expensive because it has to process the additional coordination information between robots in different connected components formed by DEC-MAS that results in little performance improvement over DEC-MAS.

4.5 Conclusion

This chapter describes a novel DEC-MAS algorithm that can efficiently coordinate multiple robots in a partially decentralized manner to gather the most informative observations for predicting an unknown, non-stationary phenomenon. In particular, we demonstrate how its efficient decentralized coordination and theoretical performance guarantee can be realized by exploiting the structural property of DPM-GPs and the locality property of each stationary GP component. Empirical evaluation on two real-world datasets featuring non-stationary phenomena shows that (a) more superior active sensing performance can be achieved by optimizing our proposed MAS criterion (4.11) or (4.17) that trades off between observing sampling units with most uncertain component labels vs. that with most uncertain measurements given the current, imprecise estimate of their labels, and (b) DEC-MAS outperforms the decentralized MAX-SUM (Rogers et al., 2011) (DEC-MES (Chen et al., 2012)) algorithm in time efficiency and scalability (active sensing) while achieving comparable active sensing performance (time efficiency).

Chapter 5

Multi-Agent Coordination to Scale Up High Dimensional Bayesian Optimization

Motivated by the challenge of high dimensionality in the large-scale optimization problem, this chapter presents ANOVA-DCOP, a novel high dimensional Bayesian optimization method. In section 5.2, we introduce the ANOVA kernel and how it can be used to decompose the correlation structure in a high dimensional problem. Furthermore, in section 5.2.3, we utilize the summation structure created by ANOVA kernel and reformulate the problem as DCOP that can be efficiently solved by bounded max-sum which is a typical multi-agent coordination method. In section 5.3, we theoretically analyze the time complexity and regret bound in searching the optimum of the unknown function. Lastly in section 5.4, we demonstrate the imperial performance of ANOVA-DCOP on two analytical functions and one real-world financial problem.

5.1 Bayesian Optimization

The optimization problem in general is to maximize a function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subset [-1,1]^d$ is a compact domain. Many complex functions have no analytical expressions, so traditional

optimization methods such as gradient descent are not applicable here. In order to optimize an unknown function, conducting simulation is required. Bayesian optimization (BO) method is useful to optimize an unknown function only based on the observations (input-out put pairs of many simulations). Without losing generality, considering a maximization problem (the minimization problem can be easily transformed into a maximization problem.), Bayesian optimization method sequentially queries the function at some $x \in \mathcal{X}$ and obtain a noisy observation $y_x = f(x) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \eta^2)$ is Gaussian white noise. The maximum point $x_* = \text{argmax}_{x \in \mathcal{X}} f(x)$ is obtained by evaluating a belief over the original function based on the observations.

In BO framework, the unknown function is assumed to be distributed as a Gaussian process with zero mean function. Suppose at time step t, BO has already collected noisy observations $y_{D_{t-1}}$ in location set $D_{t-1} = x_{1:t-1}$ from time step 1 to $t-1$, then the unknown function's value at x is distributed by a posterior Gaussian distribution $\mathcal{N}(\mu_{x|D_{t-1}}, \Sigma_{xx|D_{t-1}})$ given all the observations $y_{D_{t-1}}$ with posterior mean $mu_{x|D_{t-1}}$ and posterior variance $\Sigma_{xx|D_{t-1}}$ as follows:

$$\begin{aligned} \mu_{x|D_{t-1}} &= \mu_x + \Sigma_{xD_{t-1}}(\Sigma_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}(y_{D_{t-1}} - \mu_{D_{t-1}}) \\ \Sigma_{xx|D_{t-1}} &= \Sigma_{xx} - \Sigma_{xD_{t-1}}(\Sigma_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}\Sigma_{D_{t-1}x} \end{aligned} \tag{5.1}$$

where η is the observation noise variance [1], $\mathbf{I}$ is an identity matrix, the covariance terms Σ_{xx}, $\Sigma_{D_{t-1}D_{t-1}}$ and $\Sigma_{xD_{t-1}}$ are computed using certain kernel functions such as squared exponential kernel function (Rasmussen and Williams, 2006):

$$\kappa(x, x') = \sigma_s^2 \exp\Big(-\frac{1}{2}\sum_{i=1}^{d}\big(\frac{x^{(i)} - x'^{(i)}}{\ell_i}\big)^2\Big) \tag{5.2}$$

where $x^{(i)}$ denotes the i-th component of the d-dimensional vector x, the set of hyperparameters $\theta = \{\sigma_s^2, \ell_1, ..., \ell_d\}$ are, respectively, signal variance and lengthscale in each dimension.

The posterior distribution tell us two kinds of information: a) the expected function value at a given input location x and b) the uncertainty on that location x. In order to choose the next

[1] The notation of the specified noise variance is different from previous two chapters because we want to avoid decomposing the noise when we decompose the kernel function.

sample point, many BO methods construct certain form of acquisition function according to the two kinds of information. We have already discussed three widely used acquisition functions in the literature review (Snoek et al., 2012): probability of improvement, expected improvement and GP-UCB. In this work, we focus on GP-UCB (Srinivas et al., 2009) which has the following form:

$$\varphi_t(x) = \mu_{x|D_{t-1}} + \beta_t^{1/2}\sigma_{x|D_{t-1}} \tag{5.3}$$

where $\sigma^2_{x|D_{t-1}} = \Sigma_{xx|D_{t-1}}$. Unlike the other two methods, GP-UCB is a parametric acquisition function that contains a parameter β_t which changes over time. When β_t is large, GP-UCB tends to explore the highly uncertain area in the input space while when β_t is small, it tends to exploit the maximum value of the current expectation of the unknown function.

Algorithm 2 illustrates the general BO procedure. In literature, many BO methods are proposed for addressing optimization problems in different scenarios. In order to compare the performance of those methods, the instantaneous regret $r_t = f(x_*) - f(x_t)$ is introduced which leads to two commonly used performance measures (Kirthevasan et al., 2015): cumulative regret $R_T = \sum_{t=1}^{T} r_t$ and simple regret $S_T = \min_{t \leq T} r_t$.

Algorithm 2: Bayesian optimization procedure

for $t = 1, 2, ..., T$ **do**
- find $x_t = \text{argmax}_{x \in \mathcal{X}} \mu_{x|D_{t-1}} + \beta_t^{1/2}\sigma_{x|D_{t-1}}$
- query $y_{x_t} = f(x_t) + \varepsilon$
- collect point $D_t \leftarrow D_{t-1} \bigcup x_t$
- collect observation $y_{D_t} \leftarrow y_{\{D_{t-1} \cup x_t\}}$

5.2 ANOVA-DCOP

One challenging scenario of using BO arises when there is a need to optimize an unknown function in high dimensions (Djolonga et al., 2013, Kirthevasan et al., 2015, Wang et al., 2016). To address this challenge by learning a sparse correlation structure in the dimensions, we propose ANOVA-DCOP, a novel BO method to optimize such high dimensional function.

The method consists of two component: the ANOVA kernel function used in the poster Gaussian distribution and DCOP, the reformulation of the acquisition function from multi-agent perspective. These two components scale up the optimization problem to hundreds of dimensions.

5.2.1 High Dimensionality

In order to scale up optimization problems to high dimensions, we need to exploit the sparse correlation structure in the dimensions. ANOVA-DCOP relies on a key structural assumption: a d-dimensional objective function can be decomposed into a summation of sub-functions of all possible subsets of the dimensions as follows:

$$f(x) = \sum_{\mathcal{I} \in 2^{\mathcal{S}}} f_{\mathcal{I}}(x^{\mathcal{I}}) \tag{5.4}$$

where $\mathcal{S} = \{1, 2, ..., d\}$ is the dimension index set and $2^{\mathcal{S}}$ is the super set of $\mathcal{S}$ for recording all possible subsets of dimensions. Denote I to be one possible subsets of the dimensions and $x^{\mathcal{I}} = \bigcup_{i \in \mathcal{I}} x^{(i)}$ where $x^{(i)}$ is the i-th dimensional element of input x. If $\mathcal{I} = \emptyset$, $f_{\mathcal{I}}(x^{\mathcal{I}}) = f_0$. One simple example of the function can be $f(x) = 1 + x^{(1)} + x^{(2)} + x^{(1)}x^{(2)}$.

Under this structural assumption, we introduce ANOVA kernel function (Durrande et al., 2013) with the motivation of decomposing the correlation structure in the dimensions:

$$\kappa(x, x') = \prod_{i=1}^{d} (1 + \kappa_i(x^{(i)}, x'^{(i)})) \tag{5.5}$$

where $\kappa_i(x^{(i)}, x'^{(i)})$ is a base kernel function that can be any type of kernel function (Rasmussen and Williams, 2006) on i-th dimension.

Accordingly, with this ANOVA kernel function, the posterior Gaussian distribution at x given observations $y_{D_{t-1}}$ is derived as:

$$\begin{aligned} \mu_{x|D_{t-1}} &= \mu_x + \kappa(x, D_{t-1})(\Sigma_{D_{t-1}D_{t-1}} + \eta^2 \mathbf{I})^{-1}(y_{D_{t-1}} - \mu_{D_{t-1}}) \\ \Sigma_{xx|D_{t-1}} &= \kappa(x, x) - \kappa(x, D_{t-1})(\Sigma_{D_{t-1}D_{t-1}} + \eta^2 \mathbf{I})^{-1}\kappa(D_{t-1}, x) \end{aligned} \tag{5.6}$$

where $\Sigma_{xx} = \kappa(x,x)$ and $\kappa(x,D_{t-1}) = [\kappa(x,x'_1),...,\kappa(x,x'_{t-1})]$ is a $t-1$ dimensional row vector where $x'_j \in D_{t-1}$.

After some algebra, we can observe that ANOVA kernel function can be rearranged as a summation form with respect to the subsets of dimensions which matches the structural assumption we made on $f(x)$.

Proposition 3. *An ANOVA kernel function can be decomposed into a list of kernel functions:*

$$\kappa(x,x') = \prod_{i=1}^{d}(1+\kappa_i(x^{(i)},x'^{(i)})) = \sum_{I\in 2^{\mathcal{S}}}\prod_{i\in I}\kappa_i(x^{(i)},x'^{(i)}) = \sum_{I\in 2^{\mathcal{S}}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},x'^{\mathcal{I}}) \tag{5.7}$$

where

$$\kappa_{\mathcal{I}}(x^{\mathcal{I}},x'^{\mathcal{I}}) = \begin{cases} 1 & \text{if } \mathcal{I}=\emptyset \\ \prod_{i\in\mathcal{I}}\kappa_i(x^{(i)},x'^{(i)}) & \text{if } \mathcal{I}\neq\emptyset \end{cases}$$

Further, if we assume a Gaussian process prior with zero mean function on $f(x)$, the prior mean can also be decomposed into the subsets of dimensions:

$$\mu_x = \sum_{\mathcal{I}\in 2^{\mathcal{S}}}\mu_{x^{\mathcal{I}}} \tag{5.8}$$

where

$$\mu_{x^{\mathcal{I}}} = 0 \text{ for } \mathcal{I}\in 2^{\mathcal{S}}$$

With the decomposed form of ANOVA kernel function, we can rewrite the posterior Gaussian distribution $\mathcal{N}(\mu_{x|D_{t-1}},\Sigma_{xx|D_{t-1}})$ at any step t of $f(x)$ with posterior mean and posterior variance as:

$$\begin{aligned} \mu_{x|D_{t-1}} &= \textstyle\sum_{\mathcal{I}\in 2^{\mathcal{S}}}\left(\mu_{x^{\mathcal{I}}} + \kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\Sigma_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}(y_{D_{t-1}}-\mu_{D_{t-1}})\right) \\ \Sigma_{xx|D_{t-1}} &= \textstyle\sum_{\mathcal{I}\in 2^{\mathcal{S}}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},x^{\mathcal{I}}) - \sum_{\mathcal{I}\in 2^{\mathcal{S}}}\sum_{\mathcal{I}'\in 2^{\mathcal{S}}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\Sigma_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}'}(D_{t-1}^{\mathcal{I}'},x^{\mathcal{I}'}) \end{aligned} \tag{5.9}$$

where $\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}}) = [\kappa_{\mathcal{I}}(x^{\mathcal{I}},x'^{\mathcal{I}}_1),...,\kappa_{\mathcal{I}}(x^{\mathcal{I}},x'^{\mathcal{I}}_{t-1})]$ is a t dimensional row vector where $x'^{\mathcal{I}}_j \in D_{t-1}^{\mathcal{I}}$ and $D_{t-1}^{\mathcal{I}}$ is a set of all the observations' with dimensions in $\mathcal{I}$.

ANOVA kernel function enumerates sub-kernel functions of all the possible subsets of dimensions. To compute the posterior distribution of a Gaussian process with ANOVA kernel, we need to sum up 2^d terms for the posterior mean and 2^{2d} terms for the posterior variance. It is therefore prohibitively expensive when the input dimension is large. Fortunately, in many real-world, high-dimensional problems, the input parameters are usually sparsely correlated. One parameter may be only strongly correlated with a few other parameters instead of being strongly correlated with all the rest parameters. Hence, we make the following assumption on sparse correlation structure in the dimensions:

Assumption 1. *At any time step $t \geq 1$, the correlation structure between two measurements at input x and x' given the observations $y_{D_{t-1}}$ can be expanded into a summation of correlations with subsets of dimensions with size up to k.*

Type	Correlation
subsets with size 0	1
subsets with size 1	$\kappa_i(x^{(i)}, x'^{(i)})$ for $i = 1,2,3,4$
subsets with size 2	$\prod_{i \in I} \kappa_i(x^{(i)}, x'^{(i)})$ for $I \in \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\}$
subsets with size 3	$\prod_{i \in I} \kappa_i(x^{(i)}, x'^{(i)})$ for $I \in \{\{1,2,3\},\{1,2,4\},\{2,3,4\}\}$

Table 5.1 Demonstration of sparse correlation structure in four dimensional input.

Let us explain the meaning of the assumption with an example. Suppose the input of a system contains four parameters $x = (x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$, if we choose $k = 3$, the correlations between measurements at x and x' are restricted within the correlations in the subsets of dimensions as illustrated in table 5.1. All the correlations with dimension size larger than k will assume to be zero.

When the true underlying correlation structure is sparse, we can choose a small $k << d$ to significantly reduce the redundant computation introduced by ANOVA kernel function. In this work, we choose $k = 2$ (only consider up to pairwise correlations) in order to fit in the specific multi-agent coordination method we use. However, our proposed method is not restricted to $k = 2$ case by applying more general multi-agent coordination method.

When $k = 2$, we are actually truncating ANOVA kernel function as:

$$\begin{aligned}\widetilde{\kappa}(x,x') &= 1+\sum_{i=1}^{d}\kappa(x^{(i)},x'^{(i)})+\sum_{1\le i<j\le d}^{d}\kappa_i(x^{(i)},x'^{(i)})\kappa_j(x^{(j)},x'^{(j)}) \\ &= \sum_{\mathcal{I}\in\mathcal{U}}\prod_{i\in\mathcal{I}}\kappa_i(x^{(i)},x'^{(i)}) \\ &= \sum_{\mathcal{I}\in\mathcal{U}}\kappa_I(x^{\mathcal{I}},x'^{\mathcal{I}})\end{aligned} \tag{5.10}$$

where $\mathcal{U} = \{\{\emptyset\}\bigcup\mathcal{S}\bigcup\{\{i,j\}|i\in\mathcal{S},j\in\mathcal{S},i\neq j\}\}$ is the indices set with terms contain less than or equal to two dimensions.

Accordingly, at time step t, $f(x)\sim\mathcal{N}(\mu_{x|D_{t-1}},\Sigma_{xx|D_{t-1}})$ where $\mu_{x|D_{t-1}}$ and $\Sigma_{xx|D_{t-1}}$ have the following form:

$$\begin{aligned}\mu_{x|D_{t-1}} &= \sum_{\mathcal{I}\in\mathcal{U}}\mu_{x_{\mathcal{I}}}+\widetilde{\kappa}(x,D_{t-1})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}(y_{D_{t-1}}-\mu_{D_{t-1}}) \\ &= \sum_{\mathcal{I}\in\mathcal{U}}\left(\mu_{x_{\mathcal{I}}}+\kappa_I(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}(y_{D_{t-1}}-\mu_{D_{t-1}})\right) \\ \Sigma_{xx|D_{t-1}} &= \sum_{\mathcal{I}\in\mathcal{U}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},x^{\mathcal{I}})-\widetilde{\kappa}(x,D_{t-1})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}\widetilde{\kappa}(D_{t-1},x) \\ &= \sum_{\mathcal{I}\in\mathcal{U}}\kappa_I(x^{\mathcal{I}},x^{\mathcal{I}})-\sum_{\mathcal{I}\in\mathcal{U},\mathcal{I}'\in\mathcal{U},\mathcal{I}=\mathcal{I}'}\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}'}(D_{t-1}^{\mathcal{I}'},x^{\mathcal{I}'})\end{aligned} \tag{5.11}$$

where $\widetilde{\Sigma}^{-1}_{D_{t-1}D_{t-1}}$ is the covariance matrix computed using the truncated ANOVA kernel function. Here in the second step of the posterior covariance in equation 5.11 is derived by applying the assumption 1 we made above: At any time step t, the correlation structure between two measurements at input x and x' given the observations $y_{D_{t-1}}$ can be expanded into a summation of correlations with subsets of dimensions with size up to $k = 2$. When $\mathcal{I}\neq\mathcal{I}'$, the posterior covariance will introduce terms involving more than two dimensions which has value zero in the correlation under assumption 1.

In the Gaussian process with truncated ANOVA kernel function, there are $\mathcal{O}(d^2) = 1+d+\frac{d(d-1)}{2}$ instead of $\mathcal{O}(2^d)$ terms that need to be summed up to compute the posterior mean and the posterior variance. Therefore the time complexity is reduced from exponential to polynomial in d, which presents a significant speed-up in terms of the computing time.

It is interesting that $\mu_{x|D_{t-1}}$ and $\Sigma_{xx|D_{t-1}}$ are linear decomposable under Assumption 1. To utilize the linear decomposable form later, we define the following two terms associated with each sub-function $f_{\mathcal{I}}(x^{\mathcal{I}})$ and make the following assumption on $f_{\mathcal{I}}(x^{\mathcal{I}})$:

Assumption 2. *Define* $\mu_{x|D_{t-1}^{\mathcal{I}}}$ *and* $\Sigma_{xx|D_{t-1}^{\mathcal{I}}}$ *as:*

$$\begin{aligned} \mu_{x^{\mathcal{I}}|D_{t-1}} &= \mu_{x^{\mathcal{I}}} + \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}(y_{D_{t-1}} - \mu_{D_{t-1}}) \\ \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}} &= \kappa_{\mathcal{I}}(x^{\mathcal{I}}, x^{\mathcal{I}}) - \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}}(D_{t-1}^{\mathcal{I}}, x^{\mathcal{I}}) \end{aligned} \tag{5.12}$$

Then, we assume that at time step t, $f(x^{\mathcal{I}}) \sim \mathcal{N}(\mu_{x^{\mathcal{I}}|D_{t-1}}, \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}})$.

With assumption 2, we can construct a relationship between decomposition of $\mathcal{N}(\mu_{x|D_{t-1}}, \Sigma_{xx|D_{t-1}})$ and $\mathcal{N}(\mu_{x^{\mathcal{I}}|D_{t-1}}, \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}})$:

Proposition 4. *Under Assumption 1, at each time step t, the original function* $f(x)$ *at* x *is distributed as* $\mathcal{N}(\mu_{x|D_{t-1}}, \Sigma_{xx|D_{t-1}})$ *with truncated ANOVA kernel. It can be decomposed into a summation of terms* $f_{\mathcal{I}}(x^{\mathcal{I}})$ *which is distributed as* $\mathcal{N}(\mu_{x^{\mathcal{I}}|D_{t-1}}, \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}})$ *defined in Assumption 2. So we have the following equality:*

$$\begin{aligned} \mu_{x|D_{t-1}} &= \textstyle\sum_{\mathcal{I}\in\mathcal{U}} \mu_{x^{\mathcal{I}}|D_{t-1}} \\ \Sigma_{xx|D_{t-1}} &= \textstyle\sum_{\mathcal{I}\in\mathcal{U}} \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}} \end{aligned} \tag{5.13}$$

The proof is in appendix C.1.

5.2.2 Acquisition Function

Bayesian optimization methods utilize an acquisition function to integrate the information in posterior mean and posterior variance. One widely used acquisition function is GP-UCB with the following form:

$$\varphi_t(x) = \mu_{x|D_{t-1}} + \beta_t^{1/2}\sqrt{\sigma^2_{x|D_{t-1}}}. \tag{5.14}$$

Using the truncated ANOVA kernel function, GP-UCB can be decomposed as:

$$\varphi_t(x) = \Big(\sum_{\mathcal{I}\in U} \mu_{x^{\mathcal{I}}|D_{t-1}}\Big) + \beta_t^{1/2}\sqrt{\sum_{\mathcal{I}\in\mathcal{U}} \sigma^2_{x^{\mathcal{I}}|D_{t-1}}}. \tag{5.15}$$

The motivation of using the ANOVA kernel is to decomposing the acquisition function into a summation structure which can be casted as a decentralized constrained optimization problem (DCOP). However, the term $\sqrt{\sum_{\mathcal{I}\in\mathcal{U}}\sigma^2_{x^{\mathcal{I}}|D_{t-1}}}$ cannot be decomposed directly.

In order to generate a summation structure, we approximate the term $\sqrt{\sum_{\mathcal{I}\in\mathcal{U}}\sigma^2_{x^{\mathcal{I}}|D_{t-1}}}$ in equation 5.15 by applying Cauchy inequality $\sqrt{\sum_{\mathcal{I}\in\mathcal{U}}\sigma^2_{x^{\mathcal{I}}|D_{t-1}}} \leq \sum_{\mathcal{I}\in\mathcal{U}}\sqrt{\sigma^2_{x^{\mathcal{I}}|D_{t-1}}}$. With this approximation, we propose the following acquisition function:

$$\widetilde{\varphi}_t(x) = \sum_{\mathcal{I}\in\mathcal{U}}\left(\mu_{x^{\mathcal{I}}|D_{t-1}} + \beta_t^{1/2}\sigma_{x^{\mathcal{I}}|D_{t-1}}\right) = \sum_{\mathcal{I}\in\mathcal{U}}\widetilde{\varphi}_t(x^{\mathcal{I}}) \tag{5.16}$$

This acquisition function has a summation structure that can be cast as a DCOP. Although we set $k=2$ in our derivation, but this acquisition function can be generalized to any $k \leq d$. Perhaps surprisingly, if we set $k=1$, it reproduces the acquisition function used in additive model (Kirthevasan et al., 2015) in which they assume that all the dimensions are mutually independent. Our work in fact loose their assumption by capturing the correlations in subsets of dimensions with size up to k. To address the high dimensional optimization problem, pairwise correlations usually are the most important ones. The correlations involving more dimensions can be approximated by pairwise correlations. So in practice, we focus on $k=2$ case. And in this special case, the result-in DCOP can be solved efficiently by a robust multi-agent coordination method named bounded max-sum.

5.2.3 Bounded Max-Sum

Distributed Constraint Optimization Problem (DCOP)(Shoham and Leyton-Brown, 2008) is a quadruple $\mathcal{P} = (\mathcal{A}, \mathcal{V}, \Omega, \mathcal{F})$ where $\mathcal{A}$ is a set of agents, $\mathcal{V} = \{x^{(1)}, ..., x^{(d)}\}$ and $\Omega = \{\Omega^{(1)}, ..., \Omega^{(d)}\}$ are the input variables and domain. $\mathcal{F} = \{f_j\}_{j=1}^{|\mathcal{F}|}$ is a set of functions. The variable and function forms a bipartite factor graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$. Denote $\mathcal{M}_i$ to be an index indicating which function nodes are connected to variable $x^{(i)}$ and $\mathcal{N}_j$ is an index indicating which variable nodes are connected to function g_j. Let $x^{\mathcal{N}_j}$ to be the scope of the function g_j so that if $x^{(i)}$ is connected to g_j, then $g_j(x^{\mathcal{N}_j})$ contains $x^{(i)}$.

To solve this problem, each variable node and function node in the bipartite factor graph is assigned to a computational agent. The agents search for the maximum value via decentralized coordination. Each agent is only able to control its own node and can directly communicate with neighboring agents. Two agents are consider to be neighbors if there is an edge connecting the node that the agents control.

Max-sum (Kim and Lesser, 2013) is a message passing algorithm for solving DCOP. Two types of messages are passing along the edges on the bipartite factor graph. Suppose an edge $e_{ij} \in \mathcal{E}$ connects a variable $x^{(i)}$ and function $g_j(x^{\mathcal{N}_j})$, these messages are defined as follows:

- From variable to function:

$$q_{i\to j}(x^{(i)}) \quad = \quad C_{ij} + \sum_{k\in\mathcal{M}_i\setminus j} r_{k\to i}(x^{(i)}) \tag{5.17}$$

  where C_{ij} is a normalizing constant to prevent the messages from increasing endlessly in cyclic graphs. Its value is set by satisfying the following equation:

$$\sum_{\mathcal{M}_i} q_{i\to j}(x^{(i)}) \quad = \quad 0 \tag{5.18}$$

- From function to variable:

$$r_{j\to i}(x^{(i)}) = \max_{x^{\ell}\setminus x^{(i)}} \{g_j(x^{\ell}) + \sum_{k\in\mathcal{N}_j\setminus i} q_{k\to i}(x^{(i)})\} \tag{5.19}$$

Max-Sum is a distributed synchronous algorithm, since the agent controlling node $x^{(i)}$ has to wait to receive messages from all its neighbors but f_j, to be able to compute its message to f_j. When the graph has no cycle, this algorithm is guaranteed to converge to global optimal solution (Rollon and Larrosa, 2012). After the convergence, each variable node can compute the function $z_i(x^{(i)}) = \max_{x^{(i)}} \sum_{k\in\mathcal{M}_i} r_{k\to i}(x^{(i)})$. The optimal assignment is $x^{(i)}_* = \text{argmax}_{x^{(i)}} z_i(x^{(i)})$.

In the context of ANOVA-DCOP, each variable node is a dimension of the input of the unknown function. Its domain $\mathcal{M}_i = \Omega^{(i)}_t$ at time step t. The function node is a little bit

tricky. In order to evenly distributed the terms of acquisition function to each function node, we design the following function to be assigned to a function node that connects two variable nodes as equation 5.20. Its domain $\mathcal{M}_j = \Omega_t^{\mathcal{I}}$ which is the combination of domain $\Omega_t^{(i)}$ and $\Omega_t^{(k)}$ at time step t.

$$g_j(x^{(i)}, x^{(k)}) = \widetilde{\varphi}_t(x^{(i)}, x^{(k)}) + \frac{1}{d-1}(\widetilde{\varphi}_t(x^{(i)}) + \widetilde{\varphi}_t(x^{(k)})) \tag{5.20}$$

Although we only consider correlation structure with $k = 2$, it may still create cycle in the graph. As a result, we need to apply bounded max-sum (Rollon and Larrosa, 2012) algorithm to remove certain edges by building a maximum spanning tree $\mathcal{T}$ based on the weights of the correlations. The weights are computed as:

$$w_{ij} = \max_{x^{(k)}} \{ \max_{x^{(i)}} g_j(x^{(i)}, x^{(k)}) - \min_{x^{(i)}} g_j(x^{(i)}, x^{(k)}) \} \tag{5.21}$$

If the edge e_{ij} is not in the maximum spanning tree $\mathcal{T}$, the function $g_j(x^{(i)}, x^{(k)})$ is transformed into:

$$\widetilde{g}_j(x^{(k)}) = \min_{x^{(i)}} g_j(x^{(i)}, x^{(k)}) \tag{5.22}$$

According to the the spanning tree, the objective function becomes:

$$g(x) = \sum_{e_{ij} \in \mathcal{T}, e_{ij} \in \mathcal{T}} g_j(x^{(i)}, x^{(k)}) + \sum_{e_{ij} \notin \mathcal{T}} \widetilde{g}_j(x^{(k)}) \tag{5.23}$$

The acquisition function of ANOVA-DCOP is distributed to function nodes as subfunction $g_j(\cdot, \cdot)$. Bounded max-sum optimizes the DCOP with objective function $g(x)$ with a summation form of $g_j(\cdot, \cdot)$. The result of bounded max-sum is exactly the optimum of the acquisition function of ANOVA-DCOP after applying the maximum spanning tree in the graph. By iteratively taking new observation and solving the DCOP problem associated with the updated acquisition function, the optimum of acquisition function will be more and more close to the optimum of original function so that the optimum of the original function can be

found eventually. The entire procedure of our proposed ANOVA-DCOP method is described in Alg. 3.

Algorithm 3: Multi-agent Bayesian optimization procedure

provide an initial spanning tree $\mathcal{T}$.
for $t = 1, 2, ..., T$ **do**
  while *not converge* **do**
    build discretized domain Ω_t.
    update variable domain $\mathcal{M}_i = \Omega_t^{(i)}$.
    update function domain $\mathcal{N}_j = \Omega_t^{\mathcal{I}}$.
    for *variable node* $x^{(i)}$ **do**
      collect message $r_{j\to i}(x^{(i)})$ from $\mathcal{M}_i$
      produce message $q_{i\to j}(x^{(i)})$ using message from $\mathcal{M}_i \backslash j$
    for *function node* g_j **do**
      collect message $q_{i\to j}(x^{(i)})$ from $\mathcal{N}_j$
      produce message $r_{j\to i}(x^{(i)})$ using message from $\mathcal{N}_j \backslash i$
  for *variable node* $x^{(i)}$ **do**
    find $x_*^{(i)} = \text{argmax}_{x^{(i)}} z_i(x^{(i)})$
  combine the result as x_t
  query $y_{x_t} = f(x_t) + \varepsilon$ through simulation
  collect point $D_t \leftarrow D_{t-1} \bigcup x_t$
  collect observation $y_{D_t} \leftarrow y_{\{D_{t-1} \bigcup x_t\}}$
  compute the weights of edges on the graph using Eqn. 5.21
  update spanning tree $\mathcal{T}$
  update function nodes using Eqn. 5.23.

5.3 Theoretical Analysis of ANOVA-DCOP

GP-UCB is a parametric acquisition function in works of BO. The parameter β_t will balance the exploration and exploitation in optimizing the unknown function. In this section, we will demonstrate that by setting a suitable value of β_t, we can theoretically bound the regret of ANOVA-DCOP. Our analysis is based on the following two assumptions:

Assumption 3. *Let $f(x)$ be sampled from a zero-mean Gaussian process (see Section X) with kernel $\widetilde{\kappa}(x,x') \triangleq \sum_{\mathcal{I}\in\mathcal{U}} \kappa_{\mathcal{I}}(x^{\mathcal{I}}, x'^{\mathcal{I}})$. We assume (a) the kernel $\widetilde{\kappa}(\cdot,x)$ is L-Lipschitz for*

every x and (b) there exists constants $a, b > 0$ so that for every $x^{\mathcal{I}}$, it follows that

$$\Pr\left(\sup_x \left|\frac{\partial f(x)}{\partial x^{\mathcal{I}}}\right| > J\right) \leq a\exp\left(-\frac{J^2}{b^2}\right). \tag{5.24}$$

Assumption 4. *Let $x_t \in \Omega_t$ denotes the selected sampling location of our algorithm at time step $t \geq 1$, we assume that x_t is $\zeta_0 t^{-1/2}$-optimal which implies $0 \leq \widetilde{\varphi}_t(\widetilde{x}_t) - \widetilde{\varphi}_t(x_t) \leq \zeta_0 t^{-1/2}$ where $\widetilde{x}_t$ denotes the true maximizer of the acquisition $\widetilde{\varphi}_t(x)$ over the entire input space $\mathcal{X}$.*

Under these two assumptions and theorem 4 in Srinivas et al. (2010), the cumulative regret of ANOVA-DCOP can be bounded as Theorem 3 as follows:

Theorem 3. *Given $\delta \in (0,1)$, the cumulative regret $R_T \triangleq (1/T)\sum_{t=1}^{T}(f(x_*) - f(x_t))$ of our algorithm can be upper-bounded by $C_2(a,b,\mathcal{U},T,L,\delta,\zeta_0,\eta) + 2\beta_T^{1/2}T^{-1/2}\sqrt{C_1|\mathcal{U}|\gamma_T}$ where γ_T is the maximum information gain after T sampling steps as defined in Srinivas et al. (2010), $C_1 = 1/\log(1+\eta^{-2})$, $\beta_T = 2k\log\left(dT^3\right) + 2\log\left(3|\mathcal{U}|\pi_T/\delta\right)$ and $C_2(a,b,\mathcal{U},T,L,\delta,\zeta_0,\eta)$ is a constant that only depends on a, b, T, L, δ, ζ_0, η and $|\mathcal{U}|$.*

This theoretical result (Proof in Appendix 3) shows that by applying truncated ANOVA kernel function with k, if we set $\beta_t \in \mathcal{O}(2k\log dt^3)$, for a optimization problem with dimension index $\mathcal{U}$ constructed from d dimensions, with high probability, we can have a upper bound within T time steps with respect to the maximum information gain γ_T. If we choose $k = 1$ and $r = 1$, the regret upper bound holds with the same complexity in the additive model (Kirthevasan et al., 2015). If we assume more complex structure on the function $f(x)$ by increasing k, the regret bound grows linearly with $\sqrt{\mathcal{U}}$.

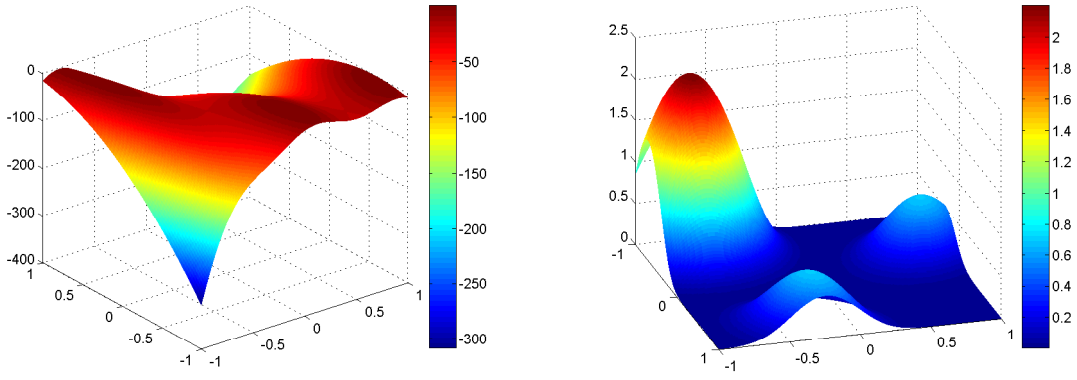
We analyze the complexity of the multi-agent collaboration as follows. Suppose we set $k = 2$ and denote $|\omega_m|$ is the maximum granularity for all time steps over dimensions $\mathcal{I} \in \mathcal{U}$. The original searching space for BO is $\mathcal{O}(\omega_m^d)$. With ANOVA kernel function, we can reduce the searching space to $\mathcal{O}(\omega_m^2)$ for each agent solving the DCOP. The searching space is significantly reduced. However, Our ANOVA-DCOP method incurs more time than original BO with time complexity $\mathcal{O}(|D_t|^3)$ in time step t. ANOVA-DCOP requires $\mathcal{O}(d\log d)$ time to build the maximum spanning tree and $\mathcal{O}(d^2)$ to compute the weights of the edges. So the total time complexity in time step t is $\mathcal{O}(d\log d + d^2 + H|D_t|^3)$ for H iterations. The extra

running time is bearable when the simulation is time-consuming, so our method is suitable when the simulation is computationally expensive or consumes lots of resources.

5.4 Experiments

In this section, we imperially evaluate ANOVA-DCOP by comparing with existing high dimensional Bayesian optimization methods in the literature such as original GP-UCB method (Srinivas et al., 2010), random embedding (Wang et al., 2016), subspace learning (Djolonga et al., 2013) and additive model (Kirthevasan et al., 2015). The performance metric we use is the simple regret $S_T = \min_{t \leq T} r_t$.

5.4.1 Analytic Function



(a) Branin (b) Logsum

Fig. 5.1 Analytic functions to be tested. The input is scaled to $[-1,1] \times [-1,1]$. The output is negated for maximization problem. Branin has maximum value -0.397887 at $(-0.75221, 0.63667)$, $(0.08555, -0.69665)$ and $(0.9233, -0.67)$. Logsum has maximum value 2.1972 at $(-0.3, 0.8)$.

We firstly test the methods on two analytic functions: Brainin function and Logsum function as in Fig. 5.1. The reason we choose these two functions is to demonstrate the performance of various BO methods under different structural assumptions on the function. Branin is a benchmark function for optimization which has been tested in many previous

works in literature. The original Branin function has the following form with two-dimensional input:

$$f(x) = (x^{(2)} - \frac{5.1{x^{(1)}}^2}{4\pi^2} + \frac{5x^{(1)}}{\pi} - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x^{(1)}) + 10 \tag{5.25}$$

where the input $(x^{(1)}, x^{(2)})$ is restrained to $[-5, 10] \times [0, 15]$. In the experiment, we scale the input to $[-1, 1] \times [-1, 1]$ and negate the output for maximization problem. The two-dimensional input is projected onto 20, 60 and 100 dimensional space by multiplying a d by 2 random matrix, where each element of the matrix is sampled from a standard Gaussian distribution. As a result, the high dimensional function we create actually only has intrinsic two dimensions. This structural assumption favors the methods which can reduce the high dimensional input to low dimensions such as subspace learning (Djolonga et al., 2013).

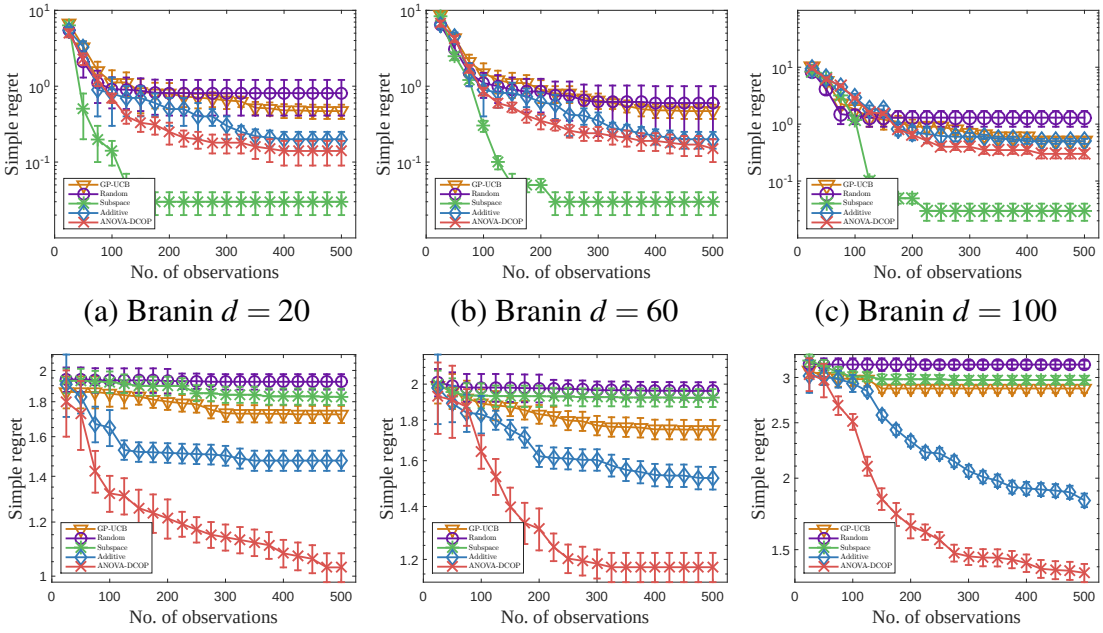
This dimension projection structure does not highlight our contribution by exploring the correlation structure between the dimensions. So we design the Logsum function that favors our method. It is a d-dimensional function which assumes each dimension is only correlated with its adjacent two dimensions:

$$\begin{aligned} f(x) &= \tfrac{1}{d-1}\textstyle\sum_{1 \le i < j \le 1} \log(1 + g(x^{(i)}, x^{(j)})) \\ g(x^{(i)}, x^{(j)}) &= 8e^{-\frac{1}{2}v_1^\top \Sigma_1^{-1} v_1} + e^{-\frac{1}{2}v_2^\top \Sigma_2^{-1} v_2} + e^{-\frac{1}{2}v_3^\top \Sigma_3^{-1} v_3} \\ v_k &= (x^{(i)} - \mu_k^{(i)}, x^{(j)} - \mu_k^{(j)})^\top \end{aligned} \tag{5.26}$$

where $\Sigma_1 = \begin{bmatrix} 0.05 & -0.04 \\ -0.04 & 0.05 \end{bmatrix}$, $\mu_1 = \begin{bmatrix} -0.3 \\ 0.8 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 0.05 & 0.04 \\ 0.04 & 0.05 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 0.8 \\ -0.3 \end{bmatrix}$, $\Sigma_3 = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -0.7 \\ -0.7 \end{bmatrix}$.

In the experiment, we use square exponential kernel function for all the other methods and for the base function of ANOVA-DCOP. The signal variance in the hyperparameters of the kernel function and noise variance is prefixed to 10% and 1% of the output value range of the testing function. We learn the lengthscale of each dimension in the hyperparameters every 50 iterations through maximizing the marginal likelihood of the realized Gaussian

process given the observations. The coefficient β_t is set to $\mathcal{O}(4\log(dt^3))$ for ANOVA-DCOP. For other methods, to the best of our knowledge, we use the settings in their original papers. For subspace learning method, the number of intrinsic dimensions is set to the true intrinsic dimensions 2 for Branin and 5 for Logsum. We presampled 20 observations to build the initial maximum spanning tree for ANOVA-DCOP, and this maximum spanning tree is updated every 50 iteration. These 20 initial observations are included in all the other methods for fair comparison.



(a) Branin $d = 20$ (b) Branin $d = 60$ (c) Branin $d = 100$

(d) Logsum $d = 20$ (e) Logsum $d = 60$ (f) Logsum $d = 100$

Fig. 5.2 Simple regret of BO methods tested two analytic functions (Branin and Logsum) within 500 time steps.

Fig. 5.2 shows the simple regret of different BO methods within 500 iterations. The top three subfigures are the results tested on Branin function and bottom three subfigures are the results tested on Logsum function. We can clearly see the difference in the performance on the two functions due to their different structural assumption.

The high dimensional input in Branin is constructed from dimension projection. Subspace learning method can learn the correlation between intrinsic dimensions and we serve the method with the true intrinsic number of dimensions. Therefore with a few observations,

it can quickly find the maximum point. On the other hand, Random embedding method, at each time step, generates a random matrix to do dimension projection which may not match to the true intrinsic dimensions. As a result, it performs not so well in our settings. Our proposed ANOVA-DCOP method and additive model have similar acquisition function. The difference is that additive model assumes all the dimensions are independent with each other while ANOVA-DCOP assumes a pairwise correlation between the dimensions. The additional information on the correlation reduces the number of iterations in searching the space. And both methods significantly reduce the searching space compare to the original GP-UCB method which results in a lower simple regret than GP-UCB.

Different from Branin, the high dimensional input in Logsum has pairwise correlations. We purposely construct the function to have no reduced subspace so both random embedding method and subspace learning method cannot successfully reduce the regret. On the other hand, the additive model has a much smaller searching space so that within the limited iteration, it outperforms the original GP-UCB method. Similarly, the pairwise correlation structure in the dimensions is easy for ANOVA-DCOP to learn. Hence, it can use the additional correlation information to reduce the regret faster than additive model.

From the experiments of two analytical functions, we can see that if there exists sparse correlation structure in the dimensions, ANOVA-DCOP can effectively find the maximum point with a few observations. But the problem is that we cannot always have such a nice structure. In the next subsection, we will evaluate the performance of ANOVA-DCOP on real application with unknown correlation structures.

5.4.2 Trading Strategy Optimization

In this section, we apply ANOVA-DCOP on a trading strategy optimization problem. We focus on a popular alpha trading strategy (Tortoriello, 2009) with many factors chosen by the investors. Every trading day, Alpha trading strategy chooses a target portfolio from a stock pool with a set of parameterized factors. By comparing the target portfolio and current portfolio, a rebalance plan is generated by selling the stocks in the target portfolio but not in current portfolio and buying the stocks in the current portfolio but not in target portfolio.

The strategy is evaluated based on the backtest performance on a sufficient large time period in the past using historical data. The backtest performance is measured by Sortino based on the tested period which has the following form:

$$\text{Sortino ratio} = \frac{r - r_f}{\sigma_d} \tag{5.27}$$

where r is the annualized return and r_f is the risk-free annualized return and σ_d is the downside standard deviation. Assume that the market has a positive return in the long term, we can not short the stocks, and we cannot hedge the risk through trading derivatives. A good strategy in this scenario will have a large return and small downside standard deviation. So we want to find the parameters that can maximize Sortino ratio.

By adjusting the parameters in the factors, the backtest performance is changed so that the Sortino value is different. As a result, we can view the Alpha trading strategy to be a blackbox function. Its input is a vector with all the parameters in the factors and its output is a singular value of Sortino. This blackbox function is time-consuming to evaluate since the backtest is usually executed on a sufficient time period on a large stock pool. We would like to find the optimal parameters in the factors with a few number of backtests. It is a suitable scenario to apply Bayesian optimization here.

In the experiment, We use the following factors listed in Table 5.2 with 22 parameters to choose the target portfolio. The stock pool is all the stocks in CSI 800 index in Chinese A-share market. The backtest is conducted within time period from Feb. 01, 2010 to Feb. 01, 2016 on the trading platform JoinQuant. The compared strategy is buy and hold CSI 300 index which is the major index in Chinese stock market.

This experiment is manually executed by computing the next best parameters' values from different BO methods, and then based on the new parameters' values, we run backtest on JoinQuant website. We limit the number of iteration to 300. In this real problem, we do not know the true intrinsic dimensions. Hence, we set the number of intrinsic dimensions to be $\sqrt{d}$ as suggested in the original paper of subspace learning (Djolonga et al., 2013).

Fig. 5.3 shows the Sortino improvement within 300 iterations and Fig. 5.4 shows the details of strategy with the parameters optimized by ANOVA-DCOP. Both random

Factor type	Rules
Fundamental signals	PE upper bound PB upper bound ROE lower bound ROB lower bound PEG lower bound EBIDA lower bound Book/Market ratio weight Fixed assets ratio weight Market capital size upper bound Circulating capital value upper bound Accounts payable/Operation revenue weight Accounts receivable/Operation revenue weight Cash on cash return weight Total profit/Total assets weight
Technical signals	RSI upper bound for selling (Price - Moving average) upper bound for buying Weekly KDJ(2 variables) lower bound for buying Daily KDJ(2 variables) lower bound for buying
Stoploss	Stoploss ratio on major index (CSI300) Stoploss ratio on individual stock

Table 5.2 Parameters in multi-factor trading strategy

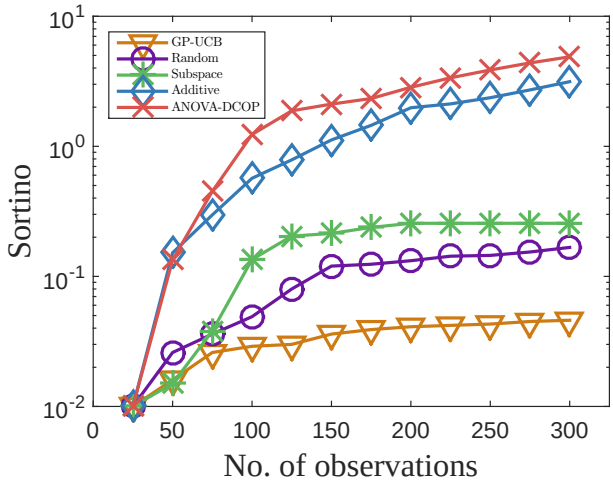


Fig. 5.3 Sortino value of the multi-factor trading strategy optimized by BO methods within 300 time steps. The simulated trading is manually conducted on JoinQuant backtest platform from Feb. 01, 2010 to Feb. 01, 2016.

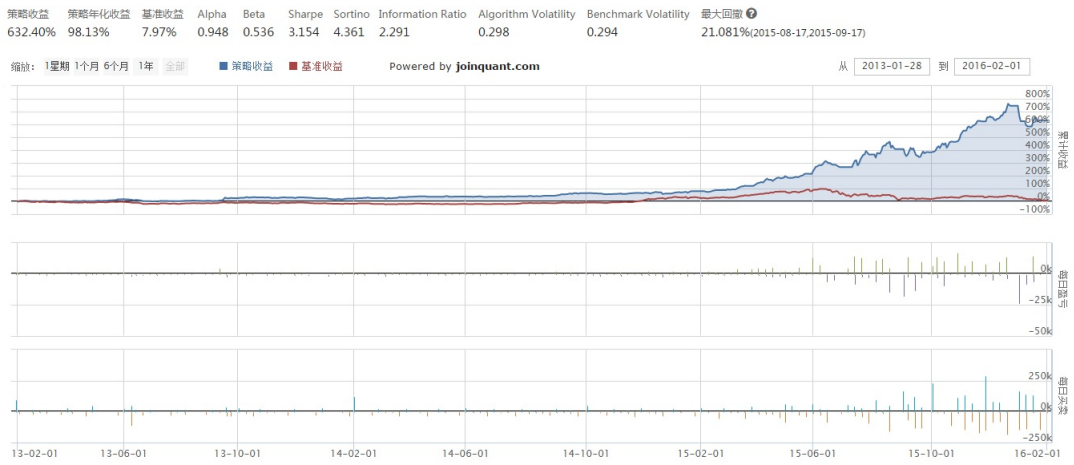


Fig. 5.4 Backtest performance of the optimized multi-factor trading strategy in Chinese A-share market v.s. CSI 300 index from Feb. 01, 2010 to Feb. 01, 2016.

embedding method and subspace learning method outperform GP-UCB since the factors we use have some correlations so they can find a smaller intrinsic dimensional space to search. For example, PE and PB use the same stock price information, Weekly KDJ and daily KDJ compute the same price sequence with different granularity. Multi-factor model is a rule based strategy. It mainly uses heuristics to select the stocks. Each heuristic contains one or two parameters, which creates the situation that some parameters are independent, and some are correlated. As a result, additive model and ANOVA-DCOP have a better performance. Moreover, ANOVA-DCOP performs better than additive model due to learning the correlation between the parameters.

5.5 Conclusion

The chapter proposed ANOVA-DCOP, a novel high dimensional Bayesian optimization method . A truncated ANOVA kernel function is introduced to decompose the sparse correlation structure of all the dimensions into a list of correlation structures that only involve subsets of dimensions. Based on the truncated ANOVA kernel, we have derived a linear decomposable acquisition function which can be cast as a DCOP. If we assume that the correlations are restricted in subsets with less than or equal to 2 dimensions, it can be

efficiently solved by bounded max-sum which is a typical multi-agent coordination method. This linear decomposable form in ANOVA-DCOP can reduce a high-dimensional problem into a list of low dimensional problems. Consequently, The complexity in searching the input space significantly reduced and we theoretically analyzed the regret of ANOVA-DCOP. We have evaluated the performance of our method with two analytical functions and a real-world trading optimization problem. The imperial results show that ANOVA-DCOP significantly improved the performance of existing high dimensional BO methods when the problem has a sparse correlation structure among the inputs.

Chapter 6

Conclusion and Future Work

This thesis has investigated the following question:

> **In the context of large-scale machine learning, how can the correlation structure of the data be exploited for constructing multi-agent coordination schemes that can improve the scalability of the machine learning models while preserving the computation accuracy?**

6.1 Summary of Contributions

While working toward a satisfactory answer to the above question, along with practical algorithms that achieve it, we have been able to make the following progress:

To address the challenge with large input domain:

- We present novel *Gaussian process decentralized data fusion algorithms* with *agent-centric support sets* for distributed cooperative perception of large-scale environmental phenomenon (section 3.2). In contrast with GP-DDF methods using fixed support set, our proposed algorithms allow every sensing agent to choose a possibly different support set and dynamically switch to another one during execution for encapsulating its own data into a local summary that, perhaps surprisingly, can still be assimilated with the other agents' local summaries (i.e., based on their current choices of support sets) into a globally consistent summary to be used for predicting the phenomenon.

- We propose a new transfer learning mechanism (section 3.2) for a team of mobile sensing agents capable of sharing and transferring information encapsulated in a summary based on a support set to that utilizing a different support set with some loss that can be theoretically bounded and analyzed. To alleviate the issue of information loss accumulating over multiple instances of transfer learning, we propose an information sharing mechanism to be incorporated into our GP-DDF algorithms.

- Our proposed algorithms can overcome the following three limitations of GP-DDF methods (Chen et al., 2012, 2013b, 2015):

    1. For any unobserved input location, an agent can choose a small, constant-sized (i.e., independent of domain size of the phenomenon) but sufficiently dense support set surrounding it to predict its measurement accurately while preserving time, space, and communication efficiencies;
    2. The agents can reduce the information loss due to summarization by choosing or dynamically switching to a support set "close" to their local data;
    3. Without needing to retain previously gathered data, an agent can choose or dynamically switch to a new support set whose summary can be constructed using information transferred from the summary based on its current support set, thus preserving scalability to big data.

- Finally, we empirically evaluate the performance of our proposed algorithms using three real-world datasets, one of which is millions in size (section 3.3).

To address the challenge with nonstationarity:

- We present a *decentralized multi-robot active sensing* (DEC-MAS) algorithm that can efficiently coordinate the exploration of multiple robots to automatically trade-off between learning the unknown, nonstationary correlation structure and minimizing the uncertainty of the environmental phenomena. Further, our DEC-MAS algorithm models a nonstationary phenomenon as a *Dirichlet process mixture of Gaussian processes* (DPM-GPs) (Section 4.1): Using the gathered observations, DPM-GPs can

learn to automatically partition the phenomenon into separate local areas, each of which comprises measurements that vary according to a stationary spatial correlation structure and can thus be modeled by a locally stationary Gaussian process.

- We demonstrate how DPM-GPs and its structural properties can be exploited to (a) formalize an active sensing criterion that trades off between gathering the most informative observations for estimating the unknown partition (i.e., a key component of the nonstationary correlation structure) vs. that for predicting the phenomenon given the current, possibly imprecise estimate of the partition (Section 4.2), and (b) support effective and efficient decentralized coordination (Section 4.3).

- We also provide a theoretical performance guarantee for DEC-MAS and analyze its time complexity (section 4.3).

- Finally, we empirically demonstrate using two real-world datasets that DEC-MAS outperforms the state-of-the-art MAS algorithms (Section 4.4).

To address the challenge with high dimensionality:

- We present a Bayesian optimization method using Gaussian process prior with ANOVA kernel function (section 5.2) that can decompose the correlation structure in high dimensions into a list of correlation structures of subsets of dimensions. Correspondingly, the high dimensional input space is decomposed into small subspaces so that a few observations can densely cover each subspace to learn and optimize the acquisition function in BO accurately.

- To the best of our knowledge, ANOVA-DCOP is the first work to introduce multi-agent coordination into high dimensional Bayesian optimization problem (section 5.2.3) by exploiting the sparse correlation structure using ANOVA kernel. We formulate the optimization of acquisition function as a decentralized constraint optimization problem (DCOP) which can be solved efficiently using multi-agent coordination. We theoretically bound the regret of the proposed algorithm and analyze its time complexity (section 5.3).

- Finally, we empirically evaluate the performance using two high dimensional functions with known optimum value and one real financial problem. The results show that our method outperforms the existing high dimensional BO methods when the problem has sparse correlation structure among the inputs (section 5.4).

6.2 Future Works

This section proposes and discusses potential research directions that could be pursued as continuation to our current work in this thesis:

- **Generalize agent-centric support set for nonstationary phenomenon** The large-scale environmental phenomenon is usually nonstationary. Therefore, many real-world problems have issues with both large input domain and nonstationarity. In GP-DDF methods with agent-centric support set, the stationary assumption is impractical in many situations. When the phenomenon is nonstationary, the transfer learning mechanism we proposed is not able to deliver the local summaries from one area to another area, which leads to huge information loss. To sharing information in the nonstationary phenomenon, a new transfer learning mechanism is required.

  Further, the information sharing mechanism we proposed is also not efficient in the sense of the number of observations for a nonstationary phenomenon. As demonstrated in our DEC-MAS, smooth area in the nonstationary field is easier to predict than the highly varying area. So instead of fixing the agent path to share the information as we did in the experiments, a more intelligent information sharing mechanism is required.

- **Multi-output DEC-MAS for multiple phenomena** The existing active sensing literature, including our DEC-MAS work in this thesis, are still restricted to single-output learning scenarios for which each measurement is a single scalar. In practice, a complex environmental field contains multiple phenomena that are strongly correlated. For example, the temperature and salinity in the ocean have a strong correlation. Therefore, one phenomenon can provide additional information to other phenomena in the same

environmental field. If some observations in one phenomenon are available in advance, we can easily learn another nonstationary phenomenon in the same field. With this motivation, it is able to extend our DEC-MAS algorithm to perform active sensing for multiple phenomena.

- **ANOVA-DCOP with constrains** In the experiments of ANOVA-DCOP, the functions that we evaluated have no constraints. However, in practice, the parameters in a complex system are often constrained by many equalities and inequalities. When ANOVA-DCOP decomposes the acquisition function according to the correlation structure in the dimensions, those constraints need to be decomposed into sub-constrains accordingly. In traditional high dimensional optimization (Chung, 2011, Frangioni and Gendron, 2013), the decomposition of constraints needs to match the decomposition of the objective function so that the high dimensional problem can be reduced to lower-dimensional sub-problems. It is challenging to make the decomposition of dimensions consistent with the acquisition function and constraints for BO methods because the correlation structure has to be learned during the optimization. Moreover, in many simulations, not only the function itself is unknown, the constraints may be unknown too, which is a highly nontrivial problem to be explored.

References

R. Akbari and K. Ziarati. A multilevel evolutionary algorithm for optimizing numerical functions. *International Journal of Industrial Engineering Computations*, 2(2):419–430, 2011.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

T.D. Bui and R.E. Turner. Tree-structured gaussian process approximations. In *Proc. NIPS*, pages 2213–2221, 2014.

N. Cao, K. H. Low, and J. M. Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proc. AAMAS*, pages 7–14, 2013.

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.

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.

J. Chen, N. Cao, K. H. Low, R. Ouyang, C. K.-Y. Tan, and P. Jaillet. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 152–161, 2013a.

J. Chen, K. H. Low, and C. K.-Y. Tan. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. Robotics: Science and Systems Conference*, 2013b.

J. Chen, K. H. Low, P. Jaillet, and Y. Yao. Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE Trans. Autom. Sci. Eng.*, 12(3):901–921, 2015.

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.

W. Chung. Dantzig–wolfe decomposition. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

J. Cortes. Distributed kriged Kalman filter for spatial estimation. *IEEE Trans. Automatic Control*, 54(12):2816–2827, 2009.

J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, 2004.

L. Csató and M. Opper. Sparse online Gaussian processes. *Neural Comput.*, 14:641–669, 2002.

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

M. P. Deisenroth and J. W. Ng. Distributed gaussian processes. In *Proc. ICML*, volume 2, page 5, 2015.

J Djolonga, A. Krause, and V. Cevher. High-dimensional gaussian process bandits. In *Proc. NIPS*, pages 1025–1033, 2013.

N. Durrande, D. Ginsbourger, O. Roustant, and L. Carraro. Anova kernels and rkhs of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57–67, 2013.

N. El Saadi and A. Bah. An individual-based model for studying the aggregation behavior in phytoplankton. *Ecological modelling*, 204(1):193–212, 2007.

A. Farinelli, A. Rogers, and N. Jennings. Bounded approximate decentralised coordination using the max-sum algorithm. 2009.

M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):381–396, 2002.

A. Frangioni and B. Gendron. A stabilized structured dantzig–wolfe decomposition method. *Mathematical Programming*, 140(1):45–76, 2013.

Walter R Gilks, Sylvia Richardson, and David J Spiegelhalter. Introducing markov chain monte carlo. *Markov chain Monte Carlo in practice*, 1:19, 1996.

D.I. Groves, R.J. Goldfarb, M. Gebre-Mariam, SG Hagemann, and F. Robert. Orogenic gold deposits: A proposed classification in the context of their crustal distribution and relationship to other gold deposit types. *Ore geology reviews*, 13(1-5):7–27, 1998.

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.

C.J. Hsieh, S. Si, and I.S. Dhillon. Fast prediction for large-scale kernel machines. In *Proc. NIPS*, pages 3689–3697, 2014.

I.CF. Ipsen and D.J. Lee. Determinant approximations. *arXiv preprint arXiv:1105.0437*, 2011.

T. S Jaakkola and D. Haussler. Probabilistic kernel regression models. In *AISTATS*, 1999.

D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

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.

Y. Kim and V. Lesser. Improved max-sum algorithm for dcop with n-ary constraints. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 191–198. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

K. Kirthevasan, Jeff S., and Barnabas P. High dimensional bayesian optimisation and bandits via additive models. In *Proc. ICML*, pages 295–304, 2015.

A. Krause and D. Golovin. Submodular function maximization. In L. Bordeaux, Y. Hamadi, and P. Kohli, editors, *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge Univ. Press, 2014.

A. Krause and C. Guestrin. Nonmyopic active learning of Gaussian processes: An exploration-exploitation approach. In *Proc. ICML*, pages 449–456, 2007.

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

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

H.J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

N.K. Larkin and D.E. Harrison. On the definition of el niño and associated seasonal average us weather anomalies. *Geophysical Research Letters*, 32(13), 2005.

Q. Le, T. Sarlós, and A. Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proc. ICML*, 2013.

N. E. Leonard, D. A. Palley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1): 48–74, 2007.

Z. Li, Y. Chao, J.C. McWilliams, and K. Ide. A three-dimensional variational data assimilation scheme for the regional ocean modeling system. *Journal of Atmospheric and Oceanic Technology*, 25(11):2074–2090, 2008.

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

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.

K. H. Low, J. M. Dolan, and P. Khosla. Active Markov information-theoretic path planning for robotic environmental sensing. In *Proceedings of the 10th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 753–760, 2011.

K. H. Low, J. Chen, J. M. Dolan, S. Chien, and D. R. Thompson. Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing. In *Proc. AAMAS*, pages 105–112, 2012.

K. H. Low, J. Chen, T. N. Hoang, N. Xu, and P. Jaillet. Recent advances in scaling up Gaussian process predictive models for large spatiotemporal data. In S. Ravela and A. Sandu, editors, *Proc. Dynamic Data-driven Environmental Systems Science Conference (DyDESS'14)*. LNCS 8964, Springer, 2015.

A. Meliou, A. Krause, C. Guestrin, and J. M. Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *Proc. AAAI*, pages 602–607, 2007a.

A. Meliou, A. Krause, C. Guestrin, and J. M. Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, pages 602–607, Vancouver, British Columbia, Canada, 2007b. AAAI Press.

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

J. Močkus. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer, 1975.

P. J. Modi, W.M. Shen, M. Tambe, and M. Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1):149–180, 2005.

R.M. Neal. Bayesian learning for neural networks. 1996.

M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proc. IPSN*, 2008.

C.J. Paciorek and M.J. Schervish. Nonstationary covariance functions for gaussian process regression. In *Proc. NIPS*. MIT Press, 2003.

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.

K.E. Parsopoulos and M.N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural computing*, 1(2-3):235–306, 2002.

A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. Technical report, 2005.

N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

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

C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In *Proc. NIPS*. MIT Press, 2002.

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

A. Rogers, A. Farinelli, R. Stranders, and N.R. Jennings. Bounded approximate decentralised coordination via the max-sum algorithm. *AIJ*, 175(2):730–759, 2011.

E. Rollon and J. Larrosa. Improved bounded max-sum for distributed constraint optimization. In *Principles and Practice of Constraint Programming*, pages 624–632. Springer, 2012.

R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.

P.D. Sampson and P. Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, 87(417):108–119, 1992.

P.D. Sampson, D. Damian, and P. Guttorp. Advances in modeling and inference for environmental processes with nonstationary spatial covariance. pages 17–32, 2001.

B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

M. Seeger and C. Williams. Fast forward selection to speed up sparse Gaussian process regression. In C. M. Bishop and B. J. Frey, editors, *Proc. AISTATS*, 2003.

Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *J. Artificial Intelligence Research*, 34:707–755, 2009.

A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

E. L. Snelson. *Flexible and efficient Gaussian process models for machine learning*. Ph.D. Thesis, University College London, London, UK, 2007.

E. L. Snelson and Z. Ghahramani. Local and global sparse Gaussian process approximation. In *Proc. AISTATS*, 2007.

E. L. Snelson and Z. Gharahmani. Sparse Gaussian processes using pseudo-inputs. In *Proc. NIPS*, pages 1259–1266, 2005.

J. Snoek, H. Larochelle, and R.P. Adams. Practical bayesian optimization of machine learning algorithms. In *Proc. NIPS*, pages 2951–2959, 2012.

N. Srinivas, A. Krause, S.M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

N. Srinivas, A. Krause, S. Kakade, and M. W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, pages 1015–1022, 2010.

M.L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

S. Sun, J. Zhao, and J. Zhu. A review of Nyström methods for large-scale machine learning. *Information Fusion*, 26:36–48, 2015.

B. Tomoiagă, M. Chindriş, A. Sumper, A. Sudria-Andreu, and R. Villafafila-Robles. Pareto optimal reconfiguration of power distribution systems using a genetic algorithm based on nsga-ii. *Energies*, 6(3):1439–1455, 2013.

R. Tortoriello. *Quantitative strategies for achieving alpha*. McGraw Hill, 2009.

V. Tresp. Mixtures of gaussian processes. In *Proc. NIPS*. MIT Press, 2001.

F. Wang, L.H. Philip, and D.W. Cheung. Complex stock trading strategy based on particle swarm optimization. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–6. IEEE, 2012.

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.

Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

S. Wei and Z. Pengda. Theoretical study of statistical fractal model with applications to mineral resource prediction. *Computers and geosciences*, 28(3):369–376, 2002.

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.

N. Xu, K. H. Low, J. Chen, K. K. Lim, and E. B. Özgül. GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. In *Proc. AAAI*, pages 2585–2592, 2014.

T. Yang, Y.F. Li, M. Mahdavi, R. Jin, and Z.H. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Proc. NIPS*, pages 476–484, 2012.

K. Zhang, I. W. Tsang, and J. T. Kwok. Improved Nyström low-rank approximation and error analysis. In *Proc. ICML*, pages 1232–1239, 2008.

Appendix A

Agent-Centric Support Set for Regression

A.1 Proof of Proposition 1

$$
\begin{aligned}
&\omega_{\mathcal{S}'|\mathcal{D}_i} \\
&= \Sigma_{\mathcal{S}'\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}(y_{\mathcal{D}_i} - \mu_{\mathcal{D}_i}) \\
&= \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}(y_{\mathcal{D}_i} - \mu_{\mathcal{D}_i}) \\
&= \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\omega_{\mathcal{S}|\mathcal{D}_i}
\end{aligned}
$$

and

$$
\begin{aligned}
&\Phi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_i} \\
&= \Sigma_{\mathcal{S}'\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}'} \\
&= \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}'} \\
&= \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Phi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}'}
\end{aligned}
$$

where the second equalities above follow from the assumption that $\mathcal{S}'$ and $\mathcal{D}_i$ are conditionally independent given $\mathcal{S}$ (i.e., $\Sigma_{\mathcal{S}'\mathcal{D}_i|\mathcal{S}} = \Sigma_{\mathcal{S}'\mathcal{D}_i} - \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{D}_i} = \underline{0}$).

A.2 Proof of Proposition 2

$$
\begin{aligned}
&\Psi_{\mathcal{SS}|\mathcal{D}_i} \\
&= \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}\Sigma_{\mathcal{D}_i\mathcal{S}} \\
&= \Sigma_{\mathcal{SD}_i}(\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i} + \Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}|\mathcal{D}_i}\Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i})\Sigma_{\mathcal{D}_i\mathcal{S}} \\
&= \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}} + \\
&\quad \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}|\mathcal{D}_i}\Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}} \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i} + \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}})^{-1}\Phi_{\mathcal{SS}|\mathcal{D}_i} \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i} + \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1}\Phi_{\mathcal{SS}|\mathcal{D}_i} \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i}(I + (\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1}\Phi_{\mathcal{SS}|\mathcal{D}_i}) \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1}(\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i} + \Phi_{\mathcal{SS}|\mathcal{D}_i}) \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1}\Sigma_{\mathcal{SS}}
\end{aligned}
$$

where the second equality follows from the matrix inverse lemma on $\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}} = (\Sigma_{\mathcal{D}_i\mathcal{D}_i} - \Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}}\Sigma_{\mathcal{SD}_i})^{-1} = \Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i} + \Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}|\mathcal{D}_i}\Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}$. As a result, $\Psi^{-1}_{SS|\mathcal{D}_i} = \Sigma^{-1}_{SS}(\Sigma_{SS} - \Phi_{SS|\mathcal{D}_i})\Phi^{-1}_{SS|\mathcal{D}_i} = \Phi^{-1}_{SS|\mathcal{D}_i} - \Sigma^{-1}_{SS}$.

$$
\begin{aligned}
&\nu_{\mathcal{S}|\mathcal{D}_i} \\
&= \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}}y_{\mathcal{D}_i} \\
&= \Sigma_{\mathcal{SD}_i}(\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i} + \Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}|\mathcal{D}_i}\Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i})y_{\mathcal{D}_i} \\
&= \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}y_{\mathcal{D}_i} + \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{SS}|\mathcal{D}_i}\Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}y_{\mathcal{D}_i} \\
&= \omega_{\mathcal{S}|\mathcal{D}_i} + \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Sigma_{\mathcal{SD}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}})^{-1}\omega_{\mathcal{S}|\mathcal{D}_i} \\
&= \omega_{\mathcal{S}|\mathcal{D}_i} + \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1}\omega_{\mathcal{S}|\mathcal{D}_i} \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i} + (\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1})\omega_{\mathcal{S}|\mathcal{D}_i} \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1} \\
&\qquad\qquad ((\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i} + I)\omega_{\mathcal{S}|\mathcal{D}_i} \\
&= \Phi_{\mathcal{SS}|\mathcal{D}_i}(\Sigma_{\mathcal{SS}} - \Phi_{\mathcal{SS}|\mathcal{D}_i})^{-1}\Sigma_{\mathcal{SS}}\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i}\omega_{\mathcal{S}|\mathcal{D}_i} \\
&= (\Sigma_{\mathcal{SS}}\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i} - I)^{-1}\Sigma_{\mathcal{SS}}\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i}\omega_{\mathcal{S}|\mathcal{D}_i} \\
&= (\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i} - \Sigma^{-1}_{\mathcal{SS}})^{-1}\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i}\omega_{\mathcal{S}|\mathcal{D}_i} \\
&= \Psi_{\mathcal{SS}|\mathcal{D}_i}\Phi^{-1}_{\mathcal{SS}|\mathcal{D}_i}\omega_{\mathcal{S}|\mathcal{D}_i}
\end{aligned}
$$

where the second equality follows from the matrix inverse lemma on $\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i|\mathcal{S}} = \Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i} + \Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}\Sigma_{\mathcal{D}_i\mathcal{S}}\Sigma^{-1}_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}\Sigma_{\mathcal{S}\mathcal{D}_i}\Sigma^{-1}_{\mathcal{D}_i\mathcal{D}_i}$. As a result, $\Psi^{-1}_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}\nu_{\mathcal{S}|\mathcal{D}_i} = \Phi^{-1}_{\mathcal{S}\mathcal{S}|\mathcal{D}_i}\omega_{\mathcal{S}|\mathcal{D}_i}$. So, (3.9) follows.

A.3 Proof of Theorem 1

The following lemma is necessary for deriving our main result here:

Lemma 4. *Define $\sigma_{xx'}$ using a squared exponential covariance function. Then, every covariance component $\sigma_{xx'}$ in $\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}}$, $\Sigma_{\mathcal{S}\mathcal{S}}$, $\Sigma_{\mathcal{S}'_t\mathcal{S}}$, and $\Sigma_{\mathcal{D}_{it'}\mathcal{S}}$ satisfies $(\sigma_{xx'}-\sigma_{ss'})^2 \leq 3e^{-1}\sigma_s^4(\|\Lambda^{-1}(x-s)\|^2+\|\Lambda^{-1}(x'-s')\|^2)$ for all $x,x',s,s' \in \mathcal{X}$.*

Proof. Since every covariance component $\sigma_{xx'}$ in $\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}}$, $\Sigma_{\mathcal{S}\mathcal{S}}$, $\Sigma_{\mathcal{S}'_t\mathcal{S}}$, and $\Sigma_{\mathcal{D}_{it'}\mathcal{S}}$ does not involve the noise variance σ_n^2, it follows from (3.1) that

$$\begin{aligned}\sigma_{xx'} &= \sigma_s^2 \exp\left(-\left\|\frac{\Lambda^{-1}(x-x')}{\sqrt{2}}\right\|^2\right)\\ &= \sigma_s^2 k\left(\left\|\frac{\Lambda^{-1}(x-x')}{\sqrt{2}}\right\|\right)\end{aligned}$$

where $k(a) \triangleq \exp(-a^2)$. Then,

$$\begin{aligned}&(\sigma_{xx'}-\sigma_{ss'})^2\\ &= \sigma_s^4\left\{k\left(\left\|\frac{\Lambda^{-1}(x-x')}{\sqrt{2}}\right\|\right)-k\left(\left\|\frac{\Lambda^{-1}(s-s')}{\sqrt{2}}\right\|\right)\right\}^2\\ &= 0.5\sigma_s^4 k'(\xi)^2(\|\Lambda^{-1}(x-x')\|-\|\Lambda^{-1}(s-s')\|)^2\\ &\leq e^{-1}\sigma_s^4(\|\Lambda^{-1}(x-s)\|+\|\Lambda^{-1}(x'-s')\|)^2\\ &\leq e^{-1}\sigma_s^4(\|\Lambda^{-1}(x-s)\|+\|\Lambda^{-1}(x'-s')\|)^2\\ &\leq 3e^{-1}\sigma_s^4(\|\Lambda^{-1}(x-s)\|^2+\|\Lambda^{-1}(x'-s')\|^2)\end{aligned}$$

where the second equality is due to mean value theorem such that $k'(\xi)$ is the first-order derivative of k evaluated at some $\xi \in (\|\Lambda^{-1}(s-s')\|/\sqrt{2}, \|\Lambda^{-1}(x-x')\|/\sqrt{2})$ without loss of generality, the first inequality follows from the fact that $k'(a)$ is maximized at $a=-1/\sqrt{2}$ and

hence $k'(\xi) \leq k'(-1/\sqrt{2}) = \sqrt{2/e}$, and the second inequality is due to triangle inequality (i.e., $\|\Lambda^{-1}(x-x')\| \leq \|\Lambda^{-1}(x-s)\| + \|\Lambda^{-1}(s-s')\| + \|\Lambda^{-1}(s'-x')\|$). □

Supposing each subset $\mathcal{D}_{is}$ ($\mathcal{S}'_s$) contains T (T') locations[1], select one location from each subset to form a new subset $\mathcal{D}_{it'} \triangleq \{x_{it's}\}_{s\in\mathcal{S}}$ ($\mathcal{S}'_t \triangleq \{x'_{ts}\}_{s\in\mathcal{S}}$) of $|\mathcal{S}|$ locations for $t' = 1$ ($t = 1$) and repeat this for $t' = 2,\ldots,T$ ($t = 2,\ldots,T'$). Then, $\mathcal{D}_i = \bigcup_{t'=1}^{T} \mathcal{D}_{it'}$ and $\mathcal{S}' = \bigcup_{t=1}^{T'} \mathcal{S}'_t$. It follows that $\Sigma_{\mathcal{S}'\mathcal{S}} = [\Sigma_{\mathcal{S}'_t\mathcal{S}}]_{t=1,\ldots,T'}$, $\Sigma_{\mathcal{S}\mathcal{D}_i} = [\Sigma_{\mathcal{S}\mathcal{D}_{it'}}]_{t'=1,\ldots,T}$, and $\Sigma_{\mathcal{S}'\mathcal{D}_i} = [\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}}]_{t=1,\ldots,T',t'=1,\ldots,T}$.

Using the definition of Frobenius norm followed by the subadditivity of a square root function,

$$\begin{aligned}
&||\Sigma_{\mathcal{S}'\mathcal{D}_i} - \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}||_F \\
&= ||\Sigma_{\mathcal{S}'\mathcal{D}_i|\mathcal{S}}||_F \\
&= \sqrt{\sum_{t=1}^{T'}\sum_{t'=1}^{T} ||\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}|\mathcal{S}}||_F^2} \\
&\leq \sum_{t=1}^{T'}\sum_{t'=1}^{T} ||\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}|\mathcal{S}}||_F \ .
\end{aligned} \tag{A.1}$$

Let $A_{\mathcal{S}'_t\mathcal{D}_{it'}} \triangleq \Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}} - \Sigma_{\mathcal{S}\mathcal{S}}$, $B_{\mathcal{S}'_t\mathcal{S}} \triangleq \Sigma_{\mathcal{S}'_t\mathcal{S}} - \Sigma_{\mathcal{S}\mathcal{S}}$, and $C_{\mathcal{D}_{it'}\mathcal{S}} \triangleq \Sigma_{\mathcal{D}_{it'}\mathcal{S}} - \Sigma_{\mathcal{S}\mathcal{S}}$. Then,

$$\begin{aligned}
&||\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}|\mathcal{S}}||_F \\
&= ||\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}} - \Sigma_{\mathcal{S}'_t\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_{it'}}||_F \\
&= ||\Sigma_{\mathcal{S}\mathcal{S}} + A_{\mathcal{S}'_t\mathcal{D}_{it'}} - \\
&\quad (\Sigma_{\mathcal{S}\mathcal{S}} + B_{\mathcal{S}'_t\mathcal{S}})\Sigma_{\mathcal{S}\mathcal{S}}^{-1}(\Sigma_{\mathcal{S}\mathcal{S}} + C_{\mathcal{D}_{it'}\mathcal{S}})^\top||_F \\
&= ||\Sigma_{\mathcal{S}\mathcal{S}} + A_{\mathcal{S}'_t\mathcal{D}_{it'}} - \Sigma_{\mathcal{S}\mathcal{S}}^\top - C_{\mathcal{D}_{it'}\mathcal{S}}^\top - B_{\mathcal{S}'_t\mathcal{S}} - \\
&\quad B_{\mathcal{S}'_t\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}C_{\mathcal{D}_{it'}\mathcal{S}}^\top||_F \\
&\leq ||A_{\mathcal{S}'_t\mathcal{D}_{it'}}||_F + ||B_{\mathcal{S}'_t\mathcal{S}}||_F + ||C_{\mathcal{D}_{it'}\mathcal{S}}||_F + \\
&\quad ||B_{\mathcal{S}'_t\mathcal{S}}||_F||C_{\mathcal{D}_{it'}\mathcal{S}}||_F||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F
\end{aligned} \tag{A.2}$$

where the inequality is due to the subadditivity and submultiplicativity of the matrix norm.

[1] If the subset sizes differ, then "virtual" locations are added to each subset to make all subsets to be of the same size as $T \triangleq \arg\max_{s\in\mathcal{S}} |\mathcal{D}_{is}|$ ($T' \triangleq \arg\max_{s\in\mathcal{S}} |\mathcal{S}'_s|$). The virtual locations added to $\mathcal{D}_{is}$ ($\mathcal{S}'_s$) are chosen as $s \in \mathcal{S}$ so that they do not induce additional errors but will loosen the bound.

Let $\varepsilon_{\mathcal{S}'_t} \triangleq (1/|\mathcal{S}|)\sum_{x\in\mathcal{S}'_t}||\Lambda^{-1}(x-c(x))||^2$ and $\varepsilon_{\mathcal{D}_{it'}} \triangleq (1/|\mathcal{S}|)\sum_{x\in\mathcal{D}_{it'}}||\Lambda^{-1}(x-c(x))||^2$. Then,

$$\begin{aligned}
&||A_{\mathcal{S}'_t\mathcal{D}_{it'}}||_F^2 \\
&= ||\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}} - \Sigma_{\mathcal{S}\mathcal{S}}||_F^2 \\
&= \sum_{s,s'\in\mathcal{S}} (\sigma_{x'_{ts}x_{it's'}} - \sigma_{ss'})^2 \\
&\leq 3e^{-1}\sigma_s^4 \sum_{s,s'\in\mathcal{S}} \left(||\Lambda^{-1}(x'_{ts}-s)||^2 + ||\Lambda^{-1}(x_{it's'}-s')||^2\right) \\
&= 3e^{-1}\sigma_s^4|\mathcal{S}|\left(\sum_{s\in\mathcal{S}}||\Lambda^{-1}(x'_{ts}-s)||^2 + \right. \\
&\qquad\qquad \left. \sum_{s'\in\mathcal{S}}||\Lambda^{-1}(x_{it's'}-s')||^2\right) \\
&= 3e^{-1}\sigma_s^4|\mathcal{S}|^2(\varepsilon_{\mathcal{S}'_t} + \varepsilon_{\mathcal{D}_{it'}})
\end{aligned} \tag{A.3}$$

since $\varepsilon_{\mathcal{S}'_t} = (1/|\mathcal{S}|)\sum_{s\in\mathcal{S}}||\Lambda^{-1}(x'_{ts}-s)||^2$ and $\varepsilon_{\mathcal{D}_{it'}} = (1/|\mathcal{S}|)\sum_{s'\in\mathcal{S}}||\Lambda^{-1}(x_{it's'}-s')||^2$. The inequality is due to Lemma 4.

$$\begin{aligned}
&||B_{\mathcal{S}'_t\mathcal{S}}||_F^2 \\
&= ||\Sigma_{\mathcal{S}'_t\mathcal{S}} - \Sigma_{\mathcal{S}\mathcal{S}}||_F^2 \\
&= \sum_{s,s'\in\mathcal{S}} (\sigma_{x'_{ts}s'} - \sigma_{ss'})^2 \\
&\leq 3e^{-1}\sigma_s^4 \sum_{s,s'\in\mathcal{S}} \left(||\Lambda^{-1}(x'_{ts}-s)||^2 + ||\Lambda^{-1}(s'-s')||^2\right) \\
&= 3e^{-1}\sigma_s^4|\mathcal{S}|\sum_{s\in\mathcal{S}}||\Lambda^{-1}(x'_{ts}-s)||^2 \\
&= 3e^{-1}\sigma_s^4|\mathcal{S}|^2\varepsilon_{\mathcal{S}'_t}
\end{aligned} \tag{A.4}$$

such that the inequality is due to Lemma 4.

$$
\begin{aligned}
&||C_{\mathcal{D}_{it'}\mathcal{S}}||_F^2 \\
&= ||\Sigma_{\mathcal{D}_{it'}\mathcal{S}} - \Sigma_{\mathcal{S}\mathcal{S}}||_F^2 \\
&= \sum_{s,s' \in \mathcal{S}} (\sigma_{x_{it's}s'} - \sigma_{ss'})^2 \\
&\leq 3e^{-1}\sigma_s^4 \sum_{s,s' \in \mathcal{S}} \left(||\Lambda^{-1}(x_{it's} - s)||^2 + ||\Lambda^{-1}(s' - s')||^2 \right) \\
&= 3e^{-1}\sigma_s^4 |\mathcal{S}| \sum_{s \in \mathcal{S}} ||\Lambda^{-1}(x_{it's} - s)||^2 \\
&= 3e^{-1}\sigma_s^4 |\mathcal{S}|^2 \varepsilon_{\mathcal{D}_{it'}}
\end{aligned}
\tag{A.5}
$$

such that the inequality is due to Lemma 4.

By substituting (A.3), (A.4), and (A.5) into (A.2),

$$
\begin{aligned}
&||\Sigma_{\mathcal{S}'_t\mathcal{D}_{it'}|\mathcal{S}}||_F \\
&\leq \sqrt{3e^{-1}\sigma_s^4 |\mathcal{S}|^2 (\varepsilon_{\mathcal{S}'_t} + \varepsilon_{\mathcal{D}_{it'}})} + \sqrt{3e^{-1}\sigma_s^4 |\mathcal{S}|^2 \varepsilon_{\mathcal{S}'_t}} + \\
&\quad \sqrt{3e^{-1}\sigma_s^4 |\mathcal{S}|^2 \varepsilon_{\mathcal{D}_{it'}}} + \\
&\quad \sqrt{3e^{-1}\sigma_s^4 |\mathcal{S}|^2 \varepsilon_{\mathcal{S}'_t}} \sqrt{3e^{-1}\sigma_s^4 |\mathcal{S}|^2 \varepsilon_{\mathcal{D}_{it'}}} ||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F \\
&= \sqrt{3/e}\sigma_s^2 |\mathcal{S}| \Big(\sqrt{\varepsilon_{\mathcal{S}'_t} + \varepsilon_{\mathcal{D}_{it'}}} + \sqrt{\varepsilon_{\mathcal{S}'_t}} + \sqrt{\varepsilon_{\mathcal{D}_{it'}}} + \\
&\qquad\qquad \sigma_s^2 ||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F |\mathcal{S}| \sqrt{3\varepsilon_{\mathcal{S}'_t}\varepsilon_{\mathcal{D}_{it'}}/e} \Big) .
\end{aligned}
\tag{A.6}
$$

By substituting (A.6) into (A.1),

$$
\begin{aligned}
&||\Sigma_{\mathcal{S}'\mathcal{D}_i} - \Sigma_{\mathcal{S}'\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{D}_i}||_F \\
&\leq \sqrt{3/e}\sigma_s^2|\mathcal{S}|\sum_{t=1}^{T'}\sum_{t'=1}^{T}\Big(\sqrt{\varepsilon_{\mathcal{S}'_t}+\varepsilon_{\mathcal{D}_{it'}}}+\sqrt{\varepsilon_{\mathcal{S}'_t}}+\sqrt{\varepsilon_{\mathcal{D}_{it'}}}+ \\
&\qquad\qquad \sigma_s^2||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F|\mathcal{S}|\sqrt{3\varepsilon_{\mathcal{S}'_t}\varepsilon_{\mathcal{D}_{it'}}/e}\Big) \\
&\leq \sqrt{3/e}\sigma_s^2|\mathcal{S}|\Bigg(\sqrt{TT'\sum_{t=1}^{T'}\sum_{t'=1}^{T}(\varepsilon_{\mathcal{S}'_t}+\varepsilon_{\mathcal{D}_{it'}})}+ \\
&\qquad \sqrt{TT'\sum_{t=1}^{T'}\sum_{t'=1}^{T}\varepsilon_{\mathcal{S}'_t}}+\sqrt{TT'\sum_{t=1}^{T'}\sum_{t'=1}^{T}\varepsilon_{\mathcal{D}_{it'}}}+ \\
&\qquad \sigma_s^2||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F|\mathcal{S}|\sqrt{TT'(3/e)\sum_{t=1}^{T'}\sum_{t'=1}^{T}\varepsilon_{\mathcal{S}'_t}\varepsilon_{\mathcal{D}_{it'}}}\Bigg) \\
&= \sqrt{3/e}\sigma_s^2|\mathcal{S}|\Bigg(\sqrt{TT'\left(T\sum_{t=1}^{T'}\varepsilon_{\mathcal{S}'_t}+T'\sum_{t'=1}^{T}\varepsilon_{\mathcal{D}_{it'}}\right)}+ \\
&\qquad \sqrt{T^2T'\sum_{t=1}^{T'}\varepsilon_{\mathcal{S}'_t}}+\sqrt{TT'^2\sum_{t'=1}^{T}\varepsilon_{\mathcal{D}_{it'}}}+ \\
&\qquad \sigma_s^2||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F|\mathcal{S}|\sqrt{TT'(3/e)\sum_{t=1}^{T'}\varepsilon_{\mathcal{S}'_t}\sum_{t'=1}^{T}\varepsilon_{\mathcal{D}_{it'}}}\Bigg) \\
&= \sqrt{3/e}\sigma_s^2|\mathcal{S}|TT'\Big(\sqrt{\varepsilon_{\mathcal{S}'}+\varepsilon_{\mathcal{D}_i}}+\sqrt{\varepsilon_{\mathcal{S}'}}+\sqrt{\varepsilon_{\mathcal{D}_i}}+ \\
&\qquad\qquad \sigma_s^2||\Sigma_{\mathcal{S}\mathcal{S}}^{-1}||_F|\mathcal{S}|\sqrt{3\varepsilon_{\mathcal{S}'}\varepsilon_{\mathcal{D}_i}/e}\Big)
\end{aligned}
$$

such that the second inequality follows from

$$\sum_{t=1}^{T}\sqrt{a_t} \leq \sqrt{T\sum_{t=1}^{T}a_t}$$

which can be obtained by applying Jensen's inequality to the concave square root function. The last equality is due to $\varepsilon_{\mathcal{S}'} = (1/T')\sum_{t=1}^{T'}\varepsilon_{\mathcal{S}'_t}$ and $\varepsilon_{\mathcal{D}_i} = (1/T)\sum_{t'=1}^{T}\varepsilon_{\mathcal{D}_{it'}}$.

A.4 GP-DDF/GP-DDF$^+$ Algorithm with Agent-Centric Support Sets based on Lazy Transfer Learning

Refer to Algorithm 4 below.

Algorithm 4: GP-DDF/GP-DDF$^+$ with agent-centric support sets based on lazy transfer learning for agent i

if agent i transits from local area with support set $\mathcal{S}$ to local area with support set $\mathcal{S}'$ **then**
- `/* Information sharing mechanism */`
- `/* Leaving local area with` $\mathcal{S}$ `*/`
- **if** other agents are in local area with support set $\mathcal{S}$ **then**
  - Construct and send local summary $(\nu_{\mathcal{S}|\mathcal{D}_i}, \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$ to an agent in this area who assimilates it with its own local summary using (3.4);
  - Delete local summary $(\nu_{\mathcal{S}|\mathcal{D}_i}, \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$;
- **else**
  - Backup local summary $(\nu_{\mathcal{S}|\mathcal{D}_i}, \Psi_{\mathcal{S}\mathcal{S}|\mathcal{D}_i})$;
- `/* Entering local area with` $\mathcal{S}'$ `*/`
- **if** other agents are in local area with support set $\mathcal{S}'$ **then**
  - Get support set $\mathcal{S}'$ from an agent in this area;
- **else**
  - **if** some agent j in the team stores a backup of local summary based on support set $\mathcal{S}'$ **then**
    - Retrieve and remove this backup of local summary based on $\mathcal{S}'$ from agent j;
  - **else**
    - Construct support set $\mathcal{S}'$;

if agent i has to predict the phenomenon **then**
- **if** data $(\mathcal{D}'_i, y_{\mathcal{D}'_i})$ is available from local area with support set $\mathcal{S}'$ **then**
  - Construct local summary $(\nu_{\mathcal{S}'|\mathcal{D}'_i}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}'_i})$ by (3.3);
- Exchange local summary with every agent $j \neq i$;
- **foreach** agent $j \neq i$ in local area with support set $\mathcal{S}'' \neq \mathcal{S}'$ **do**
  - `/* Transfer learning mechanism */`
  - Derive local summary $(\nu_{\mathcal{S}'|\mathcal{D}_j}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_j})$ based on $\mathcal{S}'$ approximately from received local summary $(\nu_{\mathcal{S}''|\mathcal{D}_j}, \Psi_{\mathcal{S}''\mathcal{S}''|\mathcal{D}_j})$ based on $\mathcal{S}''$ using transfer learning mechanism in Algorithm 1 (Section 3.2);
- Compute global summary $(\dot{\nu}_{\mathcal{S}'}, \dot{\Psi}_{\mathcal{S}'\mathcal{S}'})$ by (3.4) using local summaries $(\nu_{\mathcal{S}'|\mathcal{D}'_i}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}'_i})$ and $(\nu_{\mathcal{S}'|\mathcal{D}_j}, \Psi_{\mathcal{S}'\mathcal{S}'|\mathcal{D}_j})$ of every agent $j \neq i$;
- Run GP-DDF (3.5) or GP-DDF$^+$ (3.6);

A.5 Hyperparameter Learning

The hyperparameters of our GP-DDF-ASS and GP-DDF$^+$-ASS algorithms are learned by maximizing the sum of log-marginal likelihoods $\sum_{\mathcal{S}} \log p(y_{\mathcal{D}}|\mathcal{S})$ over the support set $\mathcal{S}$ of every different local area via gradient ascent with respect to a common set of signal variance, noise variance, and length-scale hyperparameters (Section 3.1) where, as derived in (Quiñonero-Candela and Rasmussen, 2005),

$$\log p(y_{\mathcal{D}}|\mathcal{S}) = -0.5(\log|\Xi_{\mathcal{D}\mathcal{D}|\mathcal{S}}| + y_{\mathcal{D}}^{\top}\Xi_{\mathcal{D}\mathcal{D}|\mathcal{S}}^{-1}y_{\mathcal{D}} + |\mathcal{D}|\log(2\pi))$$

such that $\Xi_{\mathcal{D}\mathcal{D}|\mathcal{S}} \triangleq \Phi_{\mathcal{D}\mathcal{D}|\mathcal{S}} + \text{blockdiag}[\Sigma_{\mathcal{D}\mathcal{D}|\mathcal{S}}] + \sigma_n^2 I$. Note that these learned hyperparameters of our GP-DDF-ASS and GP-DDF$^+$-ASS algorithms correspond to the case where our proposed lazy transfer learning mechanism incurs minimal information loss.

A.6 Real-World Plankton Density Phenomenon

The MODIS plankton density dataset (Fig. A.1) is bounded within lat. 30-31N and lon. 245.36-246.11E (i.e., off the west coast of USA) with a data size of 4941. The domain of this phenomenon is discretized into a 61×81 grid of locations that are associated with log-chlorophyll-a measurements in mg/m^3. It is partitioned into $K = 16$ disjoint local areas of size about 15 by 20, each of which is assigned N/K mobile sensing agents. The N/K agents in every local area then move together in a pre-defined lawnmower pattern from one local area to the next adjacent one such that they visit all the $K = 16$ local areas exactly twice to gather data/observations from this phenomenon and end in the same local area initially assigned to them. Whenever the N/K agents transit into the next local area, they will move randomly within to gather the local data/observations; the results are averaged over 30 runs.

The performance of our *GP-DDF and GP-DDF$^+$ algorithms with agent-centric support sets* (respectively, GP-DDF-ASS and GP-DDF$^+$-ASS), each of which is of size 50 and randomly distributed over a different local area of the plankton density phenomenon, are compared against that of the local GPs method Choudhury et al. (2002), Das and Srivastava

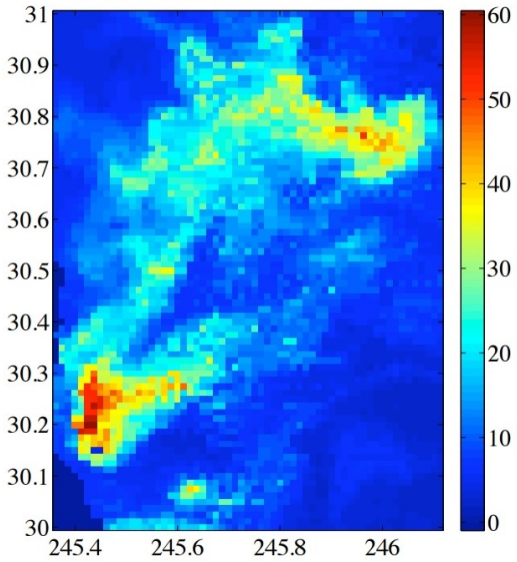
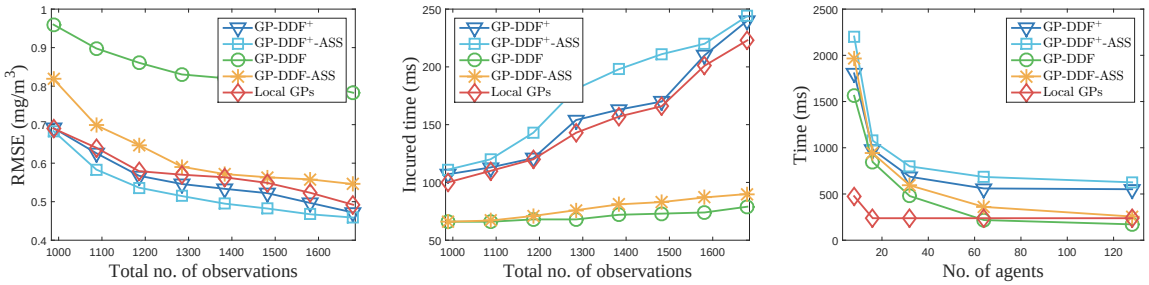


Fig. A.1 Plankton density phenomenon bounded within lat. 30-31N and lon. 245.36-246.11E.

(2010) and state-of-the-art GP-DDF and GP-DDF$^+$ Chen et al. (2015) with a common support set of size 50 randomly distributed over the entire plankton density phenomenon and known to all agents.

Predictive Performance.

Fig. A.2a shows results of decreasing RMSE achieved by tested algorithms with an increasing total number of observations for $N = 32$ agents. The observations and analysis are similar to that reported in Section 3.3 (specifically, under 'Predictive Performance'). It can also be observed that the performance gap between GP-DDF-ASS and GP-DDF$^+$-ASS appears to be smaller than that for the indoor lighting quality and temperature phenomenon shown in Figs. 4.1a and 4.1c, respectively: Compared to the indoor lighting quality (temperature phenomenon), the plankton density phenomenon has a relatively larger length-scale (much smaller domain size and consequently closer agent-centric support sets), thereby making transfer learning more effective, which agrees with the observation in our theoretical analysis for Theorem 1 (Section 3.2), and reducing the performance advantage of GP-DDF$^+$-ASS over GP-DDF-ASS in exploiting local data.



(a) (b) (c)

Fig. A.2 Graphs of (a) RMSE and (b) total incurred time vs. total no. of observations, and (c) graphs of total incurred time vs. no. of agents achieved by tested algorithms for plankton density phenomenon.

Time Efficiency.

Fig. A.2b shows results of increasing total time incurred by tested algorithms with an increasing total number of observations for $N = 32$ agents. The experimental setup, observations, and analysis are again similar to that reported in Section 3.3 (specifically, under 'Time Efficiency').

Scalability in the Number of Agents.

Fig. A.2c shows results of total time incurred by tested algorithms with an increasing number N of agents to gather a total number of 1235 observations. It can be observed that the total time incurred by GP-DDF$^+$-ASS, GP-DDF$^+$, GP-DDF-ASS, and GP-DDF decrease with more agents, as explained in Section 3.1; recall further that they become more robust to agent failure with more agents assigned to every local area to reduce its risk of being empty and hence its likelihood of inducing a backup. In addition, GP-DDF-ASS and GP-DDF$^+$-ASS, respectively, incur only slightly more time than GP-DDF and GP-DDF$^+$ due to their information sharing mechanism described in Section 3.2 (specifically, the first if-then construct in Algorithm 4 in Appendix A.4). Note that the total time incurred by local GPs remains constant with respect to the number of agents because a fixed number of about 77 observations are gathered by all agents in each local area and used by any agent for prediction in the same local area.

Appendix B

DEC-MAS for Active Learning

B.1 Proof of Theorem 2

Let $\widetilde{\Sigma}_{X_kX_k|D_k,\theta_k} \triangleq \Sigma_{X_kX_k|D_k,\theta_k} - \widehat{\Sigma}_{X_kX_k|D_k,\theta_k}$ and ρ_k be the spectral radius of $\widehat{\Sigma}^{-1}_{X_kX_k|D_k,\theta_k}\widetilde{\Sigma}_{X_kX_k|D_k,\theta_k}$. We first bound ρ_k from above.

For any X_k, $\widehat{\Sigma}^{-1}_{X_kX_k|D_k,\theta_k}\widetilde{\Sigma}_{X_kX_k|D_k,\theta_k}$ comprises diagonal blocks of size $|X_{kn}| \times |X_{kn}|$ with components of value 0 for $n = 1,\ldots,N$ and off-diagonal blocks of the form $\Sigma^{-1}_{X_{kn}X_{kn}|D_k,\theta_k}$ $\Sigma_{X_{kn}X_{kn'}|D_k,\theta_k}$ for $n,n' = 1,\ldots,N$ and $n \neq n'$. Any pair of robots $r \in \mathcal{V}_n$ and $r' \in \mathcal{V}_{n'}$ reside in different connected components of coordination graph $\mathcal{G}$ and are therefore not adjacent. So, by (4.13),

$$\max_{i,i'} \left| \left[\Sigma_{X_{kn}X_{kn'}|D_k,\theta_k} \right]_{ii'} \right| \leq \varepsilon \tag{B.1}$$

for $n,n' = 1,\ldots,N$ and $n \neq n'$. Using (4.18) and (B.1), each component in any off-diagonal block of $\widehat{\Sigma}^{-1}_{X_kX_k|D_k,\theta_k}\widetilde{\Sigma}_{X_kX_k|D_k,\theta_k}$ can be bounded as follows:

$$\max_{i,i'} \left| \left[\Sigma^{-1}_{X_{kn}X_{kn}|D_k,\theta_k}\Sigma_{X_{kn}X_{kn'}|D_k,\theta_k} \right]_{ii'} \right| \leq |X_{kn}|\,\xi\varepsilon \tag{B.2}$$

for $n,n' = 1,\ldots,N$ and $n \neq n'$. It follows from (B.2) that

$$\max_{i,i'} \left| \left[\widehat{\Sigma}^{-1}_{X_kX_k|D_k,\theta_k}\widetilde{\Sigma}_{X_kX_k|D_k,\theta_k} \right]_{ii'} \right| \leq \max_n |X_{kn}|\,\xi\varepsilon \leq \eta\xi\varepsilon\ . \tag{B.3}$$

The last inequality is due to $\max_n |X_{kn}| \leq \max_n |\mathcal{V}_n| \leq \eta$. Then,

$$\begin{aligned} \rho_k &\leq \left|\left|\widehat{\Sigma}^{-1}_{X_kX_k|D_k,\theta_k}\widetilde{\Sigma}_{X_kX_k|D_k,\theta_k}\right|\right|_2 \\ &\leq |X_k| \max_{i,i'} \left|\left[\widehat{\Sigma}^{-1}_{X_kX_k|D_k,\theta_k}\widetilde{\Sigma}_{X_kX_k|D_k,\theta_k}\right]_{ii'}\right| \\ &\leq |\mathcal{V}|\eta\xi\varepsilon\ . \end{aligned} \tag{B.4}$$

The first two inequalities are due to standard properties of matrix norm. The last inequality follows from (B.3).

The rest of this proof uses the following result of (Ipsen and Lee, 2011) that is revised to reflect our notations:

Theorem 5. *If* $|X_k|\rho_k^2 < 1$, *then* $\log\left|\Sigma_{X_kX_k|D_k,\theta_k}\right| \leq \log\left|\widehat{\Sigma}_{X_kX_k|D_k,\theta_k}\right| \leq \log\left|\Sigma_{X_kX_k|D_k,\theta_k}\right| - \log\left(1-|X_k|\rho_k^2\right)$ *for any* X_k.

Using Theorem 5 followed by (B.4),

$$\begin{aligned} \log\left|\widehat{\Sigma}_{X_kX_k|D_k,\theta_k}\right| - \log\left|\Sigma_{X_kX_k|D_k,\theta_k}\right| &\leq \log\frac{1}{1-|X_k|\rho_k^2} \\ &\leq \log\frac{1}{1-(|\mathcal{V}|^{1.5}\eta\xi\varepsilon)^2} \end{aligned} \tag{B.5}$$

for any X_k.

$$\begin{aligned} &\widetilde{\mathbb{H}}[Y_{\widetilde{X}}, Z_{\widetilde{X}}|y_D, z_D, \theta] - \widetilde{\mathbb{H}}[Y_{\widehat{X}}, Z_{\widehat{X}}|y_D, z_D, \theta] \\ &= \sum_{x\in\widetilde{X}} \mathbb{H}[Z_x|z_D,\theta] - \sum_{x\in\widehat{X}} \mathbb{H}[Z_x|z_D,\theta] \\ &\quad + \sum_{k=1}^{K} \mathbb{H}[Y_{\widetilde{X}_k}|\widehat{z}_{\widetilde{X}_k}, y_{D_k}, \theta_k] - \mathbb{H}[Y_{\widehat{X}_k}|\widehat{z}_{\widehat{X}_k}, y_{D_k}, \theta_k] \\ &\leq \sum_{x\in\widetilde{X}} \mathbb{H}[Z_x|z_D,\theta] - \sum_{x\in\widehat{X}} \mathbb{H}[Z_x|z_D,\theta] \\ &\quad + \sum_{k=1}^{K} \widehat{\mathbb{H}}[Y_{\widetilde{X}_k}|\widehat{z}_{\widetilde{X}_k}, y_{D_k}, \theta_k] - \mathbb{H}[Y_{\widehat{X}_k}|\widehat{z}_{\widehat{X}_k}, y_{D_k}, \theta_k] \\ &\leq \sum_{x\in\widehat{X}} \mathbb{H}[Z_x|z_D,\theta] - \sum_{x\in\widehat{X}} \mathbb{H}[Z_x|z_D,\theta] \\ &\quad + \sum_{k=1}^{K} \widehat{\mathbb{H}}[Y_{\widehat{X}_k}|\widehat{z}_{\widehat{X}_k}, y_{D_k}, \theta_k] - \mathbb{H}[Y_{\widehat{X}_k}|\widehat{z}_{\widehat{X}_k}, y_{D_k}, \theta_k] \\ &\leq \frac{K}{2}\log\frac{1}{1-(|\mathcal{V}|^{1.5}\eta\xi\varepsilon)^2} \end{aligned}$$

The first equality is due to (4.12). The first, second, and last inequalities follow from (4.14) and Theorem 5, (4.16), and (B.5), respectively.

B.2 Heuristics to Improve Gibbs Sampling

Gibbs sampling for estimating the component labels z_D is time-consuming because it has to evaluate $|D|K$ posterior probabilities (4.7) in every iterative sweep. We propose the following two heuristics to speed up its convergence.

Firstly, environmental sensing applications (El Saadi and Bah, 2007, Groves et al., 1998, Wei and Pengda, 2002) have shown that the magnitude of local mean $\overline{y}_x \triangleq \sum_{x' \in N_x} y_{x'}/|N_x|$ (where $x \in D$) is highly informative towards partitioning a non-stationary phenomenon into separate local areas with different locally stationary spatial correlation structures because the local means between different local areas tend to vary considerably. So, before Gibbs sampling, an informative prior set z_D of component labels can be determined by clustering the local means $\overline{y}_x$ concatenated with their corresponding feature vectors x for all $x \in D$. For clustering, we use the Gaussian mixture model of (Figueiredo and Jain, 2002) that can automatically select an appropriate number of clusters (i.e., K) based on the minimum message length criterion.

Secondly, during Gibbs sampling, observations residing deep within a local area comprising measurements induced by a GP component tend to yield highly certain component labels (i.e., with low entropy) and are thus very unlikely to change their labels. To reduce computations, they can be skipped during Gibbs sampling.

Appendix C

ANOVA-DCOP for Optimization

C.1 Proof of Proposition 4

From Assumption 2, we know that each term $f_I(x^{\mathcal{I}}) \sim \mathcal{N}(\mu_{x^{\mathcal{I}}|D_{t-1}}, \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}})$. And $\mu_{x^{\mathcal{I}}|D_{t-1}}$, $\Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}}$ are defined as:

$$\begin{aligned} \mu_{x^{\mathcal{I}}|D_{t-1}} &= \mu_{x^{\mathcal{I}}} + \kappa_I(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}(y_{D_{t-1}} - \mu_{D_{t-1}}) \\ \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}} &= \kappa_{\mathcal{I}}(x^{\mathcal{I}}, x^{\mathcal{I}}) - \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}}(D_{t-1}^{\mathcal{I}}, x^{\mathcal{I}}) \end{aligned} \tag{C.1}$$

By summing up all the terms in $\mathcal{U}$, we have:

$$\begin{aligned} \sum_{\mathcal{I}\in\mathcal{U}} \mu_{x^{\mathcal{I}}|D_{t-1}} &= \sum_{\mathcal{I}\in\mathcal{U}} \Big(\mu_{x^{\mathcal{I}}} + \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}(y_{D_{t-1}} - \mu_{D_{t-1}})\Big) \\ &= \sum_{\mathcal{I}\in\mathcal{U}} \mu_{x^{\mathcal{I}}} + \sum_{\mathcal{I}\in\mathcal{U}} \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}(y_{D_{t-1}} - \mu_{D_{t-1}}) \\ &= \sum_{\mathcal{I}\in\mathcal{U}} \mu_{x^{\mathcal{I}}} + \Big(\sum_{\mathcal{I}\in\mathcal{U}} \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})\Big)(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}(y_{D_{t-1}} - \mu_{D_{t-1}}) \\ \sum_{\mathcal{I}\in\mathcal{U}} \Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}} &= \sum_{\mathcal{I}\in\mathcal{U}} \Big(\kappa_{\mathcal{I}}(x^{\mathcal{I}}, x^{\mathcal{I}}) - \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}}(D_{t-1}^{\mathcal{I}}, x^{\mathcal{I}})\Big) \\ &= \sum_{\mathcal{I}\in\mathcal{U}} \kappa_{\mathcal{I}}(x^{\mathcal{I}}, x^{\mathcal{I}}) - \sum_{\mathcal{I}\in\mathcal{U}} \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}} + \eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}}(D_{t-1}^{\mathcal{I}}, x^{\mathcal{I}}) \end{aligned} \tag{C.2}$$

From Proposition 3 and the truncation approximation of ANOVA kernel, we know that $\mu_x = \sum_{\mathcal{I}\in\mathcal{U}} \mu_{x^{\mathcal{I}}}$ and $\widetilde{\kappa}(x,x) = \sum_{\mathcal{I}\in\mathcal{U}} \kappa_{\mathcal{I}}(x^{\mathcal{I}}, x^{\mathcal{I}})$ from the definition of ANOVA kernel. Using the property of ANOVA kernel between x and all $x_j \in D_{t-1}$, we have: $\widetilde{\kappa}(x, D_{t-1}) = \sum_{\mathcal{I}\in\mathcal{U}} \kappa_{\mathcal{I}}(x^{\mathcal{I}}, D_{t-1}^{\mathcal{I}})$.

Therefore,

$$\begin{aligned}\textstyle\sum_{\mathcal{I}\in\mathcal{U}}\mu_{x^{\mathcal{I}}|D_{t-1}} &= \textstyle\sum_{\mathcal{I}\in\mathcal{U}}\mu_{x^{\mathcal{I}}}+\Big(\sum_{\mathcal{I}\in\mathcal{U}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})\Big)(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}(y_{D_{t-1}}-\mu_{D_{t-1}})\\ &= \mu_x+\widetilde{\kappa}(x,D_{t-1})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}(y_{D_{t-1}}-\mu_{D_{t-1}})\\ &= \mu_{x|D_{t-1}}\end{aligned} \tag{C.3}$$

Additionally, Assumption 1 restricts the correlations only exists in the subsets of dimensions with size up to $k=2$ at any time step t. Consequently, We remove the correlations when $\mathcal{I}\neq\mathcal{I}'$ because it may introduce correlations more than $k=2$ dimensions:

$$\begin{aligned}&\textstyle\sum_{\mathcal{I}\in\mathcal{U}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}}(D_{t-1}^{\mathcal{I}},x^{\mathcal{I}})\\ =\ &\textstyle\sum_{\mathcal{I}\in\mathcal{U},\mathcal{I}'\in\mathcal{U},\mathcal{I}=\mathcal{I}'}\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}'}(D_{t-1}^{\mathcal{I}'},x^{\mathcal{I}'})\end{aligned} \tag{C.4}$$

It will leads to the linear decomposable form:

$$\begin{aligned}&\textstyle\sum_{\mathcal{I}\in\mathcal{U}}\Sigma_{x^{\mathcal{I}}x^{\mathcal{I}}|D_{t-1}}\\ =\ &\textstyle\sum_{\mathcal{I}\in\mathcal{U}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},x^{\mathcal{I}})-\sum_{\mathcal{I}\in\mathcal{U}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}}(D_{t-1}^{\mathcal{I}},x^{\mathcal{I}})\\ =\ &\textstyle\sum_{\mathcal{I}\in\mathcal{U}}\kappa_{\mathcal{I}}(x^{\mathcal{I}},x^{\mathcal{I}})-\sum_{\mathcal{I}\in\mathcal{U},\mathcal{I}'\in\mathcal{U},\mathcal{I}=\mathcal{I}'}\kappa_{\mathcal{I}}(x^{\mathcal{I}},D_{t-1}^{\mathcal{I}})(\widetilde{\Sigma}_{D_{t-1}D_{t-1}}+\eta^2\mathbf{I})^{-1}\kappa_{\mathcal{I}'}(D_{t-1}^{\mathcal{I}'},x^{\mathcal{I}'})\\ =\ &\Sigma_{xx|D_{t-1}}\end{aligned} \tag{C.5}$$

C.2 Proof of Theorem 3

Let $\{\pi_t\}_{t=1}^{\infty}$ and $x_t\triangleq\bigcup_i x_t^{(i)}$ denote a convergent series such that $\sum_{t=1}^{\infty}\pi_t^{-1}=1$ (e.g., $\pi_t\triangleq\pi^2t^2/6$) and the input location selected by maximizing $\widetilde{\varphi}_t(x)$ over a discretization Ω_t of $\mathcal{X}$ at time t, respectively. Then, let $\widetilde{x}_t\triangleq\bigcup_i\widetilde{x}_t^{(i)}$ be the true maximiser of $\widetilde{\varphi}_t(x)$ over $\mathcal{X}$, we further assume that $0\leq\widetilde{\varphi}_t(\widetilde{x}_t)-\widetilde{\varphi}_t(x_t)\leq\zeta_0t^{-1/2}$ where ζ_0 is a constant that does not depend on t. That is, for every step t, x_t is assumed to be $\zeta_0t^{-1/2}$-optimal. More specifically, for every time step t, we construct Ω_t as the Cartesian product of each dimension's discretization $\Omega_t^{(i)}$, i.e. $\Omega_t\triangleq\Omega_t^{(1)}\times\Omega_t^{(2)}\times\ldots\times\Omega_t^{(d)}$. For notational convenience, given a subset of dimension $\mathcal{I}\triangleq\{i_1,i_2,\ldots,i_k\}\in\mathcal{U}$, we also define (a) $\Omega_t^{\mathcal{I}}\triangleq\Omega_t^{(i_1)}\times\Omega_t^{(i_2)}\times\ldots\times\Omega_t^{(i_k)}$ as the

discretization of the sub-space of input spanning over dimensions $i_1, i_2, \ldots, i_k$; (b) $\omega_t^{\mathcal{I}} \triangleq |\Omega_t^{\mathcal{I}}|$ as its granularity and (c) $\omega_t \triangleq \max_{\mathcal{I}\in\mathcal{U}} \omega_t^{\mathcal{I}}$ as the maximum granularity at time t over all $\mathcal{I} \in \mathcal{U}$. In the following discussion, we will show how $\Omega_t^{(i)}$ can be constructed to achieve a tight upper-bound on the cumulative regret of our algorithm.

For generalization purpose, we derive the proof with respect to subsets with size less or equal to k and compact domain $x \in [0,r]^d$ which is consistent with the work of Srinivas et al. (2010). Our domain in the main thesis can be scaled to $[0,r]^d$ by setting $r = 2$ and shifting 1 unit.

Lemma 6. *For any $\delta \in (0,1)$, let $\beta_t \triangleq 2\log(\omega_t |\mathcal{U}| \pi_t/\delta)$, then with probability at least $1-\delta$, we have $\left|f(x) - \mu_{x|\mathcal{D}_{t-1}}\right| \leq \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}$ for all $x \in \Omega_t$ and $t \geq 1$.*

Proof. By definition, at any time step $t \geq 1$, given the previous observations $y_{\mathcal{D}_{t-1}}$, a subset of input dimensions $\mathcal{I} \in \mathcal{U}$ and a candidate $x^{\mathcal{I}}$, we have $f_{\mathcal{I}}\left(x^{\mathcal{I}}\right) \sim \mathcal{N}\left(\mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}, \sigma^2_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right)$. Hence, let $r \triangleq \left(f_{\mathcal{I}}\left(x^{\mathcal{I}}\right) - \mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right)/\sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}$, it trivially follows that $r \sim \mathcal{N}(0,1)$. Thus, applying the tail inequality $p(|r| > c) \leq \exp(-c^2/2)$ Srinivas et al. (2010) for the standard normal variable r with $c \triangleq \beta_t^{\frac{1}{2}}$ yields $\Pr\left(\left|f_{\mathcal{I}}\left(x^{\mathcal{I}}\right) - \mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right| > \beta_t^{\frac{1}{2}} \sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right) \leq \exp(-\beta_t/2)$ or equivalently:

$$\Pr\left(\left|f_{\mathcal{I}}\left(x^{\mathcal{I}}\right) - \mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right| \leq \beta_t^{\frac{1}{2}} \sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right) \geq 1 - \exp\left(-\beta_t/2\right) \tag{C.6}$$

for each tuple of $\left(\mathcal{I}, x^{\mathcal{I}}, t\right)$. Then, applying the union bound over $x^{\mathcal{I}} \in \Omega_t^{\mathcal{I}}$ consequently yields

$$\begin{aligned}\Pr\left(\forall x^{\mathcal{I}} \in \Omega_t^{\mathcal{I}}, \left|f_{\mathcal{I}}\left(x^{\mathcal{I}}\right) - \mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right| \leq \beta_t^{\frac{1}{2}} \sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right) &\geq 1 - \left|\Omega_t^{\mathcal{I}}\right| \exp\left(-\beta_t/2\right) \\ &\geq 1 - \omega_t \exp\left(-\beta_t/2\right) \end{aligned} \tag{C.7}$$

for each tuple of $(\mathcal{I}, t)$. Similarly, applying the union bound again over $\mathcal{I} \in \mathcal{U}$ and $t \in \mathbb{Z}^+$ subsequently implies

$$\Pr\left(\forall \mathcal{I} \in \mathcal{U}, t \geq 1, x^{\mathcal{I}} \in \Omega_t^{\mathcal{I}}, \left|f_{\mathcal{I}}\left(x^{\mathcal{I}}\right) - \mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right| \leq \beta_t^{\frac{1}{2}} \sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right) \geq 1 - |\mathcal{U}| \sum_{t=1}^{\infty} \omega_t \exp\left(-\beta_t/2\right)$$

Plugging $\beta_t \triangleq 2\log(\omega_t|\mathcal{U}|\pi_t/\delta)$ into the above inequality thus produces

$$\Pr\left(\forall \mathcal{I}\in\mathcal{U}, t\geq 1, x^{\mathcal{I}}\in\Omega_t^{\mathcal{I}}, \left|f_{\mathcal{I}}\left(x^{\mathcal{I}}\right)-\mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right|\leq \beta_t^{\frac{1}{2}}\sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right) \;\geq\; 1-\delta\sum_{t=1}^{\infty}\pi_t^{-1}=1-\delta \tag{C.8}$$

whose last equality holds due to our choice of $\{\pi_t\}_t$ such that $\sum_t \pi_t^{-1}=1$. That is, with probability at least $1-\delta$, $\left|f_{\mathcal{I}}\left(x^{\mathcal{I}}\right)-\mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right|\leq \beta_t^{\frac{1}{2}}\sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}$ simultaneously for all tuples of $(\mathcal{I},t,x^{\mathcal{I}})$.

Effectively, this means with probability at least $1-\delta$,

$$\begin{aligned}\left|f(x)-\mu_{x|\mathcal{D}_{t-1}}\right| &= \left|\sum_{\mathcal{I}\in\mathcal{U}}\left(f_{\mathcal{I}}\left(x^{\mathcal{I}}\right)-\mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right)\right| \\ &\leq \sum_{\mathcal{I}\in\mathcal{U}}\left|f_{\mathcal{I}}\left(x^{\mathcal{I}}\right)-\mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\right| \;\leq\; \beta_t^{1/2}\sum_{\mathcal{I}\in\mathcal{U}}\sigma_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}\end{aligned} \tag{C.9}$$

for all $t\in\mathbb{Z}^+$ and $x\in\Omega_t$. To elaborate, the first and last equalities in Eq. (C.9) are true due to our assumption that $\forall x, f(x)\triangleq\sum_{\mathcal{I}\in\mathcal{U}} f_{\mathcal{I}}\left(x^{\mathcal{I}}\right)$ and its implication on the decomposability of the predictive mean $\mu_{x|\mathcal{D}_{t-1}}=\sum_{\mathcal{I}\in\mathcal{U}}\mu_{x^{\mathcal{I}}|\mathcal{D}_{t-1}}$ (see Section X). Finally, the first and second inequalities of Eq. (C.9) follow directly from the triangle inequality and our previously established result in Eq. (C.8), thus completing our proof.

Lemma 7. *Given $\delta\in(0,1)$, $\mathcal{X}\subseteq[0,r]^d$ and $J\triangleq b\sqrt{\log\left(3a|\mathcal{U}|/\delta\right)}$, we have*

$$\Pr\left(\forall x\in\mathcal{X},\forall t\geq 1, \left|f(x)-\mu_{[x]_t|\mathcal{D}_{t-1}}\right|\leq Jr/t^3+\beta_t^{1/2}\sum_{\mathcal{I}\in\mathcal{U}}\sigma_{[x^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \;\geq\; 1-2\delta/3 \tag{C.10}$$

where $[x]_t$, $[x^{\mathcal{I}}]_t$ and $[x^i]_t$ denote the closest points (in terms of the L_1 distance) to x, $x^{\mathcal{I}}$ and $x^{(i)}$ in Ω_t, $\Omega_t^{\mathcal{I}}$ and $\Omega_t^{(i)}$, respectively; and $\beta_t=2k\log\left(dt^3\right)+2\log\left(3|\mathcal{U}|\pi_t/\delta\right)$ with $k=\max_{\mathcal{I}\in\mathcal{U}}|\mathcal{I}|$.

Proof. Applying the union bound over $\mathcal{I}\in\mathcal{U}$ to Eq. (5.24) (see Assumption 3) with $J\triangleq b\sqrt{\log\left(3a|\mathcal{U}|/\delta\right)}$ yields

$$\Pr\left(\forall\mathcal{I}\in\mathcal{U}, \sup_x\left|\frac{\partial f(x)}{\partial x^{\mathcal{I}}}\right|>J\right) \;\leq\; \sum_{\mathcal{I}\in\mathcal{U}}\Pr\left(\sup_x\left|\frac{\partial f(x)}{\partial x^{\mathcal{I}}}\right|>J\right) \;\leq\; a|\mathcal{U}|\exp\left(-\frac{J^2}{b^2}\right) \;=\; \frac{\delta}{3}$$

where the last equality follows from the definition of J. This immediately implies

$$\Pr\left(\forall \mathcal{I} \in \mathcal{U}, \sup_x \left|\frac{\partial f(x)}{\partial x^{\mathcal{I}}}\right| \leq J\right) \;\geq\; 1-\frac{\delta}{3} \tag{C.11}$$

with $J \triangleq b\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}$. Since $\mathcal{U}$ also contains all subsets that correspond to a single input dimension, the premise that $\forall \mathcal{I} \in \mathcal{U}, \sup_x \left|\frac{\partial f(x)}{\partial x^{\mathcal{I}}}\right| \leq J$ directly implies a J-Lipschitz condition on $f(x)$ with $J \triangleq b\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}$. Eq. (C.11) then guarantees that this is true with high probability:

$$\Pr\Big(\forall x,x' \in \mathcal{X}, \left|f(x)-f(x')\right| \;\leq\; b\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}\left\|x-x'\right\|_1\Big) \;\geq\; 1-\frac{\delta}{3}\,. \tag{C.12}$$

Also, since the input space $\mathcal{X} \subseteq [0,r]^d$ is assumed to be compact and bounded, we can construct a discretization $\Omega_t^{(i)}$ for each dimension $1 \leq i \leq d$ such that $\|x^{(i)} - [x^{(i)}]_t\| \leq r/\tau_t$ simply by extracting τ_t uniformly spaced points from $\mathcal{X}^{(i)}$. Consequently, we have $\|x-[x]_t\|_1 \triangleq \sum_{i=1}^d |x^{(i)} - [x^{(i)}]_t| \leq dr/\tau_t$. Thus, choosing $\tau_t = dt^3$ yields $\|x-[x]_t\|_1 \leq r/t^3$, $\omega_t^{\mathcal{I}} \triangleq \left|\Omega_t^{\mathcal{I}}\right| = (dt^3)^{|\mathcal{I}|}$ and $\omega_t \triangleq \max_{\mathcal{I}\in\mathcal{U}} \omega_t^{\mathcal{I}} = (dt^3)^k$ where $k \triangleq \max_{\mathcal{I}\in\mathcal{U}} |\mathcal{I}|$. Eq. (C.12) then implies

$$\Pr\Big(\forall x \in \mathcal{X}, \forall t \geq 1, |f(x)-f([x]_t)| \;\leq\; br\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}/t^3\Big) \;\geq\; 1-\frac{\delta}{3}\,. \tag{C.13}$$

On the other hand, since $\forall x \in \mathcal{X}$ and $t \geq 1$, we have $[x]_t \in \Omega_t$ and hence, Lemma 6 can be applied with $\delta/3$[1] to yield:

$$\Pr\left(\forall x \in \mathcal{X}, \forall t \geq 1, \left|f([x]_t) - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| \;\leq\; \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{[x^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \;\geq\; 1-\frac{\delta}{3} \tag{C.14}$$

[1] This means replacing δ with $\delta/3$ in the Lemma 6 and exploiting the resulting probabilistic statement in the context of Lemma 7.

where $\beta_t \triangleq 2\log\left(3\omega_t \left|\mathcal{U}\right| \pi_t/\delta\right) = 2k\log\left(dt^3\right) + 2\log\left(3\left|\mathcal{U}\right|\pi_t/\delta\right)$. Applying the union bound to combine Eq. (C.13) and Eq. (C.14) then implies

$$\begin{aligned}\left|f(x) - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| &\leq \left|f(x) - f\left([x]_t\right)\right| + \left|f\left([x]_t\right) - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| \\ &\leq br\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}/t^3 + \beta_t^{1/2}\sum_{\mathcal{I}\in\mathcal{U}}\sigma_{[x^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} \qquad \text{(C.15)}\end{aligned}$$

for all $x \in \mathcal{X}$ and $t \geq 1$ simultaneously with probability at least $1 - 2\delta/3$. Finally, substituting $J \triangleq b\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}$ into Eq. (C.15) completes our proof.

Lemma 8. *Given $\delta \in (0,1)$ and $\widetilde{\kappa}(\cdot,x)$ is L-Lipschitz with respect to all $x \in \mathcal{X}$ (see Assumption 3), we have*

$$\Pr\left(\forall t \geq 1, \forall x \in \mathcal{X}, \left|\mu_{x|\mathcal{D}_{t-1}} - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| \leq \frac{Lr}{\eta^2 t^{\frac{5}{2}}}\left(f(x_*) + \eta\sqrt{2\log(\pi_t/2\delta)}\right)\right) \geq 1-\delta$$

Proof. Using the expression of predictive mean in Eq.(X), we have

$$\left|\mu_{x|\mathcal{D}_{t-1}} - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| = \left|\left(\widetilde{\kappa}(x,\mathcal{D}_{t-1}) - \widetilde{\kappa}([x]_t,\mathcal{D}_{t-1})\right)\left(\widetilde{\kappa}(\mathcal{D}_{t-1},\mathcal{D}_{t-1}) + \eta^2\mathbf{I}\right)^{-1} y_{\mathcal{D}_{t-1}}\right| .$$

Then, let $\mathbf{u}^\top \triangleq \widetilde{\kappa}(x,\mathcal{D}_{t-1}) - \widetilde{\kappa}([x]_t,\mathcal{D}_{t-1})$, $\mathbf{A} \triangleq \widetilde{\kappa}(\mathcal{D}_{t-1},\mathcal{D}_{t-1}) + \eta^2\mathbf{I}$ and $\mathbf{v} \triangleq y_{\mathcal{D}_{t-1}}$, the above equation can be concisely rewritten as

$$\left|\mu_{x|\mathcal{D}_{t-1}} - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| = \left|\mathbf{u}^\top\mathbf{A}^{-1}\mathbf{v}\right| \leq \left\|\mathbf{u}\right\|_2\left\|\mathbf{A}^{-1}\mathbf{v}\right\|_2 \leq c\left\|\mathbf{u}\right\|_2\left\|\mathbf{v}\right\|_2 \qquad \text{(C.16)}$$

where $c \triangleq \left\|\mathbf{A}^{-1}\right\|_{\text{op}} \triangleq \inf\left\{c' \geq 0 : \forall\mathbf{v}', \left\|\mathbf{A}\mathbf{v}'\right\|_2 \leq c\left\|\mathbf{v}'\right\|_2\right\}$ denotes the operator norm of $\mathbf{A}^{-1}$ and the first inequality follows from the Cauchy-Schwarz inequality while the second inequality follows from our definition of c. Furthermore, since $\mathbf{A} - \eta^2\mathbf{I} = \widetilde{\kappa}(\mathcal{D}_{t-1},\mathcal{D}_{t-1}) \succ 0$ (i.e., positive definite) by our choice of kernel, its inverse's operator norm c is less than η^{-2} Kirthevasan et al. (2015). Hence,

$$\begin{aligned}\left|\mu_{x|\mathcal{D}_{t-1}} - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| &\leq \eta^{-2}\left\|\widetilde{\kappa}(x,\mathcal{D}_{t-1}) - \widetilde{\kappa}([x]_t,\mathcal{D}_{t-1})\right\|_2\left\|y_{\mathcal{D}_{t-1}}\right\|_2 \\ &\leq \frac{L}{\eta^2}\left\|x - [x]_t\right\|_2\left\|y_{\mathcal{D}_{t-1}}\right\|_2 \leq \frac{L}{\eta^2}\left\|x - [x]_t\right\|_1\left\|y_{\mathcal{D}_{t-1}}\right\|_2 \qquad \text{(C.17)}\end{aligned}$$

Also, by our construction of the discretization Ω_t of $\mathcal{X}$ at time step t (see Lemma 7), we have $\|x-[x]_t\|_1 \leq r/t^3$. Thus, Eq. (C.17) above can be further simplified as

$$\left|\mu_{x|\mathcal{D}_{t-1}} - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| \;\leq\; \frac{rL}{\eta^2 t^3}\left\|y_{\mathcal{D}_{t-1}}\right\|_2 . \tag{C.18}$$

On the other hand, note that $y_{x_\ell} = f(x_\ell) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0,\eta^2)$. This immediately implies $v \triangleq (y_{x_\ell} - f(x_\ell))/\eta \sim \mathcal{N}(0,1)$ and hence, applying the Gaussian tail bound $p(v > m_\ell) \leq (1/2)\exp(-m_\ell^2/2)$ Srinivas et al. (2010) with $m_\ell \triangleq \sqrt{2\log(\pi_\ell/2\delta)}$ yields

$$\Pr\left(y_{x_\ell} - f(x_\ell) > \eta\sqrt{2\log(\pi_\ell/2\delta)}\right) \;\leq\; \frac{\delta}{\pi_\ell} . \tag{C.19}$$

Besides, by definition, $f(x_*) \geq f(x_\ell)$ so $y_{x_\ell} - f(x_\ell) \leq y_{x_\ell} - f(x_*)$ and hence,

$$\Pr\left(y_{x_\ell} - f(x_*) > \eta\sqrt{2\log(\pi_\ell/2\delta)}\right) \;\leq\; \Pr\left(y_{x_\ell} - f(x_\ell) > \eta\sqrt{2\log(\pi_\ell/2\delta)}\right) \leq \frac{\delta}{\pi_\ell} \tag{C.20}$$

That is, for each x_ℓ selected by our algorithm at time step $\ell \geq 1$, Eq. (C.20) above guarantees that with probability at least $1-\delta/\pi_\ell$,

$$y_{x_\ell} \;\leq\; f(x_*) + \eta\sqrt{2\log(\pi_\ell/2\delta)} . \tag{C.21}$$

This means the chance that Eq. (C.21) holds simultaneously for all $\ell \geq 1$ is at least $1 - \delta\sum_{\ell=1}^{\infty}\pi_\ell^{-1} = 1-\delta$ (due to our choice of $\{\pi_\ell\}_\ell$ in Section X). When this happens, we have:

$$\begin{aligned}\left\|y_{\mathcal{D}_{t-1}}\right\|_2 \triangleq \left(\sum_{x_\ell\in\mathcal{D}_{t-1}} y_{x_\ell}^2\right)^{1/2} &\leq \left((t-1)\left(f(x_*) + \eta\sqrt{2\log(\pi_\ell/2\delta)}\right)^2\right)^{1/2} \\ &\leq t^{1/2}\left(f(x_*) + \eta\sqrt{2\log(\pi_t/2\delta)}\right)\end{aligned} \tag{C.22}$$

where the last step trivially follows because obviously $\pi_t \geq \pi_\ell$ when $t \geq \ell$ and $t-1 < t$. Plugging Eq. (C.22) into Eq. (C.18) above yields

$$\begin{aligned}\left|\mu_{x|\mathcal{D}_{t-1}} - \mu_{[x]_t|\mathcal{D}_{t-1}}\right| &\leq \frac{rL}{\eta^2 t^3}\left\|y_{\mathcal{D}_{t-1}}\right\|_2 \\ &\leq \frac{rL}{\eta^2 t^{5/2}}\left(f(x_*) + \eta\sqrt{2\log(\pi_t/2\delta)}\right) .\end{aligned} \tag{C.23}$$

Lastly, Eq. (C.23) holds with probability at least $1-\delta$ since Eq. (C.22) holds with probability at least $1-\delta$. This completes our proof.

Proof of Theorem 3

Proof. Given $\delta \in (0,1)$, it follows from Lemma 7 that with probability at least $1-2\delta/3$,

$$\begin{aligned} \left|f(x_*) - \mu_{[x_*]_t|\mathcal{D}_{t-1}}\right| &\leq Jr/t^3 + \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{[x_*^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} \quad \text{(C.24)} \\ \left|f(x_t) - \mu_{[x_t]_t|\mathcal{D}_{t-1}}\right| &\leq Jr/t^3 + \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} \quad \text{(C.25)} \end{aligned}$$

where $J = b\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}$ and $\beta_t = 2k\log\left(dt^3\right) + 2\log\left(3\left|\mathcal{U}\right|\pi_t/\delta\right)$ for all $t \geq 1$. This means

$$\begin{aligned} f(x_*) &\leq \mu_{[x_*]_t|\mathcal{D}_{t-1}} + Jr/t^3 + \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{[x_*^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} \quad \text{(C.26)} \\ f(x_t) &\geq \mu_{[x_t]_t|\mathcal{D}_{t-1}} - Jr/t^3 - \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} \quad \text{(C.27)} \end{aligned}$$

with probability at least $1-2\delta/3$. Hence, we can bound the instantaneous regret

$$\begin{aligned} r_t \triangleq f(x_*) - f(x_t) &\leq \mu_{[x_*]_t|\mathcal{D}_{t-1}} - \mu_{[x_t]_t|\mathcal{D}_{t-1}} + 2Jr/t^3 + \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \left(\sigma_{[x_*^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} + \sigma_{[x^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \\ &= \widetilde{\varphi}_t([x_*]_t) - \widetilde{\varphi}_t([x_t]_t) + 2Jr/t^3 + 2\beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} \\ &\leq \widetilde{\varphi}_t(x_t) + \zeta_0 t^{-\frac{1}{2}} - \widetilde{\varphi}_t([x_t]_t) + 2Jr/t^3 + 2\beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}} \\ &= \mu_{x_t|\mathcal{D}_{t-1}} - \mu_{[x_t]_t|\mathcal{D}_{t-1}} + 2Jr/t^3 + \zeta_0 t^{-\frac{1}{2}} + \beta_t^{1/2} \sum_{\mathcal{I}\in\mathcal{U}} \left(\sigma_{x_t^{\mathcal{I}}} + \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \end{aligned} \quad \text{(C.28)}$$

for all $t \geq 1$ with probability at least $1-2\delta/3$. Note that the second step of Eq. (C.28) is due to our definition of $\widetilde{\varphi}_t(x)$ in Section X while the third step follows from Assumption 4 which states that $\widetilde{\varphi}_t([x_*]_t) \leq \widetilde{\varphi}_t(\widetilde{x}_t) \leq \widetilde{\varphi}_t(x_t) + \zeta_0 t^{-\frac{1}{2}}$ (since $\widetilde{x}_t$ is the true maximizer of $\widetilde{\varphi}_t(x)$ and x_t

is assumed to be $\zeta_0 t^{-\frac{1}{2}}$-optimal). Finally, applying Lemma 8 with $\delta/3$[2], we have

$$\mu_{x_t|\mathcal{D}_{t-1}} - \mu_{[x_t]_t|\mathcal{D}_{t-1}} \;\leq\; \frac{Lr}{\eta^2 t^{5/2}}\left(f(x_*) + \eta\sqrt{2\log\left(3\pi_t/2\delta\right)}\right) \tag{C.29}$$

for all $t \geq 1$ with probability at least $1-\delta/3$. Combining both Eq. (C.28) and Eq. (C.29) via the union bound consequently yields

$$r_t \leq \frac{Lr}{\eta^2 t^{5/2}}\left(f(x_*) + \eta\sqrt{2\log\left(3\pi_t/2\delta\right)}\right) + 2Jr/t^3 + \zeta_0 t^{-\frac{1}{2}} + \beta_t^{1/2}\sum_{\mathcal{I}\in\mathcal{U}}\left(\sigma_{x_t^{\mathcal{I}}|\mathcal{D}_{t-1}} + \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \tag{C.30}$$

for all $t \geq 1$ with probability at least $1-\delta$ where $J = b\sqrt{\log\left(3a\left|\mathcal{U}\right|/\delta\right)}$ and $\beta_t = 2k\log\left(dt^3\right) + 2\log\left(3\left|\mathcal{U}\right|\pi_t/\delta\right)$. Therefore, summing both sides of the above inequality over $t = 1, 2, \ldots, T$, the cumulative regret can be bounded as

$$\begin{aligned} R_T \;&\triangleq\; \frac{1}{T}\sum_{t=1}^{T} r_t \;\leq\; C_2\left(a,b,\mathcal{U},T,L,\delta,\zeta_0,\eta\right) \;+\; \frac{1}{T}\sum_{t=1}^{T}\left(\beta_t^{1/2}\sum_{\mathcal{I}\in\mathcal{U}}\left(\sigma_{x_t^{\mathcal{I}}|\mathcal{D}_{t-1}} + \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right)\right) \\ &\leq\; C_2\left(a,b,\mathcal{U},T,L,\delta,\zeta_0,\eta\right) \;+\; \beta_T^{1/2}T^{-1}\sum_{t=1}^{T}\sum_{\mathcal{I}\in\mathcal{U}}\left(\sigma_{x_t^{\mathcal{I}}|\mathcal{D}_{t-1}} + \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \end{aligned} \tag{C.31}$$

with probability at least $1-\delta$ where $C_2\left(a,b,\mathcal{U},T,L,\delta,\zeta_0,\eta\right)$ is a constant that only depends on a, b, $|\mathcal{U}|$, T, L, δ, ζ_0 and η. Finally, to upper-bound the last term in the above equation, note that

$$\begin{aligned} \left(\sum_{t=1}^{T}\sum_{\mathcal{I}\in\mathcal{U}}\left(\sigma_{x_t^{\mathcal{I}}|\mathcal{D}_{t-1}} + \sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right)\right)^2 \;&\leq\; T\left|\mathcal{U}\right|\sum_{t=1}^{T}\sum_{\mathcal{I}\in\mathcal{U}}\left(\sigma^2_{x_t^{\mathcal{I}}|\mathcal{D}_{t-1}} + \sigma^2_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \\ &=\; T\left|\mathcal{U}\right|\sum_{t=1}^{T}\left(\sigma^2_{x_t|\mathcal{D}_{t-1}} + \sigma^2_{[x_t]_t|\mathcal{D}_{t-1}}\right) \end{aligned} \tag{C.32}$$

where the first step follows from Jensen inequality and the last step holds due to the decomposability of predictive variance (see Section X). To upper-bound $\sigma^2_{x_t|\mathcal{D}_{t-1}}$ and $\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}}$, we exploit the fact that $u^2/\log(1+u^2) \leq v^2/\log(1+v^2)$ when $u^2 \leq v^2$. Thus, letting $u^2 \triangleq \eta^{-2}\sigma^2_{x_t|\mathcal{D}_{t-1}} \leq \eta^{-2} \triangleq v^2$ (the last step goes through because by definition, $\sigma^2_{x_t|\mathcal{D}_{t-1}} \leq 1$)

[2]This means replacing δ with $\delta/3$ and applying the resulting probabilistic inequality to the Theorem 3.

and rearranging terms, we have

$$\sigma^2_{x_t|\mathcal{D}_{t-1}} \;\leq\; \log\left(1+\eta^{-2}\sigma^2_{x_t|\mathcal{D}_{t-1}}\right)/\log\left(1+\eta^{-2}\right) \;=\; C_1\log\left(1+\eta^{-2}\sigma^2_{x_t|\mathcal{D}_{t-1}}\right) \tag{C.33}$$

and similarly, using the exact argument for $\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}}$ yields

$$\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}} \;\leq\; \log\left(1+\eta^{-2}\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}}\right)/\log\left(1+\eta^{-2}\right) \;=\; C_1\log\left(1+\eta^{-2}\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}}\right) \tag{C.34}$$

Summing both sides of Eq. (C.33) and Eq. (C.34) over $t = 1, 2, \ldots, T$ yields

$$\sum_{t=1}^{T}\sigma^2_{x_t|\mathcal{D}_{t-1}} \;\leq\; 2C1\sum_{t=1}^{T}\frac{1}{2}\log\left(1+\eta^{-2}\sigma^2_{x_t|\mathcal{D}_{t-1}}\right) \;=\; 2C_1\mathbb{I}(y_{\mathcal{D}_T};f_{\mathcal{D}_T}) \;\leq\; 2C_1\gamma_T \tag{C.35}$$

where $\gamma_T \triangleq \max_{\mathcal{A}\subseteq\mathcal{X}:|\mathcal{A}|=T}\mathbb{I}(y_{\mathcal{A}};f_{\mathcal{A}})$ denotes the maximum information gain over the course of T sampling steps as defined in Srinivas et al. (2010). The second step of Eq. (C.35) above is also a borrowed result of Srinivas et al. (2010). Again, using the exact argument for $\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}}$ also yields $\sum_{t=1}^{T}\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}} \leq 2C_1\gamma_T$. Hence, combining these results, we have

$$\sum_{t=1}^{T}\left(\sigma^2_{x_t|\mathcal{D}_{t-1}}+\sigma^2_{[x_t]_t|\mathcal{D}_{t-1}}\right) \;\leq\; 4C_1\gamma_T \tag{C.36}$$

Plugging Eq. (C.36) into Eq. (C.32) consequently yields

$$\sum_{t=1}^{T}\sum_{\mathcal{I}\in\mathcal{U}}\left(\sigma_{x_t^{\mathcal{I}}|\mathcal{D}_{t-1}}+\sigma_{[x_t^{\mathcal{I}}]_t|\mathcal{D}_{t-1}}\right) \;\leq\; 2\sqrt{C_1T|\mathcal{U}|\gamma_T} \tag{C.37}$$

Lastly, plugging Eq. (C.37) into Eq. (C.31) yields

$$R_T \;\triangleq\; \frac{1}{T}\sum_{t=1}^{T}r_t \;\leq\; C_2(a,b,\mathcal{U},T,L,\delta,\zeta_0,\eta) \;+\; 2\beta_T^{1/2}T^{-1/2}\sqrt{C_1|\mathcal{U}|\gamma_T} \tag{C.38}$$

Appendix D

Useful Results

D.1 Matrix Inverse Lemma

For a positive definite matrix

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{D.1}$$

We have the following identity:

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \tag{D.2}$$

D.2 Union Bound

Theorem 9. *Let $A_1, \ldots, A_n$ to be a countable set of events, then*

$$p(\bigcup_{i=1}^{n} A_i) \leq \sum_{i=1}^{n} p(A_i) \tag{D.3}$$

D.3 Jensen Inequality

Theorem 10. *Let X be a random variable. If f is a convex function, then $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$.*

For example:

$$\sqrt{\sum_i x_i^2} \leq \sum_i x_i \tag{D.4}$$

D.4 Gaussian Tail Bound

Theorem 11. *Let X be a normal random variable: $X \sim \mathcal{N}(0,1)$. Then, we have*

$$p(x > c) \leq \frac{1}{2} e^{-\frac{c^2}{2}} \tag{D.5}$$

and

$$p(|x| > c) \leq e^{-\frac{c^2}{2}} \tag{D.6}$$

D.5 Riemann Zeta Function

Definition 6. *The Riemann zeta function is defined for any complex number s with real part > 1 by the following formula:*

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} \tag{D.7}$$

In special case $s = 2$:

$$\zeta(s = 2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \tag{D.8}$$

D.6 Frobenius Norm

Definition 7. *The Frobenius norm, is matrix norm of an $m \times n$ matrix A defined as the square root of the sum of the absolute squares of its elements,*

$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} |a_{ij}|^2} \tag{D.9}$$

D.7 Operator Norm

Definition 8. *Given two normed vector spaces V and W (over the same base field, either the real numbers R or the complex numbers C), a linear map $A : V \to W$ is continuous if and only if there exists a real number c such that*

$$||Av|| \leq c||v|| \, for \, \forall v \in V \tag{D.10}$$

Correspondingly the operator norm can be defined as:

$$||A||_{op} = \inf\{c \geq 0 : ||Av|| \leq c||v|| \, for \, \forall v \in V\} \tag{D.11}$$