ALGORITHMIC AND GRAPH-THEORETIC APPROACHES FOR OPTIMAL SENSOR SELECTION IN LARGE-SCALE SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Lintao Ye

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Shreyas Sundaram, Chair

School of Electrical Engineering, Purdue University

Dr. Jianghai Hu

School of Electrical Engineering, Purdue University

Dr. Xiaojun Lin

School of Electrical Engineering, Purdue University

Dr. Inseok Hwang

School of Aeronautics and Astronautics, Purdue University

Dr. Sandip Roy

School of Electrical Engineering and Computer Science, Washington State University

Approved by:

Dr. Dimitrios Peroulis

Head of the School Graduate Program

To my parents.

ACKNOWLEDGMENTS

It was my fortune to have many people who helped me during the journey of my study at Purdue in the past five years. Without their support, this work would never be possible.

First and foremost, I would like to thank my Ph.D. advisor Prof. Shreyas Sundaram. It was him who gave me the chance to start my Ph.D. study. I was impressed by his enthusiasm for research and his expertise in a vast array of subjects. I appreciate the time and patience with which he taught me things not only about research but also lessons that will be useful for my future career. During our weekly meetings, he was always ready to provide me with helpful feedback, and guide me towards the right direction. He also helped me improve my writing skills, and shared his experience in doing presentations. In short, I could never imagine a better doctoral advisor.

I would like to thank all my committee members: Professors Jianghai Hu, Xiaojun Lin, Inseok Hwang, and Sandip Roy. They provided valuable feedback on this work. I also had a great time collaborating with Prof. Philip E. Paré from Purdue University. As a graduate student at Purdue, I had the opportunity to take many great courses from different schools and departments. I am grateful to all my teachers and teaching assistants, who helped me build a solid background for my research. I am also indebted to all the staff of the ECE department, who create an enjoyable environment for working and studying.

I thank all my lab mates: Ashish, Aritra, Amritha, Kananart, Mustafa, Nathan, Lei, Russel, Tong, Maria, Nadia, and Baike, both for sharing laughter and maintaining a congenial research atmosphere. I will miss the group gatherings that we have had. I also would like to thank some wonderful people that I have come across in West Lafayette: Mingming, Huayi, Yibiao, Sen, Shuqi, Jie, Yechen, Yilun, Li, Xinwu,

Xuan, Ziqiao, Ahmad, Nikhil, and many others. Thank you for giving me company and suggestions, and making my life in West Lafayette more colorful.

Finally, I have to thank my parents. Without their unconditional support along the way, I could never make these achievements. I am also grateful to them for letting me make my own decisions and always standing behind me. Their encouragement helped me keep moving forward. My heartfelt thank you also goes to all my family members.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

ABSTRACT

Ye, Lintao Ph.D., Purdue University, December 2020. Algorithmic and Graph-Theoretic Approaches for Optimal Sensor Selection in Large-Scale Systems. Major Professor: Shreyas Sundaram.

Using sensor measurements to estimate the states and parameters of a system is a fundamental task in understanding the behavior of the system. Moreover, as modern systems grow rapidly in scale and complexity, it is not always possible to deploy sensors to measure all of the states and parameters of the system, due to cost and physical constraints. Therefore, selecting an optimal subset of all the candidate sensors to deploy and gather measurements of the system is an important and challenging problem. In addition, the systems may be targeted by external attackers who attempt to remove or destroy the deployed sensors. This further motivates the formulation of resilient sensor selection strategies. In this thesis, we address the sensor selection problem under different settings as follows.

First, we consider the optimal sensor selection problem for linear dynamical systems with stochastic inputs, where the Kalman filter is applied based on the sensor measurements to give an estimate of the system states. The goal is to select a subset of sensors under certain budget constraints such that the trace of the steady-state error covariance of the Kalman filter with the selected sensors is minimized. We characterize the complexity of this problem by showing that the Kalman filtering sensor selection problem is NP-hard and cannot be approximated within any constant factor in polynomial time for general systems. We then consider the optimal sensor attack problem for Kalman filtering. The Kalman filtering sensor attack problem is to attack a subset of selected sensors under certain budget constraints in order to maximize the trace of the steady-state error covariance of the Kalman filter with sensors after

the attack. We show that the same results as the Kalman filtering sensor selection problem also hold for the Kalman filtering sensor attack problem. Having shown that the general sensor selection and sensor attack problems for Kalman filtering are hard to solve, our next step is to consider special classes of the general problems. Specifically, we consider the underlying directed network corresponding to a linear dynamical system and investigate the case when there is a single node of the network that is affected by a stochastic input. In this setting, we show that the corresponding sensor selection and sensor attack problems for Kalman filtering can be solved in polynomial time. We further study the resilient sensor selection problem for Kalman filtering, where the problem is to find a sensor selection strategy under sensor selection budget constraints such that the trace of the steady-state error covariance of the Kalman filter is minimized after an adversary removes some of the deployed sensors. We show that the resilient sensor selection problem for Kalman filtering is NP-hard, and provide a pseudo-polynomial-time algorithm to solve it optimally.

Next, we consider the sensor selection problem for binary hypothesis testing. The problem is to select a subset of sensors under certain budget constraints such that a certain metric of the Neyman-Pearson (resp., Bayesian) detector corresponding to the selected sensors is optimized. We show that this problem is NP-hard if the objective is to minimize the miss probability (resp., error probability) of the Neyman-Pearson (resp., Bayesian) detector. We then consider three optimization objectives based on the Kullback-Leibler distance, J-Divergence and Bhattacharyya distance, respectively, in the hypothesis testing sensor selection problem, and provide performance bounds on greedy algorithms when applied to the sensor selection problem associated with these optimization objectives.

Moving beyond the binary hypothesis setting, we also consider the setting where the true state of the world comes from a set that can have cardinality greater than two. A Bayesian approach is then used to learn the true state of the world based on the data streams provided by the data sources. We formulate the Bayesian learning data source selection problem under this setting, where the goal is to minimize the

cost spent on the data sources such that the learning error is within a certain range. We show that the Bayesian learning data source selection is also NP-hard, and provide greedy algorithms with performance guarantees.

Finally, in light of the COVID-19 pandemic, we study the parameter estimation measurement selection problem for epidemics spreading in networks. Here, the measurements (with certain costs) are collected by conducting virus and antibody tests on the individuals in the epidemic spread network. The goal of the problem is then to optimally estimate the parameters (i.e., the infection rate and the recovery rate of the virus) in the epidemic spread network, while satisfying the budget constraint on collecting the measurements. Again, we show that the measurement selection problem is NP-hard, and provide approximation algorithms with performance guarantees.

1. INTRODUCTION

A fundamental task in understanding the behavior of a system is to estimate the states and parameters of the system using measurements (resp., data streams) from sensors (resp., data sources). Moreover, as the size of the system increases, it not always possible to measure all the states and parameters of the system, due to the fact that collecting those measurements incurs certain costs. This motivates the problem of selecting an optimal set of sensors in order to optimize certain metrics of the estimate based on the measurements from the selected sensors. Equivalently, the problem can be formulated as minimizing the cost spent on collecting the measurements such that the estimation error is within a certain range. The sensor sensor selection problem has attracted much attention from researchers from different fields, including the control (e.g., [1,2]), signal processing (e.g., [3,4]) and computer science community (e.g., [5]).

Moreover, in the case of large-scale critical infrastructure systems, the sensors that have been selected and deployed on the systems are also susceptible to a variety of potential attacks, including false data injection attacks (e.g., [6]) and Denial-of-Service (DoS) attacks (e.g., [7]). One class of DoS attacks corresponds to removing a set of installed sensors from the system, i.e., the measurements of the attacked sensors are rendered unusable (e.g., [8,9]). We also consider this type of attack in this work. Specifically, we study the problem of attacking the installed sensors (by removing a subset of them) under given attack budget constraints in order to maximally degrade the estimation performance. This problem is referred to as the sensor attack problem.

Combining the sensor selection and attack problems together, we study the resilient sensor selection problem. Specifically, we consider the scenario where a strategic attacker can attack a subset of the sensors selected by the designer. The goal (of the designer) is then to find a resilient sensor selection (under budget constraints)

in order to optimize the estimation performance corresponding to the sensors that survive the attack.

In this thesis, we systematically address the sensor selection problem under different settings: 1) Kalman filtering for linear dynamical systems with stochastic inputs; 2) hypothesis testing for signal detection; 3) Bayesian learning; 4) parameter estimation in epidemic spread networks. For each of these problems, we first characterize the computational complexity of the problem, thereby identifying fundamental limitations for any algorithm for such problems. We then identify special classes of the above problems that can be solved optimally using polynomial-time algorithms, and provide polynomial-time approximation algorithms to solve general instances of the problem with theoretical performance guarantees. In what follows, we provide a brief overview of our results for each of these settings, and delve into our main contributions.

1.1 Overview of Results

1.1.1 Sensor Selection and Attack for Kalman Filtering

One specific instance of the design-time sensor selection problem arises in the context of linear Gauss-Markov systems, where the corresponding Kalman filter (with the selected sensors) is used to estimate the states of the systems (e.g., [1,10–12]). In Chapter 3, we study the problem of selecting a subset of sensors (under given selection budget constraints) to minimize the trace of the steady-state error covariance (also known as the mean square estimation error) of the corresponding Kalman filter. We also investigate the problem of attacking the selected sensors (under given attack budget constraints) to maximize the trace of the steady-state error covariance of the Kalman filter associated with the sensors after the attack. We refer to these two problems as the Kalman Filtering Sensor Selection (KFSS) problem and the Kalman Filtering Sensor Attack (KFSA) problem, respectively.

In Chapter 3, we show that the KFSS (resp., KFSA) problem is NP-hard, and there are no polynomial-time constant-factor approximation algorithms for the KFSS (resp., KFSA) problem in general. In other words, there are no polynomial-time algorithms for any instances of the KFSS (resp., KFSA) problem that can find a sensor selection (resp., sensor attack) that is always guaranteed to yield a mean square estimation error (MSEE) that is within any constant finite factor of the MSEE for the optimal selection (resp., attack) (if P $\neq$ NP). The above result immediately implies that there is no performance guarantee for greedy algorithms for the KFSS (resp., KFSA) problem. We further show explicitly that greedy algorithms, which are widely used to solve NP-hard optimization problems, can provide arbitrarily poor performance for the KFSS (resp., KFSA) problem.

The above complexity results motivate us to consider special instances of the KFSS (resp., KFSA) problem, in order to seek efficient algorithms to solve the problem. In Chapter 4, we consider the underlying directed network associated with a linear dynamical system. Specifically, the states of the system represent nodes in a directed network, and interact according to the topology of the network. The nodes of the network are possibly affected by stochastic inputs. Such networked systems with stochastic inputs have received much attention from researchers recently (e.g., [13–17]). Moreover, we focus on the case where there is a single node of the network that is affected by a stochastic input. Our model encompasses diffusion networks, which arise in many different areas, including information and influence diffusion over social networks [18], spreading of diseases in populations [19] and diffusion of chemicals in certain environments [20]. Thus, we study the sensor selection problem and the sensor attack problem for Kalman filtering for networked systems where there is a single node in the network that has a stochastic input. We refer to these two problems as the Graph-based Kalman Filtering Sensor Selection (GKFSS) problem and Graph-based Kalman Filtering Sensor Attack (GKFSA) problem, respectively. Having shown in Chapter 3 that there are no polynomial-time constant-factor approximation algorithms for any instances of the sensor selection problem or

the sensor attack problem (if P $\neq$ NP), we show in Chapter 4 that the GKFSS problem and the GKFSA problem can be solved in polynomial time, leveraging the graph structure in the GKFSS and GKFSA problems.

In Chapter 4, we also study the resilient sensor selection problem for Kalman filtering in the same setting, which we refer to as the Resilient Graph-based Kalman Filtering Sensor Selection (RGKFSS) problem. Specifically, the problem is to find a resilient sensor selection strategy under the budget constraints in order to minimize the trace of the steady-state error covariance of the Kalman filter corresponding to the sensors that are remaining after the attack. Again, we show that the RGKFSS problem is NP-hard, and propose a pseudo-polynomial-time algorithm to solve it optimally, using the insights obtained from the GKFSS and GKFSA problems.

1.1.2 Sensor Selection for Hypothesis Testing

Another instance of the sensor selection problem arises in binary hypothesis testing for signal detection (e.g., [3, 21]). There are several detectors for signal detection, including, for instance, the Neyman-Pearson detector and the Bayesian detector [22]. Specifically, the goal of the hypothesis testing sensor selection problem is to select a subset of sensors (under a given budget constraint) such that a certain metric of the detection performance of the detector is optimized. In Chapter 5, we first consider the miss probability (resp., error probability) in the Neyman-Pearson detector (resp., Bayesian detector) corresponding to the selected sensors as the optimization metric in the hypothesis testing sensor selection problem; the problems are referred to as the Neyman-Pearson Hypothesis testing Sensor Selection (NPHSS) and the Bayesian Hypothesis testing Sensor Selection (BHSS) problems. We show that the NPHSS and BHSS problems are NP-hard. Since the miss probability of the Neyman-Pearson detector and the error probability of the Bayesian detector do not yield closed form expressions in general, we further consider three optimization metrics in the hypothesis testing sensor selection problem, which are based on the Kullback-Leibler (KL)

distance, J-Divergence and Bhattacharyya distance, respectively. We refer to the resulting sensor selection problems as the KL Distance Sensor Selection (KLDSS) problem, the J-Divergence Sensor Selection (JDSS) problem, and the Bhattacharyya Distance Sensor Selection (BDSS) problem, respectively. While we show that the KLDSS, JDSS and BDSS problems are still NP-hard, we leverage the closed form expressions of the objective functions in these problems, and provide a greedy algorithm to solve the problems with provable performance guarantees.

1.1.3 Data Source Selection for Bayesian Learning

In Chapter 6, we generalize our analysis for the binary hypothesis testing setting to a general setting where the true state of the world comes from a set that can have cardinality greater than two. Under this setting, a central task in machine learning is to learn the true state of the world based on data streams provided by data sources. Here, we do not restrict ourselves to measurements (i.e., data streams) coming from sensors, since in practice the data streams can come from a variety of sources, including experiment outcomes [23], medical tests [24], and sensor measurements [5], etc. We then consider tackling this task using the classic Bayesian learning rule, where we start with a prior belief about the true state of the world and update our belief based on the data streams from the data sources (e.g., [25]). The (steady-state) learning performance is then captured by the difference between the belief obtained from the Bayesian learning rule and the true state of the world. Following the arguments in previous chapters, we formulate the Bayesian Learning Data Source Selection (BLDS) problem, where the goal is to minimize the cost spent on the selected data sources while ensuring that the error of the learning process is within a prescribed range. Similarly, we show that the BLDS is NP-hard, and can be solved using a standard greedy algorithm with performance guarantees. Moreover, we propose a fast greedy algorithm to solve the BLDS problem that improves the running times of the stan-

dard greedy algorithm, and achieves performance guarantees that are comparable to those of the standard greedy algorithm.

1.1.4 Measurement Selection for Parameter Estimation in Epidemic Spread Networks

In Chapter 7, we apply our analysis for the sensor selection problems in previous chapters to models of epidemics spreading over networks which have been widely studied by researchers from different fields (e.g., [18, 26–30]). There are two key parameters that govern such models: the infection rate of a given node, and the recovery rate of that node. In the case of a novel virus, these parameters may not be known a priori, and must be identified or estimated from gathered data, including for instance the number of infected and recovered individuals in the network at certain points of time. For instance, in the COVID-19 pandemic, when collecting the data on the number of infected individuals or the number of recovered individuals in the network, one possibility is to perform virus or antibody tests on the individuals, with each test incurring a cost. Therefore, in the problem of parameter estimation in epidemic spread networks, it is important and of practical interest to take the costs of collecting the data (i.e., measurements) into account in the problem formulation.

The above discussions motivate us to consider the measurement selection problem for parameter estimation problem in epidemic spread networks, which shares natural similarities to the sensor selection problems that we studied in the previous chapters. Note that measurements are collected using sensors in the sensor selection problem, while the measurements are gathered by performing virus or antibody tests on the individuals in the measurement selection problem. Under the setting when exact measurements of the infected and recovered proportions of the population at certain nodes in the network can be obtained, we formulate the Parameter Identification Measurement Selection (PIMS) problem as minimizing the cost spent on collecting the measurements, while ensuing that the parameters can be uniquely identified (within

a certain time interval in the epidemic dynamics). In settings where the measurements are stochastic (thereby precluding exact identification of the parameters), we formulate the Parameter Estimation Measurement Selection (PEMS) problem. The goal is to optimize certain estimation performance metrics based on the collected measurements, while satisfying the budget on collecting the measurements. We show that the PIMS and PEMS problems are NP-hard. Leveraging the network structure in the PIMS problem, we propose an approximation algorithm for the PIMS problem with performance guarantees. Moreover, for the PEMS problem, we also provide a greedy algorithm with performance guarantees.

1.1.5 General Contributions

Although the algorithms that we study in this thesis are proposed for specific problems as we described above, our analysis of the algorithms can be extended to more general problems. For example, our analysis of the greedy algorithm applied to the KDLSS, JDSS and BDSS problems studied in Chapter 5 and the greedy algorithm applied to the PEMS problem studied in Chapter 7 generalizes the analysis of the greedy algorithms for submodular function maximization under budget constraints (e.g., [31]) to nonsubmodular function maximization under a budget constraint. Moreover, our analysis of the fast greedy algorithm proposed for the BLDS problem studied in Chapter 6 also works for the general submodular set covering problem (e.g., [32]).

1.2 Other Related Problems

There are other problems studied in the literature that share some common points with the sensor selection problem. For instance, a dual problem to the design-time sensor selection problem that has been studied by researchers from the control community is the design-time actuator selection problem (e.g., [33, 34]). If different sets of sensors can be selected at different time steps, the problem is known as the sensor scheduling problem (e.g., [35–37]). In computer science, researchers have studied the

subset selection problem which is related to feature selection and dictionary selection in machine learning (e.g., [38–40]), where the problem is to select a subset of random variables from a large set in order to obtain the best linear prediction of another random variable of interest. All these problems are about selecting elements (under given constraints) from a candidate set in order to optimize a certain objective corresponding to the selected elements.

1.3 Thesis Outline

In Chapter 3, we characterize fundamental limitations of any polynomial-time algorithm for the Kalman filtering sensor selection (resp., attack) problem. In Chapter 4, we first identify special instances of the Kalman filtering sensor selection (resp., attack) problem in a networked system setting, and provide polynomial-time algorithms to solve these instances optimally. We then propose a pseudo-polynomial-time algorithm to solve the resilient Kalman filtering sensor selection problem optimally in the networked system setting. In Chapter 5, we show that the binary hypothesis testing sensor selection problem is NP-hard, and provide greedy algorithms to solve them with performance guarantees. In Chapter 6, we show that the Bayesian learning data source selection problem is NP-hard, and propose greedy algorithms to solve it with performance guarantees. In Chapter 7, we show that parameter estimation measurement selection problem in epidemic spread networks is NP-hard, and provide approximation algorithms to solve it with theoretical guarantees using the network structure in the problem.

This work was supported in part by NSF grant CMMI-1635014.

2. BACKGROUND

2.1 Notation and Terminology

The sets of integers, real numbers and complex numbers are denoted as $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{C}$, respectively. The set of integers that are greater than (resp., greater than or equal to) $a \in \mathbb{R}$ is denoted as $\mathbb{Z}_{>a}$ (resp., $\mathbb{Z}_{\geq a}$). For $x \in \mathbb{C}$, let $|x|$ denote its magnitude. The set of real numbers that are greater than (resp., greater than or equal to) $b \in \mathbb{R}$ is denoted as $\mathbb{R}_{>b}$ (resp., $\mathbb{R}_{\geq b}$). For any $x \in \mathbb{R}$, let $\lceil x \rceil$ denote the least integer greater than or equal to x, and let $\lfloor x \rfloor$ denote the greatest integer that is less than or equal to x. For any integer $n \geq 1$, denote $[n] \triangleq \{1, \ldots, n\}$. For a set $\mathcal{A}$, let $|\mathcal{A}|$ be its cardinality. A function $\varphi_1(n)$ is $O(\varphi_2(n))$ if there exist positive constant c and N such that $|\varphi_1(n)| \leq c|\varphi_2(n)|$ for all $n \geq N$.

For a square matrix $P \in \mathbb{R}^{n \times n}$, let P^T, $\mathrm{rank}(P)$, $\mathrm{rowspace}(P)$, $\det(P)$ and $\mathrm{trace}(P)$ (or $\mathrm{tr}(P)$) be its transpose, rank, rowspace, determinant and trace, respectively. The eigenvalues of P are ordered with nonincreasing magnitude (i.e., $|\lambda_1(P)| \geq \cdots \geq |\lambda_n(P)|$). The maximum (resp., minimum) value of the diagonal elements of P is denoted as $d_1(P)$ (resp., $d_n(P)$). Let P_{ij} (or $(P)_{ij}$) denote the element in the ith row and jth column of P, and let P_i (or $(P)_i$) denote the ith row of P. Denote a diagonal matrix $P \in \mathbb{R}^{n \times n}$ as $\mathrm{diag}(P_{11}, \ldots, P_{nn})$. The identity matrix with dimension $n \times n$ is denoted as I_n. The zero matrix with dimension $m \times n$ is denoted as $\mathbf{0}_{m \times n}$; the subscripts are dropped if the dimension is clear from the context. The set of n by n positive definite (resp., positive semi-definite) matrices is denoted as $\mathbb{S}^n_{++}$ (resp., $\mathbb{S}^n_{+}$). A positive semi-definite matrix P is denoted by $P \succeq \mathbf{0}$; $P \succeq Q$ if $P - Q \succeq \mathbf{0}$. In a matrix, $*$ denotes elements of the matrix that are of no interest. For a vector v, let v_i (or $(v)_i$) denote the ith element of v; define the support of v to be $\mathrm{supp}(v) = \{i : v_i \neq 0\}$. Denote the Euclidean norm of v by $\|v\|_2$. Define $\mathbf{e}_i$ to

be a row vector where the ith element is 1 and all the other elements are zero; the dimension of the vector can be inferred from the context. Define $\mathbf{1}_n$ to be a column vector of dimension n with all the elements equal to 1. The set of $0-1$ indicator vectors of dimension n is denoted as $\{0,1\}^n$.

For a random vector $X \in \mathbb{R}^n$, let $\mathbb{E}[X] \in \mathbb{R}^n$ and $\mathrm{Cov}(X) = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T] \in \mathbb{R}^{n \times n}$ denote its mean vector and covariance, respectively. For two random vectors $X \in \mathbb{R}^{n_1}$ and $Y \in \mathbb{R}^{n_2}$, let $\Sigma_{XY} = \mathrm{Cov}(X,Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])^T] \in \mathbb{R}^{n_1 \times n_2}$ denote the cross-covariance between them. The probability density function of a Gaussian distribution with mean $\theta \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{S}_+^n$ is denoted as $\mathcal{N}(\theta, \Sigma)$.

2.2 Review of Complexity Theory

We review the following fundamental concepts from complexity theory [41].

Definition 2.2.1 *A polynomial-time algorithm for a problem is an algorithm that returns a solution to the problem in a polynomial (in the size of the problem) number of computations.*

Definition 2.2.2 *A decision problem is a problem whose answer is "yes" or "no". The set P contains those decision problems that can be solved by a polynomial-time algorithm. The set NP contains those decision problems whose "yes" answers can be verified using a polynomial-time algorithm.*

Definition 2.2.3 *An optimization problem is a problem whose objective is to maximize or minimize a certain quantity, possibly subject to constraints.*

Definition 2.2.4 *A problem $\mathcal{P}_1$ is NP-complete if (a) $\mathcal{P}_1 \in NP$ and (b) for any problem $\mathcal{P}_2$ in NP, there exists a polynomial-time algorithm that converts (or "reduces") any instance of $\mathcal{P}_2$ to an instance of $\mathcal{P}_1$ such that the answer to the constructed instance of $\mathcal{P}_1$ provides the answer to the instance of $\mathcal{P}_2$. $\mathcal{P}_1$ is NP-hard if it satisfies (b), but not necessarily (a).*

The above definition indicates that if one had a polynomial-time algorithm for an NP-complete (or NP-hard) problem, then one could solve *every* problem in NP in polynomial time. Specifically, suppose we had a polynomial-time algorithm to solve an NP-hard problem $\mathcal{P}_1$. Then, given any problem $\mathcal{P}_2$ in NP, one could first reduce any instance of $\mathcal{P}_2$ to an instance of $\mathcal{P}_1$ in polynomial time (such that the answer to the constructed instance of $\mathcal{P}_1$ provides the answer to the given instance of $\mathcal{P}_2$), and then use the polynomial-time algorithm for $\mathcal{P}_1$ to obtain the answer to $\mathcal{P}_2$.

The above discussion also reveals that to show that a given problem $\mathcal{P}_1$ is NP-hard, one simply needs to show that any instance of some other *NP-hard (or NP-complete)* problem $\mathcal{P}_2$ can be reduced to an instance of $\mathcal{P}_1$ in polynomial time (in such a way that the answer to the constructed instance of $\mathcal{P}_1$ provides the answer to the given instance of $\mathcal{P}_2$). For then, an algorithm for $\mathcal{P}_1$ can be used to solve $\mathcal{P}_2$, and hence, to solve all problems in NP (by NP-hardness of $\mathcal{P}_2$).

The following is a fundamental result in computational complexity theory [41].

Lemma 2.2.1 *If $P \neq NP$, there is no polynomial-time algorithm for any NP-complete (or NP-hard) problem.*

For optimization problems that are NP-hard, polynomial-time approximation algorithms are of particular interest. The definition of a constant-factor approximation algorithm is given as follows.

Definition 2.2.5 *A constant-factor approximation algorithm for an optimization problem is an algorithm that always returns a solution within a certain constant factor of the optimal solution.*

3. COMPLEXITY AND APPROXIMABILITY OF OPTIMAL SENSOR SELECTION AND ATTACK FOR KALMAN FILTERING

3.1 Introduction

In large-scale control system design, the number of sensors or actuators that can be selected and installed is typically limited by a design budget constraint. Moreover, system designers often need to select among a set of possible sensors and actuators, with varying qualities and costs. Consequently, a key problem is to determine an appropriate set of sensors or actuators in order to achieve certain objectives. This problem has recently received much attention from researchers (e.g., [33, 34, 42–48]). In the context of linear Gauss-Markov systems, where the corresponding Kalman filter (with the selected sensors) is used to estimate the states of the systems (e.g., [1, 12]). The problem then becomes how to select sensors dynamically (at run-time) or select sensors statically (at design-time) to minimize certain metrics of the corresponding Kalman filter. The former scenario is known as the sensor scheduling problem, where different sets of sensors can be chosen at different time steps (e.g., [35–37]). The latter scenario is known as the design-time sensor selection problem, where the set of the selected sensors is not allowed to change over time (e.g., [10, 11, 49]).

Since these problems are NP-hard in general (e.g., [2]), approximation algorithms that provide solutions within a certain factor of the optimal are then proposed to tackle them. Among these approximation algorithms, greedy algorithms have been widely used (e.g, [5, 50]), since such algorithms have provable performance guarantees if the cost function is submodular or supermodular (e.g., [40, 51]).

Additionally, in many applications, the sensors that have been selected and installed on the system are susceptible to a variety of potential attacks. For instance,

an adversary (attacker) can inject false data to corrupt the state estimation, which is known as the false data injection attack (e.g., [6,52,53]). Another type of attack is the Denial-of-Service (DoS) attack, where an attacker tries to diminish or eliminate the installed sensors' capacity to achieve its expected objective [7], including, for example, wireless jamming (e.g., [54,55]) and memory exhaustion through flooding (e.g., [56]). One class of DoS attacks corresponds to removing a set of installed sensors from the system, i.e., the measurements of the attacked sensors are not used. This was also studied in [8] and [9], and will be the type of attack that we consider here.

Related Work

In [12] and [57], the authors studied the design-time sensor selection problem for discrete-time linear time-varying systems over a finite time horizon. The objective is to minimize the number of selected sensors while guaranteeing a certain level of performance (or alternatively, to minimize the estimation error with a cardinality constraint on the selected sensors). The authors then analyzed the performance of greedy algorithms for this problem. However, their results cannot be directly applied to the problems that we consider here, since we aim to optimize the steady-state estimation error.

The papers [11] and [2] considered the same design-time sensor selection problem as the one we consider here. In [11], the authors expressed the problem as a semidefinite program. However, they did not provide theoretical guarantees on the performance of the proposed algorithm. The paper [2] showed that the problem is NP-hard and gave examples showing that the cost function is not submodular (or supermodular) in general. The authors also provided upper bounds on the performance of algorithms for the problem; these upper bounds were functions of the system matrices. Although [2] showed via simulations that greedy algorithms performed well for several randomly generated systems, the question of whether such algorithms (or other polynomial-time algorithms) could provide constant-factor approximation ratios for

the problem was left open. We resolve this question in this chapter by showing that there does not exist any (polynomial-time) constant-factor approximation algorithm for this problem.

In [8], the authors studied the problem of attacking a given observation selection in Gaussian process regression [39] to maximize the *posteriori* variance of the predictor variable. It was shown that this problem is NP-hard. Moreover, they also gave an instance of this problem such that a greedy algorithm for finding an optimal attack will perform arbitrarily poorly. In [58], the authors considered the scenario where the attacker can target a different set of sensors at each time step to maximize certain metrics of the error covariance of the Kalman filter at the final time step. Some suboptimal algorithms were provided with simulation results. Different from [8] and [58], we study the problem where the attacker removes a set of installed sensors to maximize the trace of the steady-state error covariance of the Kalman filter associated with the surviving sensors, and provide fundamental limitations on achievable performance by any possible algorithm for this problem.

In this chapter, we consider both the sensor selection problem and the sensor attack problem for Kalman filtering of discrete-time linear dynamical systems. First, we study the problem of selecting sensors at design-time on the system (under given selection budget constraints) to minimize the trace of either the steady-state *a priori* or *a posteriori* error covariance of the corresponding Kalman filter. We refer to these problems as the priori and posteriori *Kalman Filtering Sensor Selection (KFSS)* problems, respectively. Second, we investigate the problem of attacking the installed sensors (by removing a subset of them, under given attack budget constraints) to maximize the trace of either the steady-state *a priori* or *a posteriori* error covariance of the Kalman filter associated with the surviving sensors. These problems are denoted as the priori and posteriori *Kalman Filtering Sensor Attack (KFSA)* problems, respectively.

Summary of Results

The results in this chapter are summarized as follows. First, we show that the priori and posteriori KFSS problems are NP-hard and *there are no polynomial-time constant-factor approximation algorithms for these problems* (unless P = NP). In other words, there are no polynomial-time algorithms that can find a sensor selection that is always guaranteed to yield a mean square estimation error (MSEE) that is within any constant finite factor of the MSEE for the optimal selection. This stands in stark contrast to other sensor selection problems studied in the literature, which leveraged submodularity of their associated cost functions to provide greedy algorithms with constant-factor approximation ratios [57]. Second, we show that the same results hold for the priori and posteriori KFSA problems, i.e., these problems are NP-hard and *there are no polynomial-time constant-factor approximation algorithms for these problems* (unless P = NP). Our inapproximability results above immediately imply that greedy algorithms cannot provide constant-factor guarantees for our problems. We further show in this chapter how greedy algorithms can provide arbitrarily poor performance even for very small instances (with three states) of the priori and posteriori KFSS (resp., KFSA) problems.

The results presented in this chapter were published in [59, 60].

3.2 Problem Formulation

Consider the discrete-time linear system

$$x[k+1] = Ax[k] + w[k], \tag{3.1}$$

where $x[k] \in \mathbb{R}^n$ is the system state, $w[k] \in \mathbb{R}^n$ is a zero-mean white noise process with $\mathbb{E}[w[k](w[k])^T] = W$ for all $k \in \mathbb{Z}_{\geq 0}$, and $A \in \mathbb{R}^{n \times n}$ is the system dynamics matrix. The initial condition $x[0]$ is assumed to be a random vector with mean $\bar{x}_0 \in \mathbb{R}^n$ and covariance $\Pi_0 \in \mathbb{S}^n_+$. We also assume that the pair $(A, W^{\frac{1}{2}})$ is stabilizable.

Consider a set $\mathcal{Q}$ that contains q sensors. Each sensor $i \in \mathcal{Q}$ provides a measurement of the system of the form

$$y_i[k] = C_i x[k] + v_i[k], \tag{3.2}$$

where $C_i \in \mathbb{R}^{s_i \times n}$ is the state measurement matrix for sensor i, and $v_i[k] \in \mathbb{R}^{s_i}$ is a zero-mean white noise process. Denote $y[k] \triangleq \left[(y_1[k])^T \ \cdots \ (y_q[k])^T\right]^T$, $C \triangleq \left[C_1^T \ \cdots \ C_q^T\right]^T$ and $v[k] \triangleq \left[(v_1[k])^T \ \cdots \ (v_q[k])^T\right]^T$. Thus, the output provided by all sensors together is given by

$$y[k] = Cx[k] + v[k], \tag{3.3}$$

where $C \in \mathbb{R}^{s \times n}$ and $s = \sum_{i=1}^{q} s_i$. We denote $\mathbb{E}[v[k](v[k])^T] = V$ and assume that the system noise and the measurement noise are uncorrelated, i.e., $\mathbb{E}[v[k](w[j])^T] = \mathbf{0}$, $\forall k, j \in \mathbb{Z}_{\geq 0}$, and $x[0]$ is independent of $w[k]$ and $v[k]$, $\forall k \in \mathbb{Z}_{\geq 0}$.

3.2.1 The Sensor Selection Problem

Consider the scenario where there are no sensors initially selected (i.e., deployed) on the system. Instead, the system designer must select a subset of sensors from $\mathcal{Q}$ on the system. Each sensor $i \in \mathcal{Q}$ has a cost $h_i \in \mathbb{R}_{\geq 0}$; define the cost vector $h \triangleq \begin{bmatrix} h_1 & \cdots & h_q \end{bmatrix}^T$. The designer has a budget $H \in \mathbb{R}_{\geq 0}$ that can be spent on choosing sensors from $\mathcal{Q}$.

After a set of sensors is selected and installed, the Kalman filter is applied to provide an estimate of the states using the measurements from the installed sensors. We define a vector $\mu \in \{0,1\}^q$ as the indicator vector of the selected sensors selected, where $\mu_i = 1$ if and only if sensor $i \in \mathcal{Q}$ is selected. Let $C(\mu)$ denote the measurement matrix of the installed sensors indicated by μ, i.e., $C(\mu) \triangleq \begin{bmatrix} C_{i_1}^T & \cdots & C_{i_p}^T \end{bmatrix}^T$, where $\text{supp}(\mu) = \{i_1, \ldots, i_p\}$. Similarly, let $V(\mu)$ denote the measurement noise covariance matrix of the installed sensors, i.e., $V(\mu) = \mathbb{E}[\tilde{v}[k](\tilde{v}[k])^T]$, where $\tilde{v}[k] = \left[(v_{i_1}[k])^T \ \cdots \ (v_{i_p}[k])^T\right]^T$. Let $\Sigma_{k/k-1}(\mu)$ and $\Sigma_{k/k}(\mu)$ denote the *a priori* error covariance matrix and the *a posteriori* error covariance matrix of the Kalman filter at time

step k, respectively, when the sensors indicated by μ are installed. We will use the following result [61].

Lemma 3.2.1 *Suppose the pair $(A, W^{\frac{1}{2}})$ is stabilizable. For a given indicator vector μ, both $\Sigma_{k/k-1}(\mu)$ and $\Sigma_{k/k}(\mu)$ will converge to finite limits $\Sigma(\mu)$ and $\Sigma^*(\mu)$, respectively, as $k \to \infty$ if and only if the pair $(A, C(\mu))$ is detectable.*

The limit $\Sigma(\mu)$ satisfies the *discrete algebraic Riccati equation (DARE)* [61]:

$$\Sigma(\mu) = A\Sigma(\mu)A^T + W - A\Sigma(\mu)C(\mu)^T\big(C(\mu)\Sigma(\mu)C(\mu)^T + V(\mu)\big)^{-1}C(\mu)\Sigma(\mu)A^T. \quad (3.4)$$

The limits $\Sigma(\mu)$ and $\Sigma^*(\mu)$ are coupled as

$$\Sigma(\mu) = A\Sigma^*(\mu)A^T + W. \quad (3.5)$$

The limit $\Sigma^*(\mu)$ of the a *posteriori* error covariance matrix satisfies the following equation [62]:

$$\Sigma^*(\mu) = \Sigma(\mu) - \Sigma(\mu)C(\mu)^T(C(\mu)\Sigma(\mu)C(\mu)^T + V(\mu))^{-1}C(\mu)\Sigma(\mu). \quad (3.6)$$

Note that we can either obtain $\Sigma^*(\mu)$ from $\Sigma(\mu)$ using Eq. (3.6) or by substituting Eq. (3.5) into Eq. (3.6) and solving for $\Sigma^*(\mu)$. The inverses in Eq. (3.4) and Eq. (3.6) are interpreted as pseudo-inverses if the arguments are not invertible.

For the case when the pair $(A, C(\mu))$ is not detectable, we define the limits $\Sigma(\mu) = +\infty$ and $\Sigma^*(\mu) = +\infty$. Moreover, for any sensor selection μ, we note from Lemma 3.2.1 that the limit $\Sigma(\mu)$ (resp., $\Sigma^*(\mu)$), if it exists, does not depend on $\bar{x}_0$ or Π_0. Thus, we can assume without loss of generality that $\bar{x}_0 = \mathbf{0}$ and $\Pi_0 = I_n$ in the sequel. The priori and posteriori Kalman Filtering Sensor Selection (KFSS) problems are defined as follows.

Problem 3.2.2 *(Priori and Posteriori KFSS Problems). Given a system dynamics matrix $A \in \mathbb{R}^{n\times n}$, a measurement matrix $C \in \mathbb{R}^{s\times n}$ containing all of the individual sensor measurement matrices, a system noise covariance matrix $W \in \mathbb{S}^n_+$, a sensor noise covariance matrix $V \in \mathbb{S}^s_+$, a cost vector $h \in \mathbb{R}^q_{\geq 0}$ and a budget $H \in \mathbb{R}_{\geq 0}$,*

the priori Kalman Filtering Sensor Selection (KFSS) problem is to find the sensor selection $\mu \in \{0,1\}^n$*, i.e., the indicator vector* μ *of the selected sensors, that solves*

$$\min_{\mu \in \{0,1\}^q} \ trace(\Sigma(\mu))$$
$$s.t.\ h^T \mu \leq H$$

where $\Sigma(\mu)$ *is given by Eq.* (3.4) *if the pair* $(A, C(\mu))$ *is detectable, and* $\Sigma(\mu) = +\infty$*, if otherwise. Similarly, the posteriori Kalman Filtering Sensor Selection (KFSS) problem is to find the sensor selection* $\mu \in \{0,1\}^n$ *that solves*

$$\min_{\mu \in \{0,1\}^q} \ trace(\Sigma^*(\mu))$$
$$s.t.\ h^T \mu \leq H$$

where $\Sigma^*(\mu)$ *is given by Eq.* (3.6) *if the pair* $(A, C(\mu))$ *is detectable, and* $\Sigma^*(\mu) = +\infty$*, if otherwise.*

3.2.2 The Sensor Attack Problem

Now consider the scenario where the set $\mathcal{Q}$ of sensors has already been selected and installed on the system. An adversary desires to attack a subset of sensors (i.e., remove a subset of sensors from the system), where each sensor $i \in \mathcal{Q}$ has an attack cost $\varpi_i \in \mathbb{R}_{\geq 0}$; define the cost vector $\varpi \triangleq \begin{bmatrix} \varpi_1 & \cdots & \varpi_q \end{bmatrix}^T$. We assume that the adversary has a budget $\Omega \in \mathbb{R}_{\geq 0}$, which is the total cost that can be spent on removing sensors from $\mathcal{Q}$.

After a subset of sensors are attacked (i.e., removed), the Kalman filter is then applied to estimate the states using the measurements from the surviving sensors (in the sense of minimizing the mean square estimation error). We define a vector $\nu \in \{0,1\}^q$ as the indicator vector of the attacked sensors, where $\nu_i = 1$ if and only if sensor $i \in \mathcal{Q}$ is attacked. Hence, the set of sensors that survive is $\mathcal{Q} \setminus \text{supp}(\nu)$. Define $v^c \in \{0,1\}^q$ to be the vector such that $\text{supp}(\nu^c) = \mathcal{Q} \setminus \text{supp}(\nu)$, i.e., $\nu_i^c = 1$ if and only if sensor $i \in \mathcal{Q}$ survives. Similarly to the sensor selection problem, we let $C(\nu^c)$ and $V(\nu^c)$ denote the measurement matrix and the measurement noise covariance matrix,

respectively, corresponding to ν^c. Furthermore, let $\Sigma_{k/k-1}(\nu^c)$ and $\Sigma_{k/k}(\nu^c)$ denote the a *priori* error covariance matrix and the a *posteriori* error covariance matrix of the Kalman filter at time step k, respectively. Denote $\lim_{k\to\infty}\Sigma_{k/k-1}(\nu^c) = \Sigma(\nu^c)$ and $\lim_{k\to\infty}\Sigma_{k/k}(\nu^c) = \Sigma^*(\nu^c)$ if the limits exist, according to Lemma 3.2.1. Note that Eq. (3.4)-(3.6) also hold if we substitute μ with ν^c.

For the case when the pair $(A, C(\nu^c))$ is not detectable, we define the limits $\Sigma(\nu^c) = +\infty$ and $\Sigma^*(\nu^c) = +\infty$. The priori and posteriori Kalman Filtering Sensor Attack (KFSA) problems are defined as follows.

Problem 3.2.3 *(Priori and Posteriori KFSA Problems). Given a system dynamics matrix $A \in \mathbb{R}^{n\times n}$, a measurement matrix $C \in \mathbb{R}^{s\times n}$, a system noise covariance matrix $W \in \mathbb{S}^n_+$, a sensor noise covariance matrix $V \in \mathbb{S}^s_+$, a cost vector $\varpi \in \mathbb{R}^q_{\geq 0}$ and a budget $\Omega \in \mathbb{R}_{\geq 0}$, the priori Kalman Filtering Sensor Attack (KFSA) problem is to find the sensor attack $\nu \in \{0,1\}^n$, i.e., the indicator vector ν of the attacked sensors, that solves*

$$\max_{\nu\in\{0,1\}^q} \ trace(\Sigma(\nu^c))$$
$$s.t.\ \varpi^T\nu \leq \Omega$$

where $\Sigma(\nu^c)$ is given by Eq. (3.4) if the pair $(A, C(\nu^c))$ is detectable, and $\Sigma(\nu^c) = +\infty$, if otherwise. Similarly, the posteriori Kalman Filtering Sensor Attack (KFSA) problem is to find the sensor attack $\nu \in \{0,1\}^n$ that solves

$$\max_{\nu\in\{0,1\}^q} \ trace(\Sigma^*(\nu^c))$$
$$s.t.\ \varpi^T\nu \leq \Omega$$

where $\Sigma^*(\nu^c)$ is given by Eq. (3.6) if the pair $(A, C(\nu^c))$ is detectable, and $\Sigma^*(\nu^c) = +\infty$, if otherwise.

It is also useful to note that although we focus on the optimal sensor problem and attack problems for Kalman filtering here, due to the duality between the Kalman filter and the linear quadratic regulator (LQR) [63], all of the analysis in this chapter will also apply if the priori KFSS and KFSA problems are rephrased as optimal

actuator selection and attack problems for LQR, respectively. We omit the details of the rephrasing in the interest of space.

Remark 3.2.4 *Our goal in this chapter is to show that for the priori and posteriori KFSS problems and the priori and posteriori KFSA problems, the optimal solutions cannot be approximated within any constant factor in polynomial time. To do this, it is sufficient for us to consider the special case when* $C_i \in \mathbb{R}^{1\times n}$, $\forall i \in \{1,\dots,q\}$, *i.e., each sensor provides a scalar measurement. Moreover, the sensor selection cost vector and the sensor attack cost vector are considered to be* $b = \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^T$ *and* $\omega = \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^T$, *respectively, i.e., the selection cost and the attack cost of each sensor are both equal to* 1. *By showing that the problems are inapproximable even for these special subclasses, we obtain that the general versions of the problems are inapproximable as well.*

3.3 Inapproximability of the KFSS and KFSA problems

In this section, we analyze the approximability of the KFSS and KFSA problems. We will start with a brief overview of some relevant concepts from the field of computational complexity, and then provide some preliminary lemmas that we will use in proving our results. That will lead into our characterizations of the complexity of KFSS and KFSA.

It was shown in [33] that the problem of selecting a subset of sensors to make the system detectable is NP-hard, which implies that KFSS is NP-hard using Lemma 3.2.1 as shown in [2]. In this chapter, we aim to show that the hardness of KFSS (resp., KFSA) does not solely come from selecting (resp., attacking) sensors to make the system detectable (resp., undetectable). To do this, we will show that there is no polynomial-time constant-factor approximation algorithm for KFSS (resp., KFSA) *even* when the corresponding system dynamics matrix A is *stable*, which guarantees the detectability of the system. Specifically, we consider a known NP-complete problem, and show how to reduce it to certain instances of KFSS (resp., KFSA)

(with stable A matrices) in polynomial time such that hypothetical polynomial-time constant-factor approximation algorithms for the latter problems can be used to solve the known NP-complete problem. Since we know from Lemma 2.2.1 that if P $\neq$ NP, there does not exist a polynomial-time algorithm for any NP-complete problem, we can conclude that if P $\neq$ NP, there is no polynomial-time constant-factor approximation algorithm for KFSS (resp., KFSA). We emphasize that our results do not imply that there is no polynomial-time constant-factor approximation algorithm for *specific* instances of KFSS (resp., KFSA). Rather, the result is that we cannot have such an algorithm for *all* instances of KFSS (resp., KFSA).

3.3.1 Preliminary Results

We will use the following results in our analysis (the proofs are provided in Sections 3.6.1 and 3.6.2, respectively).

Lemma 3.3.1 *Consider a discrete-time linear system as defined in Eq.* (3.1) *and Eq.* (3.3). *Suppose the system dynamics matrix is of the form* $A = diag(\lambda_1, \ldots, \lambda_n)$ *with* $0 \leq |\lambda_i| < 1$, $\forall i \in \{1, \ldots, n\}$, *the system noise covariance matrix* W *is diagonal, and the sensor noise covariance matrix* $V \in \mathbb{S}_+^q$. *Then, the following hold for all sensor selections* μ.

(a) For all $i \in \{1, \ldots, n\}$, $(\Sigma(\mu))_{ii}$ *and* $(\Sigma^*(\mu))_{ii}$ *satisfy*

$$W_{ii} \leq (\Sigma(\mu))_{ii} \leq \frac{W_{ii}}{1 - \lambda_i^2}, \tag{3.7}$$

and

$$0 \leq (\Sigma^*(\mu))_{ii} \leq \frac{W_{ii}}{1 - \lambda_i^2}, \tag{3.8}$$

respectively.

(b) If $\exists i \in \{1, \ldots, n\}$ *such that* $W_{ii} \neq 0$ *and the* i*th column of* C *is zero, then* $(\Sigma(\mu))_{ii} = (\Sigma^*(\mu))_{ii} = \frac{W_{ii}}{1-\lambda_i^2}$.

(c) If $V = \mathbf{0}_{q \times q}$ *and there exists* $i \in \{1, \ldots, n\}$ *such that* $\mathbf{e}_i \in$ *rowspace*$(C(\mu))$, *then* $(\Sigma(\mu))_{ii} = W_{ii}$ *and* $(\Sigma^*(\mu))_{ii} = 0$.

Lemma 3.3.2 *Consider a discrete-time linear system as defined in Eq.* (3.1) *and Eq.* (3.3)*. Suppose the system dynamics matrix is of the form* $A = diag(\lambda_1, 0, \ldots, 0) \in \mathbb{R}^{n\times n}$*, where* $0 < |\lambda_1| < 1$*, and the system noise covariance matrix is* $W = I_n$*.*

(a) Suppose the measurement matrix is of the form $C = \begin{bmatrix} 1 & \gamma \end{bmatrix}$ *with sensor noise variance* $V = \sigma_v^2$*, where* $\gamma \in \mathbb{R}^{1\times(n-1)}$ *and* $\sigma_v \in \mathbb{R}_{\geq 0}$*. Then, the MSEE of state* 1*, denoted as* Σ_{11}*, satisfies*

$$\Sigma_{11} = \frac{1 + \alpha^2\lambda_1^2 - \alpha^2 + \sqrt{(\alpha^2 - \alpha^2\lambda_1^2 - 1)^2 + 4\alpha^2}}{2}, \tag{3.9}$$

where $\alpha^2 \triangleq \|\gamma\|_2^2 + \sigma_v^2$*.*

(b) Suppose the measurement matrix is of the form $C = \begin{bmatrix} \mathbf{1}_{n-1} & \rho I_{n-1} \end{bmatrix}$ *with sensor noise covariance* $V = \mathbf{0}_{(n-1)\times(n-1)}$*, where* $\rho \in \mathbb{R}$*. Then, the MSEE of state* 1*, denoted as* Σ'_{11}*, satisfies*

$$\Sigma'_{11} = \frac{\lambda_1^2\rho^2 + n' - \rho^2 + \sqrt{(\rho^2 - \lambda_1^2\rho^2 - n')^2 + 4n'\rho^2}}{2n'}, \tag{3.10}$$

where $n' = n - 1$*.*

Moreover, if we view Σ_{11} *and* Σ'_{11} *as functions of* α^2 *and* ρ^2*, denoted as* $\Sigma_{11}(\alpha^2)$ *and* $\Sigma'_{11}(\rho^2)$*, respectively, then* $\Sigma_{11}(\alpha^2)$ *and* $\Sigma'_{11}(\rho^2)$ *are strictly increasing functions of* $\alpha^2 \in \mathbb{R}_{\geq 0}$ *and* $\rho^2 \in \mathbb{R}_{\geq 0}$*, with* $\lim_{\alpha\to\infty}\Sigma_{11}(\alpha^2) = \frac{1}{1-\lambda_1^2}$ *and* $\lim_{\rho\to\infty}\Sigma'_{11}(\rho^2) = \frac{1}{1-\lambda_1^2}$*, respectively.*

3.3.2 Inapproximability of the Priori and Posteriori KFSS Problems

In this section, we characterize the complexity of the priori and posteriori KFSS problems by showing that there are no polynomial-time algorithms that can always yield a solution that is within any constant factor of the optimal (unless P = NP). Specifically, consider any given instance of KFSS. For any given algorithm $\mathcal{A}$ (resp., $\mathcal{A}'$) of the priori (resp., posteriori) KFSS problem, we define the following ratios:

$$r_{\mathcal{A}}(\Sigma) \triangleq \frac{\text{trace}(\Sigma_{\mathcal{A}})}{\text{trace}(\Sigma_{opt})}, \tag{3.11}$$

and

$$r_{\mathcal{A}'}(\Sigma^*) \triangleq \frac{\text{trace}(\Sigma^*_{\mathcal{A}'})}{\text{trace}(\Sigma^*_{opt})}, \tag{3.12}$$

where Σ_{opt} (resp., Σ^*_{opt}) is the optimal solution to the priori (resp., posteriori) KFSS problem and $\Sigma_{\mathcal{A}}$ (resp., $\Sigma^*_{\mathcal{A}'}$) is the solution to the priori (resp., posteriori) KFSS problem given by algorithm $\mathcal{A}$ (resp., $\mathcal{A}'$).

In [2], the authors showed that there is an upper bound for $r_{\mathcal{A}}(\Sigma)$ (resp., $r_{\mathcal{A}'}(\Sigma^*)$) for any sensor selection algorithm $\mathcal{A}$ (resp., $\mathcal{A}'$), in terms of the system matrices. However, the question of whether it is possible to find a polynomial-time algorithm $\mathcal{A}$ (resp., $\mathcal{A}'$) that is guaranteed to provide an approximation ratio $r_{\mathcal{A}}(\Sigma)$ (resp., $r_{\mathcal{A}'}(\Sigma^*)$) that is *independent* of the system parameters has remained open up to this point. In particular, it is desirable to find *constant-factor* approximation algorithms, where the ratio $r_{\mathcal{A}}(\Sigma)$ (resp., $r_{\mathcal{A}'}(\Sigma^*)$) is upper-bounded by some (system-independent) constant. Here, we provide a negative result showing that for the priori (resp., posteriori) KFSS problem, there is no polynomial-time constant-factor approximation algorithm in general, i.e., for all polynomial-time algorithms $\mathcal{A}$ (resp., $\mathcal{A}'$) and $\forall K \in \mathbb{R}_{\geq 1}$, there are instances of the priori (resp., posteriori) KFSS problem where $r_{\mathcal{A}}(\Sigma) > K$ (resp., $r_{\mathcal{A}'}(\Sigma^*) > K$).

Remark 3.3.3 *Note that the "constant" in "constant-factor approximation algorithm" refers to the fact that the cost of the solution provided by the algorithm is upper-bounded by some (system-independent) constant times the cost of the optimal solution. The algorithm can, however, use the system parameters when finding the solution. For example, an optimal algorithm for the KFSS problem will be a* 1*-factor approximation, and would use the system matrices, sensor costs, and budget to find the optimal solution. Similarly, a polynomial-time* K*-factor approximation algorithm for KFSS would use the system parameters to produce a solution whose cost is guaranteed to be no more than* K *times the cost of the optimal solution. As indicated above, we will show that no such algorithm exists for any constant* K *(unless* $P = NP$*).*

To show the inapproximability of the priori KFSS problem, we relate it to the EXACT COVER BY 3-SETS (X3C) problem described below [41].

Definition 3.3.1 *(X3C) Given a finite set $D = \{d_1, \ldots, d_{3m}\}$ and a collection $\mathcal{C} = \{c_1, \ldots, c_\tau\}$ of 3-element subsets of D, an exact cover for D is a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ such that every element of D occurs in exactly one member of $\mathcal{C}'$.*

We will use the following result [41].

Lemma 3.3.4 *Given a finite set $D = \{d_1, \ldots, d_{3m}\}$ and a collection $\mathcal{C} = \{c_1, \ldots, c_\tau\}$ of 3-element subsets of D, the problem of determining whether $\mathcal{C}$ contains an exact cover for D is NP-complete.*

Remark 3.3.5 *Note that if $\tau < m$, there does not exist an exact cover for D. Hence, we assume $\tau \geq m$. Since each member in $\mathcal{C}$ is a subset of D with exactly 3 elements, if there exists an exact cover for D, then it must consist of exactly m members of $\mathcal{C}$.*

As argued in Remark 3.2.4, in order to show that the priori KFSS problem cannot be approximated within any constant factor in polynomial time, it is sufficient for us to show that certain special instances of this problem are inapproximable. Specifically, consider any instance of the X3C problem. Using the results in Lemmas 3.3.1-3.3.2, we will first construct an instance of the priori KFSS problem in polynomial time such that the difference between the solution to KFSS when the answer to X3C is "yes" and the solution to KFSS when the answer to X3C is "no" is large enough. Thus, we can then apply any hypothetical polynomial-time constant-factor approximation algorithm for the priori KFSS problem to the constructed priori KFSS instance and obtain the answer to the X3C instance. Since we know from Lemma 3.3.4 that the X3C problem is NP-complete, we obtain from Lemma 2.2.1 the following result.

Theorem 3.3.6 *If $P \neq NP$, then there is no polynomial-time constant-factor approximation algorithm for the priori KFSS problem.*

Proof Assume that there exists such an approximation algorithm $\mathcal{A}$, i.e., $\exists K \in \mathbb{R}_{\geq 1}$ such that $r_{\mathcal{A}}(\Sigma) \leq K$ for all instances of the priori KFSS problem, where $r_{\mathcal{A}}(\Sigma)$ is defined in Eq. (3.11). We will show that $\mathcal{A}$ can be used to solve the X3C problem, which will lead to a contradiction.

Given an arbitrary instance of the X3C problem described in Definition 3.3.1 and Lemma 3.3.4, for each element $c_i \in \mathcal{C}$, we define $g_i \in \mathbb{R}^{3m}$ to encode which elements of D are contained in c_i. Specifically, for $i \in \{1, 2, \ldots, \tau\}$ and $j \in \{1, 2, \ldots, 3m\}$, $(g_i)_j = 1$ if $d_j \in D$ is in c_i, and $(g_i)_j = 0$ otherwise. Denote $G \triangleq \begin{bmatrix} g_1 & \cdots & g_\tau \end{bmatrix}^T$. Thus $G^T x = \mathbf{1}_{3m}$ has a solution $x \in \{0,1\}^\tau$ such that x has m nonzero entries if and only if the answer to the X3C instance is "yes" [64].

Given the above instance of X3C, we then construct an instance of the priori KFSS problem as follows. Denote $Z = \lceil K \rceil (m+1)(\sigma_v^2 + 3)$, where we set $\sigma_v = 1$. Define the system dynamics matrix as $A = \mathrm{diag}(\lambda_1, 0, \ldots, 0) \in \mathbb{R}^{(3m+1)\times(3m+1)}$, where $\lambda_1 = \frac{Z-1/2}{Z}$. Note that $Z \in \mathbb{Z}_{>1}$ and $0 < \lambda_1 < 1$. The set $\mathcal{Q}$ is defined to contain $\tau + 1$ sensors with collective measurement matrix

$$C = \begin{bmatrix} 1 & \varepsilon \mathbf{1}_{3m}^T \\ \mathbf{0} & G \end{bmatrix}, \tag{3.13}$$

where G is defined based on the given instance of X3C as above. The constant ε is chosen as $\varepsilon = 2Z\left\lceil \sqrt{Z-1} \right\rceil + 1$. The system noise covariance matrix is set to be $W = I_{3m+1}$. The measurement noise covariance matrix is set as $V = \sigma_v^2 \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \frac{1}{\varepsilon^2} I_\tau \end{bmatrix}$. The sensor selection cost vector is set as $b = \mathbf{1}_{\tau+1}$, and the sensor selection budget is set as $B = m + 1$. Note that the sensor selection vector for this instance is denoted by $\mu \in \{0,1\}^{\tau+1}$. For the above construction, since the only nonzero eigenvalue of A is λ_1, we know from Lemma 3.3.1(a) that $\sum_{i=2}^{3m+1} (\Sigma(\mu))_{ii} = \sum_{i=2}^{3m+1} W_{ii} = 3m$ for all μ.

We claim that algorithm $\mathcal{A}$ will return a sensor selection vector μ such that $\mathrm{trace}(\Sigma(\mu)) \leq K(m+1)(\sigma_v^2 + 3)$ if and only if the answer to the X3C problem is "yes".

We prove the above claim as follows. Suppose that the answer to the instance of the X3C problem is "yes". Then $G^T x = \mathbf{1}_{3m}$ has a solution such that x has m nonzero entries. Denote the solution as x^* and denote $\text{supp}(x^*) = \{i_1, \ldots, i_m\}$. Define $\tilde{\mu}$ to be the sensor selection vector that indicates selecting the first and the (i_1+1)th to the (i_m+1)th sensors, i.e., sensors that correspond to rows C_1, $C_{i_1+1}, \ldots, C_{i_m+1}$ from (3.13). Since $G^T x^* = \mathbf{1}_{3m}$, we have $[1 \;\; -\varepsilon x^{*T}]C = \mathbf{e}_1$ for C defined in Eq. (3.13). Noting that $\text{supp}(x^*) = \{i_1, \ldots, i_m\}$, it then follows that $\mathbf{e}_1 \in \text{rowspace}(C(\tilde{\mu}))$. We can then perform elementary row operations on $C(\tilde{\mu})$ (which does not change the steady-state *a priori* error covariance matrix of the corresponding Kalman filter) and obtain $\Gamma C(\tilde{\mu}) \triangleq \tilde{C}(\tilde{\mu}) = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & * \end{bmatrix}$ with the corresponding measurement noise covariance $\Gamma V(\mu)\Gamma^T \triangleq \tilde{V}(\tilde{\mu}) = \begin{bmatrix} \sigma_v^2(m+1) & * \\ * & * \end{bmatrix}$, where $\Gamma = \begin{bmatrix} 1 & -\varepsilon \mathbf{1}_m^T \\ \mathbf{0} & I_m \end{bmatrix}$. Let $\tilde{\Sigma}$ denote the error covariance obtained from sensing matrix $(\tilde{C}(\tilde{\mu}))_1 = \mathbf{e}_1$ with measurement noise variance $\tilde{\sigma}_v^2 \triangleq \sigma_v^2(m+1)$, which corresponds to the first sensor in $\tilde{C}(\tilde{\mu})$. We then know from Lemma 3.3.2(a) that

$$\tilde{\Sigma}_{11} = \frac{1 + \tilde{\sigma}_v^2\lambda_1^2 - \tilde{\sigma}_v^2 + \sqrt{(\tilde{\sigma}_v^2 - \tilde{\sigma}_v^2\lambda_1^2 - 1)^2 + 4\tilde{\sigma}_v^2}}{2},$$

which further implies

$$\begin{aligned} \tilde{\Sigma}_{11} &\leq \frac{1 + \sqrt{(\tilde{\sigma}_v^2(1-\lambda_1^2))^2 - 2\tilde{\sigma}_v^2(1-\lambda_1^2) + 1 + 4\tilde{\sigma}_v^2}}{2} \\ &\leq \frac{1 + \sqrt{(\tilde{\sigma}_v^2(1-\lambda_1^2))^2 + 1 + 4\tilde{\sigma}_v^2}}{2} \\ &\leq \frac{1 + \sqrt{\tilde{\sigma}_v^4 + 4\tilde{\sigma}_v^2 + 4}}{2} \leq \frac{1 + \tilde{\sigma}_v^2 + 2}{2}. \end{aligned} \tag{3.14}$$

Using similar arguments to those above, we have $\sum_{i=2}^{3m+1} \tilde{\Sigma}_{ii} = 3m$. We then obtain from (3.14) that

$$\text{trace}(\tilde{\Sigma}) \leq \tilde{\sigma}_v^2 + 3 + 3m = (m+1)(\sigma_v^2 + 3). \tag{3.15}$$

Since adding more sensors does not increase the MSEE of the corresponding Kalman filter, we have from (3.15) $\text{trace}(\Sigma(\tilde{\mu})) \leq (m+1)(\sigma_v^2+3)$, which further implies

$\text{trace}(\Sigma(\mu^*)) \leq (m+1)(\sigma_v^2+3)$, where μ^* is an optimal sensor selection of the priori KFSS problem. Since $\mathcal{A}$ has approximation ratio K, it returns a sensor selection μ such that $\text{trace}(\Sigma(\mu)) \leq K(m+1)(\sigma_v^2+3)$.

Conversely, suppose that the answer to the X3C instance is "no". Then, for any union of $l \leq m$ ($l \in \mathbb{Z}_{\geq 0}$) subsets in $\mathcal{C}$, denoted as $\mathcal{C}_l$, there exist $\kappa \geq 1$ ($\kappa \in \mathbb{Z}$) elements in D that are not covered by $\mathcal{C}_l$, i.e., for any $l \leq m$ and $\mathcal{L} \triangleq \{i_1, \ldots, i_l\} \subseteq \{1, \ldots, \tau\}$, $G_{\mathcal{L}} \triangleq \begin{bmatrix} g_{i_1} & \cdots & g_{i_l} \end{bmatrix}^T$ has $\kappa \geq 1$ zero columns. We then show that $\text{trace}(\Sigma(\mu)) > K(m+1)(\sigma_v^2+3)$ for all sensor selections μ (that satisfy the budget constraint). We divide our arguments into two cases.

First, for any sensor selection μ_1 that does not select the first sensor, the first column of $C(\mu_1)$ is zero (from the form of C defined in Eq. (3.13)). We then know from Lemma 3.3.1(b) that $(\Sigma(\mu_1))_{11} = \frac{1}{1-\lambda_1^2}$. Hence, by our choice of λ_1, we have

$$\begin{aligned} (\Sigma(\mu_1))_{11} &= \frac{Z^2}{Z-1/4} > Z \geq K(m+1)(\sigma_v^2+3) \\ \Rightarrow \text{trace}(\Sigma(\mu_1)) &> K(m+1)(\sigma_v^2+3), \end{aligned} \tag{3.16}$$

where (3.16) follows from $\sum_{i=2}^{3m+1}(\Sigma(\mu_1))_{ii} = 3m > 0$ for all possible sensor selections.

Second, consider sensor selections μ_2 that select the first sensor. To proceed, we first assume that the measurement noise covariance is $V = \mathbf{0}_{(\tau+1)\times(\tau+1)}$. Denote $\text{supp}(\mu_2) = \{1, i_1, \ldots, i_l\}$, where $l \leq m$ and define $G(\mu_2) = \begin{bmatrix} g_{i_1-1} & \cdots & g_{i_l-1} \end{bmatrix}^T$. We then have

$$C(\mu_2) = \begin{bmatrix} 1 & \varepsilon \mathbf{1}_{3m}^T \\ \mathbf{0} & G(\mu_2) \end{bmatrix},$$

where $G(\mu_2)$ has $\kappa \geq 1$ zero columns. As argued in Lemma 3.6.1 in Section 3.6.3, there exists an orthogonal matrix $E \in \mathbb{R}^{(3m+1)\times(3m+1)}$ of the form $E = \left[\begin{smallmatrix} 1 & 0 \\ 0 & N \end{smallmatrix}\right]$ such that

$$\tilde{C}(\mu_2) \triangleq C(\mu_2)E = \begin{bmatrix} 1 & \varepsilon\gamma & \varepsilon\beta \\ \mathbf{0} & \mathbf{0} & \tilde{G}(\mu_2) \end{bmatrix}.$$

In the above expression, $\tilde{G}(\mu_2) \in \mathbb{R}^{l\times r}$ is of full column rank, where $r = \text{rank}(G(\mu_2))$. Furthermore, $\gamma \in \mathbb{R}^{1\times(3m-r)}$ and at least κ of its elements are 1's, and $\beta \in \mathbb{R}^{1\times r}$. We

then perform a similarity transformation on the system with E, which does not affect the trace of the steady-state *a priori* error covariance matrix of the corresponding Kalman filter,[1] and does not change A, W and V. We further perform additional elementary row operations to transform $\tilde{C}(\mu_2)$ into the matrix

$$\tilde{C}'(\mu_2) = \begin{bmatrix} 1 & \varepsilon\gamma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{G}(\mu_2) \end{bmatrix}.$$

Since A and W are both diagonal, and $V = \mathbf{0}$, we can obtain from Eq. (3.4) that the steady-state *a priori* error covariance corresponding to the sensing matrix $\tilde{C}'(\mu_2)$, denoted as $\tilde{\Sigma}'(\mu_2)$, is of the form

$$\tilde{\Sigma}'(\mu_2) = \begin{bmatrix} \tilde{\Sigma}_1'(\mu_2) & \mathbf{0} \\ \mathbf{0} & \tilde{\Sigma}_2'(\mu_2) \end{bmatrix},$$

where $\tilde{\Sigma}_1'(\mu_2) \in \mathbb{R}^{(3m+1-r)\times(3m+1-r)}$ satisfies

$$\tilde{\Sigma}_1'(\mu_2) = A_1\tilde{\Sigma}_1'(\mu_2)A_1^T + W_1 - A_1\tilde{\Sigma}_1'(\mu_2)\tilde{C}^T\left(\tilde{C}\tilde{\Sigma}_1'(\mu_2)\tilde{C}^T\right)^{-1}\tilde{C}\tilde{\Sigma}_1'(\mu_2)A_1^T,$$

where $A_1 = \mathrm{diag}(\lambda_1, 0, \ldots, 0) \in \mathbb{R}^{(3m+1-r)\times(3m+1-r)}$, $\tilde{C} = [1\ \varepsilon\gamma]$ and $W_1 = I_{3m+1-r}$. Denoting $\alpha^2 = \varepsilon^2\|\gamma\|_2^2 \geq \kappa\varepsilon^2 \geq \varepsilon^2$, we then obtain from Lemma 3.3.2(a) that

$$\begin{aligned}(\tilde{\Sigma}'(\mu_2))_{11} &= \frac{1+\alpha^2\lambda_1^2-\alpha^2+\sqrt{(\alpha^2-\alpha^2\lambda_1^2-1)^2+4\alpha^2}}{2} \\ &\geq \frac{1+\varepsilon^2\lambda_1^2-\varepsilon^2+\sqrt{(\varepsilon^2-\varepsilon^2\lambda_1^2-1)^2+4\varepsilon^2}}{2}. \end{aligned} \tag{3.17}$$

[1] This can be easily verified using Eq. (3.4) as E is an orthogonal matrix.

By our choices of λ_1 and ε, we have the following:

$$\begin{aligned}
&\varepsilon^2 > 4Z^2(Z-1) \;\Rightarrow\; (1-\frac{Z-1/4}{Z})\varepsilon^2 > Z^2 - Z \\
&\Rightarrow \varepsilon^2 > Z^2 + Z\varepsilon^2\frac{Z-1/4}{Z^2} - Z \\
&\Rightarrow \varepsilon^2 > Z^2 + Z(\varepsilon^2(1-\lambda_1^2) - 1) \\
&\Rightarrow (\varepsilon^2 - \varepsilon^2\lambda_1^2 - 1)^2 + 4\varepsilon^2 > (\varepsilon^2 - \varepsilon^2\lambda_1^2 - 1)^2 + 4Z^2 + 4Z(\varepsilon^2 - \varepsilon^2\lambda_1^2 - 1) \\
&\Rightarrow (\varepsilon^2 - \varepsilon^2\lambda_1^2 - 1)^2 + 4\varepsilon^2 > (2Z + \varepsilon^2 - \varepsilon^2\lambda_1^2 - 1)^2 \\
&\Rightarrow \sqrt{(\varepsilon^2 - \varepsilon^2\lambda_1^2 - 1)^2 + 4\varepsilon^2} > 2Z + \varepsilon^2 - \varepsilon^2\lambda_1^2 - 1 \\
&\Rightarrow \frac{1 + \varepsilon^2\lambda_1^2 - \varepsilon^2 + \sqrt{(\varepsilon^2 - \varepsilon^2\lambda_1^2 - 1)^2 + 4\varepsilon^2}}{2} > Z. \qquad (3.18)
\end{aligned}$$

Since $Z \geq K(m+1)(\sigma_v^2+3)$, (3.17) and (3.18) imply $(\tilde{\Sigma}'(\mu_2))_{11} > K(m+1)(\sigma_v^2+3)$, which further implies $\text{trace}(\tilde{\Sigma}'(\mu_2)) > K(m+1)(\sigma_v^2+3)$. Since $\text{trace}(\tilde{\Sigma}'(\mu_2)) = \text{trace}(\Sigma(\mu_2))$ as argued above, we obtain that $\text{trace}(\Sigma(\mu_2)) > K(m+1)(\sigma_v^2+3)$. We then note the fact that the MSEE of the Kalman filter with noiseless measurements is no greater than that with any noisy measurements (for fixed A, W and C), when the system noise and the measurement noise are uncorrelated. Therefore, for $V = \sigma_v^2 \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \frac{1}{\varepsilon^2} I_\tau \end{bmatrix}$, we also have $\text{trace}(\Sigma(\mu_2)) > K(m+1)(\sigma_v^2+3)$ for all μ_2.

It then follows from the above arguments that $\text{trace}(\Sigma(\mu)) > K(m+1)(\sigma_v^2+3)$ for all sensor selections μ, which implies that algorithm $\mathcal{A}$ would also return a sensor selection μ such that $\text{trace}(\Sigma(\mu)) > K(m+1)(\sigma_v^2+3)$. This completes the proof of the converse direction of the claim above.

Hence, it is clear that algorithm $\mathcal{A}$ can be used to solve the X3C problem by applying it to the above instance of the priori KFSS problem. Since X3C is NP-complete, there is no polynomial-time algorithm for it if P $\neq$ NP, and we get a contradiction. This completes the proof of the theorem. ■

The following result is a direct consequence of the above arguments; the proof is provided in Section 3.6.4.

Corollary 3.3.7 *If $P \neq NP$, then there is no polynomial-time constant-factor approximation algorithm for the posteriori KFSS problem.*

Remark 3.3.8 *Note that Theorem* 3.3.6 *and Corollary* 3.3.7 *provide stronger results than showing Problem* 3.2.2 *is NP-hard, as the NP-hardness of Problem* 3.2.2 *follows from taking the (constant factor) $K = 1$ in the above analysis.*

3.3.3 Inapproximability of the Priori and Posteriori KFSA Problems

In this section, we analyze the achievable performance of algorithms for the priori and posteriori KFSA problems. Consider any given instance of the priori (resp., posteriori) KFSA problem. For any given algorithm $\mathcal{A}$ (resp., $\mathcal{A}'$) for the priori (resp., posteriori) KFSA problem, we define the following ratios:

$$r_{\mathcal{A}}(\tilde{\Sigma}) \triangleq \frac{\text{trace}(\tilde{\Sigma}_{opt})}{\text{trace}(\tilde{\Sigma}_{\mathcal{A}})}, \tag{3.19}$$

and

$$r_{\mathcal{A}'}(\tilde{\Sigma}^*) \triangleq \frac{\text{trace}(\tilde{\Sigma}^*_{opt})}{\text{trace}(\tilde{\Sigma}^*_{\mathcal{A}'})}, \tag{3.20}$$

where $\tilde{\Sigma}_{opt}$ (resp., $\tilde{\Sigma}^*_{opt}$) is the optimal solution to the priori (resp., posteriori) KFSA problem and $\tilde{\Sigma}_{\mathcal{A}}$ (resp., $\tilde{\Sigma}^*_{\mathcal{A}'}$) is the solution to the priori (resp., posteriori) KFSA problem given by algorithm $\mathcal{A}$ (resp., $\mathcal{A}'$). It is worth noting that using the arguments in [2], the same upper bounds for $r_{\mathcal{A}}(\tilde{\Sigma})$ and $r_{\mathcal{A}'}(\tilde{\Sigma}^*)$ can be obtained as those for $r_{\mathcal{A}}(\Sigma)$ and $r_{\mathcal{A}'}(\Sigma^*)$ in [2], respectively, where these bounds depend on the system matrices.

Nevertheless, we show that there is again no polynomial-time constant-factor algorithm for the priori (resp., posteriori) KFSA problem (if P $\neq$ NP), i.e., for all $K \in \mathbb{R}_{\geq 1}$ and polynomial-time algorithms $\mathcal{A}$ (resp., $\mathcal{A}'$), there are instances of the priori (resp., posteriori) KFSA problem where $r_{\mathcal{A}}(\tilde{\Sigma}) > K$ (resp., $r_{\mathcal{A}'}(\tilde{\Sigma}^*) > K$). To establish this result, we relate the KFSA problem to the X3C problem as described in Definition 3.3.1 and Lemma 3.3.4.

Theorem 3.3.9 *If $P \neq NP$, then there is no polynomial-time constant-factor approximation algorithm for the priori KFSA problem.*

Proof Assume that there exists such a polynomial-time constant-factor approximation algorithm $\mathcal{A}$, i.e., $\exists K \in \mathbb{R}_{\geq 1}$ such that $r_{\mathcal{A}}(\tilde{\Sigma}) \leq K$ for all instances of the priori KFSA problem, where $r_{\mathcal{A}}(\tilde{\Sigma})$ is defined in Eq. (3.19). We will show that $\mathcal{A}$ can be used to solve the X3C problem, leading to a contradiction.

Consider any instance of the X3C problem to be a finite set $D = \{d_1, \cdots, d_{3m}\}$ and a collection $\mathcal{C} = \{c_1, \ldots, c_\tau\}$ of 3-element subsets of D, where $\tau \geq m$. Recall in the proof of Theorem 3.3.6 that we use a column vector $g_i \in \mathbb{R}^{3m}$ to encode which elements of D are contained in c_i, where $(g_i)_j = 1$ if $d_j \in D$ is in c_i, and $(g_i)_j = 0$ otherwise, for $i \in \{1, 2, \ldots, \tau\}$ and $j \in \{1, 2, \ldots, 3m\}$. The matrix $G \in \mathbb{R}^{\tau \times 3m}$ was defined in the proof of Theorem 3.3.6 as $G = \begin{bmatrix} g_1 & \cdots & g_\tau \end{bmatrix}^T$. In this proof, we will make use of the matrix $F \triangleq G^T$; note that each column of F contains exactly three 1's.

Given the above instance of the X3C problem, we then construct an instance of the priori KFSA as follows. Denote $Z = \lceil K \rceil (\tau + 2)(\delta_v^2 + 1)$, where we set $\delta_v = 1$. Define the system dynamics matrix as $A = \text{diag}(\lambda_1, 0, \ldots, 0) \in \mathbb{R}^{(\tau+1)\times(\tau+1)}$, where $\lambda_1 = \frac{Z-1/2}{Z}$. Note that $Z \in \mathbb{Z}_{>1}$ and $0 < \lambda_1 < 1$. The set $\mathcal{Q}$ consists of $3m + \tau$ sensors with collective measurement matrix

$$C = \begin{bmatrix} \mathbf{1}_{3m} & \rho F \\ \mathbf{0} & I_\tau \end{bmatrix}, \tag{3.21}$$

where F is defined above and I_τ is used to encode the collection $\mathcal{C}$, i.e., $\mathbf{e}_j$ represents $c_j \in \mathcal{C}$ for all $j \in \{1, 2, \ldots, \tau\}$. The constant ρ is chosen as $\rho = 2Z\left\lceil \sqrt{m(Z-1)} \right\rceil + 1$. The system noise covariance matrix is set to be $W = I_{\tau+1}$. The measurement noise covariance is set as $V = \delta_v^2 \begin{bmatrix} I_{3m} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\rho^2} I_\tau \end{bmatrix}$. The sensor attack cost vector is set as $\omega = \mathbf{1}_{3m+\tau}$, and the sensor attack budget is set as $\Omega = m$. Note that the sensor attack vector is given by $\nu \in \{0, 1\}^{3m+\tau}$. For the above construction, since the only

nonzero eigenvalue of A is λ_1, we know from Lemma 3.3.1(a) that $\sum_{i=2}^{\tau+1}(\Sigma(\nu^c))_{ii} = \sum_{i=2}^{\tau+1} W_{ii} = \tau$ for all ν.

We claim that algorithm $\mathcal{A}$ will return a sensor attack ν such that $\text{trace}(\Sigma(\nu^c)) > (\tau+2)(\delta_v^2+1)$ if and only if the answer to the X3C problem is "yes".

We prove the above claim as follows. Suppose that the answer to the X3C problem is "yes". Similarly to the proof of Theorem 3.3.6, we first assume that $V = \mathbf{0}_{(3m+\tau)\times(3m+\tau)}$. Denote an exact cover as $\mathcal{C}' = \{c_{j_1}, \ldots, c_{j_m}\}$, where $\{j_1, \ldots, j_m\} \subseteq \{1, 2, \ldots, \tau\}$. Define $\tilde{\nu}$ to be the sensor attack such that $\text{supp}(\tilde{\nu}) = \{3m+j_1, \ldots, 3m+j_m\}$. We then renumber the states of the system from state 2 to state τ such that for all $i \in \{1, 2, \ldots, m\}$, the columns of the submatrix I_τ of C in Eq. (3.21) representing c_{j_i} in $\mathcal{C}'$, i.e., the columns of I_τ that correspond to $\text{supp}(\tilde{\nu})$, come first. Note that renumbering the states does not change the trace of the steady-state *a priori* error covariance of the corresponding Kalman filter. We then have from Eq. (3.21) the following:

$$C(\tilde{\nu}^c) = \begin{bmatrix} \mathbf{1}_{3m} & \rho F_1 & \rho F_2 \\ \mathbf{0} & \mathbf{0} & I_{\tau-m} \end{bmatrix}, \tag{3.22}$$

where $F_1 \in \mathbb{R}^{3m\times m}$ and $F_2 \in \mathbb{R}^{3m\times(\tau-m)}$ satisfy $F = \begin{bmatrix} F_1 & F_2 \end{bmatrix}$, and $I_{\tau-m}$ is the submatrix of I_τ that corresponds to $\text{supp}(\tilde{\nu}^c) \cap \{3m+1, \ldots, 3m+\tau\}$, i.e., the elements of $\mathcal{C}$ that are not in $\mathcal{C}'$.[2] Since the sensor attack $\tilde{\nu}$ targets the rows of C that correspond to the elements of the exact cover $\mathcal{C}'$ for D, we have F_1, after some row permutations of $C(\tilde{\nu}^c)$, is given by $F_1 = \begin{bmatrix} \mathbf{e}_1^T & \mathbf{e}_1^T & \mathbf{e}_1^T \cdots & \mathbf{e}_m^T & \mathbf{e}_m^T & \mathbf{e}_m^T \end{bmatrix}^T$. We perform additional elementary row operations and merge identical rows (which does not change the steady-state *a priori* error covariance matrix of the corresponding Kalman filter) to transform $C(\tilde{\nu}^c)$ into the matrix

$$\tilde{C}(\tilde{\nu}^c) = \begin{bmatrix} \mathbf{1}_m & \rho I_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\tau-m} \end{bmatrix}. \tag{3.23}$$

[2]Note that if the submatrix of I_τ corresponding to $\text{supp}(\tilde{\nu}^c) \cap \{3m+1, \ldots, 3m+\tau\}$ is not identity, we can always permute the rows of $C(\tilde{\nu}^c)$ to make it identity.

Since A and W are both diagonal, and $V = \mathbf{0}$, we can obtain from Eq. (3.4) that the steady-state *a priori* error covariance corresponding to $\tilde{C}(\tilde{\nu}^c)$, denoted as $\tilde{\Sigma}(\tilde{\nu}^c)$, is of the form

$$\tilde{\Sigma}(\tilde{\nu}^c) = \begin{bmatrix} \tilde{\Sigma}_1(\tilde{\nu}^c) & \mathbf{0} \\ \mathbf{0} & \tilde{\Sigma}_2(\tilde{\nu}^c) \end{bmatrix},$$

where $\tilde{\Sigma}_1(\tilde{\nu}^c) \in \mathbb{R}^{(m+1)\times(m+1)}$ satisfies

$$\tilde{\Sigma}_1(\tilde{\nu}^c) = A_1\tilde{\Sigma}_1(\tilde{\nu}^c)A_1^T + W_1 - A_1\tilde{\Sigma}_1(\tilde{\nu}^c)\tilde{C}^T\big(\tilde{C}\tilde{\Sigma}_1(\tilde{\nu}^c)\tilde{C}^T\big)^{-1}\tilde{C}\tilde{\Sigma}_1(\tilde{\nu}^c)A_1^T,$$

where $A_1 = \mathrm{diag}(\lambda_1, 0, \ldots, 0) \in \mathbb{R}^{(m+1)\times(m+1)}$, $\tilde{C} = \begin{bmatrix} \mathbf{1}_m & \rho I_m \end{bmatrix}$ and $W_1 = I_{m+1}$. We then know from Lemma 3.3.2(b) that $(\Sigma(\tilde{\nu}^c))_{11} = (\tilde{\Sigma}(\tilde{\nu}^c))_{11}$ satisfies

$$(\Sigma(\tilde{\nu}^c))_{11} = \frac{\lambda_1^2\rho^2 + m - \rho^2 + \sqrt{(\rho^2 - \lambda_1^2\rho^2 - m)^2 + 4m\rho^2}}{2m}. \tag{3.24}$$

By our choices of λ_1 and ρ, we have

$$\begin{aligned}
&\rho^2 > 4Z^2m(Z-1) \Rightarrow (1 - \frac{Z-1/4}{Z})\rho^2 > Z^2m - Zm \\
\Rightarrow\ &\rho^2 > mZ^2 + Z\rho^2\frac{Z-1/4}{Z^2} - Zm \\
\Rightarrow\ &4m\rho^2 > 4m^2Z^2 + 4mZ(\rho^2(1-\lambda_1^2) - m) \\
\Rightarrow\ &(\rho^2 - \lambda_1^2\rho^2 - m)^2 + 4m\rho^2 > 4m^2Z^2 + 4mZ(\rho^2 - \lambda_1^2\rho^2 - m) + (\rho^2 - \lambda_1^2\rho^2 - m)^2 \\
\Rightarrow\ &(\rho^2 - \lambda_1^2\rho^2 - m)^2 + 4m\rho^2 > (2mZ + \rho^2 - \lambda_1^2\rho^2 - m)^2 \\
\Rightarrow\ &\sqrt{(\rho^2 - \lambda_1^2\rho^2 - m)^2 + 4m\rho^2} > 2mZ + \rho^2 - \lambda_1^2\rho^2 - m \\
\Rightarrow\ &\frac{\lambda_1^2\rho^2 + m - \rho^2 + \sqrt{(\rho^2 - \lambda_1^2\rho^2 - m)^2 + 4m\rho^2}}{2m} > Z.
\end{aligned} \tag{3.25}$$

Noting that $Z \geq K(\tau+2)(\delta_v^2+1)$, we then know from (3.24) and (3.25) that $(\Sigma(\tilde{\nu}^c))_{11} > K(\tau+2)(\delta_v^2+1)$, which further implies that $\mathrm{trace}(\Sigma(\tilde{\nu}^c)) > K(\tau+2)(\delta_v^2+1)$. Following the same arguments as those in the proof of Theorem 3.3.6, we see that for $V = \delta_v^2\begin{bmatrix} I_{3m} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\rho^2}I_\tau \end{bmatrix}$, $\mathrm{trace}(\Sigma(\tilde{\nu}^c)) > K(\tau+2)(\delta_v^2+1)$ also holds, which implies $\mathrm{trace}(\Sigma(\nu^{*c})) > K(\tau+2)(\delta_v^2+1)$, where ν^* is an optimal sensor attack for the priori KFSA problem. Since algorithm $\mathcal{A}$ has approximation ratio K, it would return a sensor attack ν such that $\mathrm{trace}(\Sigma(\nu^c)) > (\tau+2)(\delta_v^2+1)$.

Conversely, suppose the answer to the X3C problem is "no". For any union of $l \leq m$ ($l \in \mathbb{Z}_{\geq 0}$) subsets in $\mathcal{C}$, denoted as $\mathcal{C}_l$, there exists at least one element in D that is not covered by $\mathcal{C}_l$. We then show that $\text{trace}(\Sigma(\nu^c)) \leq (\tau+2)(\delta_v^2+1)$ for all sensor attacks ν (that satisfy the budget constraint). We split our discussion into three cases.

First, consider any sensor attack ν_1 that targets l sensors merely from C_1 to C_{3m} in Eq. (3.21), i.e., $|\text{supp}(\nu_1)| = l$ and $\text{supp}(\nu_1) \subseteq \{1,\ldots,3m\}$, where $l \leq m$. We then obtain

$$C(\nu_1^c) = \begin{bmatrix} \mathbf{1}_{3m-l} & \rho F(\nu_1^c) \\ \mathbf{0} & I_\tau \end{bmatrix},$$

where $F(\nu_1^c) \in \mathbb{R}^{(3m-l)\times\tau}$ is defined to be the submatrix of F that corresponds to $\text{supp}(\nu_1^c) \cap \{1,\ldots,3m\}$, i.e., the rows of F that are left over by ν_1. We perform elementary row operations to transform $C(\nu_1^c)$ into

$$\tilde{C}(\nu_1^c) \triangleq \Psi C(\nu_1^c) = \begin{bmatrix} \mathbf{1}_{3m-l} & \mathbf{0} \\ \mathbf{0} & I_\tau \end{bmatrix} \tag{3.26}$$

with the corresponding measurement noise covariance

$$\tilde{V}(\nu_1^c) \triangleq \Psi V(\nu_1^c)\Psi^T = \delta_v^2 \begin{bmatrix} I_{3m-l} + F(\nu_1^c)(F(\nu_1^c))^T & -\frac{1}{\rho}F(\nu_1^c) \\ -\frac{1}{\rho}(F(\nu_1^c))^T & \frac{1}{\rho^2}I_\tau \end{bmatrix}, \tag{3.27}$$

where $\Psi = \begin{bmatrix} I_{3m-l} & -\rho F(\nu_1^c) \\ \mathbf{0} & I_\tau \end{bmatrix}$. Since there are at most τ nonzero elements (which are all 1's) in the first row of $F(\nu_1^c)$, it follows that $(F(\nu_1^c)(F(\nu_1^c))^T)_{11}$ (i.e., the element in the first row and first column of the matrix $F(\nu_1^c)(F(\nu_1^c))^T$) is at most τ. We then see from Eq. (3.27) that $(\tilde{V}(\nu_1^c))_{11}$, denoted as $\tilde{\delta}_v^2(\nu_1^c)$, satisfies

$$\tilde{\delta}_v^2(\nu_1^c) \leq (\tau+1)\delta_v^2. \tag{3.28}$$

Second, consider any sensor attack ν_2 that targets l sensors merely from C_{3m+1} to $C_{3m+\tau}$ in Eq. (3.21), i.e., $|\text{supp}(\nu_2)| = l$ and $\text{supp}(\nu_2) \subseteq \{3m+1,\ldots,3m+\tau\}$, where

$l \leq m$. Via similar arguments to those for obtaining Eqs. (3.22), (3.26) and (3.27), we can perform elementary row operations to transform

$$C(\nu_2^c) = \begin{bmatrix} \mathbf{1}_{3m} & \rho F_1' & \rho F_2' \\ \mathbf{0} & \mathbf{0} & I_{\tau-l} \end{bmatrix}$$

into

$$\tilde{C}(\nu_2^c) = \begin{bmatrix} \mathbf{1}_{3m} & \rho F_1' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\tau-l} \end{bmatrix}$$

with the corresponding measurement noise covariance $\tilde{V}(\nu_2^c) = \begin{bmatrix} \tilde{\delta}_v^2(\nu_2^c) & * \\ * & * \end{bmatrix}$, where

$$\tilde{\delta}_v^2(\nu_2^c) \leq (\tau - l + 1)\delta_v^2. \tag{3.29}$$

Note that $F_1' \in \mathbb{R}^{3m \times l}$ and $F_2' \in \mathbb{R}^{3m \times (\tau-l)}$ satisfy $F = \begin{bmatrix} F_1' & F_2' \end{bmatrix}$. Recall that for any union of $l \leq m$ subsets in $\mathcal{C}$, denoted as $\mathcal{C}_l$, there exists at least one element in D that is not covered by $\mathcal{C}_l$. We can then assume without loss of generality that one such element is d_1, which implies that the first row of F_1' is zero.

Third, consider any sensor attack ν_3 that targets sensors from both C_1 to C_{3m} and C_{3m+1} to $C_{3m+\tau}$ in Eq. (3.21). Suppose the attack ν_3 attacks l_1 sensors from C_1 to C_{3m} and l_2 sensors from C_{3m+1} to $C_{3m+\tau}$, i.e., $\text{supp}(\nu_3) = \{j_1', \ldots, j_{l_1}', 3m + j_1'', \ldots, 3m + j_{l_2}''\} \subseteq \{1, 2, \ldots, 3m + \tau\}$, where $l_1, l_2 \in \mathbb{Z}_{\geq 1}$, $l_1 + l_2 = l \leq m$, $\{j_1', \ldots, j_{l_1}'\} \subseteq \{1, \ldots, 3m\}$ and $\{j_1'', \ldots, j_{l_2}''\} \subseteq \{1, \ldots, \tau\}$. By similar arguments to those above, we can perform elementary row operations to transform

$$C(\nu_3^c) = \begin{bmatrix} \mathbf{1}_{3m-l_1} & \rho F_1(\nu_3^c) & \rho F_2(\nu_3^c) \\ \mathbf{0} & \mathbf{0} & I_{\tau-l_2} \end{bmatrix}$$

into

$$\tilde{C}(\nu_3^c) = \begin{bmatrix} \mathbf{1}_{3m-l_1} & \rho F_1(\nu_3^c) & 0 \\ \mathbf{0} & \mathbf{0} & I_{\tau-l_2} \end{bmatrix},$$

where $F_1(\nu_3^c) \in \mathbb{R}^{(3m-l_1)\times l_2}$, $F_2(\nu_3^c) \in \mathbb{R}^{(3m-l_1)\times(\tau-l_2)}$ satisfy $F(\nu_3^c) = \begin{bmatrix} F_1(\nu_3^c) & F_2(\nu_3^c) \end{bmatrix}$ with $F(\nu_3^c)$ defined in the same way as $F(\nu_1^c)$. Moreover, the measurement noise covariance corresponding to $\tilde{C}(\nu_3^c)$ is given by $\tilde{V}(\nu_3^c) = \begin{bmatrix} \tilde{\delta}_v^2(\nu_3^c) & * \\ * & * \end{bmatrix}$, where

$$\tilde{\delta}_v^2(\nu_3^c) \leq (\tau - l_2 + 1)\delta_v^2. \tag{3.30}$$

Since any l_2 subsets in $\mathcal{C}$ can cover at most $3l_2$ elements in D, there are at least $3m - 3l_2$ elements in D that are not covered by the l_2 subsets in $\mathcal{C}$. Also note that

$$3m - 3l_2 - l_1 = 3m - 2l_2 - l = 2(m - l_2) + m - l > 0,$$

where the last inequality follows from the facts that $l_1 + l_2 = l \leq m$ and $l_1, l_2 \in \mathbb{Z}_{\geq 1}$. Hence, by attacking l_1 sensors from C_1 to C_{3m} and l_2 sensors from C_{3m+1} to $C_{3m+\tau}$, we have at least $3m - 3l_2 - l_1 > 0$ row(s) of $F_1(\nu_3^c)$ that are zero. Again, we can assume without loss of generality that the first row of $F_1(\nu_3^c)$ is zero.

In summary, for any sensor attack ν, we let $\hat{\Sigma}(\nu_i^c)$ denote the steady-state *a priori* error covariance obtained from measurement matrix $(\tilde{C}(\nu_i^c))_1 = \mathbf{e}_1$ with measurement noise variance $\tilde{\delta}_v^2(\nu_i^c)$ (which corresponds to the first sensor in $\tilde{C}(\nu_i^c)$), $\forall i \in \{1, 2, 3\}$, where ν_1, ν_2 and ν_3 are given as above. Following similar arguments to those for (3.14), we have $(\hat{\Sigma}(\nu_i^c))_{11} \leq \tilde{\delta}_v^2(\nu_i^c) + 2$, $\forall i \in \{1, 2, 3\}$. Since $\sum_{i=2}^{\tau+1}(\hat{\Sigma}(\nu_i^c))_{ii} = \sum_{i=2}^{\tau+1} W_{ii} = \tau$ holds for all $i \in \{1, 2, 3\}$ via similar arguments to those above, we obtain that

$$\text{trace}(\hat{\Sigma}(\nu_i^c)) \leq \tilde{\delta}_v^2(\nu_i^c) + 2 + \tau, \forall i \in \{1, 2, 3\}. \tag{3.31}$$

Again note that adding more sensors does not increase the MSEE of the corresponding Kalman filter, and the above operations performed on the sensing matrix C do not change the trace of the steady-state *a priori* error covariance of the corresponding Kalman filter as well. We then see from Eqs. (3.28)-(3.31) that $\text{trace}(\Sigma(\nu^c)) \leq (\tau + 1)\delta_v^2 + 2 + \tau \leq (\tau + 2)(\delta_v^2 + 1)$ for all ν. It follows that algorithm $\mathcal{A}$ would also return a sensor attack ν such that $\text{trace}(\Sigma(\nu^c)) \leq (\tau + 2)(\delta_v^2 + 1)$. This proves the converse direction of the claim above.

Therefore, we know that $\mathcal{A}$ can be used to solve the X3C problem by applying it to the above instance of the priori KFSA problem. Since X3C is NP-complete, there is no polynomial-time algorithm for it if P $\neq$ NP, yielding a contradiction. This completes the proof of the theorem. ■

The arguments above also imply the following result whose proof is included in Section 3.6.5.

Corollary 3.3.10 *If $P \neq NP$, then there is no polynomial-time constant-factor approximation algorithm for the posteriori KFSA problem.*

Remark 3.3.11 *The NP-hardness of Problem* 3.2.3 *follows from taking the (constant factor) $K = 1$ in the arguments above.*

3.4 Failure of Greedy Algorithms

Our results in Theorem 3.3.6 and Theorem 3.3.9 indicate that no polynomial-time algorithm can be guaranteed to yield a solution that is within any constant factor of the optimal solution to the priori (resp., posteriori) KFSS and KFSA problems. In particular, these results apply to the greedy algorithms that are often studied for sensor selection in the literature [2, 8], where sensors are iteratively selected (resp., attacked) in order to produce the greatest decrease (resp., increase) in the error covariance at each iteration. In this section we will focus on such greedy algorithms for the priori (resp., posteriori) KFSS problem and the priori (resp., posteriori) KFSA problem, and show explicitly how these greedy algorithms can fail to provide good solutions; this provides additional insight into the factors that cause the KFSS and KFSA problems to be challenging.

3.4.1 Failure of Greedy Algorithms for the Priori and Posteriori KFSS Problems

It was shown via simulations in [2] that greedy algorithms for KFSS work well in practice (e.g., for randomly generated systems). In this section, we provide an explicit example showing that greedy algorithms for the priori and posteriori KFSS problems can perform arbitrarily poorly, even for small systems (containing only three states). We consider the greedy algorithm for the priori (resp., posteriori) KFSS problem given in Algorithm 3.4.1, for instances where all sensors have selection costs equal to 1, and the sensor selection budget $H \in \{1, \ldots, q\}$ (i.e., up to H sensors can be selected). For any such instance of the priori (resp., posteriori) KFSS problem, define $r_{gre}(\Sigma) = \frac{\text{trace}(\Sigma_{gre})}{\text{trace}(\Sigma_{opt})}$ (resp., $r_{gre}(\Sigma^*) = \frac{\text{trace}(\Sigma^*_{gre})}{\text{trace}(\Sigma^*_{opt})}$), where Σ_{gre} (resp., Σ^*_{gre}) is the solution of Eq. (3.4) (resp., Eq. (3.6)) corresponding to the sensors returned by Algorithm 3.4.1.

Algorithm 3.4.1 Greedy Algorithm for Problem 3.2.2

Input: An instance of priori (resp., posteriori) KFSS

Output: A set $\mathcal{S}$ of selected sensors

1: $k \leftarrow 1$, $\mathcal{S} \leftarrow \emptyset$

2: **for** $k \leq H$ **do**

3: $\quad j \in \arg\min_{i \notin \mathcal{S}} \text{trace}(\Sigma(\mathcal{S} \cup \{i\}))$ (resp., $j \in \arg\min_{i \notin \mathcal{S}} \text{trace}(\Sigma^*(\mathcal{S} \cup \{i\}))$)

4: $\quad \mathcal{S} \leftarrow \mathcal{S} \cup \{j\}$, $k \leftarrow k + 1$

Example 3.4.1 *Consider an instance of the priori (resp., posteriori) KFSS problem with matrices $W = I_3$ and $V = \mathbf{0}_{3\times 3}$, and A, C defined as*

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; C = \begin{bmatrix} 1 & \epsilon & \epsilon \\ 1 & 0 & \epsilon \\ 0 & 1 & 1 \end{bmatrix},$$

where $0 < |\lambda_1| < 1$, $\lambda_1 \in \mathbb{R}$, *and* $\epsilon \in \mathbb{R}_{>0}$. *In addition, we have the selection budget* $H = 2$, *the cost vector* $h = [1\ 1\ 1]^T$ *and the set of candidate sensors* $\mathcal{Q} = \{1, 2, 3\}$, *where sensor* i *corresponds to the* i*th row of matrix* C, *for* $i \in \{1, 2, 3\}$.

We then have the following result whose proof is provided in Section 3.6.6.

Theorem 3.4.2 *For the instance of the priori (resp., posteriori) KFSS problem defined in Example* 3.4.1, *the ratios* $r_{gre}(\Sigma) = \frac{trace(\Sigma_{gre})}{trace(\Sigma_{opt})}$ *and* $r_{gre}(\Sigma^*) = \frac{trace(\Sigma^*_{gre})}{trace(\Sigma^*_{opt})}$ *satisfy*

$$\lim_{\epsilon \to \infty} r_{gre}(\Sigma) = \frac{2}{3} + \frac{1}{3(1 - \lambda_1^2)}, \tag{3.32}$$

and

$$\lim_{\epsilon \to \infty} r_{gre}(\Sigma^*) = \frac{1}{1 - \lambda_1^2} + 1, \tag{3.33}$$

respectively.

Examining Eq. (3.32) (resp., Eq. (3.33)), we see that for the given instance of the priori (resp., posteriori) KFSS problem, we have $r_{gre}(\Sigma) \to \infty$ (resp., $r_{gre}(\Sigma^*) \to \infty$) as $\epsilon \to \infty$ and $\lambda_1 \to 1$. Thus, $r_{gre}(\Sigma)$ (resp., $r_{gre}(\Sigma^*)$) can be made arbitrarily large by choosing the parameters in the instance appropriately.

To explain the result in Theorem 3.4.2, we first note that the only nonzero eigenvalue of the diagonal A defined in Example 3.4.1 is λ_1, and so we know from Lemma 3.3.1(a) that state 2 and state 3 of the system as defined in Example 3.4.1 each contribute at most 1 to trace($\Sigma(\mu)$) (resp., trace($\Sigma^*(\mu)$)) for all μ. Hence, in order to minimize trace($\Sigma(\mu)$) (resp., trace($\Sigma^*(\mu)$)), we need to minimize the MSEE of state 1. Moreover, the measurements of state 2 and state 3 can be viewed as measurement noise that corrupts the measurements of state 1. It is then easy to observe from the form of matrix C defined in Example 3.4.1 that sensor 2 is the single best sensor among the three sensors since it provides measurements of state 1 with less noise than sensor 1 (and sensor 3 does not measure state 1 at all). Thus, the greedy algorithm for the priori (resp., posteriori) KFSS problem defined as Algorithm 3.4.1 selects sensor 2 in its first iteration. Nonetheless, we notice from C defined in Example 3.4.1 that the optimal set of two sensors that minimizes trace($\Sigma(\mu)$) (resp., trace($\Sigma^*(\mu)$)) contains

sensor 1 and sensor 3, which together give us exact measurements (without measurement noise) on state 1 (after some elementary row operations). Since the greedy algorithm selects sensor 2 in its first iteration, no matter which sensor it selects in its second iteration, the two chosen sensors can only give a noisy measurement of state 1 (if we view the measurements of state 2 and state 3 as measurement noise), and the variance of the measurement noise can be made arbitrary large if we take $\epsilon \to \infty$ in C defined in Example 3.4.1. Hence, the greedy algorithm fails to perform well due to its myopic choice in the first iteration.

It is also useful to note that the above behavior holds for any algorithm that outputs a sensor selection that contains sensor 2 for the above example.

3.4.2 Failure of Greedy Algorithms for the Priori and Posteriori KFSA Problems

Algorithm 3.4.2 Greedy Algorithm for Problem 3.2.3

Input: An instance of priori (resp., posteriori) KFSA

Output: A set $\mathcal{S}$ of targeted sensors

1: $k \leftarrow 1$, $\mathcal{S} \leftarrow \emptyset$

2: **for** $k \leq \Omega$ **do**

3: $j \in \arg\max_{i \notin \mathcal{S}} \operatorname{trace}(\Sigma(\mathcal{Q} \setminus (\mathcal{S} \cup \{i\})))$ (resp., $j \in \arg\max_{i \notin \mathcal{S}} \operatorname{trace}(\Sigma^*(\mathcal{Q} \setminus (\mathcal{S} \cup \{i\}))))$

4: $\mathcal{S} \leftarrow \mathcal{S} \cup \{j\}$, $k \leftarrow k + 1$

In [8], the authors showed that a simple greedy algorithm can perform arbitrarily poorly for an instance of the observation attack problem in Gaussian process regression. Here, we consider a simple greedy algorithm for the priori (resp., posteriori) KFSA problem given in Algorithm 3.4.2, for instances where all sensors have an attack cost of 1, and the sensor attack budget $\Omega \in \{1, \ldots, q\}$ (i.e., up to Ω sensors can be attacked). For any such instance of the priori (resp., posteriori) KFSA problem,

define $r_{gre}(\tilde{\Sigma}) = \frac{\text{trace}(\tilde{\Sigma}_{opt})}{\text{trace}(\tilde{\Sigma}_{gre})}$ (resp., $r_{gre}(\tilde{\Sigma}^*) = \frac{\text{trace}(\tilde{\Sigma}^*_{opt})}{\text{trace}(\tilde{\Sigma}^*_{gre})}$), where $\tilde{\Sigma}_{gre}$ (resp., $\tilde{\Sigma}^*_{gre}$) is the solution to the priori (resp., posteriori) KFSA problem given by Algorithm 3.4.2. We then show that Algorithm 3.4.2 can perform arbitrarily poorly for a simple instance of the priori (resp., posteriori) KFSA problem as described below.

Example 3.4.3 *Consider an instance of the priori (resp., posteriori) KFSA problem with matrices $W = I_3$, $V = \mathbf{0}_{4\times 4}$, and A, C defined as*

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; C = \begin{bmatrix} 1 & \epsilon & \epsilon \\ 1 & 0 & \epsilon \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $0 < |\lambda_1| < 1$, $\lambda_1 \in \mathbb{R}$ and $h \in \mathbb{R}_{>0}$. In addition, the attack budget $\Omega = 2$, the cost vector $\varpi = [1\ 1\ 1\ 1]^T$, and the set of sensors $\mathcal{Q} = \{1, 2, 3, 4\}$ has already been installed on the system, where sensor i corresponds to the ith row of matrix C, for $i \in \{1, 2, 3, 4\}$.

We then have the following result; the proof is provided in Section 3.6.7.

Theorem 3.4.4 *For the instance of the priori (resp., posteriori) KFSA problem defined in Example 3.4.3, the ratios $r_{gre}(\tilde{\Sigma}) = \frac{trace(\tilde{\Sigma}_{opt})}{trace(\tilde{\Sigma}_{gre})}$ and $r_{gre}(\tilde{\Sigma}^*) = \frac{trace(\tilde{\Sigma}^*_{opt})}{trace(\tilde{\Sigma}^*_{gre})}$ satisfy*

$$\lim_{\epsilon \to 0} r_{gre}(\tilde{\Sigma}) = \frac{2}{3} + \frac{1}{3(1-\lambda_1^2)}, \tag{3.34}$$

and

$$\lim_{\epsilon \to 0} r_{gre}(\tilde{\Sigma}^*) = \frac{1}{1-\lambda_1^2}, \tag{3.35}$$

respectively.

Inspecting Eq. (3.34) (resp., Eq. (3.35)), we observe that for the given instance of the priori (resp., posteriori) KFSA problem, we have $r_{gre}(\tilde{\Sigma}) \to \infty$ (resp., $r_{gre}(\tilde{\Sigma}^*) \to \infty$) as $\epsilon \to 0$ and $\lambda_1 \to 1$. Thus, $r_{gre}(\tilde{\Sigma})$ (resp., $r_{gre}(\tilde{\Sigma}^*)$) can be made arbitrarily large by choosing the parameters in the instance appropriately.

Here, we explain the results in Theorem 3.4.4 as follows. Using similar arguments as before, we know from the structure of matrix A defined in Example 3.4.3 that in order to maximize $\text{trace}(\Sigma(\nu^c))$ (resp., $\text{trace}(\Sigma^*(\nu^c))$), we need to maximize the MSEE of state 1, i.e., make the measurements of state 1 "worse". Again, the measurements of state 2 and state 3 can be viewed as measurement noise that corrupts the measurements of state 1. No matter which of sensor 1, sensor 2, or sensor 3 is attacked, the resulting measurement matrix $C(\nu^c)$ is full column rank, which yields an exact measurement of state 1. We also observe that if sensor 4 is targeted, the surviving sensors can only provide measurements of state 1 that are corrupted by measurements of states 2 and state 3. Hence, the greedy algorithm for the priori (resp., posteriori) KFSA problem defined as Algorithm 3.4.2 targets sensor 4 in its first iteration, since it is the single best sensor to attack from the four sensors. Nevertheless, sensor 1 and sensor 2 form the optimal set of sensors to be attacked to maximize $\text{trace}(\Sigma(\nu^c))$ (resp., $\text{trace}(\Sigma^*(\nu^c))$), since the surviving sensors provide no measurement of state 1. Since the greedy algorithm targets sensor 4 in its first iteration, no matter which sensor it targets in the second step, the surviving sensors can always provide some measurements of state 1 with noise (if we view the measurements of state 2 and state 3 as measurement noise), and the variance of the noise will vanish if we take $\epsilon \to 0$ in matrix C defined in Example 3.4.3. Hence, the myopic behavior of the greedy algorithm makes it perform poorly.

Furthermore, it is useful to note that the above result holds for any algorithm that outputs a sensor attack that does not contain sensor 1 or sensor 2 for the above example.

Remark 3.4.5 *Note that in Example* 3.4.1 *(resp., Example* 3.4.3*), we set* $V = \mathbf{0}_{3\times 3}$ *(resp.,* $V = \mathbf{0}_{4\times 4}$*). It is straightforward to show, using similar arguments to those in the proof of Theorem* 3.4.2 *(resp., Theorem* 3.4.4*), that when we instead set* $V = \delta I_3$ *(resp.,* $V = \delta I_4$*), where* $\delta \in \mathbb{R}_{>0}$*, the results in Eq.* (3.32)-(3.33) *(resp., Eq.* (3.34)-(3.35)*) hold if we let* $\delta \to 0$*.*

3.5 Chapter Summary

In this chapter, we studied sensor selection and attack problems for (steady-state) Kalman filtering of linear dynamical systems. We showed that these problems are NP-hard and have no polynomial-time constant-factor approximation algorithms, even under the assumption that the system is stable and each sensor has identical selection cost. To illustrate this point, we provided explicit examples showing how greedy algorithms can perform arbitrarily poorly on these problems, even when the system only has three states. Our results shed new insights into the problem of sensor selection and attack for Kalman filtering and show, in particular, that this problem is more difficult than other variants of the sensor selection problem that have submodular (or supermodular) cost functions.

3.6 Proofs of Key Results

3.6.1 Proof of Lemma 3.3.1

Since A and W are diagonal, the system represents a set of n scalar subsystems of the form

$$x_i[k+1] = \lambda_i x_i[k] + w_i[k], \forall i \in \{1, \ldots, n\},$$

where $x_i[k]$ is the ith state of $x[k]$ and $w_i[k]$ is a zero-mean white noise process with variance $\sigma_{w_i}^2 = W_{ii}$. As A is stable, the pair $(A, C(\mu))$ is detectable and the pair $(A, W^{\frac{1}{2}})$ is stabilizable for all sensor selections μ. Thus, the limits $\lim_{k\to\infty}(\Sigma_{k/k-1}(\mu))_{ii}$ and $\lim_{k\to\infty}(\Sigma_{k/k}(\mu))_{ii}$ exist for all i and μ (based on Lemma 3.2.1), and are denoted as $(\Sigma(\mu))_{ii}$ and $(\Sigma^*(\mu))_{ii}$, respectively.

Proof of (a). Since A and W are diagonal, we know from Eq. (3.5) that

$$(\Sigma(\mu))_{ii} = \lambda_i^2(\Sigma^*(\mu))_{ii} + W_{ii},$$

which implies $(\Sigma(\mu))_{ii} \geq W_{ii}$, $\forall i \in \{1, \ldots, n\}$. Moreover, it is easy to see that $(\Sigma(\mu))_{ii} \leq (\Sigma(\mathbf{0}))_{ii}$, $\forall i \in \{1, \ldots, n\}$. Since $C(\mathbf{0}) = \mathbf{0}$, we obtain from Eq. (3.4)

$$\Sigma(\mathbf{0}) = A\Sigma(\mathbf{0})A^T + W.$$

which implies that $(\Sigma(\mathbf{0}))_{ii} = \frac{W_{ii}}{1-\lambda_i^2}$ since A is diagonal. Hence, $(\Sigma(\mu))_{ii} \leq \frac{W_{ii}}{1-\lambda_i^2}$, $\forall i \in \{1, \ldots, n\}$. Similarly, since we also have $(\Sigma^*(\mu))_{ii} \leq (\Sigma^*(\mathbf{0}))_{ii}$, we obtain from Eq. (3.6)

$$\Sigma^*(\mathbf{0}) = A\Sigma^*(\mathbf{0})A^T + W.$$

Hence, $0 \leq (\Sigma^*(\mu))_{ii} \leq \frac{W_{ii}}{1-\lambda_i^2}$, $\forall i \in \{1, \ldots, n\}$.

Proof of (b). Assume without loss of generality that the first column of $C(\mu)$ is zero, since we can simply renumber the states to make this the case without affecting the trace of the error covariance matrix. Hence, we have $C(\mu)$ of the form

$$C(\mu) = \begin{bmatrix} \mathbf{0} & C_1(\mu) \end{bmatrix}.$$

Moreover, since A and W are diagonal, it follows from Eq. (3.4) that $\Sigma(\mu)$ is of the form

$$\Sigma(\mu) = \begin{bmatrix} \Sigma_1(\mu) & \mathbf{0} \\ \mathbf{0} & \Sigma_2(\mu) \end{bmatrix},$$

where $\Sigma_1(\mu) = (\Sigma(\mu))_{11}$ and satisfies

$$(\Sigma(\mu))_{11} = \lambda_i^2(\Sigma(\mu))_{11} + W_{11},$$

which implies $(\Sigma(\mu))_{11} = \frac{W_{11}}{1-\lambda_1^2}$. Furthermore, it follows from Eq. (3.6) that $\Sigma^*(\mu)$ is of the form

$$\Sigma^*(\mu) = \begin{bmatrix} \Sigma_1^*(\mu) & \mathbf{0} \\ \mathbf{0} & \Sigma_2^*(\mu) \end{bmatrix},$$

where $\Sigma_1^*(\mu) = (\Sigma^*(\mu))_{11}$ and satisfies

$$(\Sigma^*(\mu))_{11} = \lambda_1^2(\Sigma^*(\mu))_{11} + W_{11},$$

which implies $(\Sigma^*(\mu))_{11} = \frac{W_{11}}{1-\lambda_1^2}$.

Proof of (c). We assume without loss of generality that $\mathbf{e}_1 \in \text{rowspace}(C(\mu))$. If we further perform elementary row operations on $C(\mu)$, which does not change the solution to Eq. (3.4) (resp., Eq. (3.6)), we obtain a matrix $\tilde{C}(\mu)$ of the form

$$\tilde{C}(\mu) = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \tilde{C}_1(\mu) \end{bmatrix},$$

and $\tilde{V}(\mu) = \mathbf{0}$. Moreover, since A and W are diagonal, we see from Eq. (3.4) that $\Sigma(\mu)$ is of the form

$$\Sigma(\mu) = \begin{bmatrix} \Sigma_1(\mu) & \mathbf{0} \\ \mathbf{0} & \Sigma_2(\mu) \end{bmatrix},$$

where $\Sigma_1(\mu) = (\Sigma(\mu))_{11}$ and satisfies

$$(\Sigma(\mu))_{11} = \lambda_1^2(\Sigma(\mu))_{11} + W_{11} - \lambda_1^2(\Sigma(\mu))_{11},$$

which implies $(\Sigma(\mu))_{11} = W_{11}$. Furthermore, it follows from Eq. (3.6) that $\Sigma^*(\mu)$ is of the form

$$\Sigma^*(\mu) = \begin{bmatrix} \Sigma_1^*(\mu) & \mathbf{0} \\ \mathbf{0} & \Sigma_2^*(\mu) \end{bmatrix},$$

where $\Sigma_1^*(\mu) = (\Sigma^*(\mu))_{11}$ and satisfies $(\Sigma^*(\mu))_{11} = (\Sigma(\mu))_{11} - (\Sigma(\mu))_{11} = 0$. ■

3.6.2 Proof of Lemma 3.3.2

Proof of (a): We first note from Lemma 3.2.1 that the limit $\Sigma(\mu)$ exists for all μ (since A is stable). Since $A = \text{diag}(\lambda_1, 0, \ldots, 0)$, we have $x_i[k+1] = w_i[k]$, $\forall i \in \{2, \ldots, n\}$ and $\forall k \in \mathbb{Z}_{\geq 0}$. Moreover, we have from Eq. (3.3) that

$$y[k] = [1 \ \mathbf{0}_{1\times(n-1)}]x[k] + v[k] + v'[k] = x_1[k] + \hat{v}[k], \ \forall k \in \mathbb{Z}_{\geq 0},$$

where $v'[k] = \sum_{i=1}^{n-1} \gamma_i x_{i+1}[k]$ and $\hat{v}[k] \triangleq v[k] + v'[k]$. Recall that we have assumed with out loss of generality that $\bar{x}_0 = \mathbf{0}$ and $\Pi_0 = I_n$. Moreover, noting that $W = I_n$ and that $x[0]$ is independent of $w[k]$ and $v[k]$ for all $k \in \mathbb{Z}_{\geq 0}$, where $w[k]$ and $v[k]$ are uncorrelated zero-mean white noise processes (as assumed), we have that $\hat{v}[k]$ is a

zero-mean white noise process with $\mathbb{E}[(\hat{v}[k])^2] = \|\gamma\|_2^2 + \sigma_v^2$. Thus, to compute the MSEE of state 1 of the Kalman filter, i.e., Σ_{11}, we can consider a scalar discrete-time linear system with $A = \lambda_1$, $C = 1$, $W = 1$ and $V = \alpha^2$, and obtain from Eq. (3.4) the scalar DARE

$$\Sigma_{11} = \lambda_1^2(1 - \frac{\Sigma_{11}}{\alpha^2 + \Sigma_{11}})\Sigma_{11} + 1, \tag{3.36}$$

where $\alpha^2 = \|\gamma\|_2^2 + \sigma_v^2$. Solving for Σ_{11} in Eq. (3.36) and omitting the negative solution lead to Eq. (3.9).

To show that Σ_{11} is strictly increasing in $\alpha^2 \in \mathbb{R}_{\geq 0}$, we can use the result of Lemma 6 in [2]. For a discrete-time linear system as defined in Eq. (3.1) and Eq. (3.3), given $A = \lambda_1$ and $W = 1$, suppose we have two sensors with the measurement matrices as $C_1 = C_2 = 1$ and the variances of the measurement noise as $V_1 = \alpha_1^2$ and $V_2 = \alpha_2^2$. Define $R \triangleq C^T V^{-1} C$ to be the sensor information matrix corresponding to a sensor with measurement matrix C and measurement noise covariance matrix V. Denote the sensor information matrix of these two sensors as R_1 and R_2. We then have $R_1 = \frac{1}{\alpha_1^2}$ and $R_2 = \frac{1}{\alpha_2^2}$. If $\alpha_1^2 > \alpha_2^2$, we know from Lemma 6 in [2] that $\Sigma_{11}(\alpha_1^2) < \Sigma_{11}(\alpha_2^2)$. Hence, $\Sigma_{11}(\alpha^2)$ is a strictly increasing function of $\alpha^2 \in \mathbb{R}_{\geq 0}$. For $\alpha > 0$, we can rewrite Eq. (3.9) as

$$\Sigma_{11}(\alpha^2) = \frac{2}{\sqrt{(1 - \lambda_1^2 - \frac{1}{\alpha^2})^2 + \frac{4}{\alpha^2}} + 1 - \lambda_1^2 - \frac{1}{\alpha^2}}. \tag{3.37}$$

By letting $\alpha \to \infty$ in Eq. (3.37), we obtain $\lim_{\alpha \to \infty} \Sigma_{11}(\alpha^2) = \dfrac{1}{1 - \lambda_1^2}$.

Proof of (b). Using similar arguments as above, we obtain from Eq. (3.3)

$$y[k] = \mathbf{1}_{n-1} x_1[k] + v'[k],$$

where $v'[k] = \rho \begin{bmatrix} x_2[k] \cdots x_n[k] \end{bmatrix}^T$, which is a zero-mean white noise process with $\mathbb{E}[v'[k](v'[k])^T] = \rho^2 I_{n-1}$. Hence, to compute the MSEE of state 1 of the Kalman filter, i.e., Σ'_{11}, we can consider a system with $A = \lambda_1$, $C = \mathbf{1}_{n-1}$, $W = 1$ and $V = \rho^2 I_{n-1}$. Solving Eq. (3.4) (using the matrix inverse lemma [65]) gives us the result in Eq. (3.10). Using similar arguments as above, we obtain $\lim_{\rho \to \infty} \Sigma'_{11}(\rho^2) = \dfrac{1}{1 - \lambda_1^2}$. ■

3.6.3 A Lemma for X3C

Lemma 3.6.1 *Consider an instance of X3C: a finite set D with $|D| = 3m$, and a collection $\mathcal{C} = \{c_1, \ldots, c_\tau\}$ of τ 3-element subsets of D, where $\tau \geq m$. For each element $c_i \in \mathcal{C}$, define a column vector $g_i \in \mathbb{R}^{3m}$ to encode which elements of D are contained in c_i, i.e., for $i \in \{1, 2, \ldots, \tau\}$ and $j \in \{1, 2, \ldots, 3m\}$, $(g_i)_j = 1$ if element j of set D is in c_i, and $(g_i)_j = 0$ otherwise. Denote $G \triangleq \begin{bmatrix} g_1 & \cdots & g_\tau \end{bmatrix}^T$. For any $l \leq m$ ($l \in \mathbb{Z}$) and $\mathcal{L} \triangleq \{i_1, \ldots, i_l\} \subseteq \{1, \ldots, \tau\}$, define $G_\mathcal{L} \triangleq \begin{bmatrix} g_{i_1} & \cdots & g_{i_l} \end{bmatrix}^T$ and denote rank$(G_\mathcal{L}) = r_\mathcal{L}$.[3]. If the answer to the X3C problem is "no", then for all $\mathcal{L}$ with $|\mathcal{L}| \leq m$, there exists an orthogonal matrix $N \in \mathbb{R}^{3m \times 3m}$ such that*

$$\begin{bmatrix} \mathbf{1}_{3m}^T \\ G_\mathcal{L} \end{bmatrix} N = \begin{bmatrix} \gamma & \beta \\ \mathbf{0} & \tilde{G}_\mathcal{L} \end{bmatrix}, \tag{3.38}$$

where $\tilde{G}_\mathcal{L} \in \mathbb{R}^{l \times r}$ is of full column rank, $\gamma \in \mathbb{R}^{1 \times (3m-r)}$ and at least $\kappa \geq 1$ ($\kappa \in \mathbb{Z}$) elements of γ are 1's , and $\beta \in \mathbb{R}^{1 \times r}$. Further elementary row operations on $\begin{bmatrix} \gamma & \beta \\ \mathbf{0} & \tilde{G}_\mathcal{L} \end{bmatrix}$ transform it into the form $\begin{bmatrix} \gamma & \mathbf{0} \\ \mathbf{0} & \tilde{G}_\mathcal{L} \end{bmatrix}$.

Proof Assume without loss of generality that there are no identical subsets in $\mathcal{C}$. Since rank$(G_\mathcal{L}) = r$, the dimension of the nullspace of $G_\mathcal{L}$ is $3m - r$. We choose an orthonormal basis of the nullspace of $G_\mathcal{L}$ and let it form the first $3m - r$ columns of N, denoted as N_1. Then, we choose an orthonormal basis of the columnspace of $G_\mathcal{L}^T$ and let it form the rest of the r columns of N, denoted as N_2. Clearly, $N = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \in \mathbb{R}^{3m \times 3m}$ is an orthogonal matrix. Furthermore, since the answer to the X3C problem is "no", for any union of $l \leq m$ ($l \in \mathbb{Z}$) subsets in $\mathcal{C}$, denoted as $\mathcal{C}_l$, there exist $\kappa \geq 1$ ($\kappa \in \mathbb{Z}$) elements in D that are not covered by $\mathcal{C}_l$, i.e., $G_\mathcal{L}$ has κ zero columns. Let these be the j_1th, ..., j_κth columns of $G_\mathcal{L}$, where $\{j_1, \ldots, j_\kappa\} \subseteq \{1, \ldots, 3m\}$. Hence, we can always choose $\mathbf{e}_{j_1}, \ldots, \mathbf{e}_{j_\kappa}$ to be in the orthonormal basis of the nullspace of $G_\mathcal{L}$, i.e., as columns of N_1. Constructing N in this way, we have $G_\mathcal{L} N_1 = \mathbf{0}$ and $G_\mathcal{L} N_2 = \tilde{G}_\mathcal{L}$, where $\tilde{G}_\mathcal{L} \in \mathbb{R}^{l \times r}$ is of full column

[3] We drop the subscript $\mathcal{L}$ on r for notational simplicity.

rank since the columns of N_2 form an orthonormal basis of the columnspace of $G_{\mathcal{L}}^T$ and $r \leq l$. Moreover, we have $\mathbf{1}_{3m}^T N_1 = \gamma$ and $\mathbf{1}_{3m}^T N_2 = \beta$, where at least κ elements of γ are 1's (since $\mathbf{1}_{3m}^T \mathbf{e}_{j_s}^T = 1$, $\forall s \in \{1, \ldots, \kappa\}$). Combining these results, we obtain Eq. (3.38). Since $\tilde{G}_{\mathcal{L}}$ is of full column rank, we can perform elementary row operations on $\begin{bmatrix} \gamma & \beta \\ \mathbf{0} & \tilde{G}_{\mathcal{L}} \end{bmatrix}$ and obtain $\begin{bmatrix} \gamma & \mathbf{0} \\ \mathbf{0} & \tilde{G}_{\mathcal{L}} \end{bmatrix}$. ■

3.6.4 Proof of Corollary 3.3.7

We have shown in Theorem 3.3.6 that for any polynomial-time algorithm $\mathcal{A}$ for the priori KFSS problem and any $K \in \mathbb{R}_{\geq 1}$, there exist instances of the priori KFSS problem such that $r_{\mathcal{A}}(\Sigma) > K$ (unless P=NP). Suppose that there exists a polynomial-time constant-factor approximation algorithm $\mathcal{A}'$ for the posteriori KFSS problem, i.e., $\exists K' \in \mathbb{R}_{\geq 1}$ such that $r_{\mathcal{A}'}(\Sigma^*) \leq K'$ for all instances of the posteriori KFSS problem, where $r_{\mathcal{A}'}(\Sigma^*)$ is as defined in Eq. (3.12). We consider an instance of the priori KFSS problem as constructed in the proof of Theorem 3.3.6. We then set the instance of the posteriori KFSS problem to be the same as the constructed instance of the priori KFSS problem. Since $A = \text{diag}(\lambda_1, 0, \ldots, 0) \in \mathbb{R}^{(3m+1)\times(3m+1)}$ and $W = I_{3m+1}$, where $0 < \lambda_1 < 1$, we have from Eq. (3.5)

$$(\Sigma(\mu))_{11} = \lambda_1^2 (\Sigma^*(\mu))_{11} + 1, \forall \mu. \tag{3.39}$$

Since we know from Lemma 3.3.1(a) that $(\Sigma(\mu))_{ii} = 1$, $\forall i \in \{2, \ldots, 3m+1\}$ and $\forall \mu$, it then follows from Eq. (3.39) that

$$\text{trace}(\Sigma(\mu)) = \lambda_1^2 (\Sigma^*(\mu))_{11} + 3m + 1, \forall \mu. \tag{3.40}$$

We also know from Lemma 3.3.1(a) that $0 \leq (\Sigma^*(\mu))_{ii} \leq 1$, $\forall i \in \{2, \ldots, 3m+1\}$ and $\forall \mu$, which implies that

$$\text{trace}(\Sigma^*(\mu)) \leq (\Sigma^*(\mu))_{11} + 3m, \forall \mu. \tag{3.41}$$

We then obtain from Eqs. (3.40)-(3.41) the following:

$$\text{trace}(\Sigma^*(\mu)) \leq \frac{3m\lambda_1^2 + \text{trace}(\Sigma(\mu)) - 3m - 1}{\lambda_1^2} \leq \frac{\text{trace}(\Sigma(\mu))}{\lambda_1^2}, \forall \mu, \tag{3.42}$$

where the second inequality follows from the fact that $0 < \lambda_1 < 1$. Denote optimal sensor selections of the priori and the posteriori KFSS problems as μ_1^* and μ_2^*, respectively. Denote the sensor selection returned by algorithm $\mathcal{A}'$ as μ'. Note that $\Sigma_{opt} = \Sigma(\mu_1^*)$ and $\Sigma_{opt}^* = \Sigma^*(\mu_2^*)$ and $\Sigma_{\mathcal{A}'}^* = \Sigma^*(\mu')$. We then have the following:

$$\begin{aligned}
& \text{trace}(\Sigma_{\mathcal{A}'}^*) \leq K' \text{trace}(\Sigma_{opt}^*) \\
\Rightarrow\ & (\Sigma^*(\mu'))_{11} + \sum_{i=2}^{3m+1} (\Sigma^*(\mu'))_{ii} \leq K' \text{trace}(\Sigma^*(\mu_2^*)) \\
\Rightarrow\ & \frac{(\Sigma(\mu'))_{11} - 1}{\lambda_1^2} \leq K' \text{trace}(\Sigma^*(\mu_2^*)) \leq K' \text{trace}(\Sigma^*(\mu_1^*)) && (3.43) \\
\Rightarrow\ & (\Sigma(\mu'))_{11} - 1 \leq K' \text{trace}(\Sigma(\mu_1^*)) && (3.44) \\
\Rightarrow\ & \text{trace}(\Sigma(\mu')) \leq K' \text{trace}(\Sigma(\mu_1^*)) + 3m + 1 && (3.45) \\
\Rightarrow\ & \frac{\text{trace}(\Sigma(\mu'))}{\text{trace}(\Sigma(\mu_1^*))} \leq K' + \frac{3m+1}{\text{trace}(\Sigma(\mu_1^*))} \leq K' + 1, && (3.46)
\end{aligned}$$

where the first inequality in (3.43) follows from Eq. (3.39) and $(\Sigma^*(\mu'))_{ii} \geq 0, \forall i$ (from Lemma 3.3.1(a)), the second inequality in (3.43) follows from the fact that μ_2^* is an optimal sensor selection for the posteriori KFSS problem, (3.44) follows from (3.42), (3.45) follows from the fact that $\sum_{i=2}^{3m+1}(\Sigma(\mu'))_{ii} = 3m$ (from Lemma 3.3.1(a)), and the second inequality in (3.46) uses the fact that $\text{trace}(\Sigma(\mu_1^*)) \geq 3m + 1$ (from Lemma 3.3.1(a)). Thus, we see from (3.46) that $r_{\mathcal{A}'}(\Sigma) \leq K' + 1$, which contradicts the fact that the priori KFSS problem cannot have a polynomial-time constant-factor approximation algorithm for instances of the given form, and completes the proof of the corollary. ■

3.6.5 Proof of Corollary 3.3.10

Note that the A and W matrices for the instance of KFSA that we constructed in the proof of Theorem 3.3.9 are the same as those for the instance of KFSS that we constructed in the proof of Theorem 3.3.6. We then follow the same arguments as those in the proof of Corollary 3.3.7. Suppose that there exists a polynomial-time constant-factor approximation algorithm $\mathcal{A}'$ for the posteriori KFSA problem, i.e.,

$\exists K' \in \mathbb{R}_{\geq 1}$ such that $r_{\mathcal{A}'}(\tilde{\Sigma}^*) \leq K'$ for all instances of the posteriori KFSA problem, where $r_{\mathcal{A}'}(\tilde{\Sigma}^*)$ is as defined in Eq. (3.20). We consider an instance of the priori KFSA problem as constructed in the proof of Theorem 3.3.9. We then set the instance of the posteriori KFSA problem to be the same as the constructed instance of the priori KFSA problem. Denote an optimal sensor attacks of the priori and the posteriori KFSA problems as ν_1^* and ν_2^*, respectively. Denote the sensor attack returned by algorithm $\mathcal{A}'$ as ν'. Note that $\tilde{\Sigma}_{opt} = \Sigma(\nu_1^{*c})$, $\tilde{\Sigma}^*_{opt} = \Sigma^*(\nu_2^{*c})$ and $\tilde{\Sigma}^*_{\mathcal{A}'} = \Sigma^*(\nu'^c)$. Also note that $\text{trace}(\Sigma^*(\nu_1^{*c})) \leq \text{trace}(\Sigma^*(\nu_2^{*c}))$, since ν_2^* is an optimal sensor attack for the posteriori KFSA problem. We then have the following:

$$
\begin{aligned}
& \text{trace}(\Sigma^*(\nu_1^{*c})) \leq \text{trace}(\Sigma^*(\nu_2^{*c})) \leq K'\text{trace}(\tilde{\Sigma}^*_{\mathcal{A}'}) \\
\Rightarrow\ & (\Sigma^*(\nu_1^{*c}))_{11} + \sum_{i=2}^{3m+1} (\Sigma^*(\nu_1^{*c}))_{ii} \leq K'\text{trace}(\Sigma^*(\nu'^c)) \\
\Rightarrow\ & \frac{(\Sigma(\nu_1^{*c}))_{11} - 1}{\lambda_1^2} \leq K'\text{trace}(\Sigma^*(\nu'^c)) \\
\Rightarrow\ & (\Sigma(\nu_1^{*c}))_{11} - 1 \leq K'\text{trace}(\Sigma(\nu'^c)) \\
\Rightarrow\ & \text{trace}(\Sigma(\nu_1^{*c})) \leq K'\text{trace}(\Sigma(\nu'^c)) + 3m + 1 \\
\Rightarrow\ & \frac{\text{trace}(\Sigma(\nu_1^{*c}))}{\text{trace}(\Sigma(\nu'^c))} \leq K' + \frac{3m+1}{\text{trace}(\Sigma(\nu'^c))} \leq K' + 1,
\end{aligned}
$$

which implies that $r_{\mathcal{A}'}(\tilde{\Sigma}) \leq K' + 1$, and yields a contradiction with the fact that the priori KFSA problem cannot have a polynomial-time constant-factor approximation algorithm for the instances of the form given as above. This completes the proof of the corollary. ■

3.6.6 Proof of Theorem 3.4.2

We first prove that Algorithm 3.4.1 for the priori KFSS problem selects sensor 2 and sensor 3 in its first and second iterations, respectively. Since the only nonzero eigenvalue of A is λ_1, we know from Lemma 3.3.1(a) that $(\Sigma(\mu))_{22} = 1$ and $(\Sigma(\mu))_{33} = 1$, $\forall \mu$, which implies that $(\Sigma_{gre})_{22} = 1$ and $(\Sigma_{gre})_{33} = 1$. Hence, we focus on determining $(\Sigma_{gre})_{11}$.

Denoting $\mu_1 = [1\ 0\ 0]^T$ and $\mu_2 = [0\ 1\ 0]^T$, we have $C(\mu_1) = [1\ h\ h]$ and $C(\mu_2) = [1\ 0\ h]$. Using the result in Lemma 3.3.2(a), we see that $\sigma_1 \triangleq (\Sigma(\mu_1))_{11}$ and $\sigma_2 \triangleq (\Sigma(\mu_2))_{11}$ satisfy

$$\sigma_1 = \frac{2}{\sqrt{(1-\lambda_1^2-\frac{1}{2h^2})^2+\frac{2}{h^2}}+1-\lambda_1^2-\frac{1}{2h^2}},$$

and

$$\sigma_2 = \frac{2}{\sqrt{(1-\lambda_1^2-\frac{1}{h^2})^2+\frac{4}{h^2}}+1-\lambda_1^2-\frac{1}{h^2}},$$

respectively. Similarly, denoting $\mu_3 = [0\ 0\ 1]^T$, we obtain $C(\mu_3) = [0\ 1\ 1]$. Since the first column of $C(\mu_3)$ is zero, we know from Lemma 3.3.1(b) that $\sigma_3 \triangleq (\Sigma(\mu_3))_{11} = \frac{1}{1-\lambda_1^2}$. If we view σ_2 as a function of h^2, denoted as $\sigma(h^2)$, we have $\sigma_1 = \sigma(2h^2)$. Since we know from Lemma 3.3.2(a) that $\sigma(h^2)$ is a strictly increasing function of $h^2 \in \mathbb{R}_{>0}$ and upper bounded by $\frac{1}{1-\lambda_1^2}$, we obtain $\sigma_2 < \sigma_1 < \sigma_3$, which implies that the greedy algorithm selects sensor 2 in its first iteration.

Denote $\mu_{12} = [1\ 1\ 0]^T$. We have $C(\mu_{12}) = \left[\begin{smallmatrix} 1 & h & h \\ 1 & 0 & h \end{smallmatrix}\right]$, on which we perform elementary row operations and obtain $\tilde{C}(\mu_{12}) = \left[\begin{smallmatrix} 0 & h & 0 \\ 1 & 0 & h \end{smallmatrix}\right]$. By direct computation from Eq. (3.4), we obtain $(\Sigma(\mu_{12}))_{11} = \sigma_2$. Moreover, we denote $\mu_{23} = [0\ 1\ 1]^T$ and obtain $C(\mu_{23}) = \left[\begin{smallmatrix} 1 & 0 & h \\ 0 & 1 & 1 \end{smallmatrix}\right]$. By direct computation from Eq. (3.4), we have $(\Sigma(\mu_{23}))_{11}$, denoted as σ_{23}, to be

$$\sigma_{23} = \frac{2}{\sqrt{(1-\lambda_1^2-\frac{2}{h^2})^2+\frac{8}{h^2}}+1-\lambda_1^2-\frac{2}{h^2}}.$$

Similarly to the argument above, we have $\sigma_{12} = \sigma(h^2)$ and $\sigma_{23} = \sigma(\frac{h^2}{2})$, where $\sigma(\frac{h^2}{2}) < \sigma(h^2)$, which implies that the greedy algorithm selects sensor 3 in its second iteration. Hence, we have $\text{trace}(\Sigma_{gre}) = \sigma_{23} + 2$.

Furthermore, it is easy to see that the optimal sensor selection (for the priori KFSS instance) is $\mu = [1\ 0\ 1]^T$, denoted as μ_{13}. Since if $\mu = \mu_{13}$, then $\mathbf{e}_1 \in \text{rowspace}(C(\mu))$ and thus we know from Lemma 3.3.1(a) and (c) that $\text{trace}(\Sigma(\mu)) = 3 = \text{trace}(W)$, which is also the minimum value of $\text{trace}(\Sigma(\mu))$ among all possible sensor selections μ. Combining the results above and taking the limit as $h \to \infty$ lead to Eq. (3.32).

We next prove that the greedy algorithm defined in Algorithm 3.4.1 for the posteriori KFSS problem selects sensor 2 and sensor 3 in its first and second iterations, respectively. Note that it is easy to see from Eq. (3.5) that $\Sigma(\mu)$ is of the form $\Sigma(\mu) = \text{diag}((\Sigma(\mu))_{11}, 1, 1)$, $\forall \mu$. Hence, we see from Eq. (3.6) that $\text{trace}(\Sigma^*(\mu_1)) = 2 + \frac{h^2}{\frac{\sigma_1}{2}+h^2}(\sigma_1 - 1)$, $\text{trace}(\Sigma^*(\mu_2)) = 2 + \frac{h^2}{\sigma_2+h^2}(\sigma_2 - 1)$ and $\text{trace}(\Sigma^*(\mu_3)) = 2 + \frac{1}{1-\lambda_1^2} - 1$, where $\sigma_1 = \sigma(2h^2)$ and $\sigma_2 = \sigma(h^2)$ are defined above. Since $\sigma(h^2)$ is a strictly increasing function of $h^2 \in \mathbb{R}_{>0}$ with $\sigma(h^2) \geq 1$ and upper bounded by $\frac{1}{1-\lambda_1^2}$, and it is easy to obtain $\frac{\sigma_1}{2} < \sigma_2$, it then follows that Algorithm 3.4.1 for the posteriori KFSS problem selects sensor 2 in its first iteration.

Similarly, Eq. (3.6) implies $\text{trace}(\Sigma^*(\mu_{12})) = 1 + \frac{h^2}{\sigma_2+h^2}(\sigma_2 - 1)$, $\text{trace}(\Sigma^*(\mu_{23})) = 1 + \frac{h^2}{2\sigma_{23}+h^2}(\sigma_{23} - 1)$ and $\text{trace}(\Sigma^*(\mu_{13})) = 1$, where $\sigma_{23} = \sigma(\frac{h^2}{2})$ is defined above. Since $\sigma(h^2)$ is strictly increasing in $h^2 \in \mathbb{R}_{>0}$ with $\sigma(h^2) \geq 1$ and upper bounded by $\frac{1}{1-\lambda_1^2}$, and it is easy to check that $\sigma_2 < 2\sigma_{23}$, it follows that the greedy algorithm selects sensor 3 in its second iteration, and $\mu = \mu_{13}$ is the optimal sensor selection (for the posteriori KFSS instance). Combining the results above and letting $h \to \infty$, we obtain Eq. (3.33). ■

3.6.7 Proof of Theorem 3.4.4

We first analyze Algorithm 3.4.2 for the priori KFSA problem. Since the only nonzero eigenvalue of A is λ_1, we know from Lemma 3.3.1(a) that $(\Sigma(\nu^c))_{22} = 1$ and $(\Sigma(\nu^c))_{33} = 1$, $\forall \nu$, which implies that $(\tilde{\Sigma}_{gre})_{22} = 1$ and $(\tilde{\Sigma}_{gre})_{33} = 1$. Hence, we only need to determine $(\tilde{\Sigma}_{gre})_{11}$.

First, denote $\nu_1 = [1\ 0\ 0\ 0]^T$, $\nu_2 = [0\ 1\ 0\ 0]^T$ and $\nu_3 = [0\ 0\ 1\ 0]^T$. Then, it is easy to see that $C(\nu^c)$ is of full column rank for all $\nu \in \{\nu_1, \nu_2, \nu_3\}$. This implies that $\mathbf{e}_1 \in \text{rowspace}(C(\nu^c))$ for all $\nu \in \{\nu_1, \nu_2, \nu_3\}$. Thus, we know from Lemma 3.3.1(c) that $(\Sigma(\nu^c))_{11} = 1$, $\forall \nu \in \{\nu_1, \nu_2, \nu_3\}$. Moreover, denoting $\nu_4 = [0\ 0\ 0\ 1]^T$, we have $C(\nu_4^c)$ (after some elementary row operations and merging identical rows) is of the

form $C(\nu_4^c) = \left[\begin{smallmatrix} 1 & 0 & h \\ 0 & 1 & 0 \end{smallmatrix}\right]$. Using the results from the proof of Theorem 3.4.2, we obtain that $\sigma_4' \triangleq (\Sigma(\nu_4^c))_{11}$ satisfies

$$\sigma_4' = \frac{1 + h^2\lambda_1^2 - h^2 + \sqrt{(h^2 - h^2\lambda_1^2 - 1)^2 + 4h^2}}{2}.$$

If we view σ_4' as a function of h^2, denoted as $\sigma'(h^2)$, we know from Lemma 3.3.2(a) that $\sigma'(h^2)$ is a strictly increasing function of $h^2 \in \mathbb{R}_{\geq 0}$ with $\sigma'(0) = 1$, which implies $\sigma_4' > 1$. Thus, Algorithm 3.4.2 for priori KFSA targets sensor 4 in its first iteration.

Second, denote $\nu_{14} = [1\ 0\ 0\ 1]^T$, $\nu_{24} = [0\ 1\ 0\ 1]^T$ and $\nu_{34} = [0\ 0\ 1\ 1]^T$. We obtain that $C(\nu^c)$ (after some elementary row operations) is of the form $C(\nu^c) = \left[\begin{smallmatrix} 1 & 0 & h \\ 0 & 1 & 0 \end{smallmatrix}\right]$, for all $\nu \in \{\nu_{14}, \nu_{24}, \nu_{34}\}$. It follows that $(\Sigma(\nu^c))_{11} = \sigma_4'$ for all $\nu \in \{\nu_{14}, \nu_{24}, \nu_{34}\}$, which implies that $\text{trace}(\tilde{\Sigma}_{gre}) = \sigma_4' + 2$.

Furthermore, the optimal sensor attack (for the priori KFSA instance) is $\nu = \nu_{12}$, where $\nu_{12} = [1\ 1\ 0\ 0]^T$, since in this case we know from Lemma 3.3.1(a) and (b) that $\Sigma(\nu^c)_{11} = \frac{1}{1-\lambda_1^2}$, which is also the maximum value of $\Sigma(\nu^c)_{11}$ that it can achieve, i.e., $\tilde{\Sigma}_{opt} = \frac{1}{1-\lambda_1^2} + 2$. Combining the results above and taking the limit as $h \to 0$, we obtain Eq. (3.34).

We next analyze Algorithm 3.4.2 for the posteriori KFSA problem. Since we know from previous arguments that $C(\nu^c)$ is of full column rank for all $\nu \in \{\nu_1, \nu_2, \nu_3\}$, it follows from Lemma 3.3.1(c) that $\text{trace}(\Sigma^*(\nu_1^c)) = \text{trace}(\Sigma^*(\nu_2^c)) = \text{trace}(\Sigma^*(\nu_3^c)) = 0$. Moreover, it is easy to obtain from Eq. (3.5) that $\Sigma(\nu^c)$ is of the form $\Sigma(\nu^c) = \text{diag}((\Sigma(\nu^c))_{11}, 1, 1)$, $\forall \nu$. We then see from Eq. (3.6) that $\text{trace}(\Sigma^*(\nu_4^c)) = 1 + \frac{h^2}{\sigma_4' + h^2}(\sigma_4' - 1)$, where $\sigma_4' = \sigma'(h^2)$ is defined above. Since $\sigma'(h^2)$ is strictly increasing in $h^2 \in \mathbb{R}_{\geq 0}$ with $\sigma'(0) = 1$, it implies that Algorithm 3.4.2 for posteriori KFSA targets sensor 4 in its first iteration. Similarly, we have from Eq. (3.6) $\text{trace}(\Sigma^*(\nu_{14}^c)) = \text{trace}(\Sigma^*(\nu_{24}^c)) = \text{trace}(\Sigma^*(\nu_{34}^c)) = 1 + \frac{h^2}{\sigma_4' + h^2}(\sigma_4' - 1)$, which implies $\text{trace}(\tilde{\Sigma}_{gre}^*) = 1 + \frac{h^2}{\sigma_4' + h^2}(\sigma_4' - 1)$.

Furthermore, denote $\nu_{23} = [0\ 1\ 1\ 0]^T$ and $\nu_{13} = [1\ 0\ 1\ 0]^T$. It is easy to show, via similar arguments to those above, that $\text{trace}(\Sigma^*(\nu^c)) = 1 + \frac{h^2}{\sigma_4' + h^2}(\sigma_4' - 1)$ for all $\nu \in \{\nu_{34}, \nu_{24}, \nu_{23}, \nu_{14}\}$, $\text{trace}(\Sigma^*(\nu_{13}^c)) = 1$, and $\text{trace}(\Sigma^*(\nu_{12}^c)) = 1 + \frac{1}{1-\lambda_1^2} - 1$. Since

$\sigma_4' = \sigma'(h^2)$ is strictly increasing in $h^2 \in \mathbb{R}_{\geq 0}$ with $\sigma'(0) = 1$, and upper bounded by $\frac{1}{1-\lambda_1^2}$, it follows that the optimal sensor attack (for the posteriori KFSA instance) is $\nu = \nu_{12}$. Combining the results above and taking the limit as $h \to 0$, we obtain Eq. (3.35). ■

4. RESILIENT SENSOR SELECTION FOR KALMAN FILTERING IN NETWORKED SYSTEMS

4.1 Introduction

In Chapter 3, the sensor selection problem for Kalman filtering was shown to be NP-hard and inapproximable within any constant factor (if P $\neq$ NP) in general. Therefore, we consider special classes of this problem in this chapter and seek polynomial-time algorithms for the optimal sensor selection problem. Specifically, we consider a discrete-time linear dynamical system whose states represent nodes in a directed network, and interact according to the topology of the network. The nodes of the network are possibly affected by stochastic inputs. Such networked systems with stochastic inputs have received much attention from researchers recently (e.g., [13–17]). In this chapter, we focus on the case when there is a single node of the network that is affected by a stochastic input. This model encompasses models of diffusion networks that arise in many different areas, including, for instance, information and influence diffusion over social networks [18], spreading of diseases in populations [19] and diffusion of chemicals in certain environments [20]. In such applications, estimating the states of the entire network is an important objective.

Moreover, we consider the scenario where a sensor located at a certain node can give measurements of the state corresponding to the node. A system designer can then select a subset of sensors (at design-time) that correspond to all the nodes over the network, in order to minimize the trace of the steady-state error covariance of the Kalman filter associated with to the selected sensors. In addition, we assume that selecting the sensor of a certain node in the network incurs a selection cost for the designer. The sensor selection costs of different nodes can potentially vary, since they

may, for instance, be located in different environment conditions. We refer to this problem as the *Graph-based Kalman Filtering Sensor Selection (GKFSS)* problem.

Additionally, as we argued in the previous chapter, the systems that we are interested in monitoring may be targeted by adversaries. Here, we again consider adversaries that perform DoS attacks on sensors by simply removing them (or equivalently, dropping all the measurement data). The goal of the adversary is to remove a subset of selected sensors under a budget constraint in order to maximize the trace of the steady-state error covariance of the Kalman filter corresponding to the surviving sensors. We assume that attacking the sensor of a certain node incurs an attack cost (which could also vary across the nodes). In contrast with existing work in the literature, we analyze the problem using the graph structure of the systems. We refer to this problem as the *Graph-based Kalman Filtering Sensor Attack (GKFSA)* problem.

Finally, combining the two problems that we considered above, we formulate and study the resilient sensor selection problem for networked systems where there is a single node in the network that has the stochastic input. In our formulation, we assume that the system designer is aware of the potential attack from an adversary who chooses to optimally attack the sensors (subject to attack budget constraints) deployed by the system designer. The system designer's goal then becomes to select sensors (under selection budget constraints) of a subset of nodes in order to minimize the trace of the steady-state error covariance of the Kalman filter corresponding to the surviving sensors after the attack. We refer to this problem as the *Resilient Graph-based Kalman Filtering Sensor Selection (RGKFSS)* problem.

Related Work

The (design-time) selection problem has been widely studied in the literature. For example, in [50, 57], the authors considered the Kalman filtering sensor selection problem over a finite number of time steps. Here, we study the problem of optimizing steady-state error covariances of the corresponding Kalman filter. In Chapter 3, we

considered the same sensor selection problem as the one considered here, but for general system dynamics. In such cases, we showed that finding an optimal selection for the general problem is NP-hard. Thus, in this chapter, we impose additional structure on the problem (by considering the graph representation of the dynamics) in order to seek optimal solutions. In [66, 67], the authors studied the sensor selection problem for estimating a static variable (parameter) that does not change over time. Here, we study the problem of selecting sensors to estimate the states of a linear dynamical system affected by stochastic inputs. In contrast to the sensor selection problem where the set of selected sensors cannot change over time, the sensor scheduling problem for Kalman filtering has also received much attention (e.g., [35–37, 68]), where different sets of sensors can be chosen at different time steps.

In networked system settings, the authors in [69] considered the sensor selection problem for continuous-time diffusion dynamics, and applied the Wiener filter to estimate the system states using sensor measurements. Here, we consider discrete-time networked system dynamics and apply the Kalman filter to estimate the system states. The authors in [14–16] studied the leader selection problem in consensus networks with stochastic inputs. The problem is to select a subset of nodes whose states are fixed over time in order to minimize the H_2 norm of the system states at steady state. In contrast, we consider the problem of selecting sensors among the nodes of systems with more general dynamics in order to minimize the trace of the steady-state error covariance of the Kalman filter.

Although both of the sensor selection and the sensor attack problems have received much attention from researchers, the resilient sensor selection is less explored. The authors in [9] considered the problem of resilient maximization of monotone submodular set functions under a cardinality constraint on the sets. They proposed a polynomial-time approximation algorithm for the problem with performance bounds that depend on the curvature of the objective function. In [8], the authors considered a resilient observation selection problem. The problem is to resiliently select observations of a scalar Gaussian process given that some of the selected observations

could be removed by an adversary. The authors showed that this problem is NP-hard and proposed a greedy algorithm with a provable performance guarantee. Here, we consider the resilient sensor selection problem for Kalman filtering of (vector) linear dynamical systems subject to general knapsack constraints. While we show this problem is NP-hard, we give an algorithm based on dynamic programming to solve the problem optimally in pseudo-polynomial time [41].

Summary of Results

First, we provide an optimal sensor selection strategy, computed in polynomial time, for the GKFSS problem using the graph structure of the system. Second, leveraging the insights for the GKFSS problem, we give an optimal sensor attack strategy, computed in polynomial time, for the GKFSA problem. Third, we show that the RGKFSS problem is NP-hard; we then provide an algorithm based on dynamic programming that can return an optimal solution to general instances of the RGKFSS problem in pseudo-polynomial time. Although the results are derived under the assumption that the sensors give perfect measurements, we show that how to apply these results to analyze the case with sensor measurement noise and provide numerical examples.

The results presented in this chapter were published as [49, 70].

4.2 Problem Formulation

We begin with the following definitions from graph theory. Further details can be found in, for example, [71] and [65].

Definition 4.2.1 *For any given matrix $A \in \mathbb{R}^{n\times n}$, the directed graph of A, denoted as $\mathcal{G}(A)$, is defined as the directed graph on n vertices (or nodes) $x_1, x_2, \ldots, x_n$ such that for all $i, j \in \{1, 2, \ldots, n\}$, there is a directed edge in $\mathcal{G}(A)$ from x_j to x_i, denoted as (x_j, x_i), if and only if $A_{ij} \neq 0$. Denoting the set of vertices and the set of edges*

of $\mathcal{G}(A)$ as $\mathcal{X}(A) \triangleq \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{E}(A)$, respectively, the graph $\mathcal{G}(A)$ is also denoted as $\mathcal{G}(A) = \{\mathcal{X}(A), \mathcal{E}(A)\}$.

Definition 4.2.2 *Consider a directed graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ with $\mathcal{X} \triangleq \{x_1, x_2, \ldots, x_n\}$. A directed path from x_{i_0} to x_{i_t} is a sequence of directed edges $(x_{i_0}, x_{i_1}), (x_{i_1}, x_{i_2}), \ldots, (x_{i_{t-1}}, x_{i_t})$ in $\mathcal{G}$. The ordered list of vertices in the directed path is $x_{i_0}, x_{i_1}, \ldots, x_{i_t}$. The length of a directed path is the number of directed edges in the directed path. A cycle is a directed path that begins and ends at the same vertex which occurs exactly twice in the ordered list of vertices in the directed path, and no other vertices occur more than once in the list. A cycle of length 1 is a self-loop at the corresponding vertex.*

Definition 4.2.3 *Consider a directed graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$. For any pair of distinct vertices $x_i, x_j \in \mathcal{X}$ such that there exists a directed path from x_i to x_j, the distance from x_i to x_j, denoted as d_{ij}, is defined as the shortest length over all such paths. Define $d_{mm} = 0$ for all $x_m \in \mathcal{X}$.*

Definition 4.2.4 *A directed graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ is strongly connected if for all pairs of distinct vertices $x_i, x_j \in \mathcal{X}$, there is a directed path from x_j to x_i in $\mathcal{G}$.*

We start with a general system model. Consider a matrix $A \in \mathbb{R}^{n \times n}$ with the associated graph $\mathcal{G}(A) = \{\mathcal{X}(A), \mathcal{E}(A)\}$ (given in Definition 4.2.1). Suppose $\mathcal{I} \triangleq \{x_{i_0}, x_{i_1}, \ldots, x_{i_{n_1-1}}\} \subseteq \mathcal{X}(A)$ is the set of nodes that have stochastic inputs, where $n_1 \in \mathbb{Z}_{\geq 1}$. We then consider the following discrete-time linear system:

$$x[k+1] = Ax[k] + Bw[k], \tag{4.1}$$

where $x[k] \in \mathbb{R}^n$ is the system state at time step k, and $B \triangleq \begin{bmatrix} \mathbf{e}_{i_0}^T & \cdots & \mathbf{e}_{i_{n_1-1}}^T \end{bmatrix} \in \mathbb{R}^{n \times n_1}$ is the input matrix. The stochastic input $w[k] \in \mathbb{R}^{n_1}$ is a zero-mean white noise process with $\mathbb{E}[w[k](w[k])^T] = W \in \mathbb{S}_+^{n_1}$. The initial state $x[0]$ is a random vector with mean $\bar{x}_0 \in \mathbb{R}^n$ and covariance $\Pi_0 \in \mathbb{S}_+^n$, and is assumed to be independent of $w[k]$ for all $k \in \mathbb{Z}_{\geq 0}$. Each state of the system, denoted as $x_i[k]$, is associated with

node x_i in $\mathcal{G}(A)$. As we mentioned in the introduction, we showed in Chapter 3 that the Kalman filtering sensor selection problem cannot be approximated within any constant factor in polynomial time (if P $\neq$ NP) for general system dynamics matrices *even* when the measurement noise is zero. Moreover, under the networked system setting, [49] showed that if there are multiple input nodes in the graph, the Kalman filtering sensor selection problem becomes NP-hard *even* when the graph only contains a set of disjoint paths of length three and each path has a single input node. Hence, in order to bypass these inherent complexity issues, we focus on networked systems with a single input node $x_{i_0} \in \mathcal{X}(A)$ (i.e., $B = \mathbf{e}_{i_0}^T$ and $\mathbb{E}[(w[k])^2] = \sigma_w^2 \in \mathbb{R}_{\geq 0}$), and seek efficient algorithms to optimally solve the corresponding sensor selection, sensor attack, and resilient sensor selection problems. We assume throughout this chapter that the pair $(A, B\sigma_w)$ is stabilizable. The generality of this assumption will be justified later in this chapter.

4.2.1 The Sensor Selection Problem

First, suppose that there is a system designer who can select sensors of a subset of the vertices of the graph $\mathcal{G}(A)$ under a budget constraint. Specifically, the sensor of node $x_i \in \mathcal{X}(A)$ has a selection cost $h_i \in \mathbb{Z}_{\geq 0}$; define the sensor selection cost vector as $h \triangleq \begin{bmatrix} h_1 & \cdots & h_n \end{bmatrix}^T$. The designer has a sensor selection budget $H \in \mathbb{Z}_{\geq 0}$ that can be spent on selecting sensors of the nodes in $\mathcal{G}(A)$. Moreover, the sensor of node $x_i \in \mathcal{X}(A)$ gives a measurement of the form

$$y_i[k] = C_i x[k] + v_i[k], \tag{4.2}$$

where $C_i = \mathbf{e}_i$ and $v_i[k] \in \mathbb{R}$ is a zero-mean white noise process. We further define $y[k] \triangleq \begin{bmatrix} y_1[k] & \cdots & y_n[k] \end{bmatrix}^T$, $C \triangleq \begin{bmatrix} C_1^T & \cdots & C_n^T \end{bmatrix}^T$ and $v[k] \triangleq \begin{bmatrix} v_1[k] & \cdots & v_n[k] \end{bmatrix}^T$. Thus, the output provided by all sensors together is given by

$$y[k] = Cx[k] + v[k], \tag{4.3}$$

where $C = I_n$. We denote $\mathbb{E}[v[k](v[k])^T] = V \in \mathbb{S}^n_+$ and consider $\mathbb{E}[v[k](w[j])^T] = \mathbf{0}$, $\forall k, j \in \mathbb{Z}_{\geq 0}$. The initial state $x[0]$ is also assumed to be independent of $v[k]$ for all $k \in \mathbb{Z}_{\geq 0}$.

After the sensors are selected, the Kalman filter is then applied to provide an estimate of the states using the measurements from the installed sensors. We define a vector $\mu \in \{0,1\}^n$ to be the indicator vector indicating the vertices where sensors are selected. Specifically, $\mu_i = 1$ if and only if a sensor is selected at node $x_i \in \mathcal{X}(A)$. Let $C(\mu)$ denote the measurement matrix of the installed sensors indicated by μ, i.e., $C(\mu) \triangleq \begin{bmatrix} C_{i_1}^T & \cdots & C_{i_p}^T \end{bmatrix}^T$, where $\text{supp}(\mu) = \{i_1, \ldots, i_p\} \subseteq \{1, \ldots, n\}$. Similarly, let $V(\mu)$ denote the measurement noise covariance matrix of the installed sensors, i.e., $V(\mu) = \mathbb{E}[\tilde{v}[k](\tilde{v}[k])^T]$, where $\tilde{v}[k] \triangleq \begin{bmatrix} (v[k])_{i_1} & \cdots & (v[k])_{i_p} \end{bmatrix}^T$. The *a priori* and the *a posteriori* error covariance matrices of the Kalman filter at time step k, when the sensors indicated by μ are selected, are denoted as $\Sigma_{k/k-1}(\mu)$ and $\Sigma_{k/k}(\mu)$, respectively. The initial *a priori* error covariance is set as $\Sigma_{0/-1}(\mu) = \Pi_0$. The limit $\Sigma(\mu) \triangleq \lim_{k\to\infty} \Sigma_{k+1/k}$ (also known as the steady-state *a priori* error covariance), if it exists, satisfies the *discrete algebraic Riccati equation (DARE)* [61]:

$$\Sigma(\mu) = A\Sigma(\mu)A^T + \sigma_w^2 BB^T - A\Sigma(\mu)C(\mu)^T\big(C(\mu)\Sigma(\mu)C(\mu)^T + V(\mu)\big)^{-1}C(\mu)\Sigma(\mu)A^T, \tag{4.4}$$

where $\sigma_w^2 \in \mathbb{R}_{\geq 0}$ and $B = \mathbf{e}_{i_0}^T$. The limit $\Sigma^*(\mu) \triangleq \lim_{k\to\infty} \Sigma_{k/k}(\mu)$ (also known as the steady-state *a posteriori* error covariance), if it exists, satisfies the following equations [62]:

$$\Sigma^*(\mu) = \Sigma(\mu) - \Sigma(\mu)C(\mu)^T\big(C(\mu)\Sigma(\mu)C(\mu)^T + V(\mu)\big)^{-1}C(\mu)\Sigma(\mu), \tag{4.5}$$

and

$$\Sigma(\mu) = A\Sigma^*(\mu)A^T + \sigma_w^2 BB^T. \tag{4.6}$$

The inverses in Eqs. (4.4) and (4.5) are interpreted as the Moore-Penrose pseudo-inverses (which we denote using the notation "†") if the arguments are not invertible [61]. We will use the following result from [61].

Lemma 4.2.1 *For a given indicator vector μ, $\Sigma_{k/k-1}(\mu)$ (resp., $\Sigma_{k/k}(\mu)$) will converge, as $k \to \infty$, to a finite limit $\Sigma(\mu)$ (resp., $\Sigma^*(\mu)$), regardless of the initial covariance $\Sigma_{0/-1}(\mu)$, if and only if the pair $(A, C(\mu))$ is detectable and the pair $(A, B\sigma_w)$ is stabilizable. Furthermore, if the limit $\Sigma(\mu)$ (resp., $\Sigma^*(\mu)$) exists, it is also the only positive semi-definite solution to Eq.* (4.4) *(resp., Eq.* (4.5)*).*

When the pair $(A, C(\mu))$ is not detectable, we define the limits $\Sigma(\mu) = +\infty$ and $\Sigma^*(\mu) = +\infty$. The priori and posteriori Graph-based Kalman Filtering Sensor Selection (GKFSS) problems are defined as follows.

Problem 4.2.2 *(Priori and Posteriori GKFSS) Consider a system dynamics matrix $A \in \mathbb{R}^{n\times n}$ with the associated graph $\mathcal{G}(A) = \{\mathcal{X}(A), \mathcal{E}(A)\}$, a single vertex $x_{i_0} \in \mathcal{X}(A)$ that has a stochastic input with variance $\sigma_w^2 \in \mathbb{R}_{\geq 0}$, the measurement matrix $C = I_n$ (containing all of the individual sensor measurement matrices), a sensor noise covariance matrix $V \in \mathbb{S}_+^n$, a sensor selection cost vector $h \in \mathbb{Z}_{\geq 0}^n$ and a sensor selection budget $H \in \mathbb{Z}_{\geq 0}$. The priori Graph-based Kalman Filtering Sensor Selection (GKFSS) problem is to find the sensor selection μ, i.e., the indicator vector μ of the vertices where sensors are selected, that solves*

$$\min_{\mu\in\{0,1\}^n} \; trace(\Sigma(\mu))$$
$$s.t.\ h^T\mu \leq H,$$

where $\Sigma(\mu)$ is given by Eq. (4.4) *if the pair $(A, C(\mu))$ is detectable, and $\Sigma(\mu) = +\infty$ otherwise. The posteriori GKFSS Problem is to find the sensor selection μ that solves*

$$\min_{\mu\in\{0,1\}^n} \; trace(\Sigma^*(\mu))$$
$$s.t.\ h^T\mu \leq H,$$

where $\Sigma^*(\mu)$ is given by Eq. (4.5) *if the pair $(A, C(\mu))$ is detectable, and $\Sigma^*(\mu) = +\infty$ otherwise.*

4.2.2 The Sensor Attack Problem

Suppose that the sensors indicated by the sensor selection $\mu \in \{0,1\}^n$ are selected and installed by the system designer, and there is an adversary who aims to attack (i.e., remove) a subset of the installed sensors. To attack a sensor selected at node $x_i \in \mathcal{X}(A)$, the adversary needs to pay a cost $f_i \in \mathbb{Z}_{\geq 0}$. Define the sensor attack cost vector as $f \triangleq \begin{bmatrix} f_1 & \cdots & f_n \end{bmatrix}^T$. The adversary has a total sensor attack budget $F \in \mathbb{Z}_{\geq 0}$ for attacking the installed sensors. We define a vector $\nu \in \{0,1\}^n$ to be the indicator vector indicating the subset of sensors that are attacked, where $\nu_i = 1$ if and only if the sensor at $x_i \in \mathcal{X}(A)$ is attacked. Note that $\mathrm{supp}(\nu) \subseteq \mathrm{supp}(\mu)$ is always assumed implicitly in the sequel. Let $C(\mu \setminus \nu)$ denote the measurement matrix of the surviving sensors corresponding to μ and ν, i.e., $C(\mu \setminus \nu) \triangleq \begin{bmatrix} C_{j_1}^T & \cdots & C_{j_q}^T \end{bmatrix}^T$, where $\{j_1, \ldots, j_q\} = \mathrm{supp}(\mu) \setminus \mathrm{supp}(\nu)$. Denote $\mathrm{supp}(\mu) \setminus \mathrm{supp}(\nu) \triangleq \mathrm{supp}(\mu \setminus \nu)$. Similarly, define $V(\mu \setminus \nu)$ to be the measurement noise covariance of the surviving sensors. The Kalman filter is then applied based on the measurements of the surviving sensors. The resulting *a priori* and *a posteriori* error covariances of the Kalman filter at time step k are denoted as $\Sigma_{k/k-1}(\mu \setminus \nu)$ and $\Sigma_{k/k}(\mu \setminus \nu)$, respectively, whose limits as $k \to \infty$ are denoted as $\Sigma(\mu \setminus \nu)$ and $\Sigma^*(\mu \setminus \nu)$, respectively.

The priori and posteriori Graph-based Kalman Filtering Sensor Attack (GKFSA) problems are then defined as follows.

Problem 4.2.3 *(Priori and Posteriori GKFSA) Consider a system dynamics matrix $A \in \mathbb{R}^{n \times n}$ with the associated graph $\mathcal{G}(A) = \{\mathcal{X}(A), \mathcal{E}(A)\}$, a single vertex $x_{i_0} \in \mathcal{X}(A)$ that has a stochastic input with variance $\sigma_w^2 \in \mathbb{R}_{\geq 0}$, the measurement matrix $C = I_n$ (containing all of the individual sensor measurement matrices), a sensor noise covariance matrix $V \in \mathbb{S}_+^n$, a sensor attack cost vector $f \in \mathbb{Z}_{\geq 0}^n$, a sensor attack budget $F \in \mathbb{Z}_{\geq 0}$, and a sensor selection vector $\mu \in \{0,1\}^n$. The priori Graph-based Kalman Filtering Sensor Attack (GKFSA) problem is to find the sensor attack ν, i.e.,*

the indicator vector ν of the vertices where the installed sensors (indicated by μ) are attacked, that solves

$$\max_{\nu\in\{0,1\}^n} \textit{trace}(\Sigma(\mu \setminus \nu))$$
$$s.t.\ f^T\nu \leq F,$$

where $\Sigma(\mu\setminus\nu)$ is given by Eq. (4.4) if the pair $(A, C(\mu\setminus\nu))$ is detectable, and $\Sigma(\mu\setminus\nu) = +\infty$ otherwise. The posteriori GKFSA problem is to find the sensor attack ν that solves

$$\max_{\nu\in\{0,1\}^n} \textit{trace}(\Sigma^*(\mu \setminus \nu))$$
$$s.t.\ f^T\nu \leq F,$$

where $\Sigma^*(\mu \setminus \nu)$ is given by Eq. (4.5) if the pair $(A, C(\mu \setminus \nu))$ is detectable, and $\Sigma^*(\mu \setminus \nu) = +\infty$ otherwise.

4.2.3 The Resilient Sensor Selection Problem

We next consider the scenario where the system designer is aware of the potential attack from a strategic adversary (who can perform optimal sensor attacks under budget constraints), and aims to choose a resilient sensor selection. We first define feasible sensor selections for the system designer as follows.

Definition 4.2.5 *A sensor selection $\mu \in \{0,1\}^n$ is said to be feasible if $h^T\mu \leq H$ (i.e., the sensor selection budget constraint is satisfied), and for all $\nu \in \{0,1\}^n$ such that $f^T\nu \leq F$, $supp(\mu\setminus\nu) \neq \emptyset$ (i.e., for all sensor attacks that satisfy the sensor attack budget constraint, at least one sensor indicated by μ is left over by the adversary).*

Remark 4.2.4 *Note that if a sensor selection μ is not feasible, there is an attack (satisfying the attacker's budget constraint) such that that the pair $(A, C(\mu \setminus \nu))$ is not detectable if the system dynamics matrix A is not stable.*

The priori and posteriori Resilient Graph-based Kalman Filtering Sensor Selection (RGKFSS) problems are then given by the following.

Problem 4.2.5 *(Priori and Posteriori RGKFSS) Consider a system dynamics matrix $A \in \mathbb{R}^{n\times n}$ with the associated graph $\mathcal{G}(A) = \{\mathcal{X}(A), \mathcal{E}(A)\}$, a single vertex $x_{i_0} \in \mathcal{X}(A)$ that has a stochastic input with variance $\sigma_w^2 \in \mathbb{R}_{\geq 0}$, the measurement matrix $C = I_n$ (containing all of the individual sensor measurement matrices), a sensor noise covariance matrix $V \in \mathbb{S}_+^n$, a sensor selection cost vector $h \in \mathbb{Z}_{\geq 0}^n$, a sensor selection budget $H \in \mathbb{Z}_{\geq 0}$, a sensor attack cost vector $f \in \mathbb{Z}_{\geq 0}^n$, and a sensor attack budget $F \in \mathbb{Z}_{\geq 0}$. The priori Resilient Graph-based Kalman Filtering Sensor Selection (RGKFSS) problem is to find the sensor selection μ that solves*

$$\min_{\mu\in\{0,1\}^n} \max_{\nu\in\{0,1\}^n} \textit{trace}(\Sigma(\mu \setminus \nu))$$
$$\textit{s.t. } h^T\mu \leq H, \textit{ and } f^T\nu \leq F,$$

where $\Sigma(\mu\backslash\nu)$ is given by Eq. (4.4) if the pair $(A, C(\mu\backslash\nu))$ is detectable, and $\Sigma(\mu\backslash\nu) = +\infty$ otherwise. The posteriori RGFKSP problem is to find the sensor selection μ that solves

$$\min_{\mu\in\{0,1\}^n} \max_{\nu\in\{0,1\}^n} \textit{trace}(\Sigma^*(\mu \setminus \nu))$$
$$\textit{s.t. } h^T\mu \leq H, \textit{ and } f^T\nu \leq F,$$

where $\Sigma^*(\mu \setminus \nu)$ is given by Eq. (4.5) if the pair $(A, C(\mu \setminus \nu))$ is detectable, and $\Sigma^*(\mu \setminus \nu) = +\infty$ otherwise.

4.3 Solving the GKFSS and GKFSA problems

In this section, we provide algorithms to optimally solve the GKFSS and GKFSA problems, respectively, when the sensor noise covariance is $V = \mathbf{0}_{n\times n}$. We will make the following assumptions on the instances of the GKFSS and GKFSA problems in the sequel.

Assumption 4.3.1 *The pair $(A, C(\mu))$ is assumed to be detectable for all sensor selections $\mu \in \{0,1\}^n$ with $supp(\mu) \neq \emptyset$. The pair $(A, B\sigma_w)$ is assumed to be stabilizable.*

Assumption 4.3.2 *The graph* $\mathcal{G}(A) = \{\mathcal{X}(A), \mathcal{E}(A)\}$ *(associated with the system dynamics matrix* $A \in \mathbb{R}^{n\times n}$*) is assumed to satisfy the property that for all* $x_j \in \mathcal{X}(A)$ *and* $x_j \neq x_{i_0}$*, there exists a directed path from* x_{i_0} *to* x_j*. The system dynamics matrix* A *is assumed to satisfy* $(A^m)_{ji_0} \neq 0$ *if* $d_{i_0 j} = m$*, where* $d_{i_0 j}$ *is the distance from* x_{i_0} *to* x_j*.*

Remark 4.3.3 *Note that Assumptions* 4.3.1-4.3.2 *are satisfied by large classes of systems. For example, it was shown in [72] that Assumption* 4.3.1 *holds if the system dynamics matrix* A *is row-stochastic and irreducible.*[1] *Assumption* 4.3.2 *holds if the system dynamics matrix* A *is nonnegative and irreducible [65]. Since any row-stochastic matrix is also nonnegative, Assumptions* 4.3.1-4.3.2 *hold for any system dynamics matrix* A *that is row-stochastic and irreducible. Furthermore, using techniques in control theory pertaining to linear structured systems (e.g., [73, 74]), one can show that Assumption* 4.3.1 *holds for almost any system dynamics matrix* A *such that the graph* $\mathcal{G}(A)$ *is strongly connected, using approaches from [75, 76]. Specifically, one can consider the system dynamics matrix* A *to be structured, i.e., each entry of the system dynamics matrix* A *is either a fixed zero or an independent free parameter (which can attain any real value including zero), where the graph* $\mathcal{G}(A)$ *is defined according to the free parameters of the structured matrix* A*. One can then show that the set of parameters for which Assumption* 4.3.1 *does not hold has Lebesgue measure zero. Moreover, using similar techniques to those above and the result from [65] that shows that Assumption* 4.3.2 *holds for all nonnegative irreducible matrices* A*, one can show that Assumption* 4.3.2 *holds for almost any choice of free parameters in the structured matrix* A *such that the graph* $\mathcal{G}(A)$ *is strongly connected. Note that the systems where Assumptions* 4.3.1-4.3.2 *hold are not limited to the cases described above.*

Remark 4.3.4 *We can generalize our analysis to system dynamics matrices* A *where* $\mathcal{G}(A)$ *has multiple strongly connected components [71]. Suppose that the input node*

[1] Note that the matrix A is irreducible if and only if the graph $\mathcal{G}(A)$ is strongly connected [65].

can only reach (via directed paths in $\mathcal{G}(A)$) nodes that are in the same strongly connected component. Then, under Assumption 4.3.1*, we only need to consider the strongly connected component of $\mathcal{G}(A)$ that contains the input node, since one can show that the mean square estimation error of the Kalman filter remains zero for the states corresponding to nodes that are not in the strongly connected component containing the input node.*

The first main result of this section is as follows.

Theorem 4.3.5 *Consider a system dynamics matrix $A \in \mathbb{R}^{n\times n}$ with the associated graph $\mathcal{G}(A) = \{\mathcal{X}(A), \mathcal{E}(A)\}$, a single vertex $x_{i_0} \in \mathcal{X}(A)$ that has a stochastic input with variance $\sigma_w^2 \in \mathbb{R}_{\geq 0}$, the measurement matrix $C = I_n$ (containing all of the individual sensor measurement matrices), and the sensor noise covariance matrix $V = \mathbf{0}_{n\times n}$. Suppose that Assumptions* 4.3.1-4.3.2 *hold. For any sensor selection $\mu \in \{0,1\}^n$ such that supp$(\mu) \neq \emptyset$, denote $\zeta = \min_{j\in supp(\mu)} d_{i_0 j} \geq 0$, where $d_{i_0 j}$ is the distance from vertex x_{i_0} to vertex x_j. The following expressions hold:*

$$\Sigma(\mu) = \sigma_w^2 \sum_{m=0}^{\zeta} A^m B B^T (A^T)^m, \tag{4.7}$$

and

$$\Sigma^*(\mu) = \begin{cases} \sigma_w^2 \displaystyle\sum_{m=0}^{\zeta-1} A^m B B^T (A^T)^m & \text{if } \zeta \geq 1, \\ \mathbf{0} \quad \text{if } \zeta = 0, \end{cases} \tag{4.8}$$

where $\Sigma(\mu)$ (resp., $\Sigma^*(\mu)$) is the steady-state a priori (resp., a posteriori) error covariance of the corresponding Kalman filter, and $B = \mathbf{e}_{i_0}^T$.

Proof The existence of $\Sigma(\mu)$ and $\Sigma^*(\mu)$ follows directly from Lemma 4.2.1 and Assumption 4.3.1. Considering any sensor selection μ such that $\zeta \geq 1$, i.e., sensors are not selected at the input vertex x_{i_0}, we first prove Eq. (4.7) by verifying that Eq. (4.7) satisfies Eq. (4.4). Note that $C_i = \mathbf{e}_i$ for all $x_i \in \mathcal{X}(A)$. Let $\mathcal{X}_\mu \subseteq \mathcal{X}(A)$ denote the set of vertices indicated by μ where sensors are selected, and let $\mathcal{X}_\zeta \subseteq \mathcal{X}(A)$ denote the set of vertices that have distance ζ from the input vertex x_{i_0}. Since performing elementary

row operations on $C(\mu)$ does not change $\Sigma(\mu)$, we assume without loss of generality that $\mu = \begin{bmatrix}\mu_1^T & \mu_2^T\end{bmatrix}^T$ such that $\mu_1 = \mathbf{1}_{|\mathcal{X}_\zeta \cap \mathcal{X}_\mu|}$ and $\mu_2 \in \{0,1\}^{n-|\mathcal{X}_\zeta \cap \mathcal{X}_\mu|}$. In other words, μ_1 contains all sensors selected at vertices that have distance ζ from the input vertex x_{i_0}, and $d_{i_0 j} > \zeta$ for all $j \in \text{supp}(\mu_2)$. The corresponding measurement matrix is given by $C(\mu) = \begin{bmatrix}C(\mu_1)\\ C(\mu_2)\end{bmatrix}$, where $C(\mu_1) \in \mathbb{R}^{|\text{supp}(\mu_1)| \times n}$ and $C(\mu_2) \in \mathbb{R}^{|\text{supp}(\mu_2)| \times n}$. Substituting Eq. (4.7) into the right hand side (RHS) of Eq. (4.4), we obtain:

RHS of Eq. (4.4)

$$
\begin{aligned}
&= \sigma_w^2 \sum_{m=1}^{\zeta+1} A^m BB^T (A^T)^m + \sigma_w^2 BB^T - \sigma_w^2 A^{\zeta+1} BB^T (A^T)^\zeta \\
&\qquad \times (C(\mu))^T (C(\mu) A^\zeta BB^T (A^T)^\zeta (C(\mu))^T)^\dagger \times C(\mu) A^\zeta BB^T (A^T)^{\zeta+1} \qquad (4.9)\\
&= \sigma_w^2 \sum_{m=0}^{\zeta+1} A^m BB^T (A^T)^m - \sigma_w^2 A^{\zeta+1} BB^T (A^T)^\zeta \left[(C(\mu_1))^T \ (C(\mu_2))^T\right] \\
&\quad \times \Big(\begin{bmatrix}C(\mu_1)\\ C(\mu_2)\end{bmatrix} A^\zeta BB^T (A^T)^\zeta \left[(C(\mu_1))^T \ (C(\mu_2))^T\right] \Big)^\dagger \begin{bmatrix}C(\mu_1)\\ C(\mu_2)\end{bmatrix} A^\zeta BB^T (A^T)^{\zeta+1} \\
&= \sigma_w^2 \sum_{m=0}^{\zeta+1} A^m BB^T (A^T)^m - \sigma_w^2 A^{\zeta+1} B \left[B^T (A^T)^\zeta (C(\mu_1))^T \ \mathbf{0}_{1 \times |\text{supp}(\mu_2)|}\right] \\
&\qquad \times \begin{bmatrix}(C(\mu_1) A^\zeta BB^T (A^T)^\zeta (C(\mu_1))^T)^\dagger & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} \begin{bmatrix}C(\mu_1) A^\zeta B\\ \mathbf{0}_{|\text{supp}(\mu_2)| \times 1}\end{bmatrix} B^T (A^T)^{\zeta+1} \qquad (4.10)\\
&= \sigma_w^2 \sum_{m=0}^{\zeta+1} A^m BB^T (A^T)^m - \sigma_w^2 A^{\zeta+1} BB^T (A^T)^\zeta (C(\mu_1))^T \\
&\qquad \times (C(\mu_1) A^\zeta BB^T (A^T)^\zeta (C(\mu_1))^T)^\dagger C(\mu_1) A^\zeta BB^T (A^T)^{\zeta+1}, \qquad (4.11)
\end{aligned}
$$

where Eq. (4.9) uses the fact that $(A^m)_{j i_0} = 0$ for all $j \in \text{supp}(\mu)$ whenever $m \in \{0, 1, \ldots, \zeta - 1\}$, which implies that $C(\mu) A^m B = \mathbf{0}$ for all $m \in \{0, 1, \ldots, \zeta - 1\}$. Similarly, Eq. (4.10) follows from the fact that $C(\mu_2) A^m B = \mathbf{0}$ for all $m \in \{0, 1, \ldots, \zeta\}$.

Denoting $\psi \triangleq C(\mu_1)A^{\zeta}B \in \mathbb{R}^{|\text{supp}(\mu_1)|}$ and noting that $\psi \neq \mathbf{0}$ from Assumption 4.3.2, one can show that $\psi^T(\psi\psi^T)^{\dagger}\psi = 1$. We then have from Eq. (4.11):

$$\begin{aligned}\text{RHS of Eq. (4.4)} &= \sigma_w^2 \sum_{m=0}^{\zeta+1} A^m BB^T (A^T)^m - \sigma_w^2 A^{\zeta+1} BB^T (A^T)^{\zeta+1} \\ &= \sigma_w^2 \sum_{m=0}^{\zeta} A^m BB^T (A^T)^m.\end{aligned}$$

Since $\sigma_w^2 \sum_{m=0}^{\zeta} A^m BB^T (A^T)^m \succeq \mathbf{0}$, we know from Lemma 4.2.1 that the limit $\Sigma(\mu)$ is given by Eq. (4.7). We then see from Eq. (4.6) that the limit $\Sigma^*(\mu)$ is given by Eq. (4.8) (when $\zeta \geq 1$).

Next, we consider any sensor selection μ such that $\zeta = 0$, i.e., a sensor is selected at the input vertex x_{i_0}. Using similar arguments to those above, we can also show that Eqs. (4.7)-(4.8) hold when $\zeta = 0$. This completes the proof of the theorem. ■

To verify the results in Theorem 4.3.5, let us consider the following example.

Example 4.3.6 *Consider the graph in Fig.* 4.1, *where* x_2 *is the input node (i.e.,* $B = \mathbf{e}_2^T$*) with variance* $\sigma_w^2 = 1$. *Suppose* $A = \begin{bmatrix} 0.5 & 2.1 & 0 & 0 \\ 0.3 & 0 & 1.5 & 0 \\ 0 & 0.6 & 0 & 0.5 \\ 0 & 0 & -0.8 & 1 \end{bmatrix}$, $C = I_4$ *and* $V = \mathbf{0}_{4\times 4}$. *Denote* $\mu_2 = [0\ 1\ 0\ 0]^T$ *and* $\mu_4 = [0\ 0\ 0\ 1]^T$. *It can be verified that* $\Sigma(\mu_2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = BB^T$, $\Sigma^*(\mu_2) = \mathbf{0}_{4\times 4}$, $\Sigma(\mu_4) = \begin{bmatrix} 5.5125 & 1.6065 & 1.26 & -0.504 \\ 1.6065 & 3.3409 & 0 & -0.7344 \\ 1.26 & 0 & 0.36 & 0 \\ -0.504 & -0.7344 & 0 & 0.2304 \end{bmatrix} = \sum_{m=0}^{2} A^m BB^T (A^T)^m$ *and* $\Sigma^*(\mu_4) = \begin{bmatrix} 4.41 & 0 & 1.26 & 0 \\ 0 & 1 & 0 & 0 \\ 1.26 & 0 & 0.36 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \sum_{m=0}^{1} A^m BB^T (A^T)^m$, *as provided by Theorem* 4.3.5.

Fig. 4.1. Graph for Example 1.

4.3.1 An Optimal Solution to GKFSS

Using the above discussions, we give the following result that characterizes an optimal solution to GKFSS (Problem 4.2.2).

Theorem 4.3.7 *Supposing that Assumptions* 4.3.1-4.3.2 *hold, an optimal solution, denoted as* μ^**, to the priori (resp., posteriori) GKFSS problem is to select a single sensor at a vertex* x_j *in order to minimize* $d_{i_0 j}$*, i.e., the distance from the input vertex* x_{i_0} *to* x_j*, while satisfying the budget constraint.*

Proof Under Assumptions 4.3.1-4.3.2, we first note from Eqs. (4.7)-(4.8) that the *a priori* and the *a posteriori* error covariance matrices only depend on ζ, i.e., the shortest distance from the input node to the sensor nodes. Hence, it is sufficient to consider sensor selections $\mu \in \{0,1\}^n$ such that $|\text{supp}(\mu)| = 1$ in terms of minimizing the trace of the *a priori* (resp., *a posteriori*) steady-state error covariance of the Kalman filter. Moreover, we know from Eq. (4.7) in Theorem 4.3.5 that $\Sigma(\mu) = \sigma_w^2 \sum_{m=0}^{\zeta} A^m B B^T (A^T)^m$, where $\zeta = \min_{j \in \text{supp}(\mu)} d_{i_0 j}$. Since the matrix $A^m B B^T (A^T)^m$ is positive semi-definite for all $m \in \mathbb{Z}_{\geq 0}$, we have $\text{trace}(A^m B B^T (A^T)^m) \geq 0$ for all $m \in \mathbb{Z}_{\geq 0}$. Hence, $\text{trace}(\Sigma(\mu))$ is minimized by finding a sensor selection μ^* with $|\text{supp}(\mu^*)| = 1$ such that ζ is minimized while satisfying the budget constraint. Using similar arguments, we can show that μ^* is also an optimal solution to the posteriori GKFSS problem. ■

Based on Theorem 4.3.7, we can find an optimal solution μ^* to the priori (resp., posteriori) GKFSS problem using polynomial-time algorithms such as the Breadth-First Search (BFS) algorithm which runs in time $O(n + |\mathcal{E}(A)|)$ [77].

4.3.2 An Optimal Solution to GKFSA

Given a sensor selection μ, we know from the insights obtained above for GKFSS that the steady-state *a priori* and the *a posteriori* error covariances of the Kalman filter (after an attack that removes some of those sensors) only depend on the surviving sensors that have the shortest distance from the input vertex x_{i_0}. We then have the following result whose proof is similar to that of Theorem 4.3.7 and is thus omitted.

Theorem 4.3.8 *Suppose that Assumptions* 4.3.1-4.3.2 *hold. Given a sensor selection* μ*, an optimal solution, denoted as* ν^**, to the priori (resp., posteriori) GKFSA problem*

can be found by maximizing the shortest distance from the input vertex x_{i_0} to the surviving sensors, i.e., solving the following optimization problem

$$\begin{aligned} &\max_{\nu\in\{0,1\}^n} \min_{j\in supp(\mu\setminus\nu)} d_{i_0 j} \\ &s.t.\ f^T\nu \leq F, \end{aligned} \tag{4.12}$$

where $d_{i_0 j}$ is the distance from vertex x_{i_0} to vertex x_j, and $d_{i_0 j} = +\infty$ if $supp(\mu\setminus\nu) = \emptyset$.

An optimal solution ν^* to the priori (resp., posteriori) GKFSA problem described by Theorem 4.3.8 can be found as follows. Given a sensor selection μ, the adversary starts by inspecting the selected sensors (indicated by μ) that have the shortest distance from the input vertex x_{i_0}. The adversary will remove all of these sensors if the sum of the corresponding sensor attack costs is less than or equal to the budget constraint F, and terminate the process if otherwise. The above process is then repeated for the selected sensors that have the second shortest distance from the input vertex x_{i_0}, based on the remaining budget. This process continues with the selected sensors that have the third shortest distance from the input vertex x_{i_0}, etc. Hence, polynomial-time algorithms such as the BFS algorithm can be used to find the optimal sensor attack ν^* for the adversary in time $O(n + |\mathcal{E}(A)|)$.

4.4 Solving the RGKFSS problem

We now turn to the RGKFSS problem (Problem 4.2.5). Recall that Theorem 4.3.7 showed that it is enough to consider only sensor selections μ with $|\text{supp}(\mu)| = 1$ for the GKFSS problem (i.e., the system designer does not necessarily need to utilize all of the sensor selection budget H). However, an optimal sensor selection μ^* for the RGKFSS problem does not necessarily satisfy $|\text{supp}(\mu^*)| = 1$, since the adversary could have enough budget to remove the single sensor selected by the system designer, which causes the trace of the *a priori* (resp., *a posteriori*) error covariance of the Kalman filter to be infinite (if the system dynamics matrix A is not stable). Note that the

steady-state *a priori* and the *a posteriori* error covariance matrices of the Kalman filter (after the attack) only depend on the surviving sensors that have the shortest distance from the input vertex x_{i_0}. Using similar arguments to those for Theorems 4.3.7-4.3.8, we see that an optimal solution to the RGKFSS problem can be found by minimizing the shortest distance from the input vertex x_{i_0} to the sensors after the corresponding optimal sensor attack, and a sensor selection μ^* is optimal for the priori RGKFSS problem if and only if it is optimal for the posteriori RGKFSS problem.

We thus focus on the priori RGKFSS problem in this section. Although we provided polynomial-time algorithms to solve the GKFSS and GKFSA problems, we will show that the RGKFSS problem is NP-hard, i.e., there exist classes of the RGKFSS problem that cannot be solved by any polynomial-time algorithm if P $\neq$ NP. To do this, we first recall from Remark 4.3.3 that Assumptions 4.3.1-4.3.2 hold for any system dynamics matrix A that is row-stochastic and irreducible. Therefore, Eq. (4.7) and Eq. (4.8) in Theorem 4.3.5 also hold for such A matrices.

To show that the RGKFSS problem is NP-hard, we reduce the subset sum problem [41] to RGKFSS.

Definition 4.4.1 *An instance of the subset sum problem is given by a finite set U and a positive integer K, where each $s \in U$ has a size $\kappa(s) \in \mathbb{Z}_{>0}$.*

We use the following result from [41].

Lemma 4.4.1 *Given an instance of the subset sum problem as described in Definition* 4.4.1*, the problem of determining whether there is a subset $U' \subseteq U$ such that $\sum_{s \in U'} \kappa(s) = K$ is NP-complete.*

We are now in place to prove the following result.

Theorem 4.4.2 *The RGKFSS problem is NP-hard even when both of the following two conditions are satisfied:* (1) *the sensor selection cost and the sensor attack cost satisfy $h_i = f_i$ for all $i \in \{1, 2, \ldots, n\}$; and* (2) *there is a feasible sensor selection for the system designer.*

Proof We prove the result by giving a polynomial-time reduction from the subset sum problem. Consider any instance of the subset sum problem defined in Definition 4.4.1. Denote $U = \{s_1, s_2, \ldots, s_{|U|}\}$. Denote the number of bits of the binary representation of the positive integer K as $b(K)$, i.e., $b(K) \triangleq \lfloor \log_2(K) \rfloor + 1$. We then construct an instance of the priori RGKFSS problem as follows. The system dynamics matrix $A \in \mathbb{R}^{(|U|+b(K))\times(|U|+b(K))}$ is chosen such that the graph $\mathcal{G}(A)$ is an undirected path of length $|U|+b(K)-1$. Specifically, we set $A_{ij} = A_{ji} = \frac{1}{3}$ for all $i \in \{1, 2, \ldots, |U|+b(K)-1\}$ and $j = i+1$, $A_{mm} = \frac{1}{3}$ for all $m \in \{2, 3, \ldots, |U|+b(K)-1\}$, $A_{mm} = \frac{2}{3}$ for all $m \in \{1, |U| + b(K)\}$, and all the other entries in A are zero. The vertex x_1 is set as the only vertex that has the stochastic input with variance $\sigma_w^2 = 1$. The sensor selection cost vector is set as $h_i = \kappa(s_i)$ for all $i \in \{1, 2, \ldots, |U|\}$, and $h_i = 2^{i-|U|-1}$ for all $i \in \{|U| + 1, |U| + 2, \ldots, |U| + b(K)\}$. The sensor attack cost is set as $f_i = h_i$ for all $i \in \{1, 2, \ldots, |U| + b(K)\}$. Note that the sensor selection vector and the sensor attack vector are given by $\mu \in \{0, 1\}^{|U|+b(K)}$ and $\nu \in \{0, 1\}^{|U|+b(K)}$, respectively. The sensor selection budget of the system designer is set as $H = K$, and the sensor attack budget of the adversary is set as $F = K - 1$. We also note that the matrix A that we constructed is row-stochastic and irreducible. Therefore, Eq. (4.7) in Theorem 4.3.5 holds for the A matrix that we constructed. We claim that the answer to the given subset sum instance is "yes" if and only if an optimal solution to the constructed instance of the priori RGKFSS problem, denoted as μ^*, satisfies $\text{trace}(\Sigma(\mu^* \setminus \nu^*)) \leq \text{trace}(\sum_{i=0}^{|U|-1} A^i B B^T A^i)$, where ν^* is an optimal sensor attack given μ^*.

Suppose that the answer to the given subset sum instance is "yes", i.e., there exists $U' \subseteq U$ such that $\sum_{s\in U'} \kappa(s) = K$. It follows that for the instance of the priori RGKFSS problem as constructed above, there exists a sensor selection vector $\tilde{\mu}$ such that $\sum_{i=1}^{|U|} h_i \tilde{\mu}_i = K \leq H$. Therefore, for any sensor attack $\tilde{\nu}$ that satisfies the sensor attack budget constraint, i.e., $\sum_{i=1}^{|U|} f_i \tilde{\nu}_i \leq F = K-1$, we have $\text{supp}(\tilde{\mu} \setminus \tilde{\nu}) \cap \{1, \ldots, |U|\} \neq \emptyset$, which implies that there exists $j \in \{1, \ldots, |U|\}$ such that $j \in \text{supp}(\tilde{\mu} \setminus \tilde{\nu})$. Noting that $A^m B B^T A^m \succeq \mathbf{0}$ for all $m \in \mathbb{Z}_{\geq 0}$ and $d_{1j} = j-1 \leq d_{1|U|} = |U|-1$, it then follows from

Eq. (4.7) that $\text{trace}(\Sigma(\tilde{\mu}\backslash\tilde{\nu})) \leq \text{trace}(\sum_{i=0}^{j-1} A^i BB^T A^i) \leq \text{trace}(\sum_{i=0}^{|U|-1} A^i BB^T A^i)$, for any sensor attack $\tilde{\nu}$ such that $\sum_{i=1}^{|U|} f_i\tilde{\nu}_i \leq F$. Since $\text{trace}(\Sigma(\mu^*\backslash\nu^*)) \leq \text{trace}(\Sigma(\tilde{\mu}\backslash\tilde{\nu}))$, we have $\text{trace}(\Sigma(\mu^* \setminus \nu^*)) \leq \text{trace}(\sum_{i=0}^{|U|-1} A^i BB^T A^i)$.

Conversely, suppose that the answer to the subset sum instance is "no", i.e., for any $U' \subseteq U$, we have $\sum_{s\in U'} \kappa(s) \neq K$. Considering the instance of the priori RGKFSS problem we constructed, for any sensor selection vector μ such that $\sum_{i=1}^{|U|+b(K)} h_i\mu_i \leq H = K$, we have $\sum_{i=1}^{|U|} h_i\mu_i \neq K$, which implies $\sum_{i=1}^{|U|} h_i\mu_i \leq K-1$. Denote $\sum_{i=1}^{|U|} h_i\mu_i \triangleq K_{|U|}$. Therefore, for any sensor selection vector μ with $\sum_{i=1}^{|U|+b(K)} h_i\mu_i \leq H$, there exists an attack $\hat{\nu}$ such that $\sum_{i=1}^{|U|} f_i\hat{\nu}_i = K_{|U|} \leq K-1$, which implies $\text{supp}(\mu \setminus \hat{\nu}) \cap \{1, 2, \ldots, |U|\} = \emptyset$. Moreover, note that $K = H \geq H - K_{|U|} > F - K_{|U|}$. Since we set the sensor selection cost vector and the sensor attack cost vector to satisfy $h_i = f_i = 2^{i-|U|-1}$ for all $i \in \{|U|+1, |U|+2, \ldots, |U|+b(K)\}$, where $b(K)$ is the number of bits for the binary representation of K, we see that for any $U' \subseteq U$, there exists $\bar{U}' \subseteq \{|U|+1, |U|+2, \ldots, |U|+b(K)\}$ such that $\sum_{s\in U'} \kappa(s) + \sum_{i\in\bar{U}'} h_i = H$. Therefore, the system designer can always use all the sensor selection budget by selecting sensors at an appropriate subset of the vertices in the vertex set $\{x_{|U|+1}, x_{|U|+2}, \ldots, x_{|U|+b(K)}\}$ and guarantee to have at least one sensor left after any attack that satisfies the sensor attack budget constraint. Formally, we see that for any sensor selection μ with $\sum_{i=1}^{|U|+b(K)} h_i\mu_i = H$, there exists $j' \in \{|U|+1, \ldots, |U|+b(K)\}$ such that $j' \in \text{supp}(\mu \setminus \nu)$, where ν is any sensor attack satisfying the sensor attack budget constraint. Meanwhile, any sensor selection μ such that $\sum_{i=1}^{|U|+b(K)} h_i\mu_i < H$ is not a feasible sensor selection. Therefore, there is always a feasible sensor selection for the system designer under the constructed instance of the priori RGKFSS problem when the answer to the subset sum instance is "no". Note that the matrix $A^m BB^T A^m \succeq \mathbf{0}$ for all $m \in \mathbb{Z}_{\geq 0}$ and $d_{1j'} = j' - 1 \geq d_{1|U|} + 1 = |U|$. Combining the arguments above together, it then follows from Eq. (4.7) that for any μ such that $\sum_{i=1}^{|U|+b(K)} h_i\mu_i = H$, we have $\text{trace}(\Sigma(\mu \setminus \nu)) \geq \text{trace}(\sum_{i=0}^{j'-1} A^i BB^T A^i) \geq \text{trace}(\sum_{i=0}^{|U|} A^i BB^T A^i)$, where ν is any sensor attack satisfying the sensor attack budget constraint. Since $(A^m)_{11} > 0$ for all $m \in \mathbb{Z}_{\geq 0}$, we have $\text{trace}(A^{|U|} BB^T A^{|U|}) > 0$

and thus trace$(\Sigma(\mu \setminus \nu)) >$ trace$(\sum_{i=0}^{|U|-1} A^i B B^T A^i)$. Since the above arguments hold for any μ with $\sum_{i=1}^{|U|+b(K)} h_i \mu_i = H$, they also hold for an optimal solution μ^* to the constructed priori RGKFSS instance, i.e., $(\Sigma(\mu^* \setminus \nu^*)) >$ trace$(\sum_{i=0}^{|U|-1} A^i B B^T A^i)$, where ν^* is an optimal sensor attack given μ^*. This completes the proof of the claim above.

Since the subset sum problem is NP-complete and RGKFSS $\notin$ NP, we conclude that RGKFSS is NP-hard even under the additional conditions as stated. ■

4.4.1 An Algorithm for RGKFSS

Algorithm 4.4.1 Algorithm for RGKFSS

Input: An instance of the RGKFSS problem.

Output: A sensor selection $\mu \in \{0,1\}^n$.

1: Find the distance $d_{i_0 j}$ for all $x_j \in \mathcal{X}(A) \setminus \{x_{i_0}\}$ via BFS and denote $d_{\max} \triangleq \max_{x_j \in \mathcal{X}(A)} d_{i_0 j}$.

2: Relabel the vertices of $\mathcal{G}(A)$ such that x_1 is the input vertex and $d_{1j} \leq d_{1t}$ for all $x_j, x_t \in \mathcal{X}(A) \setminus \{x_1\}$ with $j \leq t$.

3: $\mu = \mathbf{0}_{n \times 1}$

4: **for** $m = 0$ to $d_{\max}$ **do**

5: Find $j_m \triangleq \max\{j : d_{1j} = m, x_j \in \mathcal{X}(A)\}$.

6: Find $\pi^*\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big)$

7: **if** $z\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big) > F$ **then**

8: $[\mu_1 \ \cdots \ \mu_{j_m}]^T = \pi^*\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big)$

9: **return** μ

It follows directly from Theorem 4.4.2 that there is no polynomial-time algorithm that would solve all instances of RGKFSS if P $\neq$ NP. However, we now provide a

pseudo-polynomial-time algorithm[2] (Algorithm 4.4.1) for RGKFSS by relating it to the knapsack problem defined as follows.

Definition 4.4.2 *Given a finite set* $U \triangleq \{s_1, s_2, \ldots, s_{|U|}\}$*, a size* $\kappa(s_i) \in \mathbb{Z}_{>0}$ *and a value* $\phi(s_i) \in \mathbb{Z}_{>0}$ *for each* $i \in \{1, 2, \ldots, |U|\}$*, and a positive integer* K*, the knapsack problem is to find an indicator vector* $\pi \in \{0,1\}^{|U|}$ *that solves*

$$\begin{aligned} &\max_{\pi \in \{0,1\}^{|U|}} \sum_{i=1}^{|U|} \phi(s_i)\pi_i \\ &s.t. \sum_{i=1}^{|U|} \kappa(s_i)\pi_i \leq K. \end{aligned} \tag{4.13}$$

Denote an instance of knapsack as a tuple $\{\phi, \kappa, K\}$ *with* $\phi \triangleq (\phi(s_1), \phi(s_2), \ldots, \phi(s_{|U|}))$ *and* $\kappa \triangleq (\kappa(s_1), \kappa(s_2), \ldots, \kappa(s_{|U|}))$.[3] *The corresponding optimal indicator vector for* (4.13) *is denoted as* $\pi^*(\phi, \kappa, K)$*, and the corresponding optimal value of the objective function in* (4.13) *is denoted as* $z(\phi, \kappa, K)$.

The steps of Algorithm 4.4.1 for RGKFSS are as follows. Algorithm 4.4.1 starts by relabeling the input vertex as vertex x_1 and relabeling the other vertices in terms of a non-decreasing order of the distances from the vertex x_1 (Lines 1-2). Denoting $d_{\max} \triangleq \max_{x_j \in \mathcal{X}(A)} d_{i_0 j}$, Algorithm 4.4.1 then finds the smallest $m \in \{0, 1, \ldots, d_{\max}\}$ such that by selecting sensors (under the budget constraint) solely at nodes that have distances less than or equal to m from x_1 (after the relabeling), the sum of the sensor attack costs of the selected sensors is greater than the sensor attack budget, i.e., there is at least one sensor that survives the corresponding optimal sensor attack. This is done by iteratively solving a knapsack problem at increasingly longer distances from the input node, where at each distance, the goal is to find a set of sensor locations that fits within the sensor selection budget constraint H but maximizes the sum of

[2] A pseudo-polynomial-time algorithm is an algorithm that runs in time that is bounded by a polynomial in the largest integer in its input [41].

[3] Note that the elements in ϕ and κ are ordered, and the ith element of ϕ (resp., κ) corresponds to the value (resp., weight) of $s_i \in U$ for all $i \in \{1, \ldots, |U|\}$. The dependency of $\{\phi, \kappa, K\}$ on U is dropped since each element of ϕ (resp., κ) represents an element of U.

the sensor attack costs. Algorithm 4.4.1 returns $\mu = \mathbf{0}_{n\times 1}$ if there is no feasible sensor selection. We now prove that Algorithm 4.4.1 returns an optimal solution to RGKFSS.

Theorem 4.4.3 *Under Assumptions* 4.3.1-4.3.2*, Algorithm* 4.4.1 *returns an optimal solution to the RGKFSS problem.*

Proof Denote an optimal solution to the RGKFSS problem as μ^* and denote the solution returned by Algorithm 4.4.1 as μ'. Suppose that μ' is a feasible sensor selection. Suppose that the vertices in $\mathcal{G}(A)$ are relabeled as indicated by Lines 1-2 in Algorithm 4.4.1, i.e., vertex x_1 is labeled as the input vertex and the other vertices are labeled in terms of a non-decreasing order of the distances from vertex x_1 (note that the relabeling of the vertices does not change any optimal solution to the RGKFSS problem other than permuting it). Assume for the sake of contradiction that $\text{trace}(\Sigma(\mu^* \setminus \nu^*)) < \text{trace}(\Sigma(\mu' \setminus \nu'))$, where ν^* and ν' are optimal sensor attacks given μ^* and μ', respectively. Denote $j^* \triangleq \max J$ and $j' \triangleq \max J'$, where $J \triangleq \arg\min_{m\in\text{supp}(\mu^*\setminus\nu^*)} d_{1m}$ and $J' \triangleq \arg\min_{m\in\text{supp}(\mu'\setminus\nu')} d_{1m}$. In other words, among those sensors that are closest to the input vertex in $\text{supp}(\mu^*\setminus\nu^*)$ (resp., $\text{supp}(\mu'\setminus\nu')$), j^* (resp., j') is the largest index. Noting that $\sum_{m=1}^{j^*} f_m\mu_m^* > F$ (otherwise an optimal sensor attack ν^* given μ^* would remove the sensor selected at vertex x_{j^*} as argued previously in Section 4.3.2), it follows from Definition 4.4.2 that $z\big((f_1,\ldots,f_{j^*}),(h_1,\ldots,h_{j^*}),H\big) > F$, which implies that $z\big((f_1,\ldots,f_{j_m}),(h_1,\ldots,h_{j_m}),H\big) > F$, where j_m is defined in Line 5 of Algorithm 4.4.1 with $m = d_{1j^*}$. We then know from the definition of Algorithm 4.4.1 that the sensor selection μ' returned by Algorithm 4.4.1 would satisfy $j' \le j_m$, which implies that $d_{1j'} \le d_{1j^*}$ (by the way that Algorithm 4.4.1 relabels the vertices). Moreover, we have from Theorem 4.3.5 the following:

$$\Sigma(\mu^* \setminus \nu^*) = \sigma_w^2 \sum_{m=0}^{d_{1j^*}} A^m BB^T (A^T)^m, \tag{4.14}$$

and

$$\Sigma(\mu' \setminus \nu') = \sigma_w^2 \sum_{m=0}^{d_{1j'}} A^m BB^T (A^T)^m, \tag{4.15}$$

hold under Assumptions 4.3.1-4.3.2. Since the matrix $A^m BB^T (A^T)^m \succeq \mathbf{0}$ for all $m \in \mathbb{Z}_{\geq 0}$, it follows from the assumption $\text{trace}(\Sigma(\mu^* \setminus \nu^*)) < \text{trace}(\Sigma(\mu' \setminus \nu'))$ and Eqs. (4.14)-(4.15) that $d_{1j^*} < d_{1j'}$. Thus, we get a contradiction.

Then, suppose that the solution μ' returned by Algorithm 4.4.1 is not feasible, i.e., $\text{supp}(\mu' \setminus \nu') = \emptyset$. Again, we assume that $\text{trace}(\Sigma(\mu^* \setminus \nu^*)) < \text{trace}(\Sigma(\mu' \setminus \nu'))$, i.e., $\text{supp}(\mu^* \setminus \nu^*) \neq \emptyset$. Via similar arguments to those above, we see that there exists $j^{*\prime} \in \{1, \ldots, n\}$ such that $z\big((f_1, \ldots, f_{j^{*\prime}}), (h_1, \ldots, h_{j^{*\prime}}), H\big) > F$, which implies that $z\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big) > F$, where j_m is defined in Line 5 of Algorithm 4.4.1 with $m = d_{1j^{*\prime}}$. Therefore, Algorithm 4.4.1 would also return a solution μ' such that $\text{supp}(\mu' \setminus \nu') \neq \emptyset$, which is a contradiction. We then conclude that $\text{trace}(\Sigma(\mu^* \setminus \nu^*)) = \text{trace}(\Sigma(\mu' \setminus \nu'))$, i.e., Algorithm 4.4.1 returns an optimal solution to the RGKFSS problem. ■

Since the knapsack problem is NP-hard, there is no polynomial-time algorithm to solve it optimally (if P $\neq$ NP) [41]. Various algorithms exist to approximate or optimally solve it, including greedy algorithms, linear programming relaxation and dynamic programming [78]. When implementing Algorithm 4.4.1, we can use existing algorithms for knapsack to find $\pi^*\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big)$ in Line 6 and $z\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big)$ in Line 7 when we range m from 0 to $d_{\max}$. Specifically, we call a pseudo-polynomial-time algorithm for knapsack (that solves it optimally) at most $d_{\max} + 1$ times to achieve this. For example, a typical dynamic programming approach for knapsack finds $\pi^*\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big)$ and $z\big((f_1, \ldots, f_{j_m}), (h_1, \ldots, h_{j_m}), H\big)$ in time $O(j_m H)$ for each $m \in \{0, \ldots, d_{\max}\}$ [78]. Since BFS runs in time $O(n + |\mathcal{E}(A)|)$, Algorithm 4.4.1 runs in time $O(d_{\max} nH + n + |\mathcal{E}(A)|)$.

4.5 Noisy Sensor Measurement Case

The results we obtained so far hold under the assumption that $V = \mathbf{0}_{n \times n}$. In this section, we provide a bound on the suboptimality of the proposed strategies when

there is sensor measurement noise. We will use the following result whose proof can be found in Section 4.7.1.

Lemma 4.5.1 *Consider a system dynamics matrix $A \in \mathbb{R}^{n\times n}$, an input matrix $B \in \mathbb{R}^{n\times n_1}$, a sensor measurement matrix $C \in \mathbb{R}^{n_2\times n}$, an input covariance matrix $W \in \mathbb{S}_+^{n_1}$, and a sensor measurement noise covariance matrix $\tilde{V} \in \mathbb{S}_+^{n_2}$. Suppose that the pair $(A, BW^{1/2})$ is stabilizable and the pair (A, C) is detectable. Let $\tilde{\Sigma}$ (resp., $\tilde{\Sigma}^*$) denote the steady-state a priori (resp., a posteriori) error covariance of the Kalman filter corresponding to the measurement noise covariance $\tilde{V}$, and let Σ (resp., Σ^*) denote the steady-state a priori (resp., a posteriori) error covariance of the corresponding Kalman filter when $V = \mathbf{0}_{n_2\times n_2}$. Then, $\tilde{\Sigma} \preceq \Sigma + E$ and $\tilde{\Sigma}^* \preceq \Sigma^* + (I_n - LC)E$, where E is given by*

$$E \triangleq \sum_{m=0}^{\infty} (A - KC)^m K\tilde{V}K^T((A - KC)^T)^m, \tag{4.16}$$

with $K \triangleq A\Sigma C^T(C\Sigma C^T)^{-1}$ and $L \triangleq \Sigma C^T(C\Sigma C^T)^{-1}$.[4]

Note that E exists and is finite since the matrix $A-KC$ is stable. See the proof in Section 4.7.1 for more details. We have the following result for the GKFSS problem.

Theorem 4.5.2 *Suppose that Assumptions* 4.3.1-4.3.2 *hold. Let $\tilde{\Sigma}(\mu)$ (resp., $\tilde{\Sigma}^*(\mu)$) be the steady-state a priori (resp., a posteriori) error covariance matrix of the Kalman filter associated with μ when $V = \tilde{V} \in \mathbb{S}_+^n$. Let $\tilde{\mu}_1^*$ (resp., $\tilde{\mu}_2^*$) denote an optimal solution to the priori (resp., posteriori) GKFSS problem when $V = \tilde{V}$, and let μ^* denote an optimal solution to the priori (resp., posteriori) GKFSS problem when $V = \mathbf{0}_{n\times n}$. Then, trace$(\tilde{\Sigma}(\mu^*)) \leq$ trace$(\tilde{\Sigma}(\tilde{\mu}_1^*))$ + trace$(E(\mu^*))$ and trace$(\tilde{\Sigma}^*(\mu^*)) \leq$ trace$(\tilde{\Sigma}^*(\tilde{\mu}_2^*))$ + trace$((E^*(\mu^*))$, where $E(\mu^*)$ and $L(\mu^*)$ are defined in Lemma* 4.5.1 *with $C = C(\mu^*)$, and $E^*(\mu^*) \triangleq (I_n - L(\mu^*)C(\mu^*))E(\mu^*)$.*

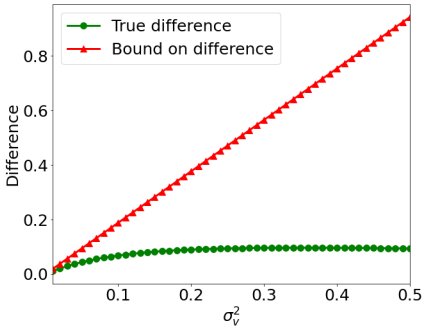
Proof First, we know from Lemma 4.5.1 that $\tilde{\Sigma}(\mu^*) \preceq \Sigma(\mu^*) + E(\mu^*)$, where $\Sigma(\mu^*)$ is the steady-state *a priori* error covariance of the Kalman filter corresponding to μ^*

[4]The inverses are interpreted as the Moore-Penrose pseudo-inverses if the arguments are not invertible [61].

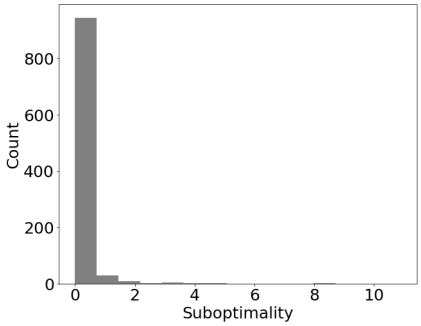
when $V = \mathbf{0}$. This implies $\text{trace}(\tilde{\Sigma}(\mu^*)) \leq \text{trace}(\Sigma(\mu^*)) + \text{trace}(E(\mu^*))$. Since μ^* is an optimal solution to the priori GKFSS problem when $V = \mathbf{0}$, we have $\text{trace}(\Sigma(\mu^*)) \leq \text{trace}(\Sigma(\tilde{\mu}_1^*))$. Moreover, one can show that the error covariance of the Kalman filter is always lower bounded (in the positive semi-definite sense) by the error covariance of the Kalman filter with zero measurement noise covariance (with the other system matrices fixed). We obtain $\text{trace}(\Sigma(\tilde{\mu}_1^*)) \leq \text{trace}(\tilde{\Sigma}(\tilde{\mu}_1^*))$. It then follows from the above arguments that $\text{trace}(\tilde{\Sigma}(\mu^*)) \leq \text{trace}(\tilde{\Sigma}(\tilde{\mu}_1^*)) + \text{trace}(E(\mu^*))$. Similarly, we can show that $\text{trace}(\tilde{\Sigma}^*(\mu^*)) \leq \text{trace}(\tilde{\Sigma}^*(\tilde{\mu}_2^*)) + \text{trace}(E^*(\mu^*))$. ■

The above result has the following interpretation. Consider an instance of the priori (resp., posteriori) GKFSS problem with $V = \tilde{V} \in \mathbb{S}_+^n$. If we simply take $V = \mathbf{0}$ and apply the algorithm described in Section 4.3.1, we will obtain an optimal solution, denoted as μ^*, to the corresponding instance of the priori (resp., posteriori) GKFSS problem (with $V = \mathbf{0}$). Theorem 4.5.2 shows that the performance (i.e., suboptimality) of this sensor selection (i.e., μ^*) for the original priori (resp., posteriori) GKFSS instance with $V = \tilde{V}$ can be bounded by $\text{trace}(\tilde{\Sigma}(\mu^*)) \leq \text{trace}(\tilde{\Sigma}(\tilde{\mu}_1^*)) + \text{trace}(E(\mu^*))$ (resp., $\text{trace}(\tilde{\Sigma}^*(\mu^*)) \leq \text{trace}(\tilde{\Sigma}^*(\tilde{\mu}_2^*)) + \text{trace}(E^*(\mu^*))$), where $\tilde{\mu}_1^*$ (resp., $\tilde{\mu}_2^*$) is an optimal solution to the instance of the priori (resp., posteriori) GKFSS problem when $V = \tilde{V}$. Moreover, we see from Eq. (4.16) that as $\tilde{V}$ goes to zero, $\text{trace}(E(\mu^*))$ (resp., $\text{trace}(E^*(\mu^*))$) will go to zero, which implies that $\text{trace}(\tilde{\Sigma}(\mu^*))$ (resp., $\text{trace}(\tilde{\Sigma}^*(\mu^*))$) will go to $\text{trace}(\tilde{\Sigma}(\tilde{\mu}_1^*))$ (resp., $\text{trace}(\tilde{\Sigma}^*(\tilde{\mu}_2^*))$). Similar performance bounds can be obtained for the GKFSA and RGKFSS problems, respectively.

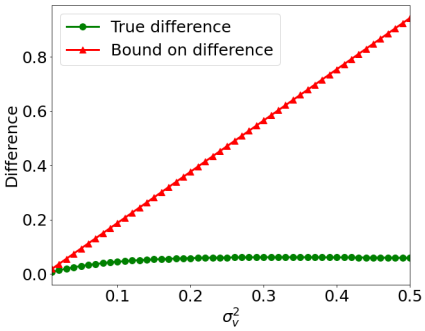
We provide simulations to show the performance of the algorithms in Section 4.3.1, Section 4.3.2, and Section 4.4, when applied to solve the GKFSS, GKFSA, and RGKFSS problems with measurement noise, respectively. Specifically, consider a strongly connected graph $\mathcal{G}(A)$ with $\mathcal{X}(A) = \{x_1, \ldots, x_{10}\}$ and $|\mathcal{E}(A)| = 15$, where node x_1 has the stochastic input with variance $\sigma_w^2 = 0.1$. Set the measurement matrix $C = I_{10}$ and the sensor noise covariance $V = \sigma_v^2 I_{10}$, where $\sigma_v^2 \in \mathbb{R}_{\geq 0}$. Under a fixed cost $h_i \in \mathbb{Z}_{\geq 0}$ to select sensor at x_i, a budget $H \in \mathbb{Z}_{\geq 0}$, a fixed cost $f_i \in \mathbb{Z}_{\geq 0}$ to attack sensor at x_i, and an attack budget $F \in \mathbb{Z}_{\geq 0}$, we randomly generate the corresponding



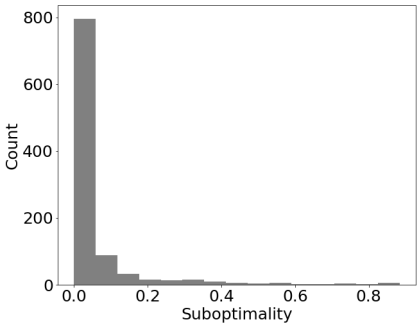
(a) GKFSS: True difference vs. Bound



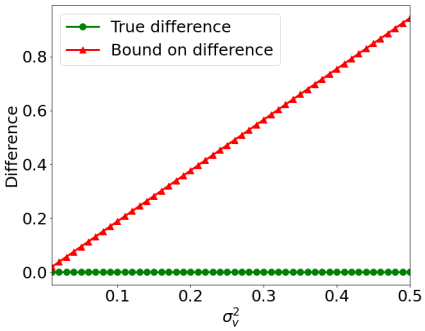
(b) GKFSS: Suboptimality



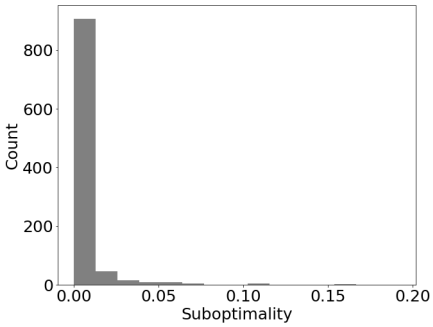
(c) GKFSA: True difference vs. Bound



(d) GKFSA: Suboptimality



(e) RGKFSS: True difference vs. Bound



(f) RGKFSS: Suboptimality

Fig. 4.2. Performance of the algorithms.

system dynamics matrix $A \in \mathbb{R}^{10\times 10}$ by selecting each nonzero element of A from a standard normal distribution. Fig. 4.2(a) and Fig. 4.2(b) show the performance of the algorithm described in Section 4.3.1, when applied to solve the (priori) GKFSS instances with $V = \sigma_v^2 I_{10}$. Specifically, Fig. 4.2(a) is obtained for a single realization of A, which compares the gap (i.e., difference) between the optimal solution to the GKFSS problem (found by brute force and denoted as OPT) and the solution returned by the algorithm (denoted as ALG), with the bound (on the difference) provided in Theorem 4.5.2, when σ_v^2 ranges from 0.01 to 0.5. Fig. 4.2(b) shows a histogram of the suboptimality of the algorithm, computed as $\frac{ALG-OPT}{OPT}$, over 1000 realizations of A, when $\sigma_v^2 = 5$. Similarly, Fig. 4.2(c)-(d) and Fig. 4.2(e)-(f) show the performance of the algorithm described in Section 4.3.2 for GKFSA and Algorithm 4.4.1 for RGKFSS, respectively. Note that we fix a sensor selection μ when solving the GKFSA instances. Moreover, the objective function of RGKFSS associated with the solution returned by Algorithm 4.4.1 is computed against the corresponding optimal sensor attack when $V = \sigma_v^2 I_{10}$. The simulations show that the bounds in Theorem 4.5.2 are conservative and that the algorithms (for zero sensor noise) give solutions that are close to optimal for the noisy measurement instances, particularly for RGKSS, even when σ_w^2/σ_v^2 becomes small.

4.6 Chapter Summary

In this chapter, we considered networked dynamical systems affected by a stochastic input. Under this setting, we first studied the problem for a system designer to optimally select the sensors of the nodes over a network subject to a budget constraint in order to minimize the trace of the steady-state error covariance of the corresponding Kalman filter. We then studied the optimal sensor attack problem where an adversary can attack the selected sensors under an attack budget constraint in order to maximize the trace of the steady-state error covariance of the Kalman filter corresponding to the surviving sensors. Using the graph structure of the networked

system, we provided polynomial-time algorithms to solve these two problems. Furthermore, we studied the resilient sensor selection for the system designer when faced with an adversary. We showed that this problem is NP-hard, and provided a pseudo-polynomial-time algorithm to solve it. Although these results are obtained when there is no sensor noise, we provided bounds on the suboptimality of the proposed strategies in the presence of sensor measurement noise.

4.7 Proofs of Key Results

4.7.1 Proof of Lemma 4.5.1

Let $\Sigma_{k/k-1}$ (resp., $\Sigma_{k/k}$) denote the *a priori* (resp., *a posteriori*) error covariance of the Kalman filter at time step k when $V = \mathbf{0}$, and let $\tilde{\Sigma}_{k/k-1}$ (resp., $\tilde{\Sigma}_{k/k}$) denote the *a priori* (resp., *a posteriori*) error covariance of the Kalman filter at time step k when $V = \tilde{V}$. Denoting $\bar{W} \triangleq BWB^T$, we have (from [61]):

$$\tilde{\Sigma}_{k+1/k} = (A - \tilde{K}_k C)\tilde{\Sigma}_{k/k-1}(A - \tilde{K}_k C)^T + \bar{W} + \tilde{K}_k \tilde{V} \tilde{K}_k^T,$$

where $k \geq 0$ and $\tilde{K}_k \triangleq A\tilde{\Sigma}_{k/k-1}C^T(C\tilde{\Sigma}_{k/k-1}C^T + \tilde{V})^{-1}$ is the corresponding Kalman gain at time step k. For any time step k, the Kalman gain $\tilde{K}_k$ satisfies

$$\tilde{K}_k = \arg\min_J \left\{(A - JC)\tilde{\Sigma}_{k/k-1}(A - JC)^T + \bar{W} + J\tilde{V}J^T\right\}, \tag{4.17}$$

where the minimization is in the positive semi-definite sense [61]. Since the pair $(A, BW^{1/2})$ (resp., (A, C)) is stabilizable (resp., detectable), we know from a more general version of Lemma 4.2.1 for general system matrices in [61] that the limit $\tilde{\Sigma} = \lim_{k\to\infty} \tilde{\Sigma}_{k+1/k}$ exists, and satisfies

$$\tilde{\Sigma} = (A - \tilde{K}C)\tilde{\Sigma}(A - \tilde{K}C)^T + \bar{W} + \tilde{K}\tilde{V}\tilde{K}^T,$$

where $\tilde{K} \triangleq A\tilde{\Sigma}C^T(C\tilde{\Sigma}C^T + \tilde{V})^{-1}$ is the corresponding (steady-state) Kalman gain. Similarly, we have

$$\Sigma = (A - KC)\Sigma(A - KC)^T + \bar{W}, \tag{4.18}$$

where $K \triangleq A\Sigma C^T(C\Sigma C^T)^{-1}$. Noting the optimality of the Kalman gains from Eq. (4.17), there exists, as argued in [61], a suboptimal filter (when $V = \tilde{V}$) with a (time-invariant) suboptimal gain given by K such that the corresponding *a priori* error covariance at time step $k+1$, denoted as $\hat{\Sigma}_{k+1/k}$, satisfies

$$\hat{\Sigma}_{k+1/k} = (A - KC)\hat{\Sigma}_{k/k-1}(A - KC)^T + \bar{W} + K\tilde{V}K^T. \tag{4.19}$$

Furthermore, the limit $\hat{\Sigma} \triangleq \lim_{k\to\infty} \hat{\Sigma}_{k+1/k}$ exists and satisfies $\hat{\Sigma} \succeq \tilde{\Sigma}$ [61]. We then obtain from Eq. (4.18) and (the steady-state version of) Eq. (4.19) the following:

$$E = (A - KC)E(A - KC)^T + K\tilde{V}K^T, \tag{4.20}$$

where $E = \hat{\Sigma} - \Sigma$. Since the matrix $A - KC$ is stable [61], it follows that there exists a unique finite positive semi-definite matrix E that satisfies Eq. (4.20) and can be written as $E = \sum_{m=0}^{\infty}(A - KC)^m K\tilde{V}K^T((A - KC)^T)^m$ (e.g., [61]). It then follows from the arguments above that $\hat{\Sigma} = E + \Sigma \succeq \tilde{\Sigma}$.

Similarly, we see from [61] that $\tilde{\Sigma}_{k/k}$ satisfies $\tilde{\Sigma}_{k/k} = (I_n - \tilde{L}_k C)\tilde{\Sigma}_{k/k-1}$, where $\tilde{L}_k \triangleq \tilde{\Sigma}_{k/k-1}C^T(C\tilde{\Sigma}_{k/k-1}C^T + \tilde{V})^{-1}$. Moreover, the limits $\tilde{\Sigma}^* \triangleq \lim_{k\to\infty} \tilde{\Sigma}_{k/k}$ and $\Sigma^* \triangleq \lim_{k\to\infty} \Sigma_{k/k}$ exist and satisfy $\tilde{\Sigma}^* = (I_n - \tilde{L}C)\tilde{\Sigma}$ and $\Sigma^* = (I_n - LC)\Sigma$, respectively, where $\tilde{L} \triangleq \tilde{\Sigma}C^T(C\tilde{\Sigma}C^T + \tilde{V})^{-1}$ and $L \triangleq \Sigma C^T(C\Sigma C^T)^{-1}$. Similarly, the *a posteriori* error covariance at time step k of the suboptimal filter (when $V = \tilde{V}$) as described above, denoted as $\hat{\Sigma}_{k/k}$, is given by

$$\hat{\Sigma}_{k/k} = (I_n - LC)\hat{\Sigma}_{k/k-1}. \tag{4.21}$$

Since the limit $\hat{\Sigma} = \lim_{k\to\infty} \hat{\Sigma}_{k+1/k}$ exists, we know from Eq. (4.21) that the limit $\hat{\Sigma}^* \triangleq \lim_{k\to\infty} \hat{\Sigma}_{k/k}$ also exists. Using similar arguments to those in [61], one can show that $\hat{\Sigma}^* \succeq \tilde{\Sigma}^*$. Thus, we have $\hat{\Sigma}^* - \Sigma^* = (I_n - LC)(\hat{\Sigma} - \Sigma) = (I_n - LC)E$, which implies $\tilde{\Sigma}^* \preceq \Sigma^* + (I_n - LC)E$. ■

5. SENSOR SELECTION FOR HYPOTHESIS TESTING UNDER BUDGET CONSTRAINTS

5.1 Introduction

In the previous chapters, we studied the Kalman filtering sensor selection problem in linear dynamical systems. In this chapter, we consider the (binary) hypothesis testing sensor selection problem in the Neyman-Pearson setting and the Bayesian setting. Each candidate sensor is assumed to be associated with a selection cost, which may vary across the sensors, and we are given a total budget that can be spent on selecting the sensors. The selected sensors first gather measurements in a distributed manner, and then transmit the measurements to a fusion center, where the fusion center can perform the hypothesis testing task (e.g., determining whether a signal exists or not) based on the measurements from the selected sensors. This scenario arises in a variety of applications, such as radar and sonar systems [79] and spectrum sensing for cognitive radio [80]. In this chapter, we first consider the Neyman-Pearson Hypothesis testing Sensor Selection (NPHSS) problem and the Bayesian Hypothesis testing Sensor Selection (BHSS) problem. The NPHSS problem is to minimize the miss probability of the Neyman-Pearson detector based on measurements from the selected sensors, while satisfying the budget constraint. The BHSS problem minimizes the error probability of the Bayesian detector based on measurements of the selected sensors, under the budget constraint. Under Gaussian measurement settings, we consider surrogate sensor selection problems based on distance measures between the conditional distributions corresponding to the measurements from the selected sensors. Specifically, the distance based sensor selection problem (under the Gaussian measurement setting) is to maximize the distance measures between the

conditional distributions corresponding to the measurements of the selected sensors, while satisfying the budget constraints.

Related Work

There is a large literature on sensor selection problems in control system design (e.g., [1, 2, 11, 33, 49, 57]), and have provided complexity characterizations and algorithms. In this chapter, we consider the hypothesis testing sensor selection problem under the Neyman-Pearson and the Bayesian settings.

The hypothesis testing sensor selection problem has also been widely studied (e.g., [3, 4, 21, 81, 82]). In particular, [3, 4, 21] studied the same problem (i.e., the distance based sensor selection problem) as we consider in this chapter, but they only considered the special instances where all the candidate sensors have the same selection cost, i.e., they considered the cardinality constraint on the set of the selected sensors.

In [3], the authors first showed that the objective function of the distance based sensor selection problem (under the Gaussian measurement setting) is not submodular in general. Then, they provided an algorithm based on Stiefel relaxation to solve the distance based sensor selection problem. However, they did not provide any theoretical guarantees on the performance of the proposed algorithm. The authors in [4] also considered the distance based sensor selection problem, and showed that the problem can be approximately solved using a semi-definite programming approach based on convex relaxation. Again, no theoretical performance guarantee was provided for the proposed heuristic.

The authors in [21] applied greedy algorithms to solve the distance based sensor selection problem. When the measurements under the two hypotheses have a common covariance, the authors provided a performance guarantee on the greedy algorithm by leveraging the notion of approximately submodular set functions introduced in [83]. However the performance guarantee becomes loose for arbitrary covariance matrices.

The authors in [83] then considered submodular surrogate objective function for the distance based sensor selection problem. Although the greedy algorithm yields the $(1 - \frac{1}{e})$ approximation guarantee (e.g., [51]) for the surrogate problem, the solution returned by the greedy algorithm for this surrogate problem does not have any provable performance guarantees in terms of the original distance based sensor selection problem. For more general instances of the distance based sensor selection problem (i.e., when the measurements have different covariances under the two hypotheses), the authors in [21] first decomposed the objective function of the problem into a difference of two (surrogate) submodular functions, and then applied a heuristic to solve the problem. However, they did not provide performance guarantees on the proposed heuristic.

Summary of Results

When considering the NPHSS problem and the BHSS problem, our contribution is to show that both of these two problems are NP-hard even when the measurement vector is Gaussian distributed. This complements the complexity result in [3], where only the distance based sensor selection problem was shown to be NP-hard.

For the distance based sensor selection problem (with varying selection costs across the sensors), we apply greedy algorithms to solve the problem, and provide theoretical performance guarantees by leveraging the notion of submodularity ratio introduced in, e.g., [84]. To achieve this, we first extend the analysis in [85] for greedy algorithms for submodular function maximization under budget constraints to nonsubmodular settings. This extended analysis works for the general problem of maximizing nonsubmodular functions under budget constraints. We then provide lower bounds on the submodularity ratio of the objective function in the distance based sensor selection problem, which in turn give performance bounds for the greedy algorithm. Our analysis provides reasonably tight performance guarantees for the greedy algorithm for general classes of instances of the problem (including the case with common co-

variance and the case with uncommon covariances). We supplement our theoretical analysis using illustrative examples and numerical simulations.

Parts of the results presented in this chapter were published in [86].

5.2 Problem Formulation

We consider the classical binary hypothesis testing problem with two possible hypotheses, denoted as H_0 and H_1, respectively. Let $\mathcal{V} \triangleq \{1, 2, \ldots, n\}$ denote the set of all candidate sensors; each sensor is capable of providing a single measurement. Let $X \triangleq \begin{bmatrix} X_1 & X_2 & \cdots & X_n \end{bmatrix}^T \in \mathbb{R}^n$ be the vector that collects measurements (of the signals) from all the sensors in $\mathcal{V}$, where $X_k \in \mathbb{R}$ is the measurement from the kth sensor in $\mathcal{V}$ for all $k \in \mathcal{V}$. The measurement vector X satisfies

$$\begin{aligned} H_0 &: X \sim p(x|H_0), \\ H_1 &: X \sim p(x|H_1), \end{aligned} \tag{5.1}$$

where $p(x|H_i)$ denotes the probability density function (pdf) of X conditioned on the state H_i for $i = 0, 1$.

We consider the scenario where we can only select a subset of sensors from $\mathcal{V}$ to deploy, due to a budget constraint. Specifically, sensor $k \in \mathcal{V}$ has a certain selection cost, denoted as $\omega_k \in \mathbb{R}_{\geq 0}$, for all $k \in \mathcal{V}$. Define $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_n \end{bmatrix}^T$ to be the sensor cost vector. We are given a total budget, denoted as $\Omega \in \mathbb{R}_{>0}$, that can be spent on selecting the sensors.

After a set of sensors is selected, we use their measurements to solve the hypothesis testing problem corresponding to (5.1). We define an indicator vector $\mu \in \{0, 1\}^n$ indicating which sensors are selected, where $\mu_k = 1$ if sensor $k \in \mathcal{V}$ is selected, and $\mu_k = 0$ if otherwise. Given an indicator vector μ with $\text{supp}(\mu) = \{j_1, \ldots, j_p\} \subseteq \{1, \ldots, n\}$, we define $X(\mu) = \begin{bmatrix} X_{j_1} & \cdots & X_{j_p} \end{bmatrix}^T$ to be the vector that contains the measurements from the selected sensors indicated by μ. Denote the pdf of $X(\mu)$

conditioned on H_i as $p(x(\mu)|H_i)$ for $i = 0, 1$. The log-likelihood ratio between the conditional pdfs $p(x(\mu)|H_1)$ and $p(x(\mu)|H_0)$ is defined as

$$\log L(x(\mu)) = \log \frac{p(x(\mu)|H_1)}{p(x(\mu)|H_0)}. \tag{5.2}$$

We first consider the Neyman-Pearson detector for hypothesis testing which minimizes the miss probability (also known as Type II error) $P_m = P_0(H_1)$ such that the false-alarm probability (also known as Type I error) $P_f = P_1(H_0)$ is within a prescribed range, where $P_0(H_1)$ (resp., $P_1(H_0)$) is the conditional probability of deciding H_1 (resp., H_0) given that H_0 (resp., H_1) is true. Given an indicator vector $\mu \in \{0,1\}^n$, we use $P_m(\mu)$ and $P_f(\mu)$ to denote the miss probability, and the false-alarm probability obtained from the measurements of the sensors indicated by μ, respectively. For a given false-alarm rate $\alpha \in \mathbb{R}_{\geq 0}$ and a given indicator vector μ, the decision rule of the Neyman-Pearson detector has the following form:

$$\log L(x(\mu)) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma(\mu), \tag{5.3}$$

where $\log L(x(\mu))$ is defined in Eq. (5.2) and $\gamma(\mu)$ is the threshold chosen such that $P_f(\mu) = \alpha$. Denoting the prior probabilities of the two hypotheses as π_0 and $\pi_1 = 1 - \pi_0$, we next consider the Bayesian hypothesis testing. The Bayes detector minimizes the Bayesian error probability given by $P_e = \pi_0 P_1(H_0) + \pi_1 P_0(H_1)$. Similarly, we use $P_e(\mu)$ to denote the Bayesian error probability corresponding to the sensors indicated by a sensor selection vector $\mu \in \{0,1\}^n$. For a given sensor selection vector μ, the Bayes detector makes a decision using the following decision rule:

$$\log L(x(\mu)) \underset{H_0}{\overset{H_1}{\gtrless}} \log \frac{\pi_0}{\pi_1}, \tag{5.4}$$

where $\log L(x(\mu))$ is defined in Eq. (5.2).

We now define the following sensor selection problems.

Problem 5.2.1 *(NPHSS and BHSS) Consider two possible states H_0 and H_1, a sensor measurement vector $X \in \mathbb{R}^n$ that satisfies* (5.1)*, a cost vector $\omega \in \mathbb{R}^n_{\geq 0}$, and a*

budget $\Omega \in \mathbb{R}_{\geq 0}$. *The Neyman-Pearson Hypothesis testing Sensor Selection (NPHSS) problem is to find a sensor selection vector* μ *that solves*

$$\begin{aligned} \min_{\mu \in \{0,1\}^n} \quad & P_m(\mu) \\ s.t. \ & \omega^T \mu \leq \Omega, \ P_f(\mu) \leq \alpha, \end{aligned}$$

where $P_m(\mu)$ *and* $P_f(\mu)$ *are the miss probability and the false-alarm probability corresponding to* μ*, respectively, and* $\alpha \in \mathbb{R}_{\geq 0}$ *a prescribed false-alarm rate. Similarly, the Bayesian Hypothesis testing Sensor Selection (BHSS) problem is to find a sensor selection vector* μ *that solves*

$$\begin{aligned} \min_{\mu \in \{0,1\}^n} \quad & P_e(\mu) \\ s.t. \ & \omega^T \mu \leq \Omega, \end{aligned}$$

where $P_e(\mu)$ *is the Bayesian error probability corresponding to* μ.

In the remaining of this chapter, we will focus on cases when the measurement vector $X \in \mathbb{R}^n$ is Gaussian distributed, i.e.,

$$\begin{aligned} H_0 &: X \sim \mathcal{N}(\theta_0, \Sigma_0), \\ H_1 &: X \sim \mathcal{N}(\theta_1, \Sigma_1), \end{aligned} \tag{5.5}$$

where $\theta_0, \theta_1 \in \mathbb{R}^n$ and $\Sigma_0, \Sigma_1 \in \mathbb{S}^n_{++}$. The Gaussian distributed case already captures many models in signal detection (e.g., [87]), which is also the main focus of prior work on sensor selection for hypothesis testing (e.g., [3], [21]). Given a sensor selection $\mu \in \{0,1\}^n$ with its support denoted by $\text{supp}(\mu) = \{j_1, \ldots, j_p\} \subseteq \{1, \ldots, n\}$, we define $\theta_i(\mu) = \begin{bmatrix} (\theta_i)_{j_1} & \cdots & (\theta_i)_{j_p} \end{bmatrix}^T$, and define $\Sigma_i(\mu)$ to be the submatrix of Σ_i that contains the rows and columns corresponding to $\text{supp}(\mu)$, for $i = 0, 1$. In other words, the conditional pdfs of $X(\mu)$ on H_0 and H_1 satisfy

$$\begin{aligned} H_0 &: X(\mu) \sim \mathcal{N}(\theta_0(\mu), \Sigma_0(\mu)), \\ H_1 &: X(\mu) \sim \mathcal{N}(\theta_1(\mu), \Sigma_1(\mu)). \end{aligned}$$

The following result characterizes the complexity of the NPHSS and BHSS problems; the proof is included in Section 5.7.1.

Theorem 5.2.2 *The NPHSS and BHSS problems are NP-hard even when the measurement vector $X \in \mathbb{R}^n$ satisfies* (5.5).

In fact, P_m, P_f and P_e do not always yield closed-form expressions even under the Gaussian distributed setting. Thus, we are further motivated to consider alternate optimization criteria, which have closed-form expressions for the NPHSS and BHSS problems, in order to seek efficient algorithms to solve the hypothesis testing sensor selection problems. Fortunately, several metrics pertaining to the distance (or divergence) between two probability distributions have been shown to serve as reasonable surrogates to the error probabilities as the optimization metrics in the NPHSS problem and the BHSS problem, such as the Kullback-Leibler (KL) distance $\mathcal{K}(H_1\|H_0)$, the J-Divergence $\mathcal{D}(H_1\|H_0)$, and the Bhattacharyya distance $\mathcal{B}(H_1\|H_0)$. The problems then become maximizing the corresponding distances between the two probability distributions (under the budget constraints). The interested readers are referred to [3, 4, 87–89] and the references therein for detailed explanations about using these surrogates as the optimization metrics in the NPHSS and BHSS problems. Note that the KL distance, J-Divergence and Bhattacharyya distance between two Gaussian distributions yield closed-form expressions. Denote the KL distance, J-Divergence and Bhattacharyya distance corresponding to μ, i.e., between $\mathcal{N}(\theta_0(\mu), \Sigma_0(\mu))$ and $\mathcal{N}(\theta_1(\mu), \Sigma_1(\mu))$, as $f_{KL}(\mu)$, $f_{JD}(\mu)$ and $f_{BD}(\mu)$, respectively, which are given by (e.g., [21]):

$$f_{KL}(\mu) = \frac{1}{2}\big(\mathrm{tr}(\tilde{\Sigma}_0^{-1}\tilde{\Sigma}_1) + (\tilde{\theta}_1 - \tilde{\theta}_0)^T\tilde{\Sigma}_0^{-1}(\tilde{\theta}_1 - \tilde{\theta}_0) + \log\frac{\det(\tilde{\Sigma}_0)}{\det(\tilde{\Sigma}_1)} - |\mathrm{supp}(\mu)|\big), \quad (5.6)$$

$$f_{JD}(\mu) = \frac{1}{2}\big(\mathrm{tr}(\tilde{\Sigma}_0^{-1}\tilde{\Sigma}_1) + \mathrm{tr}(\tilde{\Sigma}_1^{-1}\tilde{\Sigma}_0) + (\tilde{\theta}_1 - \tilde{\theta}_0)^T\tilde{\Sigma}_0^{-1}(\tilde{\theta}_1 - \tilde{\theta}_0) + (\tilde{\theta}_1 - \tilde{\theta}_0)^T\tilde{\Sigma}_1^{-1}(\tilde{\theta}_1 - \tilde{\theta}_0)\big) - |\mathrm{supp}(\mu)|, \quad (5.7)$$

$$f_{BD}(\mu) = \frac{1}{8}(\tilde{\theta}_1 - \tilde{\theta}_0)^T\tilde{\Sigma}^{-1}(\tilde{\theta}_1 - \tilde{\theta}_0) + \frac{1}{2}\log\frac{\det(\tilde{\Sigma})}{\sqrt{\det(\tilde{\Sigma}_1)\det(\tilde{\Sigma}_0)}}, \quad (5.8)$$

where $\tilde{\theta}_i \triangleq \theta_i(\mu)$ and $\tilde{\Sigma}_i \triangleq \Sigma_i(\mu)$ for $i = 0, 1$, and $\tilde{\Sigma} \triangleq \frac{1}{2}(\tilde{\Sigma}_0 + \tilde{\Sigma}_1)$.

Remark 5.2.3 *Noting from Eqs.* (5.6)-(5.8) *that* $f_{KL}(\mu)$, $f_{JD}(\mu)$ *and* $f_{BD}(\mu)$ *depend on* $\theta_0(\mu)$ *and* $\theta_1(\mu)$ *only through* $\big(\theta_1(\mu)-\theta_0(\mu)\big)$, *we assume without loss of generality that* $\theta_0 = \mathbf{0}_n$ *in the sequel. Moreover, it is well-known that these distance measures are always nonnegative (e.g., [89]), i.e.,* $f_{KL}(\mu), f_{JD}(\mu), f_{BD}(\mu) \in \mathbb{R}_{\geq 0}$ *for all* $\mu \in \{0,1\}^n$.

Following the arguments in [3,4,21], we now formally define the following distance based sensor selection problems.

Problem 5.2.4 *(KLDSS, JDSS and BDSS) Consider two possible states* H_0 *and* H_1, *a sensor measurement vector* $X \in \mathbb{R}^n$ *that satisfies* (5.5), *a cost vector* $\omega \in \mathbb{R}^n_{\geq 0}$, *and a budget* $\Omega \in \mathbb{R}_{\geq 0}$. *The KL Distance Sensor Selection (KLDSS) problem, the J-Divergence Sensor Selection (JDSS) problem, and the Bhattacharyya Distance Sensor Selection (BDSS) problem are to find sensor selection vectors* μ *that solve*

$$\begin{aligned} \max_{\mu\in\{0,1\}^n} \quad & f_{OBJ}(\mu) \\ s.t. \quad & \omega^T\mu \leq \Omega, \end{aligned} \tag{5.9}$$

where $f_{OBJ}(\cdot)$ *is taken to be* $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ *and* $f_{BD}(\cdot)$, *given by Eq.* (5.6), *Eq.* (5.7), *and Eq.* (5.8), *respectively.*

It was shown in [3] that the KLDSS problem and the BDSS problem are NP-hard even when $\Sigma_0 = \Sigma_1$. In fact, supposing $\Sigma_0 = \Sigma_1$ in the KLDSS, JDSS and BDSS problems, we see that the expressions in Eqs. (5.6)-(5.8) reduce to

$$f_{KL}(\mu) = \frac{1}{2}\tilde{\theta}_1^T\tilde{\Sigma}_0^{-1}\tilde{\theta}_1, \tag{5.10}$$

$$f_{JD}(\mu) = \tilde{\theta}_1^T\tilde{\Sigma}^{-1}\tilde{\theta}_1, \tag{5.11}$$

$$f_{BD}(\mu) = \frac{1}{8}\tilde{\theta}_1^T\tilde{\Sigma}^{-1}\tilde{\theta}_1, \tag{5.12}$$

respectively. Note that the form of the expressions in Eqs. (5.10)-(5.12) coincide with the objective function in the subset selection problem (up to constant multiplicative factors) considered in [39], which is known to be NP-hard [38]. Hence, we can conclude that the KLDSS, JDSS and BDSS problems are all NP-hard even when

the two hypotheses have a common covariance, i.e., $\Sigma_0 = \Sigma_1$. Due to the fact that these alternate problem formulations to the original hypothesis testing sensor problem (Problem 5.2.1) are NP-hard, there are still no algorithms that can solve these alternate problems optimally in polynomial time (unless P=NP). Nevertheless, the closed-form expressions given in Eqs. (5.6)-(5.8) give us chances to prove theoretical performance guarantees for (approximation) algorithms when applied to the KLDSS, JDSS and BDSS problems. We will pursue this direction in the sequel and focus particularly on greedy algorithms with their applications to the KLDSS, JDSS and BDSS problems.

5.3 Greedy Algorithm

Algorithm 5.3.1 Greedy Algorithm for Problem (P)

Input: $\mathcal{V}$, $f : 2^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$, Q, $c(v), \forall v \in \mathcal{V}$

Output: S_g

1: $S_g \leftarrow \emptyset$
2: **while** $c(S_g) \leq Q$ **do**
3: $\quad v^* = \arg\max_{v \in \mathcal{V} \setminus S_g} \frac{f(S_g \cup \{v\}) - f(S_g)}{c(v)}$
4: $\quad$ **if** $c(S_g) + c(v^*) \leq Q$ **then**
5: $\quad\quad S_g \leftarrow S_g \cup \{v^*\}$
6: $\quad$ **else**
7: $\quad\quad$ **return** $\mathcal{S}_g$

To begin our analysis, let us consider the following problem:

$$\begin{aligned} &\max_{A \subseteq \mathcal{V}} f(A) \\ &s.t. c(A) \leq Q, \end{aligned} \tag{P}$$

where $f : 2^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$ is a set function that is assumed, without loss of generality, to be normalized, i.e., $f(\emptyset) = 0$. The constraint $c(A) \leq Q$ represents a budget

constraint on the sum of costs of elements in A. Specifically, each element $v \in \mathcal{V}$ is associated with a cost denoted by $c(v) \in \mathbb{R}_{\geq 0}$, and the sum of the costs of elements in A needs to satisfy $c(A) \triangleq \sum_{v \in A} c(v) \leq Q$, where $Q \in \mathbb{R}_{\geq 0}$ and $c(\emptyset) \triangleq 0$. Noting the one-to-one correspondence between a sensor selection $\mu \in \{0,1\}^n$ and a subset of selected sensors indicated by μ,[1] the KLDSS, JDSS and BDSS problems defined in Problem 5.2.4 fall into the class of problems defined by (P). For the case when $f(\cdot)$ is monotone nondecreasing and submodular, [85] provides performance guarantees for the greedy algorithm, defined in Algorithm 5.3.1, when applied to solve Problem (P). Note that the set function $f(\cdot)$ is said to be monotone nondecreasing if for all $A \subseteq \mathcal{V}$ and for all $v \in \mathcal{V}$, $f(A \cup \{v\}) - f(A) \geq 0$. The set function $f(\cdot)$ is said to be submodular if and only if $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$ for all A, B such that $A \subseteq B \subseteq \mathcal{V}$ and for all $v \in \mathcal{V} \setminus B$. However, one can come up with instances where the objective functions $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$ (in the KLDSS, JDSS and BDSS problems, respectively) are not submodular (e.g., [3]). In order to also provide approximation guarantees of the greedy algorithm when applied to solve the KLDSS, JDSS and BDSS problems, we will leverage the notion of submodularity ratio defined as follows (see, e.g., [40, 84]), which characterizes how close a set function is to being submodular.

Definition 5.3.1 *The submodularity ratio of a nonnegative set function $f : 2^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$ is the largest $\gamma \in \mathbb{R}$ that satisfies*

$$\sum_{v \in A \setminus B} \big(f(\{v\} \cup B) - f(B)\big) \geq \gamma\big(f(A \cup B) - f(B)\big), \tag{5.13}$$

for all $A, B \subseteq \mathcal{V}$.

Remark 5.3.1 *We note from [84] that for a monotone nondecreasing $f(\cdot)$, $\gamma \in [0,1]$ by Definition* 5.3.1*; and $f(\cdot)$ is submodular if and only if $\gamma = 1$.*

[1] We will also write $f_{OBJ}(\mu)$ as $f_{OBJ}(A)$ in the sequel, where $A = \text{supp}(\mu) \subseteq \mathcal{V}$, and $f_{OBJ}(\cdot)$ is specified in Problem 5.2.4. Similarly, we will write $\theta_i(A)$ and $\Sigma_i(A)$ for $i = 0, 1$.

Based on Definition 5.3.1, we extend the analysis of the greedy algorithm for maximizing submodular functions under budget constraints to nonsubmodular settings. We have the following result; the proof technique is inspired by [85].

Theorem 5.3.2 *Consider an instance of Problem* (P)*, where the set function* $f : 2^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$ *is monotone nondecreasing with submodularity ratio* $\gamma \in \mathbb{R}_{\geq 0}$*. The greedy algorithm (Algorithm* 5.3.1*) has the following approximation guarantee when solving Problem* (P)*:*

$$f(S_g) \geq (1 - e^{-\gamma \frac{c(S_g)}{Q}}) f(S^*), \tag{5.14}$$

where S_g *is the output of the greedy algorithm, and* S^* *is the optimal solution to Problem* (P)*.*

Proof Since (5.14) always holds when $\gamma = 0$, we will assume $\gamma > 0$ in the proof as follows. Denote $S_g = \{s_1, s_2, \ldots, s_\tau\}$, where $\tau = |S_g|$. For all $j \in \{0, \ldots, \tau\}$, let $S_j = \{s_1, \ldots, s_j\}$, where $S_0 = \emptyset$. Note that the greedy choice satisfies $s_j = \arg\max_{v \in \mathcal{V} \setminus S_{j-1}} \frac{f(S_{j-1} \cup \{v\}) - f(S_{j-1})}{c(v)}$ for all $j \in \{1, \ldots, \tau\}$. We will first prove the following claim.

Claim 5.3.3 *For all* $A \subseteq \mathcal{V}$ *and for all* $j \in \{0, \ldots, \tau - 1\}$*, we have*

$$f(S^*) \leq f(S_j) + \frac{Q}{\gamma} \cdot \frac{f(S_{j+1}) - f(S_j)}{c(s_{j+1})}. \tag{5.15}$$

To prove the above claim, we note from (5.13) that

$$\begin{aligned} f(S^* \cup S_j) - f(S_j) &\leq \frac{1}{\gamma} \sum_{s \in S^* \setminus S_j} c(s) \cdot \frac{f(\{s\} \cup S_j) - f(S_j)}{c(s)}, \\ &\leq \frac{1}{\gamma} \sum_{s \in S^* \setminus S_j} c(s) \cdot \frac{f(S_{j+1}) - f(S_j)}{c(s_{j+1})}, \qquad (5.16) \\ &\leq \frac{Q}{\gamma} \cdot \frac{f(S_{j+1}) - f(S_j)}{c(s_{j+1})}, \qquad (5.17) \end{aligned}$$

where (5.16) follows from the greedy choice, and (5.17) follows from $c(S^* \setminus S_j) \leq c(S^*) \leq Q$. Noting that $f(S^*) \leq f(S^* \cup S_j)$, we complete the proof of Claim 5.3.3.

We are now in place to prove (5.14). Let $D_j = f(S^*) - f(S_j)$ for all $j \in \{0, \ldots, \tau\}$. We have from Claim 5.3.3 the following:

$$D_{j-1} \leq \frac{Q}{\gamma} \cdot \frac{f(S_j) - f(S_{j-1})}{c(s_j)} = \frac{Q}{\gamma} \cdot \frac{D_{j-1} - D_j}{c(s_j)},$$
$$\implies D_j \leq D_{j-1}(1 - \frac{c(s_j)\gamma}{Q}). \tag{5.18}$$

Rolling up (5.18), we obtain

$$D_\tau \leq D_0 \prod_{j=1}^{\tau} (1 - \frac{c(s_j)\gamma}{Q}). \tag{5.19}$$

Moreover, noting that $\sum_{j=1}^{\tau} c(s_j) = c(S_g)$, one can show that $\prod_{j=1}^{\tau}(1 - \frac{c(s_j)\gamma}{Q}) \leq \prod_{j=1}^{\tau}(1 - \frac{c(S_g)\gamma}{\tau Q})$ (e.g., [31]). It then follows from (5.19) that

$$f(S^*) - f(S_g) \leq f(S^*)(1 - \frac{c(S_g)\gamma}{\tau Q})^\tau < f(S^*)e^{-\gamma \frac{c(S_g)}{Q}},$$

which completes the proof of the theorem. ■

Clearly, the approximation guarantee of the greedy algorithm for solving Problem (P) provided in (5.14) depends on the submodularity ratio $\gamma \in \mathbb{R}_{\geq 0}$ of $f(\cdot)$, the budget constraint $Q \in \mathbb{R}_{\geq 0}$, and the sum of the costs of the elements in S_g, i.e., $c(S_g)$. Supposing $c_1 \triangleq \max_{v \in \mathcal{V}} c(v) \leq Q$, one can show that $c(S_g) \geq Q - c_1$, which implies that $f(S_g) \geq (1 - e^{-\gamma \frac{Q - c_1}{Q}}) f(S^*)$. Moreover, suppose it is allowed to violate the budget constraint such that $c(S_g) \leq Q'$, where $Q' = c_1 + l$ with $l \in \mathbb{R}_{\geq 0}$. One can then show that $c(S_g) \geq l$, which implies that $f(S_g) \geq (1 - e^{-\gamma \frac{l}{Q}}) f(S^*)$. Note that if $c(v) = 1$ for all $v \in \mathcal{V}$, then $c(S_g) = Q$ and the approximation guarantee reduces to $f(S_g) \geq (1 - e^{-\gamma}) f(S^*)$ as obtained in, e.g., [40].

Recall that our goal is to provide performance guarantees for the greedy algorithms when applied to the KLDSS, JDSS and BDSS problems. In the following section, we will show that the objective functions of the three problems, i.e., $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$, are all monotone nondecreasing. More importantly, we will characterize the submodularity ratios of $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$. Since obtaining the exact value of the submodularity ratio of a set function from Definition 5.3.1 would require an

exhaust search over all $A, B \subseteq \mathcal{V}$, we are interested in obtaining (strictly-positive) lower bounds on the submodularity ratio that can be computed in polynomial time. Substituting the lower bounds in (5.14) also leads to performance bounds of the greedy algorithm.

5.4 Bounding the Submodularity Ratios

5.4.1 Common Covariance

As we mentioned in Section 5.2, in the special case when the two hypotheses have the same covariance, i.e., $\Sigma_0 = \Sigma_1 = \Sigma_c$, where $\Sigma_c \in \mathbb{S}^n_{++}$, the objective functions $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$ are given by Eqs. (5.10), (5.11) and (5.12), respectively, which are equivalent to the objective function in the subset selection problem (up to multiplicative constant factors) [39]. In this case, the measurement vector is given by

$$\begin{aligned} H_0 &: X \sim \mathcal{N}(\mathbf{0}, \Sigma_c), \\ H_1 &: X \sim \mathcal{N}(\theta_1, \Sigma_c), \end{aligned} \tag{5.20}$$

where $\theta_1 \in \mathbb{R}^n$, $\Sigma_c \in \mathbb{S}^n_{++}$, and we recall from Remark 5.2.3 that we have assumed without loss of generality that $\theta_0 = \mathbf{0}$. Using similar arguments to those in [40], we have the following result which characterizes lower bounds on the submodularity ratios of $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$; the proof is included in Section 5.7.2.

Proposition 5.4.1 *Suppose the measurement vector $X \in \mathbb{R}^n$ satisfies* (5.20)*. Then, $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$ are monotone nondecreasing, and the submodularity ratios of $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$, denoted by $\gamma_{KL} \in \mathbb{R}_{\geq 0}$, $\gamma_{JD} \in \mathbb{R}_{\geq 0}$ and $\gamma_{BD} \in \mathbb{R}_{\geq 0}$, respectively, satisfy $\gamma_{KL} \geq \frac{\lambda_n(\Sigma_c)}{d_1(\Sigma_c)}$, $\gamma_{JD} \geq \frac{\lambda_n(\Sigma_c)}{d_1(\Sigma_c)}$ and $\gamma_{BD} \geq \frac{\lambda_n(\Sigma_c)}{d_1(\Sigma_c)}$.*

5.4.2 Uncommon Means and Uncommon Covariances

Moving forward to more general problem instances, our next goal is to give lower bounds on the submodularity ratios of $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$, when the two hy-

potheses can have uncommon means and uncommon covariances. That is, the measurement vector $X \in \mathbb{R}^n$ satisfies

$$\begin{aligned} H_0 &: X \sim \mathcal{N}(\mathbf{0}, \Sigma_0), \\ H_1 &: X \sim \mathcal{N}(\theta_1, \Sigma_1), \end{aligned} \tag{5.21}$$

where $\theta_1 \in \mathbb{R}^n$ and $\Sigma_0, \Sigma_1 \in \mathbb{S}^n_{++}$. We make the following assumption on the instances of the KLDSS, JDSS and BDSS problems that we will consider in this section.

Assumption 5.4.1 *Let $\Sigma_0 = diag(\sigma_1^2, \ldots, \sigma_n^2)$, where $\sigma_i \in \mathbb{R}_{>0}$ for all $i \in \{1, \ldots, n\}$.*

Remark 5.4.2 *Assumption* 5.4.1 *holds in the problem of detecting a Gaussian signal in Gaussian noise, where the noise terms from different sensors are uncorrelated [87]. Here, the measurement vector $X \in \mathbb{R}^n$ satisfies*

$$\begin{aligned} H_0 &: X = N, \\ H_1 &: X = S + N, \end{aligned} \tag{5.22}$$

where $N \sim \mathcal{N}(\mathbf{0}, \Sigma_N)$ and $S \sim \mathcal{N}(\theta_S, \Sigma_S)$, where $\theta_S \in \mathbb{R}^n$, $\Sigma_N, \Sigma_S \in \mathbb{S}^n_{++}$, and Σ_N is diagonal. The measurement noise $N \in \mathbb{R}^n$ and the signal $S \in \mathbb{R}^n$ are assumed to be uncorrelated.

Since the two hypotheses can have different covariances, the objective functions $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$ no longer have similar expressions. Thus, we treat $f_{KL}(\cdot)$, $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$ separately as follows. Proofs of the following results can be found in Sections 5.7.3, 5.7.4 and 5.7.5, respectively.

Proposition 5.4.2 *Suppose the measurement vector $X \in \mathbb{R}^n$ satisfies* (5.21)*. Then, the objective function $f_{KL}(\cdot)$ of the KLDSS problem under Assumption* 5.4.1 *is monotone nondecreasing, and the submodularity ratio of $f_{KL}(\cdot)$, denoted as $\gamma_{KL} \in \mathbb{R}_{\geq 0}$, satisfies*

$$\gamma_{KL} \geq \frac{\min_{v \in \mathcal{V}} C_{KL}(v)}{\log \frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + \min_{v \in \mathcal{V}} C_{KL}(v)}, \tag{5.23}$$

where

$$C_{KL}(v) \triangleq \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - \log \frac{\Sigma_1(v)}{\Sigma_0(v)} - 1, \forall v \in \mathcal{V}. \tag{5.24}$$

Proposition 5.4.3 *Suppose the measurement vector $X \in \mathbb{R}^n$ satisfies* (5.21)*. Then, the objective function $f_{JD}(\cdot)$ of the JDSS problem under Assumption* 5.4.1 *is monotone nondecreasing, and the submodularity ratio of $f_{JD}(\cdot)$, denoted as $\gamma_{JD} \in \mathbb{R}_{\geq 0}$, satisfies*

$$\gamma_{JD} \geq \min\big\{C_{JD}, \frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}\big\}, \tag{5.25}$$

where

$$C_{JD} \triangleq \min_{v\in\mathcal{V}} \frac{\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2}{\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)} - 2}. \tag{5.26}$$

Proposition 5.4.4 *Suppose the measurement vector $X \in \mathbb{R}^n$ satisfies* (5.21) *and that Assumption* 5.4.1 *holds. (a) The objective function $f_{BD}(\cdot)$ of the BDSS problem is monotone nondecreasing, and the submodularity ratio of $f_{BD}(\cdot)$, denoted by $\gamma_{BD} \in \mathbb{R}_{\geq 0}$, satisfies*

$$\gamma_{BD} \geq \min\big\{\frac{C_{BD}}{\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + C_{BD}}, \frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}\big\}, \tag{5.27}$$

where $C_{BD} \triangleq \min\{C^0_{BD}, C^1_{BD}\}$ with

$$C^0_{BD} \triangleq \log\big(1 + \frac{1}{4}\min_{v\in\mathcal{V}} \frac{(\Sigma_0(v) - \Sigma_1(v))^2}{\Sigma_0(v)\Sigma_1(v)}\big), \tag{5.28}$$

and

$$C^1_{BD} \triangleq \log\big(1 + \frac{\alpha^2 \frac{\lambda_n(\Sigma_0)}{\lambda_n(\Sigma_0)+\lambda_1(\Sigma_1)}}{\lambda_1(\Sigma_1)\max_{v\in\mathcal{V}}\Sigma_1(v) - \alpha^2}\big), \tag{5.29}$$

where $\alpha = \min_{i.j\in\mathcal{V}, i\neq j, (\Sigma_1)_{ij}\neq 0}(\Sigma_1)_{ij}$. (b) Further suppose $\Sigma_1 = \Sigma_0 + \Sigma_S$, where $\Sigma_S \in \mathbb{S}^n_{++}$. Then,

$$\gamma_{BD} \geq \min\big\{\frac{\bar{C}_{BD}}{\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + \bar{C}_{BD}}, \frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}\big\}, \tag{5.30}$$

where $\bar{C}_{BD} \triangleq \min\{\bar{C}^0_{BD}, \bar{C}^1_{BD}\}$ with

$$\bar{C}^0_{BD} \triangleq \log\big(1 + \frac{1}{4}\min_{v\in\mathcal{V}} \frac{(\Sigma_0(v) + \Sigma_1(v))^2}{\Sigma_0(v)\Sigma_1(v)}\big), \tag{5.31}$$

and

$$\bar{C}^1_{BD} \triangleq \log\big(1 + \frac{\bar{\alpha}^2 \frac{\lambda_n(\Sigma_0)}{\lambda_n(\Sigma_1)+\lambda_1(\Sigma_0)}}{\lambda_1(\Sigma_1)\max_{v\in\mathcal{V}}\Sigma_1(v) - \bar{\alpha}^2} + \frac{(\lambda_n(\Sigma_S))^2}{4\max_{v\in\mathcal{V}}\big(\Sigma_0(v)(\Sigma_1(v) - \frac{\bar{\alpha}^2}{\lambda_1(\Sigma_1)})\big)}\big), \tag{5.32}$$

where $\bar{\alpha} \triangleq \min_{i,j\in\mathcal{V}, i\neq j}(\Sigma_1)_{ij}$.

Given the problem parameters, the lower bounds on the submodularity ratios provided in Propositions 5.4.1-5.4.4 can be computed in $O(n^3)$ time, since finding $\lambda_1(\Sigma_1)$ and $\lambda_n(\Sigma_1)$ requires $O(n^3)$ in the worst case (e.g., [90]). Also note that the lower bounds are all nonnegative. In order to further illustrate how the bounds (particularly in Propositions 5.4.2-5.4.4) depend on the parameters of the KLDSS, JDSS and BDSS problems, we consider the following example corresponding to Remark 5.4.2.

Example 5.4.3 *Consider the problem of detecting the signal $S \sim \mathcal{N}(\theta_S, \Sigma_S)$ in the noise $N \sim \mathcal{N}(\mathbf{0}, \Sigma_N)$, where $S \in \mathbb{R}^n$ and $N \in \mathbb{R}^n$ are assumed to be uncorrelated. Suppose $\theta_S = \beta \mathbf{1}_n$, $\Sigma_S \in \mathbb{S}^n_{++}$ and $\Sigma_N = \sigma^2 I_n$, where $\beta \in \mathbb{R}_{\geq 0}$ and $\sigma \in \mathbb{R}_{>0}$.*

Note that the sensor measurement vector $X \in \mathbb{R}^n$ corresponding to the instances given by Example 5.4.3 satisfies (5.22), which gives $\theta_1 = \theta_S$, $\Sigma_0 = \sigma^2 I_n$ and $\Sigma_1 = \Sigma_N + \Sigma_S$. One can show that the bound on γ_{KL} given in (5.23) simplifies into

$$\gamma_{KL} \geq \frac{\frac{\beta^2}{\sigma^2} + \frac{d_n(\Sigma_S)}{\sigma^2} - \log \frac{\sigma^2 + d_n(\Sigma_S)}{\sigma^2}}{\log \frac{d_1(\Sigma_S)+\sigma^2}{\lambda_n(\Sigma_S)+\sigma^2} + \frac{\beta^2}{\sigma^2} + \frac{d_n(\Sigma_S)}{\sigma^2} - \log \frac{\sigma^2 + d_n(\Sigma_S)}{\sigma^2}}. \tag{5.33}$$

Supposing β, σ, $\lambda_n(\Sigma_S)$ and $d_1(\Sigma_S)$ are fixed, one can show that the bound in (5.33) will increase as $d_n(\Sigma_S)$ increases. Similarly, supposing σ and Σ_S are fixed, one can show that the bound in (5.33) will increase as β increases.

We then turn to the bound on γ_{JD} given in (5.25). First, one can show that C_{JD} in the bound corresponding to the instances given by Example 5.4.3 has the form:

$$C_{JD} = \frac{\frac{\beta^2}{\sigma^2} + \frac{d_1(\Sigma_S)+\sigma^2}{\sigma^2} + \frac{\sigma^2}{d_1(\Sigma_S)+\sigma^2} - 2}{\frac{\beta^2}{\sigma^2} + \frac{d_1(\Sigma_S)+\sigma^2}{\sigma^2} + \frac{\sigma^2}{\lambda_n(\Sigma_S)+\sigma^2} - 2}. \tag{5.34}$$

Supposing β, σ and $\lambda_n(\Sigma_S)$ are fixed, one can show that C_{JD} given by Eq. (5.34) will increase as $d_1(\Sigma_S)$ decreases. Noting that $\frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)} = \frac{\lambda_n(\Sigma_S)+\sigma^2}{d_1(\Sigma_S)+\sigma^2}$ will also increase as $d_1(\Sigma_1)$ decreases, we see that the lower bound in (5.25) will increase as $d_1(\Sigma_S)$ decreases. Similarly, supposing σ and Σ_S are fixed, one can show that C_{JD} given in Eq. (5.34) will increase as β increases. However, the increment of C_{JD} would only lead to a potential increment of the bound in (5.25), since $\frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}$ does not depend on β.

Finally, let us consider the bound on γ_{BD} given in (5.30) under the instances of Example 5.4.3. Specifically, one can first obtain from Eqs. (5.31)-(5.32) that

$$\bar{C}^0_{BD} = \log\big(1 + \frac{(2\sigma^2 + d_n(\Sigma_S))^2}{4\sigma^2(\sigma^2 + d_n(\Sigma_S))}\big), \tag{5.35}$$

and

$$\bar{C}^1_{BD} = \log\big(1 + \frac{\bar{\alpha}^2 \frac{\sigma^2}{2\sigma^2+\lambda_n(\Sigma_S)}}{(\sigma^2 + \lambda_1(\Sigma_S))(d_1(\Sigma_S) + \sigma^2) - \bar{\alpha}^2} + \frac{(\lambda_n(\Sigma_S))^2}{4\sigma^2\big(\sigma^2 + d_1(\Sigma_S) - \frac{\bar{\alpha}^2}{\lambda_1(\Sigma_S)+\sigma^2}\big)}. \tag{5.36}$$

Supposing σ, $\lambda_n(\Sigma_S)$, $d_n(\Sigma_S)$, and $\bar{\alpha}$ are fixed, we see that $\bar{C}^0_{BD}$ is fixed, and $\bar{C}^1_{BD}$ will increase if $d_1(\Sigma_S)$ (or $\lambda_1(\Sigma_S)$) decreases. It follows that $\bar{C}_{BD} = \min\{\bar{C}^0_{BD}, \bar{C}^1_{BD}\}$ will increase if $d_1(\Sigma_S)$ (or $\lambda_1(\Sigma_S)$) decreases. Moreover, one can show that $\frac{\bar{C}_{BD}}{\log \frac{d_1(\Sigma_S)+\sigma^2}{\lambda_n(\Sigma_S)+\sigma^2} + \bar{C}_{BD}}$ will increase as $\bar{C}_{BD}$ increases (supposing $\frac{d_1(\Sigma_S)+\sigma^2}{\lambda_n(\Sigma_S)+\sigma^2}$ is fixed). Also note that $\frac{d_1(\Sigma_S)+\sigma^2}{\lambda_n(\Sigma_S)+\sigma^2}$ will decrease as $d_1(\Sigma_S)$ decreases. Combining the above arguments yields that the bound in (5.30) will increase if $d_1(\Sigma_S)$ decreases.

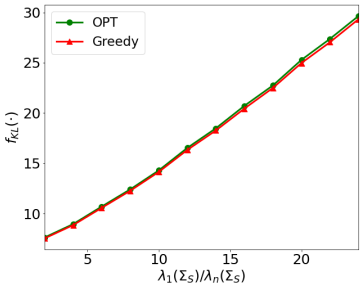
In summary, for the instances in Example 5.4.3, the bounds in Propositions 5.4.2-5.4.4 depend on the difference between $d_1(\Sigma_S)$ and $d_n(\Sigma_S)$ (or the difference between $\lambda_1(\Sigma_S)$ and $\lambda_n(\Sigma_S)$). Note that $d_1(\Sigma_S)$ and $d_n(\Sigma_S)$ correspond to the largest (resp., smallest) variance of the signals detected by different sensors. Thus, smaller differences among the signal variances of different sensors potentially lead to larger values of the bounds on the submodularity ratios.

Remark 5.4.4 *We note from Theorem* 5.3.2 *that tighter lower bounds on the submodularity ratio yield tighter performance bounds of the greedy algorithm, which potentially imply better performances of the greedy algorithm.*

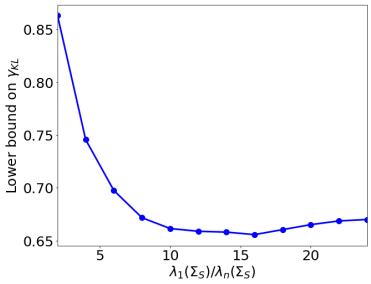
5.5 Numerical Examples

We further illustrate the lower bounds on the submodularity ratios given by Propositions 5.4.2-5.4.4 using synthetic examples, which allow us to show how those lower bounds behave according to different parameters of the KLDSS, JDSS and BDSS

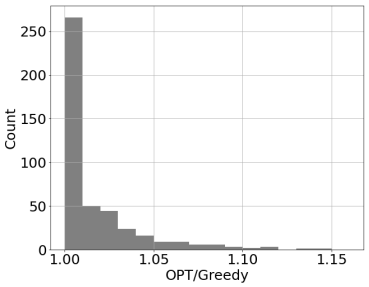
problems (Problem 5.2.4), which in turn influence the performance of the greedy algorithm. Following Example 5.4.3, we consider instances of Problem 5.2.4 with the form $\Sigma_1 = \Sigma_0 + \Sigma_S$, where $\Sigma_0, \Sigma_S \in \mathbb{S}^n_{++}$, and Σ_0 is diagonal. We then generate the instances of Problem 5.2.4 using the following procedure. Considering a system of 15 sensors, we set the budget $\Omega = 20$, and generate a random cost vector $\omega \in \mathbb{Z}^{15}_{>0}$, which results in $\max_{v\in\mathcal{V}} \omega_v = 8$. Ω and ω are then kept fixed in the sequel. We set $\Sigma_0 = \sigma^2 I_{15}$, where $\sigma \in \mathbb{R}_{\geq 0}$. As argued in Section 5.4, the lower bounds on the submodularity ratios depend on the difference between $\lambda_1(\Sigma_S)$ and $\lambda_n(\Sigma_S)$. Therefore, in order to generate $\Sigma_S \in \mathbb{S}^n_{++}$ in a controlled fashion while observing the behavior of the bounds under different problem parameters, we keep $\lambda_n(\Sigma_S) = 1$ and vary $\lambda_1(\Sigma_S)$ within certain ranges. For each value of $\lambda_1(\Sigma_S)$, we obtain 500 instances of Problem 5.2.4 via random generations of $\Sigma_S \in \mathbb{S}^{15}_{++}$ and $\theta_1 \in \mathbb{R}^{15}$.



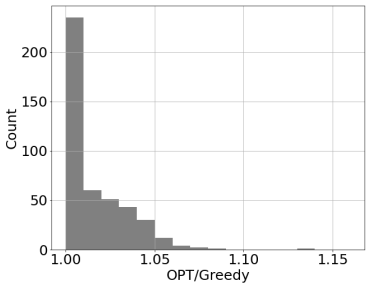
(a) OPT vs. Greedy



(b) Submodularity ratio



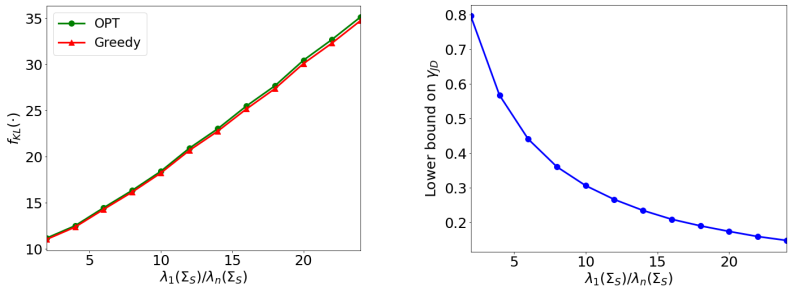
(c) $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 2$



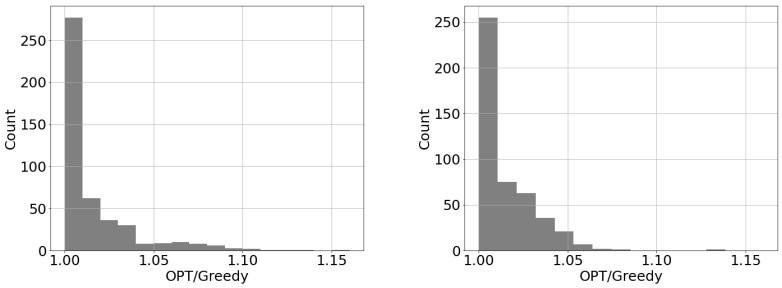
(d) $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 20$

Fig. 5.1. Results for KLDSS when $\sigma = 1$.

The results for the KLDSS problem are reported in Fig. 5.1, where we set $\sigma = 1$. Here, Fig. 5.1(a) plots the optimal solutions and the greedy solutions for different values of $\frac{\lambda_n(\Sigma_S)}{\lambda_1(\Sigma_S)}$, where for each value of $\frac{\lambda_n(\Sigma_S)}{\lambda_1(\Sigma_S)}$, the optimal solution and the greedy solution are averaged over the 500 instances of the KLDSS problem. Fig. 5.1(b) plots the averaged lower bound on γ_{KL} (provided in Proposition 5.4.2). Fig. 5.1(c)-(d) plot the histograms of the ratio $\frac{f_{KL}(\mu^*)}{f_{KL}(\mu_g)}$ for $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 2$ and $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 20$, respectively, where μ^* is the optimal sensor selection and μ_g is the greedy sensor selection.



(a) OPT vs. Greedy (b) Submodularity ratio

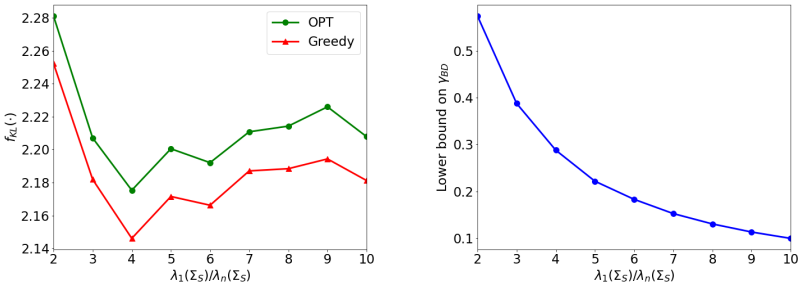


(c) $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 2$ (d) $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 20$

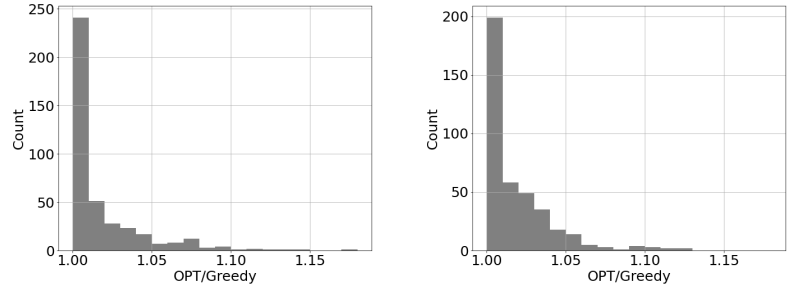
Fig. 5.2. Results for JDSS when $\sigma = 1$.

Specifically, Fig. 5.1(a) shows that the greedy algorithm performs near optimally for the instances of the KLDSS problem generated using the above procedure. As $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)}$ increases, we see from Fig. 5.1(b) that the lower bound on γ_{KL} tends to decrease, which is aligned with our analysis in Section 5.4. Note that since the cost vector satisfies $\max_{v\in\mathcal{V}} \omega_v = 8$, it follows from Theorem 5.3.2 (as argued in Section 5.3) that the greedy sensor selection μ_g satisfies $f_{KL}f(\mu_g) \geq (1 - e^{-0.6\gamma_{KL}})f_{KL}(\mu^*)$. For

instance, $\gamma_{KL} = 0.6$ yields $f_{KL}f(\mu_g) \geq 0.30 f_{KL}(\mu^*)$, i.e., $\frac{f_{KL}(\mu^*)}{f_{KL}(\mu_g)} \leq 3.31$. Finally, Fig. 5.1(c)-(d) showcase the performances of the greedy algorithm when applied to the 500 instances of KLDSS, for two different values of $\frac{\lambda_n(\Sigma_S)}{\lambda_1(\Sigma_S)}$. We observe that although the greedy algorithm tends to perform near optimally for $\frac{\lambda_n(\Sigma_S)}{\lambda_1(\Sigma_S)} = 2$ and $\frac{\lambda_n(\Sigma_S)}{\lambda_1(\Sigma_S)} = 20$, the overall performance of the greedy algorithm slightly worsens when $\frac{\lambda_n(\Sigma_S)}{\lambda_1(\Sigma_S)}$ becomes large. This observation reveals that the lower bound on γ_{KL} in Proposition 5.4.2 also provides guidance on how the performance of the greedy algorithm changes according to different problem parameters, as we discussed in Remark 5.4.4.



(a) OPT vs. Greedy (b) Submodularity ratio



(c) $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 2$ (d) $\frac{\lambda_1(\Sigma_S)}{\lambda_n(\Sigma_S)} = 10$

Fig. 5.3. Results for BDSS when $\sigma = 0.5$.

Similarly, we obtain Fig. 5.2 and Fig. 5.3 for the JDSS problem and the BDSS problem, respectively, where we set $\sigma = 0.5$ when generating the instances of BDSS. Similar analysis to that above for the KLDSS problem can be applied to the JDSS and BDSS problems. Putting Figs. 5.1-5.3 together, we observe that the lower bounds on γ_{KL} and γ_{JD} yield higher values for a wider range of $\frac{\lambda_n(\Sigma_S)}{\lambda_1(\Sigma_S)}$ and for a larger value of σ,

compared to the lower bound on γ_{BD}. In fact, as we can see from Fig. 5.3(c)-(d), the overall performance of the greedy algorithm gets worse when applied to the instances of the BDSS problem, compared to the performances of the greedy algorithm when applied to the KLDSS and JDSS instances.

5.6 Chapter Summary

In this chapter, we studied the hypothesis testing problem sensor selection problem under the Neyman-Pearson setting and the Bayesian setting. We first showed that the Neyman-Pearson hypothesis testing sensor selection problem and the Bayesian hypothesis testing sensor selection problem are NP-hard even when the measurement vector is Gaussian. Next, we studied the distance based sensor selection problem, and provided theoretical performance guarantees for the greedy algorithm by leveraging the notion of the submodularity ratio. Our analysis also extended the existing result on the performance guarantees for the greedy algorithm for maximizing submodular functions under budgeted constraints to nonsubmodular settings.

5.7 Proofs of Key Results

5.7.1 Proof of Theorem 5.2.2

We will show the NP-hardness of NPHSS and BHSS via reductions from the Subset Selection (SS) problem (e.g., [39]), which is known to be NP-hard (e.g., [38]).

Problem 5.7.1 *(SS) Consider a vector $b \in \mathbb{R}^n$, where $b_i \neq 0$ for all $i \in \{1, \ldots, n\}$, a matrix $C \in \mathbb{S}^n_{++}$, and $s \in \mathbb{Z}_{\geq 0}$. The Subset Selection (SS) problem is to find an indicator vector $\mu \in \{0,1\}^n$ that solves*

$$\max_{\mu \in \{0,1\}^m} \big(b(\mu)\big)^T \big(C(\mu)\big)^{-1} b(\mu)$$
$$s.t.\ |supp(\mu)| \leq s,$$

where $C(\mu)$ is the submatrix of C that contains the rows and columns corresponding to supp(μ) and $b(\mu) \triangleq \begin{bmatrix} b_{j_1} & \dots & b_{j_p} \end{bmatrix}^T$, where supp$(\mu) = \{j_1, \dots, j_p\} \subseteq \{1, \dots, n\}$.

We first show that NPHSS is NP-hard via a reduction from SS. Considering any instance of SS with $b \in \mathbb{R}^n$, $C \in \mathbb{S}^n_{++}$, and $s \in \mathbb{Z}_{\geq 0}$, where $b_i \neq 0$ for all $i \in \{1, \dots, n\}$, we construct an instance of NPHSS as follows. The measurement vector X is Gaussian distributed conditioned on H_i for $i = 0, 1$, i.e.,

$$H_0 : X \sim \mathcal{N}(\theta_0, C),$$
$$H_1 : X \sim \mathcal{N}(\theta_1, C),$$

where we set $\theta_0 = \mathbf{0}_n$ and $\theta_1 = b$. The cost vector is set as $\omega = \mathbf{1}_n$ and the budget is set as $\Omega = s$. The required false-alarm rate for the Neyman-Pearson detector is set as $\alpha = \frac{1}{2}$.

Considering any sensor selection $\mu \in \{0, 1\}^m$, we obtain from Eq. (5.2):

$$\log L(x(\mu)) = \big(\theta_1(\mu)\big)^T \big(C(\mu)\big)^{-1} x(\mu) - \frac{1}{2} \big(\theta_1(\mu)\big)^T \big(C(\mu)\big)^{-1} \theta_1(\mu). \tag{5.37}$$

Let $T(\mu) \triangleq \big(\theta_1(\mu)\big)^T \big(C(\mu)\big)^{-1} x(\mu) \in \mathbb{R}$, where the pdf of $T(\mu)$ conditioned on H_i for $i = 0, 1$, is given as

$$H_0 : T(\mu) \sim \mathcal{N}(0, \sigma(\mu)),$$
$$H_1 : T(\mu) \sim \mathcal{N}(\sigma(\mu), \sigma(\mu)),$$

where $\sigma(\mu) \triangleq \big(\theta_1(\mu)\big)^T \big(C(\mu)\big)^{-1} \theta_1(\mu) > 0$ for all $\mu \neq \mathbf{0}$, since $\big(C(\nu)\big)^{-1}$ is positive definite and $(\theta_1)_i \neq 0, \forall i$. We see from (5.3) and (5.37) that the Neyman-Pearson detector is of the form

$$T(\mu) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma'(\mu), \tag{5.38}$$

where $\gamma'(\mu) \triangleq \gamma(\mu) + \frac{1}{2}\sigma(\mu)$. We then know from [87] Case III.B.2 that $\gamma'(\mu)$ satisfies

$$\gamma'(\mu) = \sqrt{\sigma(\mu)} \Phi^{-1}(1 - \alpha) = \sqrt{\sigma(\mu)} \Phi^{-1}(\frac{1}{2}) = 0,$$

where $\Phi(\cdot)$ is the cumulative distribution function (cdf) of the standard normal distribution, and $\Phi^{-1}(\cdot)$ is the inverse of $\Phi(\cdot)$. The corresponding detection probability is given by

$$\begin{aligned} P_D(\mu) &= P(T(\mu) > \gamma'(\mu)|H_1) \\ &= 1 - \Phi\Big(\frac{\gamma'(\mu) - \sigma(\mu)}{\sqrt{\sigma(\mu)}}\Big) = 1 - \Phi\big(-\sqrt{\sigma(\mu)}\big) \\ &= \Phi\Big(\sqrt{\big(\theta_1(\mu)\big)^T \big(C(\mu)\big)^{-1} \theta_1(\mu)}\Big), \end{aligned} \tag{5.39}$$

where $P(T(\mu) > \gamma'(\mu)|H_1)$ is the conditional probability of $T(\mu) > \gamma'(\mu)$ given that H_1 is true. Noting that $\Phi(x)$ is monotonically nondecreasing on $x \in \mathbb{R}$, Eq. (5.39) then yields that in order to maximize $P_D(\mu)$ over sensor selections μ that satisfy the budget constraint, we have to maximize $\big(\theta_1(\mu)\big)^T \big(C(\mu)\big)^{-1} \theta_1(\mu)$. By our construction of the NPHSS instance, it follows that an indicator vector μ for SS is optimal if and only if μ is optimal for the corresponding NPHSS instance that we construct. Since SS is NP-hard, the NPHSS problem is also NP-hard. Using similar arguments, we can show that the BHSS problem is also NP-hard, which completes the proof of the theorem. ■

5.7.2 Proof of Proposition 5.4.1

We will use the following result.

Lemma 5.7.2 *Consider a random vector* $X = \begin{bmatrix} X_1 & \dots & X_n \end{bmatrix}^T \in \mathbb{R}^n$ *with covariance* $\Sigma \in \mathbb{S}^n_{++}$. *For all* $p < n$ ($p \in \mathbb{Z}_{>0}$), *consider an indicator vector* $\mu \in \{0,1\}^n$ *with* $supp(\mu) = \{1, \dots, p\}$. *Let* $X(\mu) = \begin{bmatrix} X_1 & \cdots & X_p \end{bmatrix}^T$ *and* $X(\mu^c) \triangleq \begin{bmatrix} X_{p+1} & \cdots & X_n \end{bmatrix}^T$, *where the covariances of* $X(\mu)$ *and* $X(\mu^c)$ *are denoted by* $\Sigma(\mu)$ *and* $\Sigma(\mu^c)$, *respectively. Partitioning* Σ *as*

$$\Sigma = \begin{bmatrix} \Sigma(\mu) & \Sigma_{X(\mu)X(\mu^c)} \\ \Sigma^T_{X(\mu)X(\mu^c)} & \Sigma(\mu^c) \end{bmatrix},$$

where $\Sigma_{X(\mu)X(\mu^c)} \triangleq Cov(X(\mu), X(\mu^c))$, *the following holds:*

$$\lambda_n(\Sigma) \leq \lambda_n\big(\Sigma(\mu) - \Sigma_{X(\mu)X(\mu^c)}(\Sigma(\mu^c))^{-1}\Sigma^T_{X(\mu)X(\mu^c)}\big).$$

Proof We first note that since $\Sigma \in \mathbb{S}^n_{++}$, $\Sigma(\mu)$ and $\Sigma(\mu^c)$ are positive definite for all $\mu \in \{0,1\}^n$ with $\text{supp}(\mu) = \{1,\ldots,p\}$, where $p < n$ ($p \in \mathbb{Z}_{>0}$). Denoting $\Sigma'(\mu) = \Sigma(\mu) - \Sigma_{X(\mu)X(\mu^c)}(\Sigma(\mu^c))^{-1}\Sigma^T_{X(\mu)X(\mu^c)}$, we have the following [91]:

$$\Sigma = \begin{bmatrix} I_p & \Sigma_{X(\mu)X(\mu^c)}(\Sigma(\mu^c))^{-1} \\ \mathbf{0} & I_{n-p} \end{bmatrix} \begin{bmatrix} \Sigma'(\mu) & \mathbf{0} \\ \mathbf{0} & \Sigma(\mu^c) \end{bmatrix} \begin{bmatrix} I_p & \mathbf{0} \\ (\Sigma(\mu^c))^{-1}\Sigma^T_{X(\mu)X(\mu^c)} & I_{n-p} \end{bmatrix}. \tag{5.40}$$

Let e_1 be an eigenvector of the eigenvalue $\lambda_n(\Sigma'(\mu)) > 0$, i.e., $\Sigma'(\mu)e_1 = \lambda_n(\Sigma'(\mu))e_1$, and define $e_0 = \begin{bmatrix} e_1 \\ -(\Sigma(\mu^c))^{-1}\Sigma^T_{X(\mu)X(\mu^c)}e_1 \end{bmatrix}$. We have from Eq. (5.40) the following:

$$\begin{aligned} \Sigma e_0 &= \begin{bmatrix} \Sigma'(\mu) & \Sigma_{X(\mu)X(\mu^c)} \\ \mathbf{0} & \Sigma(\mu^c) \end{bmatrix} \begin{bmatrix} I_p & \mathbf{0} \\ (\Sigma(\mu^c))^{-1}\Sigma^T_{X(\mu)X(\mu^c)} & I_{n-p} \end{bmatrix} \begin{bmatrix} e_1 \\ -(\Sigma(\mu^c))^{-1}\Sigma^T_{X(\mu)X(\mu^c)}e_1 \end{bmatrix} \\ &= \begin{bmatrix} \Sigma'(\mu) & \Sigma_{X(\mu)X(\mu^c)} \\ \mathbf{0} & \Sigma(\mu^c) \end{bmatrix} \begin{bmatrix} e_1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \Sigma'(\mu)e_1 \\ \mathbf{0} \end{bmatrix}. \end{aligned} \tag{5.41}$$

Using (5.41), we have

$$\begin{aligned} e_0^T\lambda_n(\Sigma)e_0 \le e_0^T\Sigma e_0 &= \begin{bmatrix} e_1^T & -e_1^T\Sigma_{X(\mu)X(\mu^c)}(\Sigma(\mu^c))^{-1} \end{bmatrix} \begin{bmatrix} \Sigma'(\mu)e_1 \\ \mathbf{0} \end{bmatrix} \\ &= e_1^T\Sigma'(\mu)e_1 = e_1^T\lambda_n(\Sigma'(\mu))e_1, \end{aligned} \tag{5.42}$$

where the first inequality follows from $\Sigma \succeq \lambda_n(\Sigma)I_n$ [91]. Noting the definition of e_0, we have $e_0^Te_0 \ge e_1^Te_1 > 0$. It then follows from (5.42) that $\lambda_n(\Sigma) \le \lambda_n\big(\Sigma(\mu) - \Sigma_{X(\mu)X(\mu^c)}(\Sigma(\mu))^{-1}\Sigma^T_{X(\mu)X(\mu^c)}\big)$. ■

Proof of Proposition 5.4.1: Following Definition 5.3.1, let us consider any subsets $A, B \subseteq \mathcal{V}$ and any $v \in A \setminus B$, where $A \setminus B \neq \emptyset$. First, suppose $B \neq \emptyset$. Denote $\tilde{A} \triangleq A \setminus B$ and let $\tilde{A} = \{v_1,\ldots,v_{|\tilde{A}|}\}$. For all $i \in \{1,\ldots,|\tilde{A}|\}$, let $y_i \triangleq \big(\Sigma_c(v_i) - \Sigma^T_{Bv_i}Z\Sigma_{Bv_i}\big)^{-1} \in \mathbb{R}_{>0}$, where $\Sigma_{Bv_i} \triangleq \text{Cov}(X(B), X(v_i)) \in \mathbb{R}^{|B|}$, and $Z \triangleq (\Sigma_c(B))^{-1}$.

Note that the measurement vector $X \in \mathbb{R}^n$ satisfies (5.20). Partitioning $\Sigma_c(\{v_i\} \cup B)$ as $\Sigma_c(\{v_i\} \cup B) = \left[\begin{smallmatrix} \Sigma_c(v_i) & \Sigma_{Bv_i}^T \\ \Sigma_{Bv_i} & \Sigma_c(B) \end{smallmatrix}\right]$, we have from Eq. (5.10) the following:

$$\begin{aligned}
&2f_{KL}(\{v_i\} \cup B) \\
=&\big[\theta_1(v_i)\ (\theta_1(B))^T\big]\big(\Sigma_c(\{v_i\} \cup B)\big)^{-1} \begin{bmatrix} \theta_1(v_i) \\ \theta_1(B) \end{bmatrix} \\
=&\big[\theta_1(v_i)\ (\theta_1(B))^T\big] \begin{bmatrix} y_i & -y_i\Sigma_{Bv_i}^T Z \\ -Z\Sigma_{Bv_i}y_i & Z + Z\Sigma_{Bv_i}y_i\Sigma_{Bv_i}^T Z \end{bmatrix} \begin{bmatrix} \theta_1(v_i) \\ \theta_1(B) \end{bmatrix} \qquad (5.43) \\
=&(\theta_1(B))^T Z\theta_1(B) + (\theta_1(B))^T Z\Sigma_{Bv_i}y_i\Sigma_{Bv_i}^T Z\theta_1(B) \\
&\qquad - 2\theta_1(v_i)y_i\Sigma_{Bv_i}^T Z\theta_1(B) + (\theta_1(v_i))^2 y_i \\
=&(\theta_1(B))^T Z\theta_1(B) + \big((\theta_1(B))^T Z\Sigma_{Bv_i} - \theta_1(v_i)\big)^2 y_i, \qquad (5.44)
\end{aligned}$$

where (5.43) uses the inverse formula for block matrices [91]. (5.44) implies that $f_{KL}(\{v\} \cup B) - f_{KL}(B) \geq 0$ for all $B \subseteq \mathcal{V}$ ($B \neq \emptyset$ and $B \neq \mathcal{V}$) and for all $v \in \mathcal{V} \setminus B$. Since $f_{KL}(B) \geq f_{KL}(\emptyset) = 0$ for all $B \subseteq \mathcal{V}$, it follows that $f_{KL}(\cdot)$ is monotone nondecreasing. Moreover, we have from (5.44) the following:

$$2\sum_{v \in \tilde{A}} \big(f_{KL}(\{v\} \cup B) - f_{KL}(B)\big) = \begin{bmatrix} m_1 & \cdots & m_{|\tilde{A}|} \end{bmatrix} \mathrm{diag}(y_1, \ldots, y_{|\tilde{A}|}) \begin{bmatrix} m_1 \\ \vdots \\ m_{|\tilde{A}|} \end{bmatrix}, \qquad (5.45)$$

where $m_i \triangleq (\theta_1(B))^T Z\Sigma_{Bv_i} - \theta_1(v_i)$ for all $i \in \{1, \ldots, |\tilde{A}|\}$. Noting that $y_i = \big(\Sigma_c(v_i) - \Sigma_{Bv_i}^T Z\Sigma_{Bv_i}\big)^{-1} \geq (\Sigma_c(v_i))^{-1} \geq \frac{1}{d_1(\Sigma_c)}$ for all $i \in \{1, \ldots, |\tilde{A}|\}$, we then see from (5.45) that

$$2\sum_{v \in \tilde{A}} \big(f_{KL}(\{v\} \cup B) - f_{KL}(B)\big) \geq \frac{1}{d_1(\Sigma_c)} \begin{bmatrix} m_1 & \cdots & m_{|\tilde{A}|} \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_{|\tilde{A}|} \end{bmatrix}. \qquad (5.46)$$

Using similar arguments to those above, we can partition $\Sigma_c(A\cup B)$ as $\Sigma_c(A\cup B) = \left[\begin{smallmatrix}\Sigma_c(\tilde{A}) & \Sigma_{B\tilde{A}}^T\\ \Sigma_{B\tilde{A}} & \Sigma_c(B)\end{smallmatrix}\right]$, where $\Sigma_{B\tilde{A}} \triangleq \mathrm{Cov}(X(B), X(\tilde{A})) \in \mathbb{R}^{|B|\times|\tilde{A}|}$, and obtain from Eq. (5.10) the following:

$$2\big(f_{KL}(A\cup B) - f_{KL}(B)\big) = \big((\theta_1(B))^T Z\Sigma_{B\tilde{A}} - (\theta_1(\tilde{A}))^T\big)Y(\Sigma_{B\tilde{A}}^T Z\theta_1(B) - \theta_1(\tilde{A})), \tag{5.47}$$

where $Y \triangleq \big(\Sigma_c(\tilde{A}) - \Sigma_{B\tilde{A}}^T Z\Sigma_{B\tilde{A}}\big)^{-1} \in \mathbb{R}^{|\tilde{A}|\times|\tilde{A}|}$. Noting that $\begin{bmatrix} m_1 & \cdots & m_{|\tilde{A}|}\end{bmatrix} = (\theta_1(B))^T Z\Sigma_{B\tilde{A}} - (\theta_1(\tilde{A}))^T$, and using the fact that $Y \preceq \lambda_1(Y) I_{|\tilde{A}|}$ [91], where $\lambda_1(Y) = 1/\lambda_n\big(\Sigma_c(\tilde{A}) - \Sigma_{B\tilde{A}}^T Z\Sigma_{B\tilde{A}}\big)$, we see from Eq. (5.47) that

$$2\big(f_{KL}(A\cup B) - f_{KL}(B)\big) \leq \frac{1}{\lambda_n(\Sigma_c)} \begin{bmatrix} m_1 & \cdots & m_{|\tilde{A}|}\end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_{|\tilde{A}|}\end{bmatrix}. \tag{5.48}$$

To obtain (5.48), we use the following chain of inequalities:

$$\lambda_n\big(\Sigma_c(\tilde{A}) - \Sigma_{B\tilde{A}}^T Z\Sigma_{B\tilde{A}}\big) \geq \lambda_n(\Sigma_c(A\cup B)) \geq \lambda_n(\Sigma_c), \tag{5.49}$$

where the first inequality follows from Lemma 5.7.2, and the second inequality follows from the Cauchy interlacing theorem for positive definite matrices [91]. Combining (5.46) and (5.48) yields $\gamma_{KL} \geq \frac{\lambda_n(\Sigma_c)}{d_1(\Sigma_c)}$.[2] Next, supposing $B = \emptyset$, one can show using similar arguments to those above that $\gamma_{KL} \geq \frac{\lambda_n(\Sigma_c)}{d_1(\Sigma_c)}$. Moreover, using the same arguments as above, we see that $f_{JD}(\cdot)$ and $f_{BD}(\cdot)$ are monotone nondecreasing, and $\gamma_{JD} \geq \frac{\lambda_n(\Sigma_c)}{d_1(\Sigma_c)}$ and $\gamma_{BD} \geq \frac{\lambda_n(\Sigma_c)}{d_1(\Sigma_c)}$ hold. ■

5.7.3 Proof of Proposition 5.4.2

Note that $f_{KL}(B) \geq f_{KL}(\emptyset) = 0$, $\forall B \subseteq \mathcal{V}$. To prove that $f_{KL}(\cdot)$ is monotone nondecreasing, it is then sufficient to show that for all $B \subseteq \mathcal{V}$ ($B \neq \emptyset$ and $B \neq \mathcal{V}$) and for all $v \in \mathcal{V}\setminus B$, $f_{KL}(\{v\}\cup B) - f_{KL}(B) \geq 0$ holds. Denote $\bar{B} \triangleq \{v\}\cup B$, and

[2] Note that if $\begin{bmatrix} m_1 & \cdots & m_{|\tilde{A}|}\end{bmatrix}\begin{bmatrix} m_1 & \cdots & m_{|\tilde{A}|}\end{bmatrix}^T = 0$, (5.13) in Definition 5.3.1 naturally holds for such $A, B \subseteq \mathcal{V}$.

let $\Sigma_{Bv} \triangleq \mathrm{Cov}(X(B), X(v)) \in \mathbb{R}^{|B|}$ be the cross-covariance of $X(B)$ and $X(v)$ under hypothesis H_1. Noting that the measurement vector $X \in \mathbb{R}^n$ satisfies (5.21) and that $\Sigma_0 \in \mathbb{S}^n_{++}$ is diagonal from Assumption 5.4.1, we have from Eq. (5.6) the following:

$$
\begin{aligned}
&2\big(f_{KL}(\bar{B}) - f_{KL}(B)\big) \\
&= \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \log\frac{\det(\Sigma_1(B))}{\det(\Sigma_1(\bar{B}))} + \log\Sigma_0(v) - 1 \\
&= \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \log\Sigma_0(v) - \log\big(\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}\big) - 1 \qquad (5.50) \\
&\geq \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \log\Sigma_0(v) - \log\Sigma_1(v) - 1 \qquad (5.51) \\
&= \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - \log\frac{\Sigma_1(v)}{\Sigma_0(v)} - 1 \geq 0, \qquad (5.52)
\end{aligned}
$$

To obtain (5.50) we use the following identity [91]:

$$
\begin{aligned}
\det(\Sigma_1(\bar{B})) &= \det\begin{bmatrix} \Sigma_1(v) & \Sigma_{Bv}^T \\ \Sigma_{Bv} & \Sigma_1(B) \end{bmatrix} \\
&= \det(\Sigma_1(B))\det\big(\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}\big). \qquad (5.53)
\end{aligned}
$$

where $\big(\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}\big) \in \mathbb{R}_{>0}$. To obtain (5.51), we use the fact $\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}^T \leq \Sigma_1(v)$. Noting that the function $h(x) \triangleq x - \log x$ achieves its unique minimum on $x > 0$ at $x = 1$ with $h(1) = 1$, i.e., $\frac{\Sigma_1(v)}{\Sigma_0(v)} - \log\frac{\Sigma_1(v)}{\Sigma_0(v)} \geq 1$, the inequality in (5.52) then follows, which proves that $f_{KL}(\cdot)$ is monotone nondecreasing.

We now bound the submodularity ratio γ_{KL}. Let us consider any $A, B \subseteq \mathcal{V}$, where $A \setminus B \neq \emptyset$. First, suppose $B \neq \emptyset$. We begin by providing a lower bound on $\sum_{v\in A\setminus B}\big(f_{KL}(\{v\}\cup B) - f_{KL}(B)\big)$. Following (5.51), we obtain:

$$
2\sum_{v\in A\setminus B}\big(f_{KL}(\{v\}\cup B) - f_{KL}(B)\big) \geq \sum_{v\in A\setminus B}\Big(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - \log\frac{\Sigma_1(v)}{\Sigma_0(v)} - 1\Big). \qquad (5.54)
$$

We then give an upper bound on $\big(f_{KL}(A\cup B)-f_{KL}(B)\big)$. Denote $\tilde{B}\triangleq A\cup B$ and $\tilde{A}\triangleq A\setminus B$. Let $\Sigma_{B\tilde{A}}\triangleq \mathrm{Cov}(X(B),X(\tilde{A}))\in\mathbb{R}^{|B|\times|\tilde{A}|}$ be the cross-covariance of $X(B)$ and $X(\tilde{A})$ under hypothesis H_1. Similarly, we have from Eq. (5.6) the following:

$$
\begin{aligned}
&2\big(f_{KL}(\tilde{B})-f_{KL}(B)\big)\\
&=\Big\{\sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)}+\frac{\Sigma_1(v)}{\Sigma_0(v)}-1\big)\Big\}+\log\frac{\big(\prod_{v\in\tilde{A}}\Sigma_0(v)\big)\det(\Sigma_1(B))}{\det(\Sigma_1(\tilde{B}))}\\
&=\Big\{\sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)}+\frac{\Sigma_1(v)}{\Sigma_0(v)}-1\big)\Big\}+\log\frac{\prod_{v\in\tilde{A}}\Sigma_0(v)}{\det\big(\Sigma_1(\tilde{A})-\Sigma_{B\tilde{A}}^T(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\big)} \qquad (5.55)\\
&\le\Big\{\sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)}+\frac{\Sigma_1(v)}{\Sigma_0(v)}-1\big)\Big\}+\log\frac{\prod_{v\in\tilde{A}}\Sigma_0(v)}{(\lambda_n(\Sigma_1))^{|\tilde{A}|}} \qquad (5.56)\\
&=\Big\{\sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)}+\frac{\Sigma_1(v)}{\Sigma_0(v)}-\log\frac{\Sigma_1(v)}{\Sigma_0(v)}-1\big)\Big\}+\log\frac{\prod_{v\in\tilde{A}}\Sigma_1(v)}{(\lambda_n(\Sigma_1))^{|\tilde{A}|}},\\
&\le\Big\{\sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)}+\frac{\Sigma_1(v)}{\Sigma_0(v)}-\log\frac{\Sigma_1(v)}{\Sigma_0(v)}-1\big)\Big\}+|\tilde{A}|\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)}, \qquad (5.57)
\end{aligned}
$$

where (5.55) uses the following identity [91]:

$$
\begin{aligned}
\det(\Sigma_1(\tilde{B}))&=\det\begin{bmatrix}\Sigma_1(\tilde{A}) & \Sigma_{B\tilde{A}}^T\\ \Sigma_{B\tilde{A}} & \Sigma_1(B)\end{bmatrix}\\
&=\det(\Sigma_1(B))\det\big(\Sigma_1(\tilde{A})-\Sigma_{B\tilde{A}}^T(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\big). \qquad (5.58)
\end{aligned}
$$

For (5.56), we use the following chain of inequalities:

$$
\lambda_n\big(\Sigma_1(\tilde{A})-\Sigma_{B\tilde{A}}^T(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\big)\ge\lambda_n(\Sigma_1(\tilde{B}))\ge\lambda_n(\Sigma_1), \qquad (5.59)
$$

where the first inequality follows from Lemma 5.7.2, and the second inequality follows from the Cauchy interlacing theorem for positive definite matrices [91].

Combining (5.54) and (5.57), we obtain from Definition 5.3.1

$$
\gamma_{KL}\ge\frac{\sum_{v\in\tilde{A}}C_{KL}(v)}{|\tilde{A}|\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)}+\sum_{v\in\tilde{A}}C_{KL}(v)}, \qquad (5.60)
$$

where $C_{KL}(v)$ is defined in Eq. (5.24). Noting that $\frac{\Sigma_1(v)}{\Sigma_0(v)}-\log\frac{\Sigma_1(v)}{\Sigma_0(v)}\ge 1$ for all $v\in\mathcal{V}$ as argued above, we have $C_{KL}(v)\ge 0$ for all $v\in\mathcal{V}$. Given any $A,B\subseteq\mathcal{V}$, one

can view the lower bound obtained in (5.60) as a function of $\sum_{v\in\tilde{A}} C_{KL}(v)$, where $|\tilde{A}|\min_{v\in\mathcal{V}} C_{KL}(v) \leq \sum_{v\in\tilde{A}} C_{KL}(v) \leq |\tilde{A}|\max_{v\in\mathcal{V}} C_{KL}(v)$. Moreover, noting that $\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} \geq 0$ (since $\Sigma_1 - \lambda_n(\Sigma_1)I_n \succeq \mathbf{0}$ [91]), one can then show that the lower bound in (5.60) (as a function of $\sum_{v\in\tilde{A}} C_{KL}(v)$) is monotone nondecreasing, which implies

$$\begin{aligned}\gamma_{KL} &\geq \frac{|\tilde{A}|\min_{v\in\mathcal{V}} C_{KL}(v)}{|\tilde{A}|\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + |\tilde{A}|\min_{v\in\mathcal{V}} C_{KL}(v)} \\ &\geq \frac{\min_{v\in\mathcal{V}} C_{KL}(v)}{\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + \min_{v\in\mathcal{V}} C_{KL}(v)}. \end{aligned} \tag{5.61}$$

Next, supposing $B = \emptyset$, one can show, using similar arguments to those above, that (5.61) holds. This completes the proof of the proposition. ■

5.7.4 Proof of Proposition 5.4.3

Noting that the measurement vector $X \in \mathbb{R}^n$ satisfies (5.21), we see from Eq. (5.7) that for all $\mu \in \{0,1\}^n$, $f_{JD}(\cdot)$ can be split into two terms:

$$f_{JD}(\mu) = f^0_{JD}(\mu) + f^1_{JD}(\mu), \tag{5.62}$$

where

$$f^0_{JD}(\mu) \triangleq \frac{1}{2}(\theta_1(\mu))^T(\Sigma_1(\mu))^{-1}\theta_1(\mu), \tag{5.63}$$

and

$$\begin{aligned} f^1_{JD}(\mu) \triangleq \frac{1}{2}\mathrm{tr}\big((\Sigma_0(\mu))^{-1}\Sigma_1(\mu)\big) + \frac{1}{2}\mathrm{tr}\big((\Sigma_1(\mu))^{-1}\Sigma_0(\mu)\big) \\ + \frac{1}{2}(\theta_1(\mu))^T(\Sigma_0(\mu))^{-1}\theta_1(\mu) - |\mathrm{supp}(\mu)|. \end{aligned} \tag{5.64}$$

Denote the submodularity ratios of $f^0_{JD}(\cdot)$ and $f^1_{JD}(\cdot)$ as $\gamma^0_{JD} \in \mathbb{R}_{\geq 0}$ and $\gamma^1_{JD} \in \mathbb{R}_{\geq 0}$, respectively. Noting the form of $f^0_{JD}(\cdot)$ given in Eq. (5.63), we see from Proposition 5.4.1 that $f^0_{JD}(\cdot)$ is monotone nondecreasing with $r^0_{JD} \geq \frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}$. Moreover, one can show that the sum of monotone nondecreasing functions is monotone nondecreasing, and the submodularity ratio of $f_{JD}(\cdot)$ satisfies $\gamma_{JD} = \min\{\gamma^0_{JD}, \gamma^1_{JD}\}$. Thus, we aim to show that $f^1_{JD}(\cdot)$ is also monotone nondecreasing and lower bound r^1_{JD}.

Similarly to the proof of Proposition 5.4.2, we first show that $f^1_{JD}(\cdot)$ is monotone nondecreasing, i.e., for all $B \subseteq \mathcal{V}$ ($B \neq \emptyset$ and $B \neq \mathcal{V}$) and for all $v \in \mathcal{V}\setminus B$, $f^1_{JD}(\{v\}\cup B) - f^1_{JD}(B) \geq 0$ holds. Denote $\bar{B} \triangleq \{v\}\cup B$, and let $\Sigma_{Bv} \triangleq \mathrm{Cov}(X(B), X(v)) \in \mathbb{R}^{|B|}$ be the cross-covariance of $X(B)$ and $X(v)$ under hypothesis H_1. Noting that $\Sigma_0 \in \mathbb{S}^n_{++}$ is diagonal from Assumption 5.4.1, we have from Eq. (5.64) the following:

$$\begin{aligned}&2\big(f^1_{JD}(\bar{B}) - f^1_{JD}(B)\big)\\ &= \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - 2 + \mathrm{tr}\big((\Sigma_1(\bar{B}))^{-1}\Sigma_0(\bar{B})\big) - \mathrm{tr}\big((\Sigma_1(B))^{-1}\Sigma_0(B)\big). \qquad (5.65)\end{aligned}$$

Note that $\Sigma_1(\bar{B})$ and $\Sigma_0(\bar{B})$ can be partitioned as $\Sigma_1(\bar{B}) = \left[\begin{smallmatrix}\Sigma_1(v) & \Sigma_{Bv}^T\\ \Sigma_{Bv} & \Sigma_1(B)\end{smallmatrix}\right]$ and $\Sigma_0(\bar{B}) = \left[\begin{smallmatrix}\Sigma_0(v) & \mathbf{0}\\ \mathbf{0} & \Sigma_0(B)\end{smallmatrix}\right]$, respectively, which implies via the inverse formula for block matrices [91] that

$$(\Sigma_1(\bar{B}))^{-1}\Sigma_0(\bar{B}) = \begin{bmatrix}(M_1(B,v))^{-1}\Sigma_0(v) & *\\ * & ((\Sigma_1(B))^{-1}\Sigma_0(B) + (\Sigma_1(\bar{B}))^{-1}_B\Sigma_0(B)\end{bmatrix}, \qquad (5.66)$$

where

$$((\Sigma_1(\bar{B}))^{-1}_B \triangleq (\Sigma_1(B))^{-1}\Sigma_{Bv}(M_1(B,v))^{-1}\Sigma_{Bv}^T(\Sigma_1(B))^{-1},$$

and $M_1(B,v) \triangleq \Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}$. It then follows from Eqs. (5.65)-(5.66) that

$$\begin{aligned}2\big(f^1_{JD}(\bar{B}) - f^1_{JD}(B)\big) &= \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - 2 + (M_1(B,v))^{-1}\Sigma_0(v)\\ &\quad + \mathrm{tr}\big((\Sigma_1(B))^{-1}\Sigma_{Bv}(M_1(B,v))^{-1}\Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_0(B)\big)\\ &\geq \frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - 2 + \frac{\Sigma_0(v)}{\Sigma_1(v)}\\ &\quad + \mathrm{tr}\big(\Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_0(B)(\Sigma_1(B))^{-1}\Sigma_{Bv}(M_1(B,v))^{-1}\big), \qquad (5.67)\end{aligned}$$

where (5.67) uses $M_1(B,v) \leq \Sigma_1(v)$ and the cyclic property of trace. Since the function $g(x) \triangleq x + \frac{1}{x}$ achieves its unique minimum on $x > 0$ at $x = 1$ with $g(1) = 2$, we have $\frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} \geq 2$, which implies via (5.67) that $f^1_{JD}(\{v\}\cup B) - f^1_{JD}(B) \geq 0$. Hence, we conclude that $f^1_{JD}(\cdot)$ is monotone nondecreasing.

We now give a lower bound on γ^1_{JD}. Let us consider any $A, B \subseteq \mathcal{V}$, where $A\setminus B \neq \emptyset$. First, suppose $B \neq \emptyset$. We begin with lower bounding $\sum_{v\in A\setminus B}\big(f^1_{JD}(\{v\}\cup B) -$

$f^1_{JD}(B)\big)$. Denote $\tilde{A} \triangleq A\backslash B$, and let $\Sigma_{B\tilde{A}} \triangleq \mathrm{Cov}(X(B), X(\tilde{A})) \in \mathbb{R}^{|B|\times|\tilde{A}|}$ be the cross-covariance of $X(B)$ and $X(\tilde{A})$ under hypothesis H_1. Continuing with the arguments leading to (5.67), we have the following:

$$
\begin{aligned}
&2\sum_{v\in\tilde{A}}\big(f^1_{JD}(\{v\}\cup B) - f^1_{JD}(B)\big)\\
&\geq \sum_{v\in\tilde{A}}\Big(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2\\
&\qquad\qquad + \frac{1}{d_1(\Sigma_1)}\mathrm{tr}\big(\Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_0(B)(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)\Big) \qquad (5.68)\\
&= \Big\{\sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2\big)\Big\}\\
&\qquad\qquad + \frac{1}{d_1(\Sigma_1)}\mathrm{tr}\big(\Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_0(B)(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\big), \qquad (5.69)
\end{aligned}
$$

where (5.68) follows from $M_1(B,v) \leq \Sigma_1(v) \leq d_1(\Sigma_1)$ for all $v \in \mathcal{V}$. To obtain (5.69), we note that $\Sigma_{B\tilde{A}} = \big[\begin{smallmatrix}\Sigma_{Bv_1} & \cdots & \Sigma_{Bv_{|\tilde{A}|}}\end{smallmatrix}\big]$, where $\Sigma_{Bv_i} = \mathrm{Cov}(X(B), X(v_i))$ under hypothesis H_1 for all $v_i \in \tilde{A} = \{v_1, \ldots, v_{|\tilde{A}|}\}$. Next, we upper bound $\big(f^1_{JD}(A\cup B) - f^1_{JD}(B)\big)$. Denote $\tilde{B} \triangleq A \cup B$, and let $M_1(B,\tilde{A}) \triangleq \Sigma_1(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}$. Similarly, we obtain from Eq. (5.64) the following:

$$
\begin{aligned}
&2\big(f^1_{JD}(\tilde{B}) - f^1_{JD}(B)\big)\\
&= \sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - 2\big) + \mathrm{tr}\big((\Sigma_1(\tilde{B}))^{-1}\Sigma_0(\tilde{B})\big) - \mathrm{tr}\big((\Sigma_1(B))^{-1}\Sigma_0(B)\big)\\
&= \sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - 2\big) + \mathrm{tr}\big((M_1(B,\tilde{A}))^{-1}\Sigma_0(\tilde{A})\big)\\
&\qquad\qquad + \mathrm{tr}\big((\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}(M_1(B,\tilde{A}))^{-1}\Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_0(B)\big) \qquad (5.70)\\
&= \sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - 2\big) + \mathrm{tr}\big(\Sigma_0(\tilde{A})(M_1(B,\tilde{A}))^{-1}\big)\\
&\qquad\qquad + \mathrm{tr}\big(\Sigma^T_{B\tilde{A}}(\Sigma_0(B))^{-1}\Sigma_0(B)(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}(M_1(B,\tilde{A}))^{-1}\big) \qquad (5.71)\\
&\leq \sum_{v\in\tilde{A}}\big(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} - 2\big) + \frac{1}{\lambda_n(\Sigma_1)}\mathrm{tr}(\Sigma_0(B))\\
&\qquad\qquad + \frac{1}{\lambda_n(\Sigma_1)}\mathrm{tr}\big(\Sigma^T_{B\tilde{A}}(\Sigma_0(B))^{-1}\Sigma_0(B)(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\big), \qquad (5.72)
\end{aligned}
$$

To obtain (5.70), we note from similar arguments to those for (5.66) that

$$(\Sigma_1(\tilde{B})^{-1}\Sigma_0(\tilde{B}) = \begin{bmatrix} (M_1(B,\tilde{A}))^{-1}\Sigma_0(\tilde{A}) & * \\ * & ((\Sigma_1(B))^{-1}\Sigma_0(B)+(\Sigma_1(\tilde{B}))_B^{-1}\Sigma_0(B) \end{bmatrix},$$

where

$$((\Sigma_1(\tilde{B}))_B^{-1} \triangleq (\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}(M_1(B,\tilde{A}))^{-1}\Sigma_{B\tilde{A}}^T(\Sigma_1(B))^{-1}.$$

We obtain (5.71) from the cyclic property of trace. For (5.72), we first obtain $\lambda_n(M_1(B,\tilde{A})) \geq \lambda_n(\Sigma_1)$ from a similar chain of inequalities to that in (5.59), which implies $\lambda_1((M_1(B,\tilde{A}))^{-1}) \leq \frac{1}{\lambda_n(\Sigma_1)}$. Then, we use the trace inequality $\mathrm{tr}(P_1P_2) \leq \lambda_1(P_2)\mathrm{tr}(P_1)$, $\forall P_1, P_2 \in \mathbb{S}_+^n$ [92], which implies (5.72).

Combining (5.69) and (5.72), we have from Definition 5.3.1 the following:

$$\gamma_{JD}^1 \geq \frac{\frac{D_{JD}^1(A,B)}{d_1(\Sigma_1)} + \sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2\right)}{\frac{D_{JD}^1(A,B)}{\lambda_n(\Sigma_1)} + \sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)} - 2\right)}, \tag{5.73}$$

where $D_{JD}^1(A,B) \in \mathbb{R}_{\geq 0}$ is defined as

$$D_{JD}^1(A,B) \triangleq \mathrm{tr}\big(\Sigma_{B\tilde{A}}^T(\Sigma_1(B))^{-1}\Sigma_0(B)(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\big),$$

for all $A, B \in \mathcal{V}$. Since $\frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} \geq 2$ as argued above, it implies $\frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)} \geq \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} \geq 2$, where the first inequality follows from $\Sigma_1 - \lambda_n(\Sigma_1)I_n \succeq \mathbf{0}$ [91]. Similarly, we have $\frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)} \leq 1$. Supposing $D_{JD}^1(A,B) > 0$, one can then obtain from (5.73)

$$\gamma_{JD}^1 \geq \min\Big\{\frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}, \frac{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2\right)}{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)} - 2\right)}\Big\}. \tag{5.74}$$

Note that if $D_{JD}^1(A,B) = 0$, we have $\gamma_J^1 \geq \frac{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2\right)}{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)} - 2\right)}$, which also implies (5.74). Moreover, using similar arguments to those for (5.74), one can show that

$$\frac{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2\right)}{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)} - 2\right)} \geq \min_{v\in\tilde{A}} \frac{\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\Sigma_1(v)} - 2}{\frac{(\theta_1(v))^2}{\Sigma_0(v)} + \frac{\Sigma_1(v)}{\Sigma_0(v)} + \frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)} - 2} \geq C_{JD}, \tag{5.75}$$

where C_{JD} is defined by Eq. (5.26). Combining (5.74)-(5.75), we obtain

$$\gamma_{JD}^1 \geq \min\{C_{JD}, \frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}\}. \tag{5.76}$$

Next, suppose $B = \emptyset$. Using similar arguments to those above, one can show that $\gamma^1_{JD} \geq \frac{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)}+\frac{\Sigma_1(v)}{\Sigma_0(v)}+\frac{\Sigma_0(v)}{\Sigma_1(v)}-2\right)}{\sum_{v\in\tilde{A}}\left(\frac{(\theta_1(v))^2}{\Sigma_0(v)}+\frac{\Sigma_1(v)}{\Sigma_0(v)}+\frac{\Sigma_0(v)}{\lambda_n(\Sigma_1)}-2\right)} \geq C_{JD}$, which also leads to (5.76). Recalling $\gamma^0_{JD} \geq \frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}$, (5.25) follows from $\gamma_{JD} = \min\{\gamma^0_{JD}, \gamma^1_{JD}\}$, completing the proof of the proposition. ■

5.7.5 Proof of Proposition 5.4.4

Proof of (a): Noting that the measurement vector $X \in \mathbb{R}^n$ satisfies (5.21), we see from Eq. (5.8) that for all $\mu \in \{0,1\}^n$, $f_{BD}(\mu)$ can be written as

$$f_{BD}(\mu) = f^0_{BD}(\mu) + f^1_{BD}(\mu), \tag{5.77}$$

where

$$f^0_{BD}(\mu) \triangleq \frac{1}{4}(\theta_1(\mu))^T(\Sigma_0(\mu)+\Sigma_1(\mu))^{-1}\theta_1(\mu), \tag{5.78}$$

and

$$f^1_{BD}(\mu) \triangleq \frac{1}{2}\log\frac{\det(\frac{1}{2}\Sigma_0(\mu)+\frac{1}{2}\Sigma_1(\mu))}{\sqrt{\det(\Sigma_0(\mu))\det(\Sigma_1(\mu))}}. \tag{5.79}$$

Denote the submodularity ratios of $f^0_{BD}(\cdot)$ and $f^1_{BD}(\cdot)$ as $\gamma^0_{BD} \in \mathbb{R}_{\geq 0}$ and $\gamma^1_{BD} \in \mathbb{R}_{\geq 0}$, respectively. Similarly to the arguments in the proof of Proposition 5.4.3, we note from Proposition 5.4.1 that $f^0_{BD}(\cdot)$ is monotone nondecreasing with $\gamma^0_{BD} \geq \frac{\lambda_n(\Sigma_1)}{d_1(\Sigma_1)}$. Moreover, the submodularity ratio of $f_{BD}(\cdot)$ satisfies $\gamma_{BD} = \min\{\gamma^0_{BD}, \gamma^1_{BD}\}$. We thus focus on proving $f^1_{BD}(\cdot)$ is monotone nondecreasing and lower bounding γ^1_{BD}.

We first show that $f^1_{BD}(\cdot)$ is monotone nondecreasing, i.e., for all $B \subseteq \mathcal{V}$ ($B \neq \emptyset$ and $B \neq \mathcal{V}$) and for all $v \in \mathcal{V} \setminus B$, $f^1_{BD}(\{v\} \cup B) - f^1_{BD}(B) \geq 0$ holds. Let $\Sigma_{Bv} \triangleq \mathrm{Cov}(X(B), X(v)) \in \mathbb{R}^{|B|}$ be the cross-covariance of $X(B)$ and $X(v)$ under hypothesis H_1, and let $\Sigma(B) \triangleq \frac{1}{2}(\Sigma_0(B)+\Sigma_1(B))$, where we note that $\Sigma(\{v\}\cup B) = \begin{bmatrix} \Sigma(v) & \frac{1}{2}\Sigma^T_{Bv} \\ \frac{1}{2}\Sigma_{Bv} & \Sigma(B) \end{bmatrix}$

since $\Sigma_0 \in \mathbb{S}^n_{++}$ is diagonal from Assumption 5.4.1. Denoting $\bar{B} \triangleq \{v\} \cup B$, we then have from Eq. (5.79) the following:

$$\begin{aligned}
&4\big(f^1_{BD}(\bar{B}) - f^1_{BD}(B)\big) \\
&= 2\log\frac{\det(\Sigma(\bar{B}))}{\det(\Sigma(B))} + \log\frac{\det(\Sigma_0(B))\det(\Sigma_1(B))}{\det(\Sigma_0(\bar{B}))\det(\Sigma_1(\bar{B}))} \\
&= 2\log\big(\Sigma(v) - \frac{1}{4}\Sigma^T_{Bv}(\Sigma(B))^{-1}\Sigma_{Bv}\big) \\
&\qquad\qquad - \log\big(\Sigma_0(v)(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv})\big) \qquad (5.80) \\
&= \log\frac{\frac{1}{4}\big(\Sigma_0(v) + \Sigma_1(v) - \Sigma^T_{Bv}(2\Sigma(B))^{-1}\Sigma_{Bv}\big)^2}{\Sigma_0(v)\big(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)}, \qquad (5.81)
\end{aligned}$$

where (5.80) follows from the identity in (5.53). Noting that $\Sigma_0(B) + \Sigma_1(B) \succeq \Sigma_1(B)$ implies $(\Sigma_0(B) + \Sigma_1(B))^{-1} \preceq (\Sigma_1(B))^{-1}$ [91], we have $\Sigma^T_{Bv}(\Sigma_0(B) + \Sigma_1(B))^{-1}\Sigma_{Bv} \leq \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}$. Hence, we see from (5.81) that

$$4\big(f^1_{BD}(\bar{B}) - f^1_{BD}(B)\big) \geq \log\frac{\frac{1}{4}\big(\Sigma_0(v) + \Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)^2}{\Sigma_0(v)\big(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)} \geq 0,$$

where the second inequality follows from the inequality $\frac{(x+y)^2}{xy} \geq 4$ for all $x, y \in \mathbb{R}_{>0}$. Thus, we conclude that $f^1_{BD}(\cdot)$ is monotone nondecreasing.

We now lower bound γ^1_{BD}. Let us consider any $A, B \subseteq \mathcal{V}$, where $A \setminus B \neq \emptyset$. First, supposing $B \neq \emptyset$, we have from (5.81)

$$4\sum_{v\in\tilde{A}}\big(f^1_{BD}(\{v\} \cup B) - f^1_{BD}(B)\big) = \sum_{v\in\tilde{A}} D^1_{BD}(B, v), \qquad (5.82)$$

where

$$D^1_{BD}(B, v) \triangleq \log\frac{\frac{1}{4}\big(\Sigma_0(v) + \Sigma_1(v) - \Sigma^T_{Bv}(2\Sigma(B))^{-1}\Sigma_{Bv}\big)^2}{\Sigma_0(v)\big(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)} \geq 0, \qquad (5.83)$$

for all $B \subseteq \mathcal{V}$ and for all $v \in \mathcal{V} \setminus B$. We then upper bound $\big(f^1_{BD}(A \cup B) - f^1_{BD}(B)\big)$. Denote $\tilde{B} \triangleq A \cup B$ and $\tilde{A} \triangleq A \setminus B$. Let $\Sigma_{B\tilde{A}} \triangleq \mathrm{Cov}(X(B), X(\tilde{A})) \in \mathbb{R}^{|B|\times|\tilde{A}|}$ be the

cross-covariance of $X(B)$ and $X(\tilde{A})$ under hypothesis H_1. Similarly, we have from Eq. (5.79) the following:

$$
\begin{aligned}
&4\big(f^1_{BD}(\tilde{B}) - f^1_{BD}(B)\big) \\
&= 2\log\frac{\det(\Sigma(\tilde{B}))}{\det(\Sigma(B))} + \log\frac{\det(\Sigma_0(B))\det(\Sigma_1(B))}{\det(\Sigma_0(\tilde{B}))\det(\Sigma_1(\tilde{B}))} \\
&= 2\log\det\left(\Sigma(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma(B))^{-1}\Sigma_{B\tilde{A}}\right) \\
&\qquad\qquad - \log\det\left(\Sigma_0(\tilde{A})(\Sigma_1(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}})\right) \qquad (5.84) \\
&\leq 2\log\prod_{v\in\tilde{A}}\left(\Sigma(v) - \Sigma^T_{Bv}(\Sigma(B))^{-1}\Sigma_{Bv}\right) \\
&\qquad\qquad - \log\det\left(\Sigma_0(\tilde{A})(\Sigma_1(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}})\right) \qquad (5.85) \\
&= \log\frac{\prod_{v\in\tilde{A}}\left(\Sigma(v) - \Sigma^T_{Bv}(\Sigma(B))^{-1}\Sigma_{Bv}\right)^2}{\prod_{v\in\tilde{A}}\left(\Sigma_0(v)(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv})\right)} \\
&\qquad\qquad + \log\frac{\prod_{v\in\tilde{A}}\left(\Sigma_0(v)(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv})\right)}{\det\left(\Sigma_0(\tilde{A})(\Sigma_1(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}})\right)} \\
&= \sum_{v\in\tilde{A}}\log\frac{\frac{1}{4}\big(\Sigma_0(v) + \Sigma_1(v) - \Sigma^T_{Bv}(2\Sigma(B))^{-1}\Sigma_{Bv}\big)^2}{\Sigma_0(v)\big(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)} \\
&\qquad\qquad + \log\frac{\prod_{v\in\tilde{A}}\left(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}\right)}{\det\left(\Sigma_1(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\right)} \qquad (5.86) \\
&\leq \sum_{v\in\tilde{A}}\log\frac{\frac{1}{4}\big(\Sigma_0(v) + \Sigma_1(v) - \Sigma^T_{Bv}(2\Sigma(B))^{-1}\Sigma_{Bv}\big)^2}{\Sigma_0(v)\big(\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)} + |\tilde{A}|\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)}, \qquad (5.87)
\end{aligned}
$$

where (5.84) follows from the identity in (5.58). To obtain (5.85), we first use the Hadamard's inequality $\det(P) \leq P_{11}\cdots P_{nn}$ for any $P \in \mathbb{S}^n_{++}$ [91], and then note that the ith diagonal element of the matrix $\left(\Sigma(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma(B))^{-1}\Sigma_{B\tilde{A}}\right)$ is given by $\left(\Sigma(v_i) - \Sigma^T_{Bv_i}(\Sigma(B))^{-1}\Sigma_{Bv_i}\right)$, where $\Sigma_{Bv_i} = \mathrm{Cov}(X(B), X(v_i))$ under hypothesis H_1 for all $v_i \in \tilde{A} = \{v_1, \ldots, v_{|\tilde{A}|}\}$. To obtain (5.86), we note that $\Sigma_0 \in \mathbb{S}^n_{++}$ is diagonal from Assumption 5.4.1, which implies $\det(\Sigma_0(\tilde{A})) = \prod_{v\in\tilde{A}}\Sigma_0(v)$. To obtain (5.87), we first note that $\Sigma_1(v) - \Sigma^T_{Bv}(\Sigma_1(B))^{-1}\Sigma_{Bv} \leq \Sigma_1(v) \leq d_1(\Sigma_1)$ for all $v \in \tilde{A}$. We then use a similar chain of inequalities to that in (5.59), and obtain $\lambda_n\big(\Sigma_1(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\big) \geq \lambda_n(\Sigma_1)$, which implies $\det\left(\Sigma_1(\tilde{A}) - \Sigma^T_{B\tilde{A}}(\Sigma_1(B))^{-1}\Sigma_{B\tilde{A}}\right) \geq (\lambda_n(\Sigma_1))^{|\tilde{A}|}$. (5.87) now follows.

Combining (5.82) and (5.87), we obtain from Definition 5.3.1

$$\gamma_{BD}^1 \geq \frac{\sum_{v\in\tilde{A}} D_{BD}^1(B,v)}{|\tilde{A}|\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + \sum_{v\in\tilde{A}} D_{BD}^1(B,v)}. \tag{5.88}$$

In the following, we will show that $D_{BD}^1(B,v) \geq \min\{C_{BD}^0, C_{BD}^1\}$ for all $B \subseteq \mathcal{V}$ ($B \neq \emptyset$ and $B \neq \mathcal{V}$) and for all $v \in \mathcal{V}\setminus B$, where C_{BD}^0 and C_{BD}^1 are given by Eq. (5.28) and Eq. (5.29), respectively. This lower bound on $D_{BD}^1(B,v)$ together with similar arguments to those for (5.61) will imply via (5.88) the following:

$$\gamma_{BD}^1 \geq \frac{C_{BD}}{\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + C_{BD}}, \tag{5.89}$$

where $C_{BD} = \min\{C_{BD}^0, C_{BD}^1\}$.

We now lower bound $D_{BD}^1(B,v)$. We first note from (5.83) that we can also write $D_{BD}^1(B,v)$ as

$$D_{BD}^1(B,v) = \log\big(\frac{\Sigma_1(v) - \Sigma_{Bv}^T(2\Sigma(B))^{-1}\Sigma_{Bv}}{\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}} + \frac{\big(\Sigma_0(v) - \Sigma_1(v) + \Sigma_{Bv}^T(2\Sigma(B))^{-1}\Sigma_{Bv}\big)^2}{4\Sigma_0(v)\big(\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)}\big). \tag{5.90}$$

Supposing $\Sigma_{Bv} = \mathbf{0}$, we see from Eq. (5.90) that

$$D_{BD}^1(B,v) = \log\big(1 + \frac{(\Sigma_0(v) - \Sigma_1(v))^2}{4\Sigma_0(v)\Sigma_1(v)}\big) \geq C_{BD}^0. \tag{5.91}$$

Moreover, we see that the following holds for all $B \subseteq \mathcal{V}$ and for all $v \in \mathcal{V}\setminus B$:

$$\begin{aligned}
&\frac{\Sigma_1(v) - \Sigma_{Bv}^T(2\Sigma(B))^{-1}\Sigma_{Bv}}{\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}} \\
&= 1 + \frac{\Sigma_{Bv}^T\big((\Sigma_1(B))^{-1} - (2\Sigma(B))^{-1}\big)\Sigma_{Bv}}{\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}} \\
&= 1 + \frac{\Sigma_{Bv}^T\big(\Sigma_1(B) + \Sigma_1(B)(\Sigma_0(B))^{-1}\Sigma_1(B)\big)^{-1}\Sigma_{Bv}}{\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}} &(5.92)\\
&\geq 1 + \frac{\frac{1}{\lambda_1(\Sigma_1) + \frac{(\lambda_1(\Sigma_1))^2}{\lambda_n(\Sigma_0)}}\Sigma_{Bv}^T\Sigma_{Bv}}{\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}} &(5.93)\\
&\geq 1 + \frac{\frac{\lambda_n(\Sigma_0)}{\lambda_1(\Sigma_1)(\lambda_n(\Sigma_0)+\lambda_1(\Sigma_1))}\Sigma_{Bv}^T\Sigma_{Bv}}{\Sigma_1(v) - \frac{1}{\lambda_1(\Sigma_1)}\Sigma_{Bv}^T\Sigma_{Bv}}, &(5.94)
\end{aligned}$$

where (5.92) uses the matrix inversion lemma [91]:

$$(\Sigma_0(B)+\Sigma_1(B))^{-1}=(\Sigma_1(B))^{-1}-\big(\Sigma_1(B)+\Sigma_1(B)(\Sigma_0(B))^{-1}\Sigma_1(B)\big)^{-1}.$$

To obtain (5.93), we first note $\Sigma_0(B)\succeq\lambda_n(\Sigma_0(B))I_{|B|}$ [91], which implies $(\Sigma_0(B))^{-1}\preceq\frac{1}{\lambda_n(\Sigma_0(B))}I_{|B|}$, which further implies, via the Cauchy interlacing theorem for positive definite matrices [91], that $(\Sigma_0(B))^{-1}\preceq\frac{1}{\lambda_n(\Sigma_0)}I_{|B|}$. Hence, $\Sigma_1(B)(\Sigma_0(B))^{-1}\Sigma_1(B)\preceq\frac{1}{\lambda_n(\Sigma_0)}(\Sigma_1(B))^2$. Similarly, $\Sigma_1(B)\preceq\lambda_1(\Sigma_1)I_{|B|}$ and $(\Sigma_1(B))^2\preceq\lambda_1((\Sigma_1(B))^2)I_{|B|}=(\lambda_1(\Sigma_1(B)))^2I_{|B|}\preceq(\lambda_1(\Sigma_1))^2I_{|B|}$. Combining the above arguments yields $\Sigma_1(B)+\Sigma_1(B)(\Sigma_0(B))^{-1}\Sigma_1(B)\preceq\big(\lambda_1(\Sigma_1)+\frac{(\lambda_1(\Sigma_1))^2}{\lambda_n(\Sigma_0)}\big)I_{|B|}$, which leads to (5.93). Similarly, (5.94) follows from $(\Sigma_1(B))^{-1}\succeq\frac{1}{\lambda_1(\Sigma_1)}I_{|B|}$. Supposing $\Sigma_{Bv}\neq\mathbf{0}$, it then follows from Eq. (5.90) and (5.94) that

$$D^1_{BD}(B,v)\geq 1+\frac{\frac{\lambda_n(\Sigma_0)}{\lambda_n(\Sigma_0)+\lambda_1(\Sigma_1)}\Sigma_{Bv}^T\Sigma_{Bv}}{\lambda_1(\Sigma_1)\Sigma_1(v)-\Sigma_{Bv}^T\Sigma_{Bv}}\geq C^1_{BD}, \tag{5.95}$$

Noting that $\Sigma_1(v)-\Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}>0$ for all $v\in\mathcal{V}\setminus B$, we have $C^1_{BD}\geq 1$ by its definition in Eq. (5.29). Since (5.91) (resp., (5.95)) holds for all $B\subseteq\mathcal{V}$ and $v\in\mathcal{V}\setminus B$ such that $\Sigma_{Bv}=\mathbf{0}$ (resp., $\Sigma_{Bv}\neq\mathbf{0}$), we conclude that $D^1_{BD}(B,v)\geq\min\{C^0_{BD},C^1_{BD}\}$ for all $B\subseteq\mathcal{V}$ ($B\neq\emptyset$ and $B\neq\mathcal{V}$) and for all $v\in\mathcal{V}\setminus B$.

Next, suppose $B=\emptyset$. Using similar arguments to those above, one can show that

$$\gamma^1_{BD}\geq\frac{\bar{C}^0_{BD}}{\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)}+\bar{C}^0_{BD}}, \tag{5.96}$$

where $\bar{C}^0_{BD}$ is defined by Eq. (5.31). Noting that $\bar{C}^0_{BD}\geq C_{BD}$, it then follows from (5.89) and (5.96) that

$$\gamma^1_{BD}\geq\frac{C_{BD}}{\log\frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)}+C_{BD}},$$

for all $A,B\subseteq\mathcal{V}$, which completes the proof of part (a).

Proof of (b): Following similar arguments to those for part (a), we will show that $D^1_{BD}\geq\min\{\bar{C}^0_{BD},\bar{C}^1_{BD}\}$, where $\bar{C}^0_{BD}$ and $\bar{C}^1_{BD}$ are defined in Eq. (5.31) and Eq. (5.32), respectively. Consider any $A,B\subseteq\mathcal{V}$, where $A\setminus B\neq\emptyset$. First, suppose

$B \neq \emptyset$. Given that $\Sigma_1 = \Sigma_0 + \Sigma_S$ ($\Sigma_S \in \mathbb{S}^n_{++}$), we will show that $D^1_{BD}(B,v) \geq \bar{C}^1_{BD}$ for all $B \subseteq \mathcal{V}$ ($B \neq \emptyset$ and $B \neq \mathcal{V}$) and for all $v \in \mathcal{V} \setminus B$. We begin with the following:

$$\begin{aligned}
&\frac{\big(\Sigma_0(v) - \Sigma_1(v) + \Sigma_{Bv}^T(2\Sigma(B))^{-1}\Sigma_{Bv}\big)^2}{4\Sigma_0(v)\big(\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}\big)} \\
&\geq \frac{\big(\Sigma_S(v) - \Sigma_{Bv}^T(2\Sigma_0(B) + \Sigma_S(B))^{-1}\Sigma_{Bv}\big)^2}{4\Sigma_0(v)\big(\Sigma_1(v) - \frac{1}{\lambda_1(\Sigma_1)}\Sigma_{Bv}^T\Sigma_{Bv}\big)} && (5.97) \\
&\geq \frac{\big(\Sigma_S(v) - \Sigma_{Bv}^T(\Sigma_S(B))^{-1}\Sigma_{Bv}\big)^2}{4\Sigma_0(v)\big(\Sigma_1(v) - \frac{1}{\lambda_1(\Sigma_1)}\Sigma_{Bv}^T\Sigma_{Bv}\big)} && (5.98) \\
&\geq \frac{(\lambda_n(\Sigma_S))^2}{4\Sigma_0(v)\big(\Sigma_1(v) - \frac{1}{\lambda_1(\Sigma_1)}\Sigma_{Bv}^T\Sigma_{Bv}\big)}, && (5.99)
\end{aligned}$$

where (5.97) follows from similar arguments to those for (5.94). To obtain (5.98), we note that $\Sigma_S(B) \preceq 2\Sigma_0(B) + \Sigma_S(B)$, which implies $(\Sigma_S(B))^{-1} \succeq (2\Sigma_0(B) + \Sigma_S(B))^{-1}$ [91]. For (5.99), we first note that $\Sigma_0 \in \mathbb{S}^n_{++}$ is diagonal from Assumption 5.4.1, which implies $\Sigma_S(\{v\} \cup B) = \left[\begin{smallmatrix} \Sigma_S(v) & \Sigma_{Bv}^T \\ \Sigma_{Bv} & \Sigma_S(B) \end{smallmatrix}\right]$, where $\Sigma_{Bv} = \mathrm{Cov}(X(B), X(v))$ under hypothesis H_1. We then use a similar chain of inequalities to that in (5.59), and obtain $\lambda_n\big(\Sigma_S(v) - \Sigma_{Bv}^T(\Sigma_S(B))^{-1}\Sigma_{Bv}\big) \geq \lambda_n(\Sigma_S)$. Moreover, using similar arguments to those leading to (5.94), we have

$$\frac{\Sigma_1(v) - \Sigma_{Bv}^T(2\Sigma(B))^{-1}\Sigma_{Bv}}{\Sigma_1(v) - \Sigma_{Bv}^T(\Sigma_1(B))^{-1}\Sigma_{Bv}} \geq 1 + \frac{\frac{\lambda_n(\Sigma_0)}{\lambda_1(\Sigma_1)(\lambda_n(\Sigma_0)+\lambda_1(\Sigma_1))}\Sigma_{Bv}^T\Sigma_{Bv}}{\Sigma_1(v) - \frac{1}{\lambda_1(\Sigma_1)}\Sigma_{Bv}^T\Sigma_{Bv}}. \tag{5.100}$$

Noting that (5.99) and (5.100) hold for all $B \subseteq \mathcal{V}$ and for all $v \in \mathcal{V} \setminus B$, it then follows from Eq. (5.90) that $D^1_{BD}(B,v) \geq \bar{C}^1_{BD}$ for all $B \subseteq \mathcal{V}$ and for all $v \in \mathcal{V} \setminus B$. Hence, we conclude that $D^1_{BD}(B,v) \geq \bar{C}^1_{BD}$ for all $B \subseteq \mathcal{V}$ ($B \neq \emptyset$ and $B \neq \mathcal{V}$), which implies

$$\gamma^1_{BD} \geq \frac{\bar{C}^1_{BD}}{\log \frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + \bar{C}^1_{BD}}. \tag{5.101}$$

Next, suppose $B = \emptyset$. Using similar arguments to those above, one can obtain

$$\gamma^1_{BD} \geq \frac{\bar{C}^0_{BD}}{\log \frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + \bar{C}^0_{BD}}. \tag{5.102}$$

It then follows from (5.101) and (5.102) that

$$\gamma^1_{BD} \geq \frac{\bar{C}_{BD}}{\log \frac{d_1(\Sigma_1)}{\lambda_n(\Sigma_1)} + \bar{C}_{BD}},$$

where $\bar{C}_{BD} = \min\{\bar{C}^0_{BD}, \bar{C}^1_{BD}\}$, for all $A, B \subseteq \mathcal{V}$. This completes the proof of part (b). ■

6. NEAR-OPTIMAL DATA SOURCE SELECTION FOR BAYESIAN LEARNING

6.1 Introduction

In this chapter, we extend our analysis for the binary hypothesis testing in Chapter 5 to a general setting where the true state of the world comes from a set that can have cardinality greater two. Under this setting, a central task in machine learning is to learn the true state of the world based on data streams provided by data sources. Here, we do not restrict ourselves to measurements (i.e., data streams) coming from sensors, since in practice the data streams can come from a variety of sources, including experiment outcomes [23], medical tests [24], and sensor measurements [5], etc.

A classical method to tackle this task is Bayesian learning, where we start with a prior belief about the true state of the world and update our belief based on the data streams from the data sources (e.g., [25]). In practice, we need to pay a cost in order to obtain the data streams from the data sources; for example, conducting certain experiments or installing a particular sensor incurs some cost that depends on the nature of the corresponding data source. Thus, a fundamental problem that arises in Bayesian learning is to select a subset of data sources with the smallest total cost, while ensuring a certain level of the learning performance based on the data streams provided by the selected data sources.

In this chapter, we focus on a standard Bayesian learning rule that updates the belief on the true state of the world recursively based on the data streams. The learning performance is then characterized by an error given by the difference between the steady-state belief obtained from the learning rule and the true state of the world. Moreover, we consider the scenario where the data sources are selected a priori before

running the Bayesian learning rule, and the set of selected data sources is fixed over time. We then formulate and study the Bayesian Learning Data Source Selection (BLDS) problem, where the goal is to minimize the cost spent on the selected data sources while ensuring that the error of the learning process is within a prescribed range.

Related Work

In [93] and [94], the authors studied the data source selection problem for Bayesian active learning. They considered the scenario where the data sources are selected in a sequential manner with a single data source selected at each time step in the learning process. The goal is then to find a policy on sequentially selecting the data sources with minimum cost, while the true state of the world can be identified based on the selected data sources. In contrast, we consider the scenario where a subset of data sources are selected a priori. Moreover, the selected data sources may not necessarily lead to the learning of the true state of the world. Thus, we characterize the performance of the learning process via its steady-state error.

The problem studied in this chapter is also related but different from the problem of ensuring sparsity in learning, where the goal is to identify the fewest number of features in order to explain the phenomena in a *given* set of data [83, 95].

Finally, as we mentioned above, our problem formulation also falls into the class of the sensor selection problems that have been studied in previous chapters. In general, the goal of these problems is either to optimize certain (problem-specific) performance metrics of the estimate associated with the measurements of the selected sensors while satisfying the sensor selection budget constraint, or minimize the cost spent on the selected sensors while the estimation performance is within a certain range.

Summary of Results

In this chapter, we first formulate the Bayesian Learning Data Source Selection (BLDS) problem, and show that the BLDS problem is NP-hard. Next, we show that the BLDS problem can be transformed into an instance of the submodular set covering problem studied in [32]. The BLDS problem can then be solved using a standard greedy algorithm with approximation guarantees, where the query complexity of the greedy algorithm is $O(n^2)$, with n to be the number of all candidate data sources. In order to improve the running times of the greedy algorithm, we further propose a fast greedy algorithm with query complexity $O(\frac{n}{\epsilon}\ln\frac{n}{\epsilon})$, where $\epsilon \in (0,1)$. The fast greedy algorithm also achieves comparable performance guarantees to those of the standard greedy algorithm. Finally, we provide illustrative examples to interpret the performance bounds obtained for the greedy algorithms applied to the BLDS problem, and give simulation results.

The results presented in this chapter are available in a preprint [96].

6.2 The Bayesian Learning Data Source Selection Problem

In this section, we formulate the data source selection problem for Bayesian learning that we will study in this chapter. Let $\Theta \triangleq \{\theta_1, \theta_2, \ldots, \theta_m\}$ be a finite set of possible states of the world, where $m \triangleq |\Theta|$. We consider a set $[n]$ of data sources that can provide data streams of the state of the world. At each discrete time step $k \in \mathbb{Z}_{\geq 1}$, the signal (or observation) provided by source $i \in [n]$ is denoted as $\omega_{i,k} \in S_i$, where S_i is the signal space of source i. Conditional on the state of the world $\theta \in \Theta$, an observation profile of the n sources at time k, denoted as $\omega_k \triangleq (\omega_{1,k}, \ldots, \omega_{n,k}) \in S_1 \times \cdots \times S_n$, is generated by the likelihood function $\ell(\cdot|\theta)$. Let $\ell_i(\cdot|\theta)$ denote the i-th marginal of $\ell(\cdot|\theta)$, which is the signal structure of data source $i \in [n]$. We make the following assumption on the observation model (e.g., see [97–100]).

Assumption 6.2.1 *For each source $i \in [n]$, the signal space S_i is finite, and the likelihood function $\ell_i(\cdot|\theta)$ satisfies $l_i(s_i|\theta) > 0$ for all $s_i \in S_i$ and for all $\theta \in \Theta$. Further-*

more, for all $\theta \in \Theta$, the observations are independent over time, i.e., $\{\omega_{i,1}, \omega_{i,2}, \dots\}$ is a sequence of independent identically distributed (i.i.d.) random variables. The likelihood function is assumed to satisfy $\ell(\cdot|\theta) = \prod_{i=1}^{n} \ell_i(\cdot|\theta)$ for all $\theta \in \Theta$, where $\ell_i(\cdot|\theta)$ is the i-th marginal of $\ell(\cdot|\theta)$.

Consider the scenario where there is a (central) designer who needs to select a subset of data sources in order to learn the true state of the world based on the observations from the selected sources. Specifically, each data source $i \in [n]$ is assumed to have an associated selection cost $h_i \in \mathbb{R}_{>0}$. Considering any $\mathcal{I} \triangleq \{n_1, n_2, \dots, n_\tau\}$ with $\tau = |\mathcal{I}|$, we let $h(\mathcal{I})$ denote the sum of the costs of the selected sources in $\mathcal{I}$, i.e., $h(\mathcal{I}) \triangleq \sum_{n_i \in \mathcal{I}} h_{n_i}$. Let $\omega_{\mathcal{I},k} \triangleq (\omega_{n_1,k}, \dots, \omega_{n_\tau,k}) \in S_{n_1} \times \cdots \times S_{n_\tau}$ be the observation profile (conditioned on $\theta \in \Theta$) generated by the likelihood function $\ell_{\mathcal{I}}(\cdot|\theta)$, where $\ell_{\mathcal{I}}(\cdot|\theta) = \prod_{i=1}^{\tau} \ell_{n_i}(\cdot|\theta)$. We assume that the designer knows $\ell_i(\cdot|\theta)$ for all $\theta \in \Theta$ and for all $i \in [n]$, and thus knows $\ell_{\mathcal{I}}(\cdot|\theta)$ for all $\mathcal{I} \subseteq [n]$ and for all $\theta \in \Theta$. After the sources are selected, the designer updates its belief of the state of the world using the following standard Bayes' rule:

$$\mu_{k+1}^{\mathcal{I}}(\theta) = \frac{\mu_0(\theta) \prod_{j=0}^{k} \ell_{\mathcal{I}}(\omega_{\mathcal{I},j+1}|\theta)}{\sum_{\theta_p \in \Theta} \mu_0(\theta_p) \prod_{j=0}^{k} \ell_{\mathcal{I}}(\omega_{\mathcal{I},j+1}|\theta_p)} \ \forall \theta \in \Theta, \tag{6.1}$$

where $u_{k+1}^{\mathcal{I}}(\theta)$ is the belief of the designer that θ is the true state at time step $k+1$, and $\mu_0(\theta)$ is the initial (or prior) belief of the designer that θ is the true state. We take $\sum_{\theta \in \Theta} \mu_0(\theta) = 1$ and $\mu_0(\theta) \in \mathbb{R}_{\geq 0}$ for all $\theta \in \Theta$. Note that $\sum_{\theta \in \Theta} \mu_k^{\mathcal{I}}(\theta) = 1$ for all $\mathcal{I} \subseteq [n]$ and for all $k \in \mathbb{Z}_{\geq 1}$, where $0 \leq \mu_k^{\mathcal{I}}(\theta) \leq 1$ for all $\theta \in \Theta$. In other words, $\mu_k^{\mathcal{I}}(\cdot)$ is a probability distribution over Θ for all $k \in \mathbb{Z}_{\geq 1}$ and for all $\mathcal{I} \subseteq [n]$. Rule (6.1) is also equivalent to the following recursive rule:

$$\mu_{k+1}^{\mathcal{I}}(\theta) = \frac{\mu_k^{\mathcal{I}}(\theta) \ell_{\mathcal{I}}(\omega_{\mathcal{I},k+1}|\theta)}{\sum_{\theta_p \in \Theta} \mu_k^{\mathcal{I}}(\theta_p) \ell_{\mathcal{I}}(\omega_{\mathcal{I},k+1}|\theta_p)} \ \forall \theta \in \Theta, \tag{6.2}$$

with $\mu_0^{\mathcal{I}}(\theta) \triangleq \mu_0(\theta)$ for all $\mathcal{I} \subseteq [n]$. For a given state $\theta \in \Theta$, we define the set of *observationally equivalent* states to θ as

$$F_\theta(\mathcal{I}) \triangleq \arg\min_{\theta_p \in \Theta} D_{KL}(\ell_{\mathcal{I}}(\cdot|\theta_p) \| \ell_{\mathcal{I}}(\cdot|\theta)),$$

where $D_{KL}(\ell_{\mathcal{I}}(\cdot|\theta_p)\|\ell_{\mathcal{I}}(\cdot|\theta))$ is the Kullback-Leibler (KL) divergence between the likelihood functions $\ell_{\mathcal{I}}(\cdot|\theta_p)$ and $\ell_{\mathcal{I}}(\cdot|\theta)$. Noting that $D_{KL}(\ell_{\mathcal{I}}(\cdot|\theta)\|\ell_{\mathcal{I}}(\cdot|\theta)) = 0$ and that the KL divergence is always nonnegative, we have $\theta \in F_\theta(\mathcal{I})$ for all $\theta \in \Theta$ and for all $\mathcal{I} \subseteq [n]$. Equivalently, we can write $F_\theta(\mathcal{I})$ as

$$F_\theta(\mathcal{I}) = \{\theta_p \in \Theta : \ \ell_{\mathcal{I}}(s_{\mathcal{I}}|\theta_p) = \ell_{\mathcal{I}}(s_{\mathcal{I}}|\theta), \forall s_{\mathcal{I}} \in S_{\mathcal{I}}\}, \tag{6.3}$$

where $S_{\mathcal{I}} \triangleq S_{n_1} \times \cdots \times S_{n_\tau}$. Note that $F_\theta(\mathcal{I})$ is the set of states that cannot be distinguished from θ based on the data streams provided by the data sources indicated by $\mathcal{I}$. Moreover, we define $F_\theta(\emptyset) \triangleq \Theta$. Noting that $\ell_{\mathcal{I}}(\cdot|\theta) = \prod_{i=1}^{\tau} \ell_{n_i}(\cdot|\theta)$ under Assumption 6.2.1, we can further obtain from Eq. (6.3) the following:

$$F_\theta(\mathcal{I}) = \bigcap_{n_i \in \mathcal{I}} F_\theta(n_i), \tag{6.4}$$

for all $\mathcal{I} \subseteq [n]$ and for all $\theta \in \Theta$. Using similar arguments to those for Lemma 1 in [101], one can show the following result.

Lemma 6.2.2 *Suppose the true state of the world is θ^*, and $\mu_0(\theta) > 0$ for all $\theta \in \Theta$. For all $\mathcal{I} \subseteq [n]$, the rule given in (6.1) ensures: (a) $\lim_{k\to\infty} \mu_k^{\mathcal{I}}(\theta_p) = 0$ almost surely (a.s.) for all $\theta_p \notin F_{\theta^*}(\mathcal{I})$; and (b) $\lim_{k\to\infty} \mu_k^{\mathcal{I}}(\theta_q) = \frac{\mu_0(\theta_q)}{\sum_{\theta \in F_{\theta^*}(\mathcal{I})} \mu_0(\theta)}$ a.s. for all $\theta_q \in F_{\theta^*}(\mathcal{I})$, where $F_{\theta^*}(\mathcal{I})$ is given by Eq. (6.4).*

Consider a true state $\theta^* \in \Theta$ and a set $\mathcal{I} \subseteq [n]$ of selected sources. In order to characterize the (steady-state) learning performance of rule (6.1), we will use the following error metric (e.g., [102]):

$$e_{\theta^*}(\mathcal{I}) \triangleq \frac{1}{2} \lim_{k\to\infty} \|\mu_k^{\mathcal{I}} - \mathbf{1}_{\theta^*}\|_1, \tag{6.5}$$

where $\mu_k^{\mathcal{I}} \triangleq \begin{bmatrix} \mu_k^{\mathcal{I}}(\theta_1) & \cdots & \mu_k^{\mathcal{I}}(\theta_m) \end{bmatrix}'$, and $\mathbf{1}_{\theta^*} \in \mathbb{R}^m$ is a (column) vector where the element that corresponds to θ^* is 1 and all the other elements are zero. Note that $\frac{1}{2}\|\mu_k^{\mathcal{I}} - \mathbf{1}_{\theta^*}\|_1$ is also known as the total variation distance between the two distributions $\mu_k^{\mathcal{I}}$ and $\mathbf{1}_{\theta^*}$ (e.g., [103]). Also note that $e_{\theta^*}(\mathcal{I})$ exists (a.s.) due to Lemma 6.2.2. We

then see from Lemma 6.2.2 that $e_{\theta^*}(\mathcal{I}) = 1 - \frac{\mu_0(\theta^*)}{\sum_{\theta \in F_{\theta^*}(\mathcal{I})} \mu_0(\theta)}$ holds almost surely. Now, let us define

$$e^s_{\theta_p}(\mathcal{I}) \triangleq 1 - \frac{\mu_0(\theta_p)}{\sum_{\theta \in F_{\theta_p}(\mathcal{I})} \mu_0(\theta)} \ \forall \theta_p \in \Theta, \tag{6.6}$$

which represents the (steady-state) total variation distance between the designer's belief $\mu_k^{\mathcal{I}}$ and $\mathbf{1}_{\theta_p}$ when θ_p is the true state of the world, for all $\theta_p \in \Theta$. We then define the Bayesian Learning Data Source Selection (BLDS) problem as follows.

Problem 6.2.3 *(BLDS) Consider a set $\Theta = \{\theta_1, \ldots, \theta_m\}$ of possible states of the world; a set $[n]$ of data sources providing data streams, where the signal space of source $i \in [n]$ is S_i and the observation from source $i \in [n]$ under state $\theta \in \Theta$ is generated by $\ell_i(\cdot|\theta)$; a selection cost $h_i \in \mathbb{R}_{>0}$ of each source $i \in [n]$; an initial belief $\mu_0(\theta) \in \mathbb{R}_{>0}$ for all $\theta \in \Theta$ with $\sum_{\theta \in \Theta} \mu_0(\theta) = 1$; and prescribed error bounds $0 \leq R_{\theta_p} \leq 1$ ($R_{\theta_p} \in \mathbb{R}$) for all $\theta_p \in \Theta$. The BLDS problem is to find a set of selected data sources $\mathcal{I} \subseteq [n]$ that solves*

$$\begin{aligned} &\min_{\mathcal{I} \subseteq [n]} h(\mathcal{I}) \\ &s.t.\ e^s_{\theta_p}(\mathcal{I}) \leq R_{\theta_p} \ \forall \theta_p \in \Theta, \end{aligned} \tag{6.7}$$

where $e^s_{\theta_p}(\mathcal{I})$ is defined in (6.6).

Note that the constraints in (6.7) capture the fact that the true state of the world is unknown to the designer in general. In other words, for any set $\mathcal{I} \subseteq [n]$ and for any $\theta_p \in \Theta$, the constraint $e^s_{\theta_p}(\mathcal{I}) \leq R_{\theta_p}$ requires the (steady-state) learning error $e^s_{\theta_p}(\mathcal{I})$ to be upper bounded by R_{θ_p} when the true state of the world is assumed to be θ_p. Moreover, the interpretation of R_{θ_p} for $\theta_p \in \Theta$ is as follows. When $R_{\theta_p} = 0$, we see from (6.6) and the constraint $e^s_{\theta_p}(\mathcal{I}) \leq R_{\theta_p}$ that $F_{\theta_p}(\mathcal{I}) = \{\theta_p\}$. In other words, the constraint $e^s_{\theta_p}(\mathcal{I}) \leq 0$ requires that any $\theta_q \in \Theta \setminus \{\theta_p\}$ is *not* observationally equivalent to θ_p, based on the observations from the data sources indicated by $\mathcal{I} \subseteq [n]$. Next, when $R_{\theta_p} = 1$, we know from (6.6) that the constraint $e^s_{\theta_p}(\mathcal{I}) \leq 1$ is satisfied for all $\mathcal{I} \subseteq [n]$. Finally, when $0 < R_{\theta_p} < 1$ and $\mu_0(\theta) = \frac{1}{m}$ for all $\theta \in \Theta$, where $m = |\Theta|$, we see from (6.6) that the constraint $e^s_{\theta_p}(\mathcal{I}) \leq R_{\theta_p}$ is equivalent to $|F_{\theta_p}(\mathcal{I})| \leq \frac{1}{1-R_{\theta_p}}$, i.e.,

the number of states that are observationally equivalent to θ_p should be less than or equal to $\frac{1}{1-R_{\theta_p}}$, based on the observations from the data source indicated by $\mathcal{I} \subseteq [n]$. In summary, the value of R_{θ_p} in the constraints represents the requirements of the designer on distinguishing state θ_p from other states in Θ, where a smaller value of R_{θ_p} would imply that the designer wants to distinguish θ_p from more states in Θ and vice versa. Supposing $R_{\theta_p} = R$ for all $\theta_p \in \Theta$, where $0 \leq R \leq 1$ and $R \in \mathbb{R}$, we see that the constraints in (6.7) can be equivalently written as $\max_{\theta_p \in \Theta} e^s_{\mathcal{I}}(\theta_p) \leq R$.

Remark 6.2.4 *The problem formulation that we described above can be extended to the scenario where the data sources are distributed among a set of agents, and the agents collaboratively learn the true state of the world using their own observations and communications with other agents. This scenario is known as distributed non-Bayesian learning (e.g., [100]). The goal of the (central) designer is then to select a subset of all the agents whose data sources will be used to collect observations such that the learning error of all the agents is within a prescribed range. More details about this extension can be found in Section* 6.4.

Next, we show that the BLDS problem is NP-hard via a reduction from the set cover problem defined in Problem 6.2.5, which is known to be NP-hard (e.g., [41], [104]).

Problem 6.2.5 *(Set Cover) Consider a set $U = \{u_1, \ldots, u_d\}$ and a collection of subsets of U, denoted as $\mathcal{C} = \{C_1, \ldots, C_k\}$. The set cover problem is to select as few as possible subsets from $\mathcal{C}$ such that every element in U is contained in at least one of the selected subsets.*

Theorem 6.2.6 *The BLDS problem is NP-hard even when all the data sources have the same cost, i.e., $h_i = 1$ for all $i \in [n]$.*

Proof We give a polynomial-time reduction from the set cover problem to the BLDS problem. Consider an arbitrary instance of the set cover problem as described in Problem 6.2.5, with the set $U = \{u_1, \ldots, u_d\}$ and the collection $\mathcal{C} = \{C_1, \ldots, C_k\}$,

where C_i's are subsets of U. Denote $C_i = \{u_{i_1}, \ldots, u_{i_{\beta_i}}\}$ for all $i \in [k]$, where $\beta_i = |C_i|$. We then construct an instance of the BLDS problem as follows. The set of possible states of the world is set to be $\Theta = \{\theta_1, \ldots, \theta_{d+1}\}$. The number of data sources is set as $n = k$, where the signal space of source i is set to be $S_i = \{0, 1\}$ for all $i \in [k]$. For any source $i \in [k]$, the likelihood function $\ell_i(\cdot|\theta)$ corresponding to source $i \in [k]$ is set to satisfy that $\ell_i(0|\theta_1) = \ell_i(1|\theta_1) = \frac{1}{2}$, $\ell_i(0|\theta_{q+1}) = \ell_i(1|\theta_{q+1}) = \frac{1}{2}$ for all $u_q \in U \setminus C_i$, and $\ell_i(0|\theta_{i_j+1}) = \frac{1}{3}$ and $\ell_i(1|\theta_{i_j+1}) = \frac{2}{3}$ for all $u_{i_j} \in C_i$. The selection cost is set as $h_i = 1$ for all $i \in [k]$. The initial belief is set to be $\mu_0(\theta_p) = \frac{1}{d+1}$ for all $p \in [d+1]$. The prescribed error bounds are set as $R_{\theta_1} = 0$ and $R_{\theta_p} = 1$ for all $p \in \{2, \ldots, d+1\}$. Note that the set of selected sources is denoted as $\mathcal{I} = \{n_1, \ldots, n_\tau\} \subseteq [k]$.

Since $R_{\theta_p} = 1$ for all $p \in \{2, \ldots, d+1\}$, the constraint $e^s_{\theta_p}(\mathcal{I}) \leq R_{\theta_p}$ is satisfied for all $\mathcal{I} \subseteq [n]$ and for all $p \in \{2, \ldots, d+1\}$. We then focus on the constraint corresponding to θ_1. Letting $R_{\theta_1} = 0$ and $\mu_0(\theta_p) = \frac{1}{d+1}$ for all $p \in [d+1]$, the constraint $e^s_{\theta_1}(\mathcal{I}) \leq R_{\theta_1}$ is equivalent to $|F_{\theta_1}(\mathcal{I})| \leq 1$, where $F_{\theta_1}(\mathcal{I}) = \bigcap_{n_i \in \mathcal{I}} F_{\theta_1}(n_i)$ with $F_{\theta_1}(n_i)$ given by Eq. (6.3). Denote $F^c_{\theta_1}(i) \triangleq \Theta \setminus F_{\theta_1}(i)$ for all $i \in [k]$. From the way we set the likelihood function $\ell_i(\cdot|\theta)$ for source $i \in [k]$ in the constructed instance of the BLDS problem, we see that $F^c_{\theta_1}(i) = \{\theta_{i_1+1}, \ldots, \theta_{i_{\beta_i}+1}\}$ for all $i \in [k]$, i.e., $C_i \in \mathcal{C}$ corresponds to $F^c_{\theta_1}(i)$ for all $i \in [k]$. Moreover, using De Morgan's laws, we have

$$F_{\theta_1}(\mathcal{I}) = \bigcap_{n_i \in \mathcal{I}} F_{\theta_1}(n_i) = \Theta \setminus \big(\bigcup_{n_i \in \mathcal{I}} F^c_{\theta_1}(n_i) \big). \tag{6.8}$$

Considering any $\mathcal{I} = \{n_1, \ldots, n_\tau\} \subseteq [k]$ with $\tau = |\mathcal{I}|$, we let $\mathcal{C}_\mathcal{I} \triangleq \{C_{n_1}, \ldots, C_{n_\tau}\}$. We will show that $\mathcal{I}$ is a feasible solution to the given set cover instance (i.e., for any $u_j \in U$, there exists $C_i \in \mathcal{C}_\mathcal{I}$ such that $u_j \in C_i$) if and only if $\mathcal{I}$ is a feasible solution to the constructed BLDS instance (i.e., the constraint $e^s_{\theta_1}(\mathcal{I}) \leq R_{\theta_1}$ is satisfied).

Suppose $\mathcal{I}$ is a feasible solution to the given set cover instance. Since $C_i \in \mathcal{C}$ corresponds to $F^c_{\theta_1}(i)$ for all $i \in [k]$, we see that for any $\theta_p \in \{\theta_2, \ldots, \theta_{d+1}\}$, there exists $n_i \in \mathcal{I}$ such that $\theta_p \in F^c_{\theta_1}(n_i)$ in the constructed BLDS instance, which implies $\bigcup_{n_i \in \mathcal{I}} F^c_{\theta_1}(n_i) = \{\theta_2, \ldots, \theta_{d+1}\}$. It follows from (6.8) that $F_{\theta_1}(\mathcal{I}) = \Theta \setminus \{\theta_2, \ldots, \theta_{d+1}\} = \{\theta_1\}$, which implies that the constraint $|F_{\theta_1}(\mathcal{I})| \leq 1$ is satisfied, i.e., the constraint

$e^s_{\theta_1}(\mathcal{I}) \leq R_{\theta_1}$ is satisfied. Conversely, suppose $\mathcal{I}$ is a feasible solution to the constructed BLDS instance, i.e., the constraint $e^s_{\theta_1}(\mathcal{I}) \leq R_{\theta_1}$ is satisfied, which implies $|F_{\theta_1}(\mathcal{I})| \leq 1$. Noting that $\theta_1 \in F_{\theta_1}(\mathcal{I})$ for all $\mathcal{I} \subseteq [k]$, we have $F_{\theta_1}(\mathcal{I}) = \{\theta_1\}$. We then see from (6.8) that $\bigcup_{n_i \in \mathcal{I}} F^c_{\theta_1}(n_i) = \{\theta_2, \ldots, \theta_{d+1}\}$, i.e., for all $\theta_p \in \{\theta_2, \ldots, \theta_{d+1}\}$, there exists $n_i \in \mathcal{I}$ such that $\theta_p \in F^c_{\theta_1}(n_i)$. It then follows from the one-to-one correspondence between C_i and $F^c_{\theta_1}(i)$ that for any $u_j \in U$, there exists $C_{n_i} \in \mathcal{C}_{\mathcal{I}}$ such that $u_j \in C_{n_i}$ in the set cover instance.

Since the selection cost is set as $h_i = 1$ for all $i \in [k]$, we see from the above arguments that $\mathcal{I}^*$ is an optimal solution to the set cover instance if and only if it is an optimal solution to the BLDS instance. Since the set cover problem is NP-hard, we conclude that the BLDS problem is NP-hard. ■

6.3 Greedy Algorithms for the BLDS Problem

In this section, we consider a greedy algorithm for the BLDS problem and study its performance guarantees. We first introduce the following definition.

Definition 6.3.1 *([51]) A set function $f : 2^{[n]} \to \mathbb{R}$ is submodular if for all $X \subseteq Y \subseteq [n]$ and for all $j \in [n] \setminus Y$,*

$$f(X \cup \{j\}) - f(X) \geq f(Y \cup \{j\}) - f(Y). \tag{6.9}$$

Equivalently, $f : 2^{[n]} \to \mathbb{R}$ is submodular if for all $X, Y \subseteq [n]$,

$$\sum_{j \in Y \setminus X} (f(X \cup \{j\}) - f(X)) \geq f(Y \cup X) - f(X). \tag{6.10}$$

To proceed, note that the constraint corresponding to θ_p in Problem 6.2.3 (i.e., (6.7)) is satisfied for all $\mathcal{I} \subseteq [n]$ if $R_{\theta_p} = 1$. Since $\mu_0(\theta_p) > 0$ for all $\theta_p \in \Theta$, we can then equivalently write the constraints as

$$\sum_{\theta \in F_{\theta_p}(\mathcal{I})} \mu_0(\theta) \leq \frac{\mu_0(\theta_p)}{1 - R_{\theta_p}}, \ \forall \theta_p \in \Theta \text{ with } R_{\theta_p} < 1. \tag{6.11}$$

Define $F_\theta^c(\mathcal{I}) \triangleq \Theta \setminus F_{\theta_p}(\mathcal{I})$ for all $\theta \in \Theta$ and for all $\mathcal{I} \subseteq [n]$, where $F_\theta(\mathcal{I})$ is given by Eq. (6.4). Note that $F_\theta^c(\mathcal{I})$ is the set of states that can be distinguished from θ, given the data sources indicated by $\mathcal{I}$. Using the fact $\sum_{\theta\in\Theta} \mu_0(\theta) = 1$, (6.11) can be equivalently written as

$$\sum_{\theta\in F_{\theta_p}^c(\mathcal{I})} \mu_0(\theta) \geq 1 - \frac{\mu_0(\theta_p)}{1 - R_{\theta_p}},\ \forall \theta_p \in \Theta \text{ with } R_{\theta_p} < 1. \tag{6.12}$$

Moreover, we note that the constraint corresponding to θ_p in (6.12) is satisfied for all $\mathcal{I} \subseteq [n]$ if $1 - \frac{\mu_0(\theta_p)}{1-R_{\theta_p}} \leq 0$, i.e., $R_{\theta_p} \geq 1 - \mu_0(\theta_p)$. Hence, we can equivalently write (6.12) as

$$\sum_{\theta\in F_{\theta_p}^c(\mathcal{I})} \mu_0(\theta) \geq 1 - \frac{\mu_0(\theta_p)}{1 - R_{\theta_p}},\ \forall \theta_p \in \bar{\Theta},$$

where $\bar{\Theta} \triangleq \{\theta_p \in \Theta : 0 \leq R_{\theta_p} < 1 - \mu_0(\theta_p)\}$. For any $\mathcal{I} \subseteq [n]$, let us define

$$f_{\theta_p}(\mathcal{I}) \triangleq \sum_{\theta\in F_{\theta_p}^c(\mathcal{I})} \mu_0(\theta),\ \forall \theta_p \in \bar{\Theta}. \tag{6.13}$$

Noting that $F_{\theta_p}(\emptyset) = \Theta$, i.e., $F_{\theta_p}^c(\emptyset) = \emptyset$, we let $f_{\theta_p}(\emptyset) = 0$. It then follows directly from (6.13) that $f_{\theta_p} : 2^{[n]} \to \mathbb{R}_{\geq 0}$ is a monotone nondecreasing set function.[1]

Remark 6.3.1 *Note that in order to ensure that there exists $\mathcal{I} \subseteq [n]$ that satisfies the constraints in* (6.12)*, we assume that $f_{\theta_p}([n]) \geq 1 - \frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ for all $\theta_p \in \bar{\Theta}$, since $f_{\theta_p}(\cdot)$ is nondecreasing for all $\theta_p \in \bar{\Theta}$.*

Lemma 6.3.2 *The set function $f_{\theta_p} : 2^{[n]} \to \mathbb{R}_{\geq 0}$ defined in* (6.13) *is submodular for all $\theta_p \in \bar{\Theta}$.*

[1] A set function $f : 2^{[n]} \to \mathbb{R}$ is monotone nondecreasing if $f(X) \leq f(Y)$ for all $X \subseteq Y \subseteq [n]$.

Proof Consider any $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq [n]$ and any $j \in [n] \setminus \mathcal{I}_2$. For all $\mathcal{I} \subseteq [n]$, we will drop the dependency of $F_{\theta_p}(\mathcal{I})$ (resp., $F^c_{\theta_p}(\mathcal{I})$) on θ_p, and write $F(\mathcal{I})$ (resp., $F^c(\mathcal{I})$) for notational simplicity in this proof. We then have the following:

$$\begin{aligned}
& f_{\theta_p}(\mathcal{I}_1 \cup \{j\}) - f_{\theta_p}(\mathcal{I}_1) \\
&= \sum_{\theta \in F^c(\mathcal{I}_1 \cup \{j\})} \mu_0(\theta) - \sum_{\theta \in F^c(\mathcal{I}_1)} \mu_0(\theta) \\
&= \sum_{\theta \in F^c(\mathcal{I}_1) \cup F^c(j)} \mu_0(\theta) - \sum_{\theta \in F^c(\mathcal{I}_1)} \mu_0(\theta) \qquad (6.14) \\
&= \sum_{\theta \in (F^c(\mathcal{I}_1) \cup F^c(j)) \setminus F^c(\mathcal{I}_1)} \mu_0(\theta) = \sum_{\theta \in F^c(j) \setminus F^c(\mathcal{I}_1)} \mu_0(\theta). \qquad (6.15)
\end{aligned}$$

To obtain (6.14), we note $F^c(\mathcal{I}_1 \cup \{j\}) = \Theta \setminus F(\mathcal{I}_1 \cup \{j\}) = \Theta \setminus (F(\mathcal{I}_1) \cap F(j))$, which implies (via De Morgan's laws) $F^c(\mathcal{I}_1 \cup \{j\}) = F^c(\mathcal{I}_1) \cup F^c(j)$. Similarly, we also have

$$f_{\theta_p}(\mathcal{I}_2 \cup \{j\}) - f_{\theta_p}(\mathcal{I}_2) = \sum_{\theta \in F^c(j) \setminus F^c(\mathcal{I}_2)} \mu_0(\theta). \qquad (6.16)$$

Since $\mathcal{I}_1 \subseteq \mathcal{I}_2$, we have $F^c(j) \setminus F^c(\mathcal{I}_2) \subseteq F^c(j) \setminus F^c(\mathcal{I}_1)$, which implies via (6.15)-(6.16)

$$f_{\theta_p}(\mathcal{I}_1 \cup \{j\}) - f_{\theta_p}(\mathcal{I}_1) \geq f_{\theta_p}(\mathcal{I}_2 \cup \{j\}) - f_{\theta_p}(\mathcal{I}_2).$$

Since the above arguments hold for all $\theta_p \in \bar{\Theta}$, we know from (6.9) in Definition 6.3.1 that $f_{\theta_p}(\cdot)$ is submodular for all $\theta_p \in \bar{\Theta}$. ■

Moreover, considering any $\mathcal{I} \subseteq [n]$, we define

$$f'_{\theta_p}(\mathcal{I}) \triangleq \min\{f_{\theta_p}(\mathcal{I}), 1 - \frac{\mu_0(\theta_p)}{1 - R_{\theta_p}}\} \ \forall \theta_p \in \bar{\Theta}, \qquad (6.17)$$

where $f_{\theta_p}(\mathcal{I})$ is defined in (6.13). Since $f_{\theta_p}(\cdot)$ is submodular and nondecreasing with $f_{\theta_p}(\emptyset) = 0$ and $f_{\theta_p}([n]) \geq 1 - \frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$, one can show that $f'_{\theta_p}(\cdot)$ is also submodular and nondecreasing with $f'_{\theta_p}(\emptyset) = 0$ and $f'_{\theta_p}([n]) = 1 - \frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$. Noting that the sum of submodular functions remains submodular, we see that $\sum_{\theta_p \in \bar{\Theta}} f'_{\theta_p}(\cdot)$ is submodular and nondecreasing. We also have the following result.

Lemma 6.3.3 *Consider any $\mathcal{I} \subseteq [n]$. The constraint $\sum_{\theta \in F^c_{\theta_p}(\mathcal{I})} \mu_0(\theta) \geq 1 - \frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ holds for all $\theta_p \in \bar{\Theta}$ if and only if $\sum_{\theta_p \in \bar{\Theta}} f'_{\theta_p}(\mathcal{I}) = \sum_{\theta_p \in \bar{\Theta}} f'_{\theta_p}([n])$, where $f'_{\theta_p}(\cdot)$ is defined in* (6.17).

Proof Suppose the constraints $\sum_{\theta\in F^c_{\theta_p}(\mathcal{I})}\mu_0(\theta)\geq 1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ hold for all $\theta_p\in\bar{\Theta}$. It follows from (6.17) that $f'_{\theta_p}(\mathcal{I})=1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ for all $\theta_p\in\bar{\Theta}$. Noting that $f_{\theta_p}([n])\geq 1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ as argued in Remark 6.3.1, we have $f'_{\theta_p}([n])=1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ for all $\theta_p\in\bar{\Theta}$, which implies $\sum_{\theta_p\in\bar{\Theta}}f'_{\theta_p}(\mathcal{I})=\sum_{\theta_p\in\bar{\Theta}}f'_{\theta_p}([n])$. Conversely, suppose $\sum_{\theta_p\in\bar{\Theta}}f'_{\theta_p}(\mathcal{I})=\sum_{\theta_p\in\bar{\Theta}}f'_{\theta_p}([n])$, i.e., $\sum_{\theta_p\in\bar{\Theta}}\left(f'_{\theta_p}(\mathcal{I})-\left(1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}\right)\right)=0$. Noting from (6.17) that $f'_{\theta_p}(\mathcal{I})\leq 1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ for all $\mathcal{I}\subseteq[n]$, we have $f'_{\theta_p}(\mathcal{I})=1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ for all $\theta_p\in\bar{\Theta}$, i.e., $f_{\theta_p}(\mathcal{I})\geq 1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}$ for all $\theta_p\in\bar{\Theta}$. This completes the proof of the lemma. ■

Based on the above arguments, for any $\mathcal{I}\subseteq[n]$, we further define

$$z(\mathcal{I})\triangleq\sum_{\theta_p\in\bar{\Theta}}f'_{\theta_p}(\mathcal{I})=\sum_{\theta_p\in\bar{\Theta}}\min\{f_{\theta_p}(\mathcal{I}),1-\frac{\mu_0(\theta_p)}{1-R_{\theta_p}}\},\tag{6.18}$$

where $f_{\theta_p}(\mathcal{I})$ is defined in (6.13). We then see from Lemma 6.3.3 that (6.7) in Problem 6.2.3 can be equivalently written as

$$\begin{aligned}&\min_{\mathcal{I}\subseteq[n]}h(\mathcal{I})\\&s.t.\ z(\mathcal{I})=z([n]),\end{aligned}\tag{6.19}$$

where one can see that $z(\cdot)$ defined in Eq. (6.18) is a nondecreasing and submodular set function with $z(\emptyset)=0$. Considering an instance of the BLDS problem, for any $\mathcal{I}\subseteq[n]$ and for any $\theta\in\Theta$, one can obtain $F_{\mathcal{I}}(\theta)$ (and $F^c_{\mathcal{I}}(\theta)$) in $O(S|\mathcal{I}||\Theta|)$ time, where $S\triangleq\max_{n_i\in\mathcal{I}}|S_i|$ with S_i to be the signal space of source $n_i\in\mathcal{I}$. Therefore, we see from (6.13) and (6.18) that for any $\mathcal{I}\subseteq[n]$, one can compute the value of $z(\mathcal{I})$ in $O(Sn|\Theta|^2)$ time.

Problem (6.19) can now be viewed as the submodular set covering problem studied in [32], where the submodular set covering problem is solved using a greedy algorithm with performance guarantees. Specifically, we consider the greedy algorithm defined in Algorithm 6.3.1 for the BLDS problem. The algorithm maintains a sequence of sets $\mathcal{I}^0_g,\mathcal{I}^1_g,\ldots,\mathcal{I}^T_g$ containing the selected elements from $[n]$, where $T\in\mathbb{Z}_{\geq 1}$. Note that Algorithm 6.3.1 requires $O(n^2)$ evaluations of function $z(\cdot)$, where $z(\mathcal{I})$ can be computed in $O(Sn|\Theta|^2)$ time for any $\mathcal{I}\subseteq[n]$ as argued above. In other words, the query complexity of Algorithm 6.3.1 is $O(n^2)$. We then have the following result

from the arguments above (i.e., Lemmas 6.3.2-6.3.3) and Theorem 1 in [32], which characterizes the performance guarantees for the greedy algorithm (Algorithm 6.3.1) when applied to the BLDS problem.

Algorithm 6.3.1 Greedy Algorithm for BLDS

Input: $[n]$, $z : 2^{[n]} \to \mathbb{R}_{\geq 0}$, $h_i \ \forall i \in [n]$

Output: $\mathcal{I}_g$

1: $t \leftarrow 0$, $\mathcal{I}_g^0 \leftarrow \emptyset$

2: **while** $z(\mathcal{I}_g^t) < z([n])$ **do**

3: $\quad j_t \in \arg\max_{i \in [n] \setminus \mathcal{I}_g^t} \frac{z(\mathcal{I}_g^t \cup \{i\}) - z(\mathcal{I}_g^t)}{h_i}$

4: $\quad \mathcal{I}_g^{t+1} \leftarrow \mathcal{I}_g^t \cup \{j_t\}$, $t \leftarrow t+1$

5: $T \leftarrow t$, $\mathcal{I}_g \leftarrow \mathcal{I}_g^T$

6: **return** $\mathcal{I}_g$

Theorem 6.3.4 *Let $\mathcal{I}^*$ be an optimal solution to the BLDS problem. Algorithm* 6.3.1 *returns a solution $\mathcal{I}_g$ to the BLDS problem (i.e.,* (6.19)*) that satisfies the following, where $\mathcal{I}_g^1, \ldots, \mathcal{I}_g^{T-1}$ are specified in Algorithm* 6.3.1.

(a) $h(\mathcal{I}_g) \leq \big(1 + \ln \max_{i \in [n], \zeta \in [T-1]} \{ \frac{z(i) - z(\emptyset)}{z(\mathcal{I}_g^\zeta \cup \{i\}) - z(\mathcal{I}_g^\zeta)} : z(\mathcal{I}_g^\zeta \cup \{i\}) - z(\mathcal{I}_g^\zeta) > 0 \}\big) h(\mathcal{I}^*)$,

(b) $h(\mathcal{I}_g) \leq \big(1 + \ln \frac{h_{j_T}(z(j_1) - z(\emptyset))}{h_{j_1}(z(\mathcal{I}_g^{T-1} \cup \{j_T\}) - z(\mathcal{I}_g^{T-1}))}\big) h(\mathcal{I}^*)$,

(c) $h(\mathcal{I}_g) \leq \big(1 + \ln \frac{z([n]) - z(\emptyset)}{z([n]) - z(\mathcal{I}_g^{T-1})}\big) h(\mathcal{I}^*)$,

(d) if $z(\mathcal{I}) \in \mathbb{Z}_{\geq 0}$ *for all* $\mathcal{I} \subseteq [n]$, $h(\mathcal{I}_g) \leq (\sum_{i=i}^{M} \frac{1}{i}) h(\mathcal{I}^*) \leq (1 + \ln M) h(\mathcal{I}^*)$, *where* $M \triangleq \max_{j \in [n]} z(j)$.

Note that the bounds in Theorem 6.3.4(a)-(c) depend on $\mathcal{I}_g^t$ from the greedy algorithm. We can compute the bounds in Theorem 6.3.4(a)-(c) in parallel with the greedy algorithm, in order to provide a performance guarantee on the output of the algorithm. The bound in Theorem 6.3.4(d) does not depend on $\mathcal{I}_g^t$, and can be computed using $O(n)$ evaluations of function $z(\cdot)$.

6.3.1 Fast greedy algorithm

Algorithm 6.3.2 Fast Greedy Algorithm for BLDS

Input: $[n]$, $z : 2^{[n]} \to \mathbb{R}_{\geq 0}$, $h_i \ \forall i \in [n]$, $\epsilon \in (0, 1)$

Output: $\mathcal{I}_f$

1: $t \leftarrow 0$, $\mathcal{I}_f^0 \leftarrow \emptyset$
2: $d \leftarrow \max_{i \in [n]} \frac{z(i) - z(\emptyset)}{h_i}$
3: **for** $(\tau = d; \tau \geq \frac{\epsilon h_{\min}}{n h_{\max}} d; \tau \leftarrow \tau(1 - \epsilon))$ **do**
4: **for** $j \in [n]$ **do**
5: **if** $\frac{z(\mathcal{I}_f^t \cup \{j\}) - z(\mathcal{I}_f^t)}{h_j} \geq \tau$ **then**
6: $\mathcal{I}_f^{t+1} \leftarrow \mathcal{I}_f^t \cup \{j\}$, $t \leftarrow t + 1$
7: **if** $z(\mathcal{I}_f^t) = z([n])$ **then**
8: $T \leftarrow t$, $\mathcal{I}_f \leftarrow \mathcal{I}_f^T$
9: **return** $\mathcal{I}_f$
10: $T \leftarrow t$, $\mathcal{I}_f \leftarrow \mathcal{I}_f^T$
11: **return** $\mathcal{I}_f$

We now give an algorithm (Algorithm 6.3.2) for BLDS that achieves $O(\frac{n}{\epsilon} \ln \frac{n}{\epsilon})$ query complexity for any $\epsilon \in (0, 1)$, which is significantly smaller than $O(n^2)$ as n scales large. In line 3 of Algorithm 6.3.2, $h_{\max} \triangleq \max_{j \in [n]} h_j$ and $h_{\min} \triangleq \min_{j \in [n]} h_j$. While achieving faster running times, we will show that the solution returned by Algorithm 6.3.2 has slightly worse performance bounds compared to those of Algorithm 6.3.1 provided in Theorem 6.3.4, and potentially slightly violates the constraint of the BLDS problem given in (6.19). Specifically, a larger value of ϵ in Algorithm 6.3.2 leads to faster running times of Algorithm 6.3.2, but yields worse performance guarantees. Moreover, note that Algorithm 6.3.1 adds a single element to $\mathcal{I}_g$ in each iteration of the while loop in lines 2-4. In contrast, Algorithm 6.3.2 considers multiple candidate elements in each iteration of the for loop in lines 3-9, and adds elements that

satisfy the threshold condition given in line 5, which leads to faster running times. Formally, we have the following result; the proof is included in Section 6.6.1.

Theorem 6.3.5 *Suppose* $\frac{h_{\max}}{h_{\min}} \leq n^H$ *holds in the BLDS instances, where* $h_{\max} = \max_{j\in[n]} h_j$, $h_{\min} = \min_{j\in[n]} h_j$, *and* $H \in \mathbb{R}_{\geq 1}$ *is a fixed constant. Let* $\mathcal{I}^*$ *be an optimal solution to the BLDS problem. For any* $\epsilon \in (0,1)$, *Algorithm* 6.3.2 *returns a solution* $\mathcal{I}_f$ *to the BLDS problem (i.e.,* (6.19)*) in query complexity* $O(\frac{n}{\epsilon}\ln\frac{n}{\epsilon})$ *that satisfies* $z(\mathcal{I}_f) \geq (1-\epsilon)z([n])$, *and has the following performance bounds, where* $\mathcal{I}_f^{T-1}$ *is given in Algorithm* 6.3.2.
(a) $h(\mathcal{I}_f) \leq \frac{1}{1-\epsilon}\big(1 + \ln\frac{z([n])}{z([n])-z(\mathcal{I}_f^{T-1})}\big)h(\mathcal{I}^*)$,
(b) if $z(\mathcal{I}) \in \mathbb{Z}_{\geq 0}$ *for all* $\mathcal{I} \subseteq [n]$, $h(\mathcal{I}_f) \leq \frac{1}{1-\epsilon}\big(1 + \ln z([n])\big)h(\mathcal{I}^*)$.

Remark 6.3.6 *The threshold-based greedy algorithm has also been proposed for the problem of maximizing a monotone nondecreasing submodular function subject to a cardinality constraint (e.g., [105]). The threshold-based greedy algorithm proposed in [105] improves the running times of the standard greedy algorithm proposed in [51], and achieves a comparable performance guarantee to that of the standard greedy algorithm in [51]. Here, we propose a threshold-based greedy algorithm (Algorithm* 6.3.2*) to solve the submodular set covering problem, which improves the running times of the standard greedy algorithm for the submodular set covering problem proposed in [32] (i.e., Algorithm* 6.3.2*), and achieves comparable performances guarantees as we showed in Theorem* 6.3.5.

6.3.2 Interpretation of Performance Bounds

Here, we give an illustrative example to interpret the performance bounds of Algorithm 6.3.1 and Algorithm 6.3.2 given in Theorem 6.3.4 and Theorem 6.3.5, respectively. In particular, we focus on the bounds given in Theorem 6.3.4(d) and Theorem 6.3.5(b). Consider an instance of the BLDS problem, where we set $\mu_0(\theta_p) = \frac{1}{m}$ for all $\theta_p \in \Theta$ with $m = |\Theta|$. In other words, there is a uniform prior belief on each state in $\Theta = \{\theta_1, \ldots, \theta_m\}$. Moreover, we set the error bounds $R_{\theta_p} = \frac{R}{m}$

for all $\theta_p \in \Theta$, where $R \in \mathbb{Z}_{\geq 0}$ and $R < m-1$. We then have $\bar{\Theta} = \Theta$, where $\bar{\Theta} = \{\theta_p \in \Theta : 0 \leq R_{\theta_p} < 1 - \mu_0(\theta_p)\}$. Recalling the definition of $z(\cdot)$ in Eq. (6.18), for any $\mathcal{I} \subseteq [n]$, we define

$$z'(\mathcal{I}) \triangleq m(m-R)z(\mathcal{I}) = m(m-R)\sum_{\theta_p \in \Theta} f'_{\theta_p}(\mathcal{I}). \tag{6.20}$$

One can check that $z'(\mathcal{I}) \in \mathbb{Z}_{\geq 0}$ for all $\mathcal{I} \subseteq [n]$. Moreover, one can show that (6.19) can be equivalently written as

$$\begin{aligned} &\min_{\mathcal{I} \subseteq [n]} h(\mathcal{I}) \\ &s.t.\ z'(\mathcal{I}) = z'([n]). \end{aligned} \tag{6.21}$$

Noting that $M' \triangleq \max_{j \in [n]} z'(j) \leq m^2(m-R)$ from (6.20), we then see from Theorem 6.3.4(d) that applying Algorithm 6.3.1 to (6.21) yields the following performance bound:

$$h(\mathcal{I}_g) \leq (\sum_{i=i}^{M'} \frac{1}{i}) h(\mathcal{I}^*) \leq \big(1 + \ln M'\big) h(\mathcal{I}^*) \leq \big(1 + 2\ln m + \ln(m-R)\big) h(\mathcal{I}^*). \tag{6.22}$$

Similarly, since $z'([n]) \leq m^2(m-R)$ also holds, Theorem 6.3.5(b) implies the following performance bound for Algorithm 6.3.2 when applied to (6.21):

$$h(\mathcal{I}_f) \leq \frac{1}{1-\epsilon}\big(1 + \ln z'([n])\big) h(\mathcal{I}^*) \leq \frac{1}{1-\epsilon}\big(1 + 2\ln m + \ln(m-R)\big) h(\mathcal{I}^*), \tag{6.23}$$

where $\epsilon \in (0,1)$. Again, we note from Theorem 6.3.5 that a smaller value of ϵ yields a tighter performance bound for Algorithm 6.3.2 (according to (6.23)) at the cost of slower running times. Thus, supposing m and ϵ are fixed, we see from (6.22) and (6.23) that the performance bounds of Algorithm 6.3.1 and Algorithm 6.3.2 become tighter as R increases, i.e., as the error bound R_{θ_p} increases. On the other hand, supposing R and ϵ are fixed, we see from (6.22) and (6.23) that the performance bounds of Algorithm 6.3.1 and Algorithm 6.3.2 become tighter as m decreases.

Finally, we note that the performance bounds given in Theorem 6.3.4 are worst-case performance bounds for Algorithm 6.3.1. Thus, in practice the ratio between a solution returned by the algorithm and an optimal solution can be smaller than the

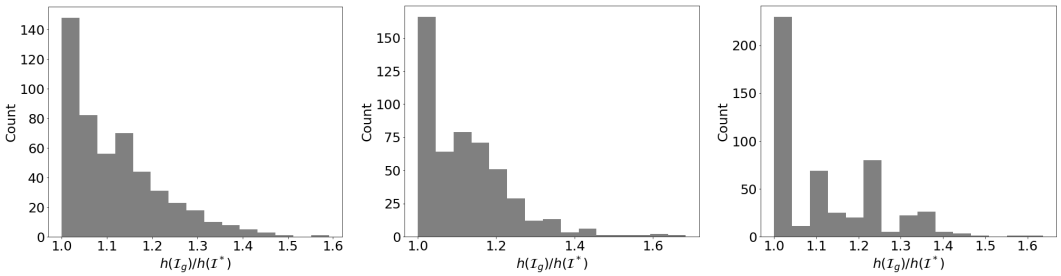
ratio predicted by Theorem 6.3.4. Nevertheless, there may also exist instances of the BLDS problem that let Algorithm 6.3.1 return a solution that meets the worst-case performance bound. Moreover, instances with tighter performance bounds (given by Theorem 6.3.4) potentially imply better performance of the algorithm when applied to those instances, as we can see from the above discussions and the numerical examples that will be provided in the next section. Therefore, the performance bounds given in Theorem 6.3.4 also provide insights into how different problem parameters of BLDS influence the actual performance of Algorithm 6.3.1. Similar arguments also hold for Algorithm 6.3.2 and the corresponding performance bounds given in Theorem 6.3.5.

6.3.3 Numerical examples

In this section, we focus on validating Algorithm 6.3.1 and the performance bounds provided in Theorem 6.3.4 using numerical examples. First, the total number of data sources is set to be 10, and the selection cost h_i is drawn uniformly from [10] for all $i \in [n]$. The cost structure is then fixed in the sequel. Similarly to Section 6.3.2, we consider BLDS instances where $\mu_0(\theta_p) = \frac{1}{m}$ for all $\theta_p \in \Theta$ with $m = |\Theta|$, and $R_{\theta_p} = \frac{R}{m}$ for all $\theta_p \in \Theta$ with $R \in \mathbb{Z}_{>0}$ and $R < m - 1$. Specifically, we set $m = 15$ and range R from 0 to 13. For each $R \in \{0, 1, \ldots, 13\}$, we further consider 500 corresponding randomly generated instances of the BLDS problem, where for each BLDS instance we randomly generate the set $F^c_{\theta_p}(i)$ (i.e., the set of states that can be distinguished from θ_p given data source i) for all $i \in [n]$ and for all $\theta_p \in \Theta$.[2] In Fig. 6.1, we plot histograms of the ratio $h(\mathcal{I}_g)/h(\mathcal{I}^*)$ for $R = 1$, $R = 5$ and $R = 10$, where $\mathcal{I}_g$ is the solution returned by Algorithm 6.3.1 and $\mathcal{I}^*$ is an optimal solution to BLDS. We

[2]Note that in the BLDS problem (Problem 6.2.3), the signal structure of each data source $i \in [n]$ is specified by the likelihood functions $\ell_i(\cdot|\theta_p)$ for all $\theta_p \in \Theta$. As we discussed in previous sections, (6.7) in Problem 6.2.3 can be equivalently written as (6.19), where one can further note that the function $z(\cdot)$ does not depend on any likelihood function $\ell_i(\cdot|\theta_p)$, and can be (fully) specified given $F^c_{\theta_p}(i)$ for all $i \in [n]$ and for all $\theta_p \in \Theta$. Thus, when constructing the BLDS instances in this section, we directly construct $F^c_{\theta_p}(i)$ for all $i \in [n]$ and for all $\theta_p \in \Theta$ in a random manner.

see from Fig. 6.1 that Algorithm 6.3.1 works well on the randomly generated BLDS instances. Moreover, we see from Fig. 6.1 that as R increases, Algorithm 6.3.1 yields better overall performance for the 500 randomly generated BLDS instances.



(a) $R = 1$. (b) $R = 5$. (c) $R = 10$.

Fig. 6.1. Histograms of the ratio $h(\mathcal{I}_g)/h(\mathcal{I}^*)$.

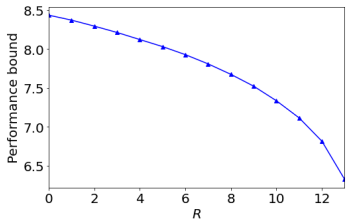


Fig. 6.2. Performance bound for Algorithm 6.3.1 given by Theorem 6.3.4(d).

Now, from the way we set $\mu_0(\theta_p)$ and R_{θ_p} in the BLDS instances constructed above, we see from the arguments in Section 6.3.2 that the performance bound for Algorithm 6.3.1 given by Theorem 6.3.4(d) can be written as $h(\mathcal{I}_g) \leq \big(1+\ln M'\big)h(\mathcal{I}^*)$, where $M' = \max_{j\in[n]} z'(j)$ and $z'(\cdot)$ is defined in (6.20). Thus, in Fig. 6.2, we plot the performance bound of Algorithm 6.3.1, i.e., $1 + \ln M'$, for R ranging from 0 to 13. Also note that for each $R \in \{0, 1, \ldots, 13\}$, we obtain the averaged value of $1+\ln M'$ over 500 random BLDS instances as we constructed above. We then see from Fig. 6.2 that the value of the performance bound of Algorithm 6.3.1 decreases, i.e., the performance bound becomes tighter, as R increases from 0 to 13. The behavior of the performance bound aligns with the actual performance of Algorithm 6.3.1 as we

presented in Fig. 6.1, i.e., a tighter performance bound potentially implies a better overall performance of the algorithm on the 500 random BLDS instances.

6.4 Extension to Non-Bayesian Learning

Let us consider a scenario where there is a set of agents, denoted as $[n]$, who wish to *collaboratively* learn the true state of the world. The agents interact over a directed graph $\mathcal{G} = ([n], \mathcal{E})$, where each vertex in $[n]$ corresponds to an agent and a directed edge $(j, i) \in \mathcal{E}$ indicates that agent i can (directly) receive information from agent j. Denote $\mathcal{N}_i \triangleq \{j : (j, i) \in \mathcal{E}, j \neq i\}$ as the set of neighbors of agent i. Suppose each agent has an associated data source with the same observation model as described in Section 6.2. Specifically, the observation (conditioned on the state $\theta \in \Theta$) provided by the data source at agent i at time step $k \in \mathbb{Z}_{\geq 1}$ is denoted as $\omega_{i,k} \in S_i$, which is generated by the likelihood function $\ell_i(\cdot|\theta)$. Each agent $i \in [n]$ is assumed to know $\ell_i(\cdot|\theta)$ for all $\theta \in \Theta$. Similarly, we consider the scenario where using the data source of agent $i \in [n]$ incurs a cost denoted as $h_i \in \mathbb{R}_{>0}$ for all $i \in [n]$, and there is a (central) designer who can select a subset $\mathcal{I} \subseteq [n]$ of agents whose data sources will be used to collect observations. We assume that the designer knows $\ell_i(\cdot|\theta)$ for all $i \in [n]$ and for all $\theta \in \Theta$. After set $\mathcal{I} \subseteq [n]$ is selected, each agent $i \in [n]$ updates its belief of the state of the world, denoted as $\mu^{\mathcal{I}}_{i,k}(\cdot)$, using the following distributed non-Bayesian learning rule as described in [100]:

$$\mu^{\mathcal{I}}_{i,k+1}(\theta) = \frac{\prod_{j=1}^{n} (\mu^{\mathcal{I}}_{j,k}(\theta))^{a_{ij}} \ell_i(\omega_{i,k+1}|\theta)}{\sum_{\theta_p \in \Theta} \prod_{j=1}^{n} (\mu^{\mathcal{I}}_{j,k}(\theta_p))^{a_{ij}} \ell_i(\omega_{i,k+1}|\theta_p)}, \tag{6.24}$$

for all $\theta \in \Theta$, where $\mu^{\mathcal{I}}_{i,k}(\theta)$ is the belief of agent i that θ is the true state at time step k when the set of sources given by $\mathcal{I} \subseteq [n]$ is selected, and a_{ij} is the weight that agent $i \in [n]$ assigns to an agent $j \in \mathcal{N}_i \cup \{i\}$. Specifically, for any two distinct agents $i, j \in [n]$, $a_{ij} > 0$ if agent i receives information from agent j and $a_{ij} = 0$ otherwise, where $\sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} = 1$. Note that if agent $i \notin \mathcal{I}$, i.e., the data source of agent i is not selected to collect observations, we set $\ell_i(s_i|\theta_p) = \ell_i(s_i|\theta_q)$ for all $\theta_p, \theta_q \in \Theta$ and for all $s_i \in S_i$. Similarly, for any $i \in [n]$, the initial belief is set to be $\mu^{\mathcal{I}}_{i,0}(\theta) = \mu_{i,0}(\theta)$

for all $\mathcal{I} \subseteq [n]$ and for all $\theta \in \Theta$, where $\sum_{\theta \in \Theta} \mu_{i,0}(\theta) = 1$ and $\mu_{i,0}(\theta) \in \mathbb{R}_{\geq 0}$ for all $\theta \in \Theta$. We then have from (6.24) that $\sum_{\theta \in \Theta} \mu_{i,k}^{\mathcal{I}}(\theta) = 1$ and $0 \leq \mu_{i,k}^{\mathcal{I}}(\theta) \leq 1$ for all $k \in \mathbb{Z}_{\geq 0}$, for all $\theta \in \Theta$ and for all $\mathcal{I} \subseteq [n]$. Moreover, for a given true state $\theta \in \Theta$, we define $F_\theta(i) = \{\theta_p \in \Theta : \ \ell_i(s_i|\theta_p) = \ell_i(s_i|\theta), \forall s_i \in S_i\}$ for all $i \in [n]$, and similarly denote $F_\theta(\mathcal{I}) = \bigcap_{n_i \in \mathcal{I}} F_\theta(n_i)$, where we note that $F_\theta(\emptyset) = \Theta$, and $\theta \in F_\theta(\mathcal{I})$ for all $\theta \in \Theta$ and for all $\mathcal{I} \subseteq [n]$. We have the following result.

Lemma 6.4.1 *Consider a set $[n]$ of agents interacting over a strongly connected graph $\mathcal{G} = ([n], \mathcal{E})$.*[3] *Suppose the true state of the world is θ^*, $\mu_{i,0}(\theta) > 0$ for all $i \in [n]$ and for all $\theta \in \Theta$, and $a_{ii} > 0$ for all $i \in [n]$ in the rule given in (6.24). For any $\mathcal{I} \in [n]$, the rule given in (6.24) ensures that (a) $\lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta_p) = 0$ a.s. for all $\theta_p \notin F_{\mathcal{I}}(\theta^*)$ and for all $i \in [n]$, and (b) $\lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta_q) = \frac{\prod_{j=1}^n \mu_{j,0}(\theta_q)^{\pi_j}}{\sum_{\theta \in F_{\mathcal{I}}(\theta^*)} \prod_{j=1}^n \mu_{j,0}(\theta)^{\pi_j}}$ a.s. for all $i \in [n]$ and $\theta_q \in F_{\mathcal{I}}(\theta^*)$, where $\pi \triangleq \begin{bmatrix} \pi_1 & \dots & \pi_n \end{bmatrix}^T$ satisfies $\pi^T A = \pi^T$ and $\|\pi\|_1 = 1$, and $A \in \mathbb{R}^{n\times n}$ is defined such that $A_{ij} = a_{ij}$ for all $i, j \in [n]$.*

Proof We begin by defining the following quantities for all $\mathcal{I} \subseteq [n]$, for all $i \in [n]$ and for all $k \in \mathbb{Z}_{\geq 0}$:

$$\delta_{i,k}^{\mathcal{I}}(\theta) \triangleq \ln \frac{\mu_{i,k}^{\mathcal{I}}(\theta)}{\mu_{i,k}^{\mathcal{I}}(\theta^*)} \text{ and } \sigma_{i,k+1}(\theta) \triangleq \ln \frac{\ell_i(\omega_{i,k+1}|\theta)}{\ell_i(\omega_{i,k+1}|\theta^*)}, \tag{6.25}$$

where $\delta_{i,0}^{\mathcal{I}}(\theta) = \delta_{i,0}(\theta) \triangleq \ln \frac{\mu_{i,0}(\theta)}{\mu_{i,0}(\theta^*)}$ for all $\mathcal{I} \subseteq [n]$. For any $\mathcal{I} \subseteq [n]$, we consider an agent $i \in [n]$ and $\theta_p \notin F_{\theta^*}(\mathcal{I})$. Following similar arguments to those for Theorem 1 in [100], we can obtain that $\lim_{k\to\infty} \delta_{i,k}^{\mathcal{I}}(\theta_p) = -\infty$ a.s., i.e., $\lim_{k\to\infty} \frac{\mu_{i,k}^{\mathcal{I}}(\theta_p)}{\mu_{i,k}^{\mathcal{I}}(\theta^*)} = 0$ a.s. Since $0 \leq \mu_{i,k}^{\mathcal{I}}(\theta) \leq 1$ for all $\theta \in \Theta$ and for all $k \in \mathbb{Z}_{\geq 0}$, it follows that $\lim_{k\to\infty} \frac{\mu_{i,k}^{\mathcal{I}}(\theta_p)}{\mu_{i,k}^{\mathcal{I}}(\theta^*)} \geq \lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta_p) \geq 0$, which implies $0 \leq \lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta_p) \leq 0$ a.s., i.e., $\lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta_p) = 0$ a.s. This proves Part (a).

We then prove Part (b). For any $\mathcal{I} \subseteq [n]$, we now consider an agent $i \in [n]$ and $\theta_q \in F_{\theta^*}(\mathcal{I})$. Based on the definition of $F_{\theta^*}(\mathcal{I})$, we note that $\sigma_{i,k+1}(\theta_q) = 0, \forall k \in \mathbb{Z}_{\geq 0}$. We then obtain from (6.24) the following:

$$\delta_{k+1}^{\mathcal{I}}(\theta_q) = A\delta_k^{\mathcal{I}}(\theta_q),$$

[3] A directed graph $\mathcal{G} = ([n], \mathcal{E})$ is said to be strongly connected if for each pair of distinct vertices $i, j \in [n]$, there exists a directed path (i.e., a sequence of directed edges) from j to i.

where $\delta_k^{\mathcal{I}}(\theta_q) \triangleq \begin{bmatrix}\delta_{1,k}^{\mathcal{I}}(\theta_q) & \cdots & \delta_{n,k}^{\mathcal{I}}(\theta_q)\end{bmatrix}^T$. Moreover, we have

$$\lim_{k\to\infty} \delta_k^{\mathcal{I}}(\theta_q) = (\lim_{k\to\infty} A^k)\delta_0(\theta_q) = \mathbf{1}_n \pi^T \delta_0(\theta_q), \tag{6.26}$$

where the last equality follows from the fact that A is an irreducible and aperiodic stochastic matrix based on the hypotheses of the lemma. Simplifying (6.26), we obtain

$$\lim_{k\to\infty} \frac{\mu_{i,k}^{\mathcal{I}}(\theta_q)}{\mu_{i,k}^{\mathcal{I}}(\theta^*)} = \frac{\prod_{j=1}^n \mu_{j,0}(\theta_q)^{\pi_j}}{\prod_{j=1}^n \mu_{j,0}(\theta^*)^{\pi_j}} > 0. \tag{6.27}$$

Summing up Eq. (6.27) for all $\theta_q \in F_{\theta^*}(\mathcal{I})$, we have

$$\lim_{k\to\infty} \frac{\sum_{\theta_q\in F_{\theta^*}(\mathcal{I})} \mu_{i,k}^{\mathcal{I}}(\theta_q)}{\mu_{i,k}^{\mathcal{I}}(\theta^*)} = \sum_{\theta_q\in F_{\theta^*}(\mathcal{I})} \frac{\prod_{j=1}^n \mu_{j,0}(\theta_q)^{\pi_j}}{\prod_{j=1}^n \mu_{j,0}(\theta^*)^{\pi_j}} > 0. \tag{6.28}$$

Noting from Part (a) that $\lim_{k\to\infty} \sum_{\theta_q\in\ F_{\theta^*}(\mathcal{I})} \mu_{i,k}^{\mathcal{I}}(\theta_1) = 1$ a.s., we see from (6.28) that $\lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta^*)$ exists and is positive, a.s., which further implies via (6.27) that $\lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta_q)$ exists and is positive, a.s. In other words, we have from (6.28) the following:

$$\frac{\mu_{i,\infty}^{\mathcal{I}}(\theta_q)}{\mu_{i,\infty}^{\mathcal{I}}(\theta^*)} = \frac{\prod_{j=1}^n \mu_{j,0}(\theta_q)^{\pi_j}}{\prod_{j=1}^n \mu_{j,0}(\theta^*)^{\pi_j}}, \tag{6.29}$$

where $\mu_{i,\infty}^{\mathcal{I}}(\theta_q) \triangleq \lim_{k\to\infty} \mu_{i,k}^{\mathcal{I}}(\theta_q)$ for all $\theta_q \in F_{\theta^*}(\mathcal{I})$. Since $\lim_{k\to\infty} \sum_{\theta\in\ F_{\theta^*}(\mathcal{I})} \mu_{i,k}^{\mathcal{I}}(\theta) = 1$ a.s. for all $i \in [n]$, part (b) then follows from Eq. (6.29). ■

Proceeding with the problem formulation described in Section 6.2, we define the following error metric for the designer:

$$\bar{e}_{\mathcal{I}}(\theta^*) = \sum_{i=1}^n e_{\mathcal{I},i}(\theta^*), \tag{6.30}$$

where θ^* is the true state of the world, $e_{\mathcal{I},i}(\theta^*) \triangleq \frac{1}{2}\lim_{k\to\infty} \|\mu_{i,k}^{\mathcal{I}} - \mathbf{1}_{\theta^*}\|_1$ and $\mu_{i,k}^{\mathcal{I}} \triangleq \begin{bmatrix}\mu_{i,k}^{\mathcal{I}}(\theta_1) & \cdots & \mu_{i,k}^{\mathcal{I}}(\theta_m)\end{bmatrix}^T$. In other words, $\bar{e}_{\mathcal{I}}(\theta^*)$ is the sum of the steady-state learning errors of all the agents in $[n]$, when the true state of the world is assumed to be θ^*. It then follows from Lemma 6.4.1 that $\bar{e}_{\mathcal{I}}(\theta^*) = n\big(1 - \frac{\prod_{j=1}^n (\mu_{j,0}(\theta^*))^{\pi_j}}{\sum_{\theta\in F_{\theta^*}(\mathcal{I})} \prod_{j=1}^n (\mu_{j,0}(\theta))^{\pi_j}}\big)$ almost surely. Denoting

$$\bar{e}_{\mathcal{I}}^s(\theta_p) \triangleq n\big(1 - \frac{\prod_{j=1}^n (\mu_{j,0}(\theta_p))^{\pi_j}}{\sum_{\theta\in F_{\theta_p}(\mathcal{I})} \prod_{j=1}^n (\mu_{i,0}(\theta))^{\pi_j}}\big)\ \forall \theta_p \in \Theta, \tag{6.31}$$

we consider the following problem for the designer:

$$\begin{aligned} &\min_{\mathcal{I}\subseteq[n]} h_{\mathcal{I}} \\ &s.t.\ \bar{e}^s_{\mathcal{I}}(\theta_p) \leq \bar{R}_{\theta_p}, \forall \theta_p \in \Theta, \end{aligned} \tag{6.32}$$

where $0 \leq \bar{R}_{\theta_p} \leq n$ and $\bar{R}_{\theta_p} \in \mathbb{R}$. Denoting $\bar{\mu}_0(\theta) \triangleq \prod_{i=1}^n \mu_{i,0}(\theta)^{\pi_i}$ for all $\theta \in \Theta$, we have from (6.31) that

$$\bar{e}^s_{\mathcal{I}}(\theta_p) = n(1 - \frac{\bar{\mu}_0(\theta_p)}{\sum_{\theta \in F_{\theta_p}(\mathcal{I})} \bar{\mu}_0(\theta)}), \forall \theta_p \in \Theta. \tag{6.33}$$

Under Assumption 6.2.1, i.e., $F_\theta(\mathcal{I}) = \bigcap_{n_i \in \mathcal{I}} F_\theta(n_i)$ for all $\mathcal{I} \subseteq [n]$ and $\theta \in \Theta$, we see from (6.6) and (6.33) that the optimization problem (6.32) can be viewed as an instance of Problem 6.2.3.

6.5 Chapter Summary

In this chapter, we considered the problem of data source selection for Bayesian learning. We showed that the data source selection problem for Bayesian learning is NP-hard in general. Next, we showed that the data source selection problem can be transformed into an instance of the submodular set covering problem, and can then be solved using a standard greedy algorithm with provable performance guarantees. We also proposed a fast greedy algorithm that improves the running times of the standard greedy algorithm, while achieving performance guarantees that are comparable to those of the standard greedy algorithm. Finally, we illustrated the obtained performance bounds for the greedy algorithms using examples, and showed that the performance bounds provide insights into the actual performances of the algorithms under different instances of the data source selection problem.

6.6 Proofs of Key Results

6.6.1 Proof of Theorem 6.3.5

Consider any $\epsilon \in (0,1)$. We first show that the query complexity of Algorithm 6.3.2 is $O(\frac{n}{\epsilon}\ln\frac{n}{\epsilon})$. Note that the for loop in lines 3-9 runs for at most $K_{\max} \triangleq \lceil \frac{1}{-\ln(1-\epsilon)} \cdot (\ln\frac{n}{\epsilon} + \ln\frac{h_{\max}}{h_{\min}}) \rceil$ iterations, where each iteration requires $O(n)$ evaluations of $z(\cdot)$. One can also show that $-\ln(1-\epsilon) - \epsilon > 0$ for $\epsilon \in (0,1)$, which implies $K_{\max} \leq \frac{1}{\epsilon} \cdot (\ln\frac{n}{\epsilon} + H\ln n) + 1 \leq \frac{1}{\epsilon}((H+1)\ln\frac{n}{\epsilon} + 1)$, where $H \in \mathbb{R}_{\geq 1}$ is a fixed constant. It then follows from the above arguments that the query complexity of Algorithm 6.3.2 is $O(\frac{n}{\epsilon}\ln\frac{n}{\epsilon})$.

Next, we show that $\mathcal{I}_f$ satisfies $z(\mathcal{I}_f) \geq (1-\epsilon)z([n])$. Note that if Algorithm 6.3.2 ends with line 9, then $z(\mathcal{I}_f) = z([n])$ and thus $z(\mathcal{I}_f) \geq (1-\epsilon)z([n])$ hold. Hence, we assume that Algorithm 6.3.2 ends with $\tau = \frac{\epsilon h_{\min}}{n h_{\max}} d$ in the for loop in lines 3-9. Also note that $z(\emptyset) = 0$. Denoting $j^* \in \arg\max_{i\in[n]} \frac{z(i)-z(\emptyset)}{h_i}$ and considering any $j \in [n] \setminus \mathcal{I}_f$, we have from the definition of Algorithm 6.3.2 the following:

$$\begin{aligned} &\frac{z(\mathcal{I}_f \cup \{j\}) - z(\mathcal{I}_f)}{h_j} < \frac{\epsilon h_{\min} z(j^*)}{n h_{\max} h_{j^*}}, \\ \implies &z(\mathcal{I}_f \cup \{j\}) - z(\mathcal{I}_f) < \frac{\epsilon}{n} z(j^*) \leq \frac{\epsilon}{n} z([n]), \end{aligned} \tag{6.34}$$

where we use the facts $h_j \leq h_{\max}$ and $h_{j^*} \geq h_{\min}$ to obtain the first inequality in (6.34), and use the fact that $z(\cdot)$ is monotone nondecreasing to obtain the second inequality in (6.34). Since (6.34) holds for all $j \in [n] \setminus \mathcal{I}_f$, it follows that

$$\sum_{j\in[n]\setminus\mathcal{I}_f} z(\mathcal{I}_f \cup \{j\}) - z(\mathcal{I}_f) < \epsilon z([n]) \implies z([n]) - z(\mathcal{I}_f) < \epsilon z([n]),$$

where we use the submodularity of $z(\cdot)$ (i.e., (6.10) in Definition 6.3.1).

We now prove part (a). Denote $\mathcal{I}_f^t = \{j_1, \ldots, j_t\} \subseteq [n]$ for all $t \in [T]$ with $\mathcal{I}_f^0 = \emptyset$ in Algorithm 6.3.2. First, suppose $T \geq 2$. Considering any $t \in [T-1]$, we have from line 5 in Algorithm 6.3.2:

$$\frac{z(\mathcal{I}_f^t \cup \{j_{t+1}\}) - z(\mathcal{I}_f^t)}{h_{j_{t+1}}} \geq \tau. \tag{6.35}$$

Moreover, consider any $j \in [n] \setminus \mathcal{I}_f^t$. Since j has not been added to $\mathcal{I}_f^t$ while the current threshold is τ, one can see that j does not satisfy the threshold condition in line 5 when the threshold was $\frac{\tau}{1-\epsilon}$, i.e.,

$$\frac{z(\mathcal{I}_f^{t'} \cup \{j\}) - z(\mathcal{I}_f^{t'})}{h_j} \leq \frac{\tau}{1-\epsilon} \implies \frac{z(\mathcal{I}_f^{t} \cup \{j\}) - z(\mathcal{I}_f^{t})}{h_j} \leq \frac{\tau}{1-\epsilon}, \tag{6.36}$$

where $t' \in \{0, \ldots, T-1\}$ with $t' < t$ is a corresponding time step in Algorithm 6.3.2 when the threshold was $\frac{\tau}{1-\epsilon}$. Note that we obtain the second inequality in (6.36) using again the submodularity of $z([n])$ (i.e., (6.9) in Definition 6.3.1). Combining (6.35) and (6.36), we have

$$\frac{z(\mathcal{I}_f^t \cup \{j_{t+1}\}) - z(\mathcal{I}_f^t)}{h_{j_{t+1}}} \geq \frac{(1-\epsilon)(z(\mathcal{I}_f^t \cup \{j\}) - z(\mathcal{I}_f^t))}{h_j}. \tag{6.37}$$

Noting that (6.37) holds for all $j \in \mathcal{I}^* \setminus \mathcal{I}_f^t$, one can show that

$$\frac{z(\mathcal{I}_f^t \cup \{j_{t+1}\}) - z(\mathcal{I}_f^t)}{h_{j_{t+1}}} \geq \frac{(1-\epsilon)\sum_{j \in \mathcal{I}^* \setminus \mathcal{I}_f^t}(z(\mathcal{I}_f^t \cup \{j\}) - z(\mathcal{I}_f^t))}{\sum_{j \in \mathcal{I}^* \setminus \mathcal{I}_f^t} h_j}, \tag{6.38}$$

which further implies, via the fact that $z(\cdot)$ is submodular and monotone nondecreasing, the following:

$$\frac{z(\mathcal{I}_f^t \cup \{j_{t+1}\}) - z(\mathcal{I}_f^t)}{h_{j_{t+1}}} \geq \frac{(1-\epsilon)(z(\mathcal{I}^* \cup \mathcal{I}_f^t) - z(\mathcal{I}_f^t))}{h(\mathcal{I}^* \setminus \mathcal{I}_f^t)} \geq \frac{(1-\epsilon)(z(\mathcal{I}^*) - z(\mathcal{I}_f^t))}{h(\mathcal{I}^*)}. \tag{6.39}$$

Rearranging the terms in (6.39), we have

$$\begin{aligned} z(\mathcal{I}^*) - z(\mathcal{I}_f^t) &\leq \frac{h(\mathcal{I}^*)}{1-\epsilon} \cdot \frac{z(\mathcal{I}^*) - z(\mathcal{I}_f^t) - (z(\mathcal{I}^*) - z(\mathcal{I}_f^{t+1}))}{h_{j_{t+1}}}, \\ \implies z(\mathcal{I}^*) - z(\mathcal{I}_f^{t+1}) &\leq (1 - \frac{(1-\epsilon)h_{j_t+1}}{h(\mathcal{I}^*)})(z(\mathcal{I}^*) - z(\mathcal{I}_f^t)). \end{aligned} \tag{6.40}$$

Moreover, we see from the above arguments that (6.40) holds for all $t \in [T-1]$. Now, considering $t = 0$ and using similar arguments to those above, we can show that (6.37) and thus (6.40) also hold. Therefore, viewing (6.40) as a recursion of $z(\mathcal{I}^*) - z(\mathcal{I}_f^t)$ for $t \in \{0, \ldots, T-1\}$, we obtain the following:

$$z(\mathcal{I}^*) - z(\mathcal{I}_f^{T-1}) \leq (z(\mathcal{I}^*) - z(\mathcal{I}_f^0)) \prod_{t=1}^{T-1} (1 - \frac{h_{j_t}(1-\epsilon)}{h(\mathcal{I}^*)}). \tag{6.41}$$

One can also show that $\prod_{t=1}^{T-1}(1-\frac{h_{j_t}(1-\epsilon)}{h(\mathcal{I}^*)}) \leq (1-\frac{h(\mathcal{I}_f^{T-1})(1-\epsilon)}{(T-1)h(\mathcal{I}^*)})^{T-1} \leq e^{-(1-\epsilon)\frac{h(\mathcal{I}_f^{T-1})}{h(\mathcal{I}^*)}}$ (e.g., [31]). Since $z(\mathcal{I}_f^0) = z(\emptyset) = 0$ and $z(\mathcal{I}^*) = z([n])$, it then follows from (6.41) that

$$\begin{aligned} & z(\mathcal{I}^*) - z(\mathcal{I}_f^{T-1}) \leq z(\mathcal{I}^*) e^{-(1-\epsilon)\frac{h(\mathcal{I}_f^{T-1})}{h(\mathcal{I}^*)}}, \\ \Longrightarrow\ & \ln(z([n] - z(\mathcal{I}_f^{T-1}))) \leq -(1-\epsilon)\frac{h(\mathcal{I}_f^{T-1})}{h(\mathcal{I}^*)} + \ln z([n]), \\ \Longrightarrow\ & h(\mathcal{I}_f^{T-1}) \leq \frac{1}{1-\epsilon} \ln \frac{z([n])}{z([n]) - z(\mathcal{I}_f^{T-1})} h(\mathcal{I}^*), \end{aligned} \tag{6.42}$$

where we note that $z([n]) - z(\mathcal{I}_f^{T-1}) > 0$, since $z(\cdot)$ is monotone nondecreasing and $z(\mathcal{I}_f^{T-1}) \neq z([n])$. In order to prove part (a) (for $T \geq 2$), it remains to show that $h_{j_T} \leq \frac{1}{1-\epsilon} h(\mathcal{I}^*)$, which together with (6.42) yield the bound in part (a). We can now use (6.38) with $t = T-1$ to obtain

$$\begin{aligned} h_{j_T} &\leq \frac{h(\mathcal{I}^* \setminus \mathcal{I}_f^{T-1})}{1-\epsilon} \cdot \frac{z(\mathcal{I}_f^T) - z(\mathcal{I}_f^{T-1})}{\sum_{j \in \mathcal{I}^* \setminus \mathcal{I}_f^{T-1}} z(\mathcal{I}_f^{T-1} \cup \{j\}) - z(\mathcal{I}_f^{T-1})} \\ &\leq \frac{h(\mathcal{I}^*)}{1-\epsilon} \cdot \frac{z(\mathcal{I}_f^T) - z(\mathcal{I}_f^{T-1})}{z(\mathcal{I}_f^{T-1} \cup \mathcal{I}^*) - z(\mathcal{I}_f^{T-1})}, \end{aligned} \tag{6.43}$$

where (6.43) follows from the submodularity of $z(\cdot)$. Since $z(\mathcal{I}_f^T) \leq z(\mathcal{I}_f^{T-1} \cup \mathcal{I}^*)$ from the facts that $z(\mathcal{I}^*) = z([n])$ and $z(\cdot)$ is monotone nondecreasing, we see from (6.43) that $h_{j_T} \leq \frac{1}{1-\epsilon} h(\mathcal{I}^*)$.

Next, suppose $T = 1$, i.e., $\mathcal{I}_f = j_1$. We will show that $h(\mathcal{I}^*) = h(\mathcal{I}_g)$. Noting from the definition of Algorithm 6.3.2 that $j_1 \in \arg\max_{i \in [n]} \frac{z(i) - z(\emptyset)}{h_i}$, we have

$$\frac{z(j_1)}{h_{j_1}} \geq \frac{z(j)}{h_j},\ \forall j \in \mathcal{I}^*.$$

It then follows from similar arguments to those for (6.38) and (6.39) that

$$\frac{z(j_1)}{h_{j_1}} \geq \frac{\sum_{j \in \mathcal{I}^*} z(j)}{\sum_{j \in \mathcal{I}^*} h_j} \geq \frac{z(\mathcal{I}^*)}{h(\mathcal{I}^*)},$$

which implies

$$\frac{h(\mathcal{I}_f)}{h(\mathcal{I}^*)} \leq \frac{z(\mathcal{I}_f)}{z(\mathcal{I}^*)} \leq 1,$$

where we use the fact $z(\mathcal{I}_f) \leq z(\mathcal{I}^*)$, since $z(\cdot)$ is monotone nondecreasing with $z(\mathcal{I}^*) = z([n])$. Thus, we have $h(\mathcal{I}_f) \leq h(\mathcal{I}^*)$. Noting that $h(\mathcal{I}^*) \leq h(\mathcal{I}_g)$ always holds due to the fact that $\mathcal{I}^*$ is an optimal solution, we conclude that $h(\mathcal{I}^*) = h(\mathcal{I}_g)$. This completes the proof of part (a).

Part (b) now follows directly from part (a) by noting that $z([n]) - z(\mathcal{I}_f^{T-1}) \geq 1$, since $z([n]) - z(\mathcal{I}_f^{T-1}) > 0$ and $z(\mathcal{I}) \in \mathbb{Z}_{\geq 1}$ for all $\mathcal{I} \subseteq [n]$. ■

7. PARAMETER ESTIMATION IN EPIDEMIC SPREAD NETWORKS USING LIMITED MEASUREMENTS

7.1 Introduction

Following our discussions in the previous chapters, we extend our analysis to models of spreading processes over networks in this chapter. Such models have been widely studied by researchers from different fields (e.g., [18, 26–30]). The case of epidemics spreading through networked populations has received a particularly significant amount of attention, especially in light of the ongoing COVID-19 pandemic (e.g., [30, 106]). A canonical example is the networked SIR model, where each node in the network represents a subpopulation or an individual, and can be in one of three states: susceptible (S), infected (I), or recovered (R) [107]. There are two key parameters that govern such models: the infection rate of a given node, and the recovery rate of that node. In the case of a novel virus, these parameters may not be known a priori, and must be identified or estimated from gathered data, including for instance the number of infected and recovered individuals in the network at certain points of time. For instance, in the COVID-19 pandemic, when collecting the data on the number of infected individuals or the number of recovered individuals in the network, one possibility is to perform virus or antibody tests on the individuals, with each test incurring a cost. Therefore, in the problem of parameter estimation in epidemic spread networks, it is important and of practical interest to take the costs of collecting the data (i.e., measurements) into account in the problem formulation.

The above discussions motivate us to consider the measurement selection problem for parameter estimation problem in epidemic spread networks, which shares natural similarities to the sensor selection problems that we studied in Chapters 3-5. Note that measurements are collected using sensors in the sensor selection problem, while the

measurements are gathered by performing virus or antibody tests on the individuals in the measurement selection problem that we will consider in this chapter. Specifically, in the parameter estimation measurement selection problem, the goal is to exactly identify (when possible) or estimate the parameters in the networked SIR model using a limited number of measurements. We divide our analysis into two scenarios: 1) when the measurements (e.g., the number of infected individuals) can be collected exactly without error; and 2) when only a stochastic measurement can be obtained.

Under the setting when exact measurements of the infected and recovered proportions of the population at certain nodes in the network can be obtained, we formulate the Parameter Identification Measurement Selection (PIMS) problem as minimizing the cost spent on collecting the measurements, while ensuring that the parameters of the SIR model can be uniquely identified (within a certain time interval in the epidemic dynamics). In settings where the measurements are stochastic (thereby precluding exact identification of the parameters), we formulate the Parameter Estimation Measurement Selection (PEMS) problem. The goal is to optimize certain estimation performance metrics based on the collected measurements, while satisfying the budget on collecting the measurements. We summarize some related work as follows.

Related Work

The authors in [108, 109] studied the parameter estimation problem in epidemic spread networks using a "Susceptible-Infected-Susceptible (SIS)" model of epidemics. When exact measurements of the infected proportion of the population at each node of the network can be obtained, the authors proposed a sufficient and necessary condition on the set of the collected measurements such that the parameters of the SIS model (i.e., the infection rate and the recovery rate) can be uniquely identified. However, this condition does not pose any constraint on the number of measurements that can be collected.

In [110], the authors considered a measurement selection problem in the SIR model. Their problem is to perform a limited number of virus tests among the population such that the probability of undetected asymptotic cases is minimized. The transmission of the disease in the SIR model considered in [110] is characterized by a Bernoulli random variable which leads to a Hidden Markov Model for the SIR dynamics.

Finally, our work is also closely related to the sensor selection problem that has been studied for control systems and signal processing (e.g., Chapters 3-5), and machine learning (e.g., [5]). The goal of these problems is to optimize certain (problem-specific) performance metrics of the estimate based on the measurements of the selected sensors, while satisfying the sensor selection budget constraints.

Summary of Results

In this chapter, we first show that the PIMS problem is NP-hard, and provide an approximation algorithm that returns a solution that is within a certain approximation ratio of the optimal. The approximation factor depends on the cost structure of the measurements and on the graph structure of the epidemic spread network. Next, we show that the PEMS problem is also NP-hard, but it is possible to transform the problem into an instance of the problem of maximizing a set function subject to a knapsack constraint. We then apply a greedy algorithm to the PEMS problem, and provide performance guarantees for the greedy algorithm. Finally, we use numerical examples to validate the obtained performance bounds of the greedy algorithm, and show that the greedy algorithm performs well in practice.

The results presented in this chapter are based on a work submitted to SIAM Journal on Control and Optimization for review.

7.2 Model of Epidemic Spread Network

Suppose a disease (or virus) is spreading over a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} \triangleq [n]$ is the set of n nodes, and $\mathcal{E}$ is the set of directed edges (and self loops) that captures the interactions among the nodes in $\mathcal{V}$. Here, each node $i \in \mathcal{V}$ is considered to be a group (or population) of individuals (e.g., a city or a country). A directed edge from node i to node j, where $i \neq j$, is denoted by (i, j). For all $i \in \mathcal{V}$, denote $\mathcal{N}_i \triangleq \{j : (j, i) \in \mathcal{E}\}$ and $\bar{\mathcal{N}}_i \triangleq \{j : (j, i) \in \mathcal{E}\} \cup \{i\}$. For all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$, we let $s_i[k]$, $x_i[k]$ and $r_i[k]$ represent the proportion of population of node $i \in \mathcal{V}$ that is susceptible, infected and recovered at time k, respectively. To describe the dynamics of the spread of the disease in $\mathcal{G}$, we will use the following discrete-time SIR model (e.g., [111]), which is a straightforward extension of the discrete-time SIS model studied in, e.g., [108]:

$$s_i[k+1] = s_i[k] - hs_i[k]\beta \sum_{j \in \bar{\mathcal{N}}_i} a_{ij} x_j[k], \tag{7.1a}$$

$$x_i[k+1] = (1 - h\delta)x_i[k] + hs_i[k]\beta \sum_{j \in \bar{\mathcal{N}}_i} a_{ij} x_j[k], \tag{7.1b}$$

$$r_i[k+1] = r_i[k] + h\delta x_i[k], \tag{7.1c}$$

where $\beta \in \mathbb{R}_{\geq 0}$ is the infection rate of the disease, $\delta \in \mathbb{R}_{\geq 0}$ is the recovery rate of the disease, $h \in \mathbb{R}_{\geq 0}$ is the sampling parameter, and $a_{ij} \in \mathbb{R}_{\geq 0}$ is the weight associated with edge (j, i). Letting $A \in \mathbb{R}^{n \times n}$ be a weight matrix, where $A_{ij} = a_{ij}$ for all $i, j \in \mathcal{V}$ such that $j \in \bar{\mathcal{N}}_i$, and $A_{ij} = 0$ otherwise, one can write Eq. (7.1) as

$$s[k+1] = s[k] - hS[k]\beta Ax[k], \tag{7.2a}$$

$$x[k+1] = (1 - h\delta)x[k] + hS[k]\beta Ax[k], \tag{7.2b}$$

$$r[k+1] = r[k] + h\delta x[k], \tag{7.2c}$$

where $s[k] \triangleq \begin{bmatrix} s_1[k] & \cdots & s_n[k] \end{bmatrix}^T \in \mathbb{R}^n$, $x[k] \triangleq \begin{bmatrix} x_1[k] & \cdots & x_n[k] \end{bmatrix}^T \in \mathbb{R}^n$, $r[k] \triangleq \begin{bmatrix} r_1[k] & \cdots & r_n[k] \end{bmatrix}^T \in \mathbb{R}^n$, and $S[k] \triangleq \operatorname{diag}(s_1[k], \ldots, s_n[k]) \in \mathbb{R}^{n \times n}$. Suppose the weight matrix A and the sampling parameter h are known. Given parameters β and

δ, and initial conditions $s[0]$, $x[0]$ and $r[0]$, one can obtain states $s_i[k]$, $x_i[k]$ and $r_i[k]$ using Eq. (7.1) for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 1}$.

7.3 Preliminaries

In this section, we provide some preliminaries that will be useful for later analysis. Recall that for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$, $s_i[k]$, $x_i[k]$ and $r_i[k]$ represent the proportion of the population of node i that is susceptible, infected and recovered at time k, respectively. Hence, we make the following assumptions on the initial conditions $s[0]$, $x[0]$ and $r[0]$.

Assumption 7.3.1 *For all $i \in \mathcal{V}$, we assume that $s_i[0] \in (0, 1]$, $x_i[0] \in [0, 1)$, $r_i[0] = 0$, and $s_i[0] + x_i[0] = 1$.*

Similarly to [108, 111], we make the following assumption on the parameters of the SIR model in Eq. (7.1).

Assumption 7.3.2 *We assume that $h, \beta, \delta \in \mathbb{R}_{>0}$ and $h\delta < 1$. For all $i, j \in \mathcal{V}$ with $(j, i) \in \mathcal{E}$ and $i \neq j$, we assume that $a_{ij} \in \mathbb{R}_{>0}$. For all $i \in \mathcal{V}$, we assume that $h\beta \sum_{j \in \bar{\mathcal{N}}_i} a_{ij} < 1$.*

Next, we recall the following definition from graph theory (e.g, [71]).

Definition 7.3.1 *Consider a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $\mathcal{V} = \{1, \ldots, n\}$. A directed path of length t from node i_0 to node i_t in $\mathcal{G}$ is a sequence of t directed edges $(i_0, i_1), \ldots, (i_{t-1}, i_t)$. For any distinct pair of nodes $i, j \in \mathcal{V}$ such that there exists a directed path from i to j, the distance from node i to node j, denoted as d_{ij}, is defined as the shortest length over all such paths.*

Based on Definition 7.3.1, we give the following definition.

Definition 7.3.2 *Define $\mathcal{S}_I \triangleq \{i : x_i[0] > 0, i \in \mathcal{V}\}$ and $\mathcal{S}_H \triangleq \{i : x_i[0] = 0, i \in \mathcal{V}\}$. For all $i \in \mathcal{S}_H$, define $d_i \triangleq \min_{j \in \mathcal{S}_I} d_{ji}$, where $d_i \geq 1$ and $d_i \triangleq +\infty$ if there is no path from j to i for any $j \in \mathcal{S}_I$. For all $i \in \mathcal{S}_I$, define $d_i \triangleq 0$.*

In other words, for any $i \in \mathcal{S}_H$, d_i is the shortest distance among all the distances from the nodes in $\mathcal{S}_I$ to i. Using similar arguments to those in, e.g., [111], one can show that $s_i[k], x_i[k], r_i[k] \in [0,1]$ with $s_i[k] + x_i[k] + r_i[k] = 1$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$ under Assumptions 7.3.1-7.3.2. Therefore, given $x_i[k]$ and $r_i[k]$, we can obtain $s_i[k] = 1 - x_i[k] - r_i[k]$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$. Leveraging the structure of the graph $\mathcal{G}$, we further obtain the following result that characterizes properties of the dynamics of $x_i[k]$ and $r_i[k]$ in the SIR model over $\mathcal{G}$ given by Eq. (7.1). The proof of the following result is included in Section 7.8.1.

Lemma 7.3.3 *Consider a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $\mathcal{V} = \{1, \ldots, n\}$ and the SIR dynamics given by Eq.* (7.1)*. Suppose Assumptions* 7.3.1-7.3.2 *hold. Then, the following results hold for all $i \in \mathcal{V}$, where $k \in \mathbb{Z}_{\geq 0}$, and $\mathcal{S}_H$ and d_i are defined in Definition* 7.3.2*.*

(a) $s_i[k] > 0$ for all $k \geq 0$.

(b) If $d_i \neq +\infty$, then $x_i[k] = 0$ for all $k < d_i$, and $x_i[k] \in (0,1)$ for all $k \geq d_i$.[1]

(c) If $d_i \neq +\infty$, then $r_i[k] = 0$ for all $k \leq d_i$, and $r_i[k] \in (0,1)$ for all $k > d_i$.

(d) If $i \in \mathcal{S}_H$ with $d_i = +\infty$, then $x_i[k] = 0$ and $r_i[k] = 0$ for all $k \geq 0$.

7.4 Measurement Model

Suppose for any node $i \in \mathcal{V}$ and for any time step $k \in \mathbb{Z}_{\geq 1}$, we can obtain a measurement of state $x_i[k]$, i.e., the proportion of the population of node $i \in \mathcal{V}$ that is infected at time $k \in \mathbb{Z}_{\geq 1}$. Denote the obtained measurement of $x_i[k]$ as $\hat{x}_i[k]$ for all $i \in \mathcal{V}$. Noting that $x_i[k] \in [0,1]$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$ as argued in Section 7.3, we assume that $\hat{x}_i[k] \in [0,1]$ also holds for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$. Moreover, given the true state $x_i[k]$, we assume that measurement $\hat{x}_i[k]$ is a (discrete) random variable whose probability mass function (pmf) is given by $p(\hat{x}_i[k]|x_i[k])$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$.[2] Similarly, let $\hat{r}_i[k]$ denote the measurement of $r_i[k]$,

[1] Note that for the case when $d_i = 0$, i.e., $i \in \mathcal{S}_I$, part (b) implies $x_i[k] > 0$ for all $k \geq 0$.

[2] Note that one can also model $\hat{x}_i[k]$ as a continuous random variable whose probability density function is denoted as $p(\hat{x}_i[k]|x_i[k])$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$.

i.e., the proportion of population of node $i \in \mathcal{V}$ that is recovered at time k, for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$. We assume that $\hat{r}_i[k] \in [0,1]$ is given by pmf $p(\hat{r}_i[k]|r_i[k])$.

In general, the above measurement models for $x_i[k]$ and $r_i[k]$ are proposed to capture the potential randomness of the obtained measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$. For instance, consider the scenario (e.g., the ongoing COVID-19 pandemic) where measurement $\hat{x}_i[k]$ is obtained by conducting viral tests on individuals in the population corresponding to node $i \in \mathcal{V}$ at time $k \in \mathbb{Z}_{\geq 1}$, where a positive testing result indicates that the tested individual is infected at time k and a negative testing result indicates that the tested individual is not infected at time k (e.g., [112]). In practice, it is not always practical to test all the individuals of node $i \in \mathcal{V}$ at time $k \in \mathbb{Z}_{\geq 1}$, due to, for instance, the lack of testing kits and the fact that performing such tests incur certain costs. Rather, the tests are given to a group of randomly sampled individuals of the population at node $i \in \mathcal{V}$ at time $k \in \mathbb{Z}_{\geq 1}$ (e.g., [113]). Moreover, the testing results are not always accurate, e.g., a test on an infected individual can yield a negative result (e.g., [114]). Therefore, it is reasonable to model measurement $\hat{x}_i[k]$ as a random variable with pmf $p(\hat{x}_i[k]|x_i[k])$. Similarly, one way to obtain measurement $\hat{r}_i[k]$ is to perform antibody tests on randomly sampled individuals in the population of node $i \in \mathcal{V}$ (e.g., [115]), where the testing results can potentially be inaccurate (e.g., [116]). Thus, we model measurement $\hat{r}_i[k]$ as a random variable with pmf $p(\hat{r}_i[k]|\theta)$. Note that the proposed measurement models also capture the special case when one can obtain exact measurements of $x_i[k]$ and $r_i[k]$. In such a case, we can view $p(\hat{x}_i[k] = x_i[k]|x_i[k]) = 1$ and $p(\hat{r}_i[k] = r_i[k]|r_i[k]) = 1$. In the following sections, we will divide our discussions into the special case when exact measurements of $x_i[k]$ and $r_i[k]$ can be obtained, and the general case when measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$ are given by $p(\hat{x}_i[k]|x_i[k])$ and $p(\hat{r}_i[k]|r_i[k])$, respectively.

7.5 Measurement Selection Problem in Exact Measurement Setting

In this section, we consider the scenario where exact measurements of $x_i[k]$ and $r_i[k]$ (for a subset of nodes in the network) can be obtained. Throughout this section, we will assume that $\mathcal{S}_I, \mathcal{S}_H \subseteq \mathcal{V}$ defined in Definition 7.3.2 is known. In other words, we assume that we know the set of nodes in $\mathcal{V}$ that have infected individuals at time step $t = 0$.

7.5.1 Problem Formulation

Given exact measurements of $x_i[k]$ and $r_i[k]$ for a subset of nodes, our goal is to estimate (or uniquely identify, if possible) the unknown parameters β and δ, i.e., the infection rate and the recover rate of the disease in the SIR dynamics (given by Eq. (7.1)) over the network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. As we will see in the following, when exact measurements of $x_i[k]$ and $r_i[k]$ can be obtained, one can uniquely identify β and δ based on the measurements. Here, we consider the scenario where collecting the measurement of $x_i[k]$ (resp., $r_i[k]$) at any node $i \in \mathcal{V}$ and at any time step $k \in \mathbb{Z}_{\geq 0}$ incurs a cost, denoted as $c_{k,i} \in \mathbb{R}_{\geq 0}$ (resp., $b_{k,i} \in \mathbb{R}_{\geq 0}$). Moreover, we can only collect the measurements of $x_i[k]$ and $r_i[k]$ for $k \in \{t_1, t_1 + 1, \ldots, t_2\}$, where $t_1, t_2 \in \mathbb{Z}_{\geq 0}$ are given with $t_2 > t_1$. Noting that Lemma 7.3.3 provides a (sufficient and necessary) condition under which $x_i[k] = 0$ (resp., $r_i[k] = 0$) holds, we see that one does not need to collect measurement of $x_i[k]$ (resp., $r_i[k]$) if $x_i[k] = 0$ (resp., $r_i[k] = 0$) from Lemma 7.3.3. Given time steps $t_1, t_2 \in \mathbb{Z}_{\geq 0}$ with $t_2 > t_1$, we now define a set

$$\mathcal{I}_{t_1:t_2} \triangleq \{x_i[k] : k \in \{t_1, \ldots, t_2\}, i \in \mathcal{V}, x_i[k] > 0\} \cup \{r_i[k] : k \in \{t_1, \ldots, t_2\}, i \in \mathcal{V}, r_i[k] > 0\}, \quad (7.3)$$

which represents the set of all candidate measurements from time step t_1 to time step t_2. To proceed, we first use Eq. (7.1b)-(7.1c) to obtain

$$\begin{bmatrix} x[t_1+1]-x[t_1] \\ \vdots \\ x[t_2]-x[t_2-1] \\ r[t_1+1]-r[t_1] \\ \vdots \\ r[t_2]-r[t_2-1] \end{bmatrix} = h \begin{bmatrix} \Phi^x_{t_1:t_2} \\ \Phi^r_{t_1:t_2} \end{bmatrix} \begin{bmatrix} \beta \\ \delta \end{bmatrix}, \tag{7.4}$$

where $\Phi^x_{t_1:t_2-1} \triangleq \begin{bmatrix} (\Phi^x_{t_1})^T & \cdots & (\Phi^x_{t_2-1})^T \end{bmatrix}^T$ with

$$\Phi^x_k \triangleq \begin{bmatrix} s_1[k]\sum_{j\in\bar{\mathcal{N}}_1} a_{1j}x_j[k] & -x_1[k] \\ \vdots & \vdots \\ s_n[k]\sum_{j\in\bar{\mathcal{N}}_n} a_{nj}x_j[k] & -x_n[k] \end{bmatrix} \ \forall k \in \{t_1,\ldots,t_2-1\}, \tag{7.5}$$

and $\Phi^r_{t_1:t_2-1} \triangleq \begin{bmatrix} (\Phi^r_{t_1})^T & \cdots & (\Phi^r_{t_2-1})^T \end{bmatrix}^T$ with

$$\Phi^r_k \triangleq \begin{bmatrix} 0 & x_1[k] \\ \vdots & \vdots \\ 0 & x_n[k] \end{bmatrix} \ \forall k \in \{t_1,\ldots,t_2-1\}. \tag{7.6}$$

Supposing the weight matrix A and the sampling parameter h are known, we can then view Eq. (7.4) as a set of $2(t_2-t_1)n$ equations in β and δ. Noting that $s_i[k]$ for all $i \in \mathcal{V}$ can be obtained from $s_i[k] = 1 - x_i[k] - r_i[k]$ as argued in Section 7.3, we see that the coefficients in the set of equations in β and δ given by Eq. (7.4), i.e., the terms in Eq. (7.4) other than β and δ, can be determined given that $x[k]$ and $r[k]$ are known for all $k \in \{t_1,\ldots,t_2\}$. Also note that given $x[k]$ and $r[k]$ for all $k \in \{t_1,\ldots,t_2\}$, we can uniquely identify β and δ using Eq. (7.4) if and only if $\text{rank}(\begin{bmatrix} (\Phi^x_{t_1:t_2-1})^T & (\Phi^r_{t_1:t_2-1})^T \end{bmatrix}) = 2$.

Next, let $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ denote a measurement selection strategy, where $\mathcal{I}_{t_1:t_2}$ is given by Eq. (7.3). We will then consider identifying β and δ using measurements contained

in $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$. To illustrate our analysis, given any $i \in \mathcal{V}$ and any $k \in \{t_1, \ldots, t_2 - 1\}$, we first consider the following equation from Eq. (7.4):

$$x_i[k+1] - x_i[k] = h \begin{bmatrix} s_i[k] \sum_{w \in \bar{\mathcal{N}}_i} a_{iw} x_w[k] & -x_i[k] \end{bmatrix} \begin{bmatrix} \beta \\ \delta \end{bmatrix}, \tag{7.7}$$

where $s_i[k] = 1 - x_i[k] - r_i[k]$, and we index the equation in Eq. (7.4) corresponding to Eq. (7.7) as (k, i, x). Note that in order to use Eq. (7.7) in identifying β and δ, one needs to determine the coefficients (i.e., the terms other than β and δ) in the equation. Also note that in order to determine the coefficients in equation (k, i, x), one can use the measurements contained in $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$, and use Lemma 7.3.3 to determine if $x_i[k] = 0$ (resp., $r_i[k] = 0$) holds. Supposing $x_i[k+1] = 0$, we see from Lemma 7.3.3 and Eq. (7.2b) that $x_i[k] = 0$ and $s_i[k] \sum_{w \in \bar{\mathcal{N}}_i} a_{iw} x_w[k] = 0$, which makes equation (k, i, x) useless in identifying β and δ. Thus, in order to use equation (k, i, x) in identifying β and δ, we need to have $x_i[k+1] \in \mathcal{I}$ with $x_i[k+1] > 0$. Next, we will show that equation (k, i, x) can still be used in identifying β and δ even if there exist coefficients in equation (k, i, x) that cannot be determined using measurements from $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ or using Lemma 7.3.3. To see this, given any $i, j \in \mathcal{V}$ with $i \neq j$ and any $k \in \{t_1, \ldots, t_2 - 1\}$, we consider the following two equations from Eq. (7.4):

$$x_i[k+1] - x_i[k] = h \begin{bmatrix} s_i[k] \sum_{w \in \bar{\mathcal{N}}_i} a_{iw} x_w[k] & -x_i[k] \end{bmatrix} \begin{bmatrix} \beta \\ \delta \end{bmatrix}, \tag{7.8a}$$

$$x_j[k+1] - x_j[k] = h \begin{bmatrix} s_j[k] \sum_{w \in \bar{\mathcal{N}}_j} a_{jw} x_w[k] & -x_j[k] \end{bmatrix} \begin{bmatrix} \beta \\ \delta \end{bmatrix}, \tag{7.8b}$$

where we index the equation in Eq. (7.4) corresponding to Eq. (7.8a) (resp., Eq. (7.8b)) as (k, i, x) (resp., (k, j, x)). Suppose $x_w[k] > 0$ and $x_w[k] \notin \mathcal{I}$ for all $w \in \mathcal{N}_i$, i.e., the coefficient $s_i[k] \sum_{w \in \bar{\mathcal{N}}_i} a_{iw} x_w[k]$ in equation (k, i, x) cannot be determined using the measurements from $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ and using Lemma 7.3.3. Moreover, suppose $\mathcal{N}_i = \mathcal{N}_j$ and $a_{iw} = a_{jw}$ for all $w \in \mathcal{N}_i$. Noting that $s_i[k] > 0$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$

from Lemma 7.3.3(a), one can then subtract Eq. (7.8b) multiplied by $1/s_j[k]$ from Eq. (7.8a) multiplied by $1/s_i[k]$, and obtain the following equation in β and δ:

$$\frac{1}{s_i[k]}(x_i[k+1]-x_i[k])-\frac{1}{s_j[k]}(x_j[k+1]-x_j[k]) \\ = h\begin{bmatrix} a_{ii}x_i[k]-a_{jj}x_j[k] & -\frac{x_i[k]}{s_i[k]}+\frac{x_j[k]}{s_j[k]}\end{bmatrix}\begin{bmatrix}\beta\\ \delta\end{bmatrix}. \quad (7.9)$$

where $s_i[k]=1-x_i[k]-r_i[k]$ and $s_j[k]=1-x_j[k]-r_j[k]$. Now, suppose $\{x_i[k+1],x_j[k+1]\}\subseteq\mathcal{I}$, and $x_i[k]$, $r_i[k]$, $x_j[k]$ and $r_j[k]$ can be determined using the measurements from $\mathcal{I}$ or using Lemma 7.3.3. We see that Eq. (7.9) can now be used in identifying β and δ. Similarly, given any $i\in\mathcal{V}$ and any $k\in\{t_1,\ldots,t_2-1\}$, we consider the following equation from Eq. (7.4):

$$r_i[k+1]-r_i[k]=h\begin{bmatrix}0 & x_i[k]\end{bmatrix}\begin{bmatrix}\beta\\ \delta\end{bmatrix}, \quad (7.10)$$

where we index the above equation as (k,i,r). Supposing $r_i[k+1]=0$, we see from Lemma 7.3.3 and Eq. (7.2c) that $r_i[k]=x_i[k]=0$, which makes equation (k,i,r) useless in identifying β and δ. Hence, in order to use equation (k,i,r) in identifying β and δ, we need to have $\{x_i[k],r_i[k+1]\}\subseteq\mathcal{I}$ with $x_i[k]>0$ and $r_i[k+1]>0$. More precisely, we observe that equation (k,i,r) can be used in identifying β and δ if and only if $\{x_i[k],r_i[k+1]\}\subseteq\mathcal{I}$, and $r_i[k]\in\mathcal{I}$ or $r_i[k]=0$ (from Lemma 7.3.3).

In general, let us denote the following two coefficient matrices corresponding to equations (k,i,x) and (k,i,r) in Eq. (7.4), respectively:

$$\Phi^x_{k,i}\triangleq\begin{bmatrix}s_i[k]\sum_{j\in\mathcal{N}_i}a_{ij}x_j[k] & -x_i[k]\end{bmatrix}, \quad (7.11a)$$

$$\Phi^r_{k,i}\triangleq\begin{bmatrix}0 & x_i[k]\end{bmatrix}, \quad (7.11b)$$

for all $k\in\{t_1,\ldots,t_2-1\}$ and for all $i\in\mathcal{V}$. Moreover, given any measurement selection strategy $\mathcal{I}\subseteq\mathcal{I}_{t_1:t_2}$, we let

$$\bar{\mathcal{I}}\triangleq\{(k,i,x):x_i[k+1]\in\mathcal{I},x_i[k]=0\}\cup\{(k,i,x):\{x_i[k+1],x_i[k]\}\subseteq\mathcal{I}\} \\ \cup\{(k,i,r):\{r_i[k+1],x_i[k]\}\subseteq\mathcal{I},r_i[k]=0\}\cup\{(k,i,r):\{r_i[k+1],r_i[k],x_i[k]\}\subseteq\mathcal{I}\} \quad (7.12)$$

be the set that contains indices of the equations from Eq. (7.4) that can be *potentially* used in identifying β and δ, based on the measurements contained in $\mathcal{I}$. In other words, the coefficients on the left-hand side of equation (k, i, x) (resp., (k, i, r)) can be determined using the measurements from $\mathcal{I}$ and using Lemma 7.3.3, for all $(k, i, x) \in \bar{\mathcal{I}}$ (resp., $(k, i, r) \in \bar{\mathcal{I}}$). Let us now consider the coefficient matrix $\Phi^x_{k,i}$ (resp., $\Phi^r_{k,i}$) corresponding to $(k, i, x) \in \bar{\mathcal{I}}$ (resp., $(k, i, r) \in \bar{\mathcal{I}}$). As we discussed above, it is possible that there exist equations in $\bar{\mathcal{I}}$ whose coefficients cannot be (directly) determined using the measurements contained in $\mathcal{I}$ or using Lemma 7.3.3, where the undetermined coefficients come from the first element in $\Phi^x_{k,i}$ given by Eq. (7.11a). Nevertheless, it is also possible that one can perform algebraic operations among the equations in $\bar{\mathcal{I}}$ such that the undetermined coefficients get cancelled. Formally, we introduce the following definition.

Definition 7.5.1 *Consider a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$, where $\mathcal{I}_{t_1:t_2}$ is given by Eq.* (7.3)*. Stack coefficient matrices $\Phi^x_{k,i} \in \mathbb{R}^{1\times 2}$ for all $(k, i, x) \in \bar{\mathcal{I}}$ and $\Phi^r_{k,i} \in \mathbb{R}^{1\times 2}$ for all $(k, i, r) \in \bar{\mathcal{I}}$ into a single matrix, where $\Phi^x_{k,i}$ and $\Phi^r_{k,}$ are given by* (7.11) *and $\bar{\mathcal{I}}$ is given by Eq.* (7.12)*. The resulting matrix is denoted as $\Phi(\mathcal{I}) \in \mathbb{R}^{|\bar{\mathcal{I}}|\times 2}$. Moreover, define $\tilde{\Phi}(\mathcal{I})$ to be the set that contains all the matrices $\Phi \in \mathbb{R}^{2\times 2}$ such that $(\Phi)_1$ and $(\Phi)_2$ can be obtained via algebraic operations among the rows in $\Phi(\mathcal{I})$, and the elements in $(\Phi)_1$ and $(\Phi)_2$ can be fully determined using the measurements from $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ and using Lemma* 7.3.3*.*

In other words, $\Phi \in \tilde{\Phi}(\mathcal{I})$ corresponds to two equations (in β and δ) obtained from Eq. (7.4) such that the coefficients in the equations can be determined using the measurements contained in $\mathcal{I}$ and using Lemma 7.3.3 (if the coefficients contain $x_i[k] = 0$ or $r_i[k] = 0$). Moreover, using similar arguments to those for obtaining Eq. (7.9), one can show that the coefficients on the left-hand side of the two equations obtained from Eq. (7.4) corresponding to Φ can also be determined using measurements from $\mathcal{I}$ and using Lemma 7.3.3. Putting the above arguments together, we see that given

a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$, β and δ can be uniquely identified if and only if there exists $\Phi \in \tilde{\Phi}(\mathcal{I})$ such that $\text{rank}(\Phi) = 2$. Equivalently, denoting

$$r_{\max}(\mathcal{I}) \triangleq \max_{\Phi \in \tilde{\Phi}(\mathcal{I})} \text{rank}(\Phi), \tag{7.13}$$

where $r_{\max}(\mathcal{I}) \triangleq 0$ if $\tilde{\Phi}(\mathcal{I}) = \emptyset$, we have β and δ can be uniquely identified using the measurements from $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ if and only if $r_{\max}(\mathcal{I}) = 2$.

Remark 7.5.1 *Note that if a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ satisfies $r_{\max}(\mathcal{I}) = 2$, it follows from the above arguments that $|\bar{\mathcal{I}}| \geq 2$, i.e., $\Phi(\mathcal{I}) \in \mathbb{R}^{|\bar{\mathcal{I}}| \times 2}$ has at least two rows.*

Recall that collecting the measurement of $x_i[k]$ (resp., $r_i[k]$) at any node $i \in \mathcal{V}$ and at any time step $k \in \mathbb{Z}_{\geq 1}$ incurs cost $c_{k,i} \in \mathbb{R}_{\geq 0}$ (resp., $b_{k,i} \in \mathbb{R}_{\geq 0}$). Given any measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$, we denote the cost associated with $\mathcal{I}$ as

$$c(\mathcal{I}) \triangleq \sum_{x_i[k] \in \mathcal{I}} c_{k,i} + \sum_{r_i[k] \in \mathcal{I}} b_{k,i}. \tag{7.14}$$

The above arguments then lead to the problem of minimizing the cost spent on collecting measurements such that parameters β and δ can be uniquely identified (within a given time interval $[t_1 : t_2]$). Formally, we define the Parameter Identification Measurement Selection (PIMS) problem in the perfect measurement setting as follows, where we assume that Assumptions 7.3.1-7.3.2 hold for the PIMS instances

Problem 7.5.2 *Consider a discrete-time SIR model given by Eq. (7.1) with a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, a weight matrix $A \in \mathbb{R}^{n \times n}$, a sampling parameter $h \in \mathbb{R}_{\geq 0}$, and sets $\mathcal{S}_I, \mathcal{S}_H \subseteq \mathcal{V}$ defined in Definition 7.3.2. Moreover, consider time steps $t_1, t_2 \in \mathbb{Z}_{\geq 1}$ with $t_1 < t_2$, and a cost $c_{k,i} \in \mathbb{R}_{\geq 0}$ of measuring $x_i[k]$ and a cost $b_{k,i} \in \mathbb{R}_{\geq 0}$ of measuring $r_i[k]$ for all $i \in \mathcal{V}$ and for all $k \in \{t_1, \ldots, t_2\}$. The Parameter Identification Measurement Selection (PIMS) problem is to find $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ that solves*

$$\begin{aligned} &\min_{\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}} c(\mathcal{I}) \\ &s.t.\ r_{\max}(\mathcal{I}) = 2, \end{aligned} \tag{7.15}$$

where $\mathcal{I}_{t_1:t_2}$ is defined in Eq. (7.3), $c(\mathcal{I})$ is defined in Eq. (7.14), and $r_{\max}(\mathcal{I})$ is defined in Eq. (7.13).

7.5.2 Complexity of the PIMS problem

In this section, we will show that the PIMS problem is NP-hard. To do it, we will relate the PIMS problem to the exact cover by 3-sets (X3C) problem, which is known to be NP-complete [41].

Definition 7.5.2 *An X3C instance is given by a set* $\mathcal{X} = \{1, 2, \ldots, 3m\}$ *and a collection* $\mathcal{Z} = \{z_1, z_2, \ldots, z_\tau\}$ *of 3-element subsets of* $\mathcal{X}$*, where* $\tau \geq m$*. An exact cover for* $\mathcal{X}$ *is a subcollection* $\mathcal{Z}' \subseteq \mathcal{Z}$ *such that every element of* $\mathcal{X}$ *occurs in exactly one member of* $\mathcal{Z}'$*.*

Lemma 7.5.3 *Consider a set* $\mathcal{X} = \{1, \ldots, 3m\}$ *and a collection* $\mathcal{Z} = \{z_1, \ldots, z_\tau\}$ *of 3-element subsets of* $\mathcal{X}$*. The problem of determining whether* $\mathcal{Z}$ *contains an exact cover for* $\mathcal{X}$ *is NP-complete.*

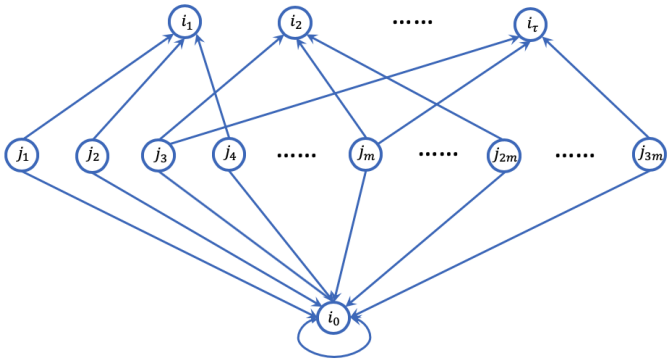


Fig. 7.1. Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ constructed in the proof of Theorem 7.5.4.

Theorem 7.5.4 *The PIMS problem is NP-hard.*

Proof We will give a polynomial-time reduction from the X3C problem to the PIMS problem. Consider an instance of the X3C problem given by a set $\mathcal{X} = \{1, \ldots, 3m\}$ and a collection $\mathcal{Z} = \{z_1, \ldots, z_\tau\}$ of 3-element subsets of $\mathcal{X}$, where $\tau \geq m$. We then construct an instance of the PIMS problem as follows. The node set of the

graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is set to be $\mathcal{V} = \{i_0, i_1, \ldots, i_\tau\} \cup \{j_1, j_2, \ldots, j_{3m}\}$. The edge set of $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is set to satisfy that $(j_q, i_l) \in \mathcal{E}$ if $q \in \mathcal{X}$ is contained in $z_l \in \mathcal{Z}$, $(j_q, i_0) \in \mathcal{E}$ for all $q \in \mathcal{X}$, and $(i_0, i_0) \in \mathcal{E}$. A plot of the graph is given in Fig. 7.1. Note that based on the construction of $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, each node $i \in \{i_1, \ldots, i_\tau\}$ represents a subset from $\mathcal{Z}$ in the X3C instance, and each node $j \in \{j_1, \ldots, j_{3m}\}$ represents an element from $\mathcal{X}$ in the X3C instance, where the edges between $\{i_1, \ldots, i_\tau\}$ and $\{j_1, \ldots, j_{3m}\}$ indicate how the elements in $\mathcal{X}$ are included in the subsets in $\mathcal{Z}$. Accordingly, the weight matrix $A \in \mathbb{R}^{(3m+\tau+1)\times(3m+\tau+1)}$ is set to satisfy that $a_{i_l j_q} = 1$ if $q \in \mathcal{X}$ is contained in $z_l \in \mathcal{Z}$, $a_{i_0 j_q} = 1$ for all $q \in \mathcal{X}$, and $a_{i_0 i_0} = 1$. We set the sampling parameter to be $h = 1/(3m+1)$. The set $\mathcal{S}_I \subseteq \mathcal{V}$ is set to be $\mathcal{S}_I = \mathcal{V}$, i.e., $x_i[0] > 0$ for all $i \in \mathcal{V}$. We set time steps $t_1 = 2$ and $t_2 = 3$. Finally, we set $b_{2,i} = b_{3,i} = 0$ for all $i \in \mathcal{V}$, $c_{2,i_l} = 1$ and $c_{3,i_l} = 0$ for all $l \in \{1, \ldots, \tau\}$, $c_{2,j_q} = c_{3,j_q} = m+1$ for all $q \in \mathcal{X}$, and $c_{2,i_0} = c_{3,i_0} = 0$. Since we set $b_{2,i} = b_{3,i} = 0$ for all $i \in \mathcal{V}$ and $c_{2,i_0} = 0$, the following equation can always be used in identifying β and δ with zero cost on collecting the corresponding measurements:

$$r_{i_0}[3] - r_{i_0}[2] = h \begin{bmatrix} 0 & x_{i_0}[2] \end{bmatrix} \begin{bmatrix} \beta \\ \delta \end{bmatrix}, \tag{7.16}$$

where we also note $x_{i_0}[2] > 0$ from Lemma 7.3.3, since $x_{i_0}[0] = 0.5$. Moreover, since $x_i[0] > 0$ for all $i \in \mathcal{V}$, we see from Lemma 7.3.3 that $x_i[k] > 0$ and $r_i[k] > 0$ for all $i \in \mathcal{V}$ and for all $k \in \{2, 3\}$. Therefore, Lemma 7.3.3 is no longer useful in determining the coefficients in the equations from Eq. (7.4).

We claim that an optimal solution, denoted as $\mathcal{I}^\star$, to the constructed PIMS instance satisfies $c(\mathcal{I}^\star) \leq m$ if and only if the solution to the X3C instance is "yes".

First, suppose the solution to the X3C instance is "yes". Denote an exact cover as $\mathcal{Z}' = \{z_{q_1}, \ldots, z_{q_m}\} \subseteq \mathcal{Z}$, where $\{q_1, \ldots, q_m\} \subseteq \{1, \ldots, \tau\}$. Let us consider a measurement selection strategy $\mathcal{I}_0 \subseteq \mathcal{I}_{t_1:t_2}$ given by

$$\mathcal{I}_0 = \Big(\bigcup_{l \in \{1,\ldots,m\}} \{x_{i_{q_l}}[2], x_{i_{q_l}}[3], r_{i_{q_l}}[2]\} \Big) \cup \{x_{i_0}[2], x_{i_0}[3], r_{i_0}[2], r_{i_0}[3]\}.$$

We then have from Eq. (7.12) $\bar{\mathcal{I}}_0 = \{(2, i_0, r), (2, i_0, x)\} \cup \{(2, i_{q_l}, x) : l \in \{1, \ldots, m\}\}$. Noting that $s_i[k] > 0$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$ from Lemma 7.3.3(a), we consider the following equations from Eq. (7.4) whose indices are contained in $\bar{\mathcal{I}}_0$:

$$\frac{1}{s_{i_0}[2]}(x_{i_0}[3] - x_{i_0}[2]) = h\begin{bmatrix} x_{i_0}[2] + \sum_{w \in \mathcal{N}_{i_0}} x_w[2] & -\frac{x_{i_0}[2]}{s_{i_0}[1]} \end{bmatrix}\begin{bmatrix} \beta \\ \delta \end{bmatrix}, \tag{7.17}$$

and

$$\begin{aligned} \frac{1}{s_{i_{q_1}}[2]}(x_{i_{q_1}}[3] - x_{i_{q_1}}[2]) &= h\begin{bmatrix} \sum_{w \in \mathcal{N}_{i_{q_1}}} x_w[2] & -\frac{x_{i_{q_1}}[2]}{s_{i_{q_1}}[2]} \end{bmatrix}\begin{bmatrix} \beta \\ \delta \end{bmatrix} \\ &\vdots \\ \frac{1}{s_{i_{q_m}}[2]}(x_{i_{q_m}}[3] - x_{i_{q_m}}[2]) &= h\begin{bmatrix} \sum_{w \in \mathcal{N}_{i_{q_m}}} x_w[2] & -\frac{x_{i_{q_m}}[2]}{s_{i_{q_m}}[2]} \end{bmatrix}\begin{bmatrix} \beta \\ \delta \end{bmatrix}, \end{aligned} \tag{7.18}$$

where we note $\mathcal{N}_{i_0} = \{j_1, \ldots, j_{3m}\}$ from the way we constructed $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Since $\mathcal{Z}' = \{z_{q_1}, \ldots, z_{q_m}\}$ is an exact cover for $\mathcal{X}$, we see from the construction of $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ that $\bigcup_{l \in \{1,\ldots,m\}} \mathcal{N}_{i_{q_l}}$ is a union of mutually disjoint (3-element) sets such that $\bigcup_{l \in \{1,\ldots,m\}} \mathcal{N}_{i_{q_l}} = \{j_1, \ldots, j_{3m}\}$. Therefore, subtracting the equations in (7.18) from Eq. (7.17), we obtain

$$\begin{aligned} &\frac{1}{s_{i_0}[2]}(x_{i_0}[3] - x_{i_0}[2]) - \sum_{l \in \{1,\ldots,m\}} \frac{1}{s_{i_{q_l}}[2]}(x_{i_{q_l}}[3] - x_{i_{q_l}}[2]) \\ &\qquad = h\begin{bmatrix} x_{i_0}[2] & -\frac{x_{i_0}[2]}{s_{i_0}[2]} + \sum_{l \in \{1,\ldots,m\}} \frac{x_{i_{q_l}}[2]}{s_{i_{q_l}}[2]} \end{bmatrix}\begin{bmatrix} \beta \\ \delta \end{bmatrix}, \end{aligned} \tag{7.19}$$

where we note $x_{i_0}[2] > 0$ as argued above. Following Definition 7.5.1, we stack coefficient matrices $\Phi^r_{2,i_0} \in \mathbb{R}^{1\times 2}$, $\Phi^x_{2,i_0} \in \mathbb{R}^{1\times 2}$ and $\Phi^x_{2,i_{q_l}} \in \mathbb{R}^{1\times 2}$ for all $l \in \{1, \ldots, m\}$ into a matrix $\Phi(\mathcal{I}_0) \in \mathbb{R}^{(m+2)\times 2}$, where $\Phi^r_{k,i}$ and $\Phi^x_{k,i}$ are defined in (7.11). Now, considering the matrix

$$\Phi_0 = \begin{bmatrix} x_{i_0}[2] & -\frac{x_{i_0}[2]}{s_{i_0}[2]} + \sum_{l \in \{1,\ldots,m\}} \frac{x_{i_{q_l}}[2]}{s_{i_{q_l}}[2]} \\ 0 & x_{i_0}[2] \end{bmatrix}, \tag{7.20}$$

we see from the above arguments that $(\Phi_0)_1$ and $(\Phi_0)_2$ can be be obtained via algebraic operations among the rows in $\Phi(\mathcal{I}_0)$, and the elements in $(\Phi_0)_1$ and $(\Phi_0)_2$ can be determined using the measurements from $\mathcal{I}_0$. Therefore, we have $\Phi_0 \in \tilde{\Phi}(\mathcal{I}_0)$, where recall from Definition 7.5.1 that $\tilde{\Phi}(\mathcal{I}_0)$ contains all the matrices $\Phi \in \mathbb{R}^{2\times 2}$ such that $(\Phi)_1$ and $(\Phi)_2$ can be obtained via algebraic operations among the rows in $\Phi(\mathcal{I}_0)$ and the elements in $(\Phi)_1$ and $(\Phi)_2$ can be determined using the measurements from $\mathcal{I}_0$. Noting that $x_{i_0}[2] > 0$, we have $\text{rank}(\Phi_0) = 2$, which implies $r_{\max}(\mathcal{I}_0) = 2$, where $r_{\max}(\mathcal{I}_0)$ is given by Eq. (7.13). Thus, $\mathcal{I}_0 \subseteq \mathcal{I}_{t_1:t_2}$ satisfies the constraint in (7.15). Since $c(\mathcal{I}_0) = m$ from the way we set the costs of collecting measurements in the PIMS instance, we have $c(\mathcal{I}^\star) \leq m$.

Conversely, suppose the solution to the X3C instance is "no", i.e., for any subcollection $\mathcal{Z}' \subseteq \mathcal{Z}$ that contains m subsets, there exists at least one element in $\mathcal{X}$ that is not contained in any subset in $\mathcal{Z}'$. We will show that for any measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ that satisfies $r_{\max}(\mathcal{I}) = 2$, $c(\mathcal{I}) > m$ holds. Equivalently, we will show that for any $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ with $c(\mathcal{I}) \leq m$, $r_{\max}(\mathcal{I}) = 2$ does not hold. Consider any $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ such that $c(\mathcal{I}) \leq m$. Noting that $c_{2,j_q} = c_{3,j_q} = m + 1$ for all $q \in \mathcal{X}$ in the constructed PIMS instance, we have $x_{j_q}[2] \notin \mathcal{I}$ and $x_{j_q}[3] \notin \mathcal{I}$ for all $q \in \mathcal{X}$. Moreover, we see that $\mathcal{I}$ contains at most m measurements from $\{x_{i_1}[2], \ldots, x_{i_\tau}[2]\}$. To proceed, let us consider any $\mathcal{I}_1 \subseteq \mathcal{I}_{t_1:t_2}$ such that

$$\mathcal{I}_1 = \{x_{i_0}[2], x_{i_{v_1}}[2], \ldots, x_{i_{v_m}}[2]\} \cup \Big(\bigcup_{l\in\{0,\ldots,\tau\}} \{x_{i_l}[3]\}\Big) \cup \Big(\bigcup_{i\in\mathcal{V}} \{r_i[2], r_i[3]\}\Big), \tag{7.21}$$

where $\{v_1, \ldots, v_m\} \subseteq \{1, \ldots, \tau\}$. In other words, $\mathcal{I}_1 \subseteq \mathcal{I}_{t_1:t_2}$ contains m measurements from $\{x_{i_1}[2], \ldots, x_{i_\tau}[2]\}$ and all the other measurements from $\mathcal{I}_{t_1:t_2}$ that have zero costs. It follows that $c(\mathcal{I}_1) = m$. Also note that for all $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ with $\{x_{i_{v_1}}[2], \ldots, x_{i_{v_m}}[2]\} \subseteq \mathcal{I}$ and $c(\mathcal{I}) \leq m$, we have $\mathcal{I} \subseteq \mathcal{I}_1$. Similarly to (7.17) and (7.18), we have the following equations from Eq. (7.4) whose indices are contained in $\bar{\mathcal{I}}_1$ (given by Eq. (7.12)):

$$\frac{1}{s_{i_0}[2]}(x_{i_0}[3] - x_{i_0}[2]) = h \begin{bmatrix} x_{i_0}[2] + \sum_{w\in\mathcal{N}_{i_0}} x_w[2] & -\frac{x_{i_0}[2]}{s_{i_0}[1]} \end{bmatrix} \begin{bmatrix} \beta \\ \delta \end{bmatrix}, \tag{7.22}$$

and

$$\begin{aligned}\frac{1}{s_{i_{v_1}}[2]}(x_{i_{v_1}}[3]-x_{i_{v_1}}[2]) &= h\begin{bmatrix}\sum_{w\in\mathcal{N}_{i_{v_1}}} x_w[2] & -\frac{x_{i_{v_1}}[2]}{s_{i_{v_1}}[2]}\end{bmatrix}\begin{bmatrix}\beta\\ \delta\end{bmatrix}\\ &\vdots\\ \frac{1}{s_{i_{v_m}}[2]}(x_{i_{v_m}}[3]-x_{i_{v_m}}[2]) &= h\begin{bmatrix}\sum_{w\in\mathcal{N}_{i_{v_m}}} x_w[2] & -\frac{x_{i_{v_m}}[2]}{s_{i_{v_m}}[2]}\end{bmatrix}\begin{bmatrix}\beta\\ \delta\end{bmatrix}.\end{aligned} \tag{7.23}$$

Noting that for any subcollection $\mathcal{Z}' \subseteq \mathcal{Z}$ that contains m subsets, there exists at least one element in $\mathcal{X}$ that is not contained in any subset in $\mathcal{Z}'$ as we argued above, we see that there exists at least one element in $\mathcal{X}$ that is not contained in any subset in $\{z_{v_1}, \ldots, z_{v_m}\}$. It then follows from the way we constructed $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ that there exists $w' \in \mathcal{N}_{i_0}$ such that $w' \notin \mathcal{N}_{i_{v_l}}$ for all $l \in \{1, \ldots, m\}$. Thus, by subtracting the equations in (7.23) (multiplied by any constants) from Eq. (7.22), $x_{w'}[2]$ will remain on the right-hand side of the equation in (7.22). Similarly, consider any equation from (7.23) indexed by $(2, i_{v_l}, x) \in \bar{\mathcal{I}}_1$, where $l \in \{1, \ldots, m\}$. First, suppose we subtract Eq. (7.22) multiplied by some positive constant and any equations in (7.23) other than equation $(2, i_{v_l}, x)$ (multiplied by any constants) from equation $(2, i_{v_l}, x)$. Since there exists $w' \in \mathcal{N}_{i_0}$ such that $w' \notin \mathcal{N}_{i_{v_l}}$ for all $l \in \{1, \ldots, m\}$ as argued above, we see that $x_{w'}[2]$ will appear on the right-hand side of equation $(2, i_{v_l}, x)$. Next, suppose we subtract any equations in (7.23) other than equation $(2, i_{v_l}, x)$ (multiplied by any constants) from equation $(2, i_{v_l}, x)$. One can check that either of the following two facts hold for the resulting equation $(2, i_{v_l}, x)$: (a) the coefficients on the right-hand side of equation $(2, i_{v_l}, x)$ contain $x_{j_q}[2] \notin \mathcal{I}_1$, where $q \in \mathcal{X}$; or (b) the coefficient matrix on the right-hand side of equation $(2, i_{v_l}, x)$ is of the form $\begin{bmatrix}0 & \star\end{bmatrix}$. Again, we stack $\Phi^r_{k,i} \in \mathbb{R}^{1\times 2}$ for all $(k, i, r) \in \bar{\mathcal{I}}_1$ and $\Phi^x_{k,i} \in \mathbb{R}^{1\times 2}$ for all $(k, i, x) \in \bar{\mathcal{I}}_1$ into a matrix $\Phi(\mathcal{I}_1)$, where we note that $\Phi^r_{k,i}$ is of the form $\begin{bmatrix}0 & \star\end{bmatrix}$ for all $(k, i, r) \in \bar{\mathcal{I}}_1$. One can then see from the above arguments that for all $\Phi \in \mathbb{R}^{2\times 2}$ (if they exist) such that $(\Phi)_1$ and $(\Phi)_2$ can be obtained from algebraic operations among the rows in $\Phi(\mathcal{I}_1)$, and the elements in $(\Phi)_1$ and $(\Phi)_2$ can be determined using the measurements from $\mathcal{I}_1$, $\text{rank}(\Phi) \leq 1$ holds. It follows that $r_{\max}(\mathcal{I}_1) < 2$, i.e., constraint $r_{\max}(\mathcal{I}_1) = 2$

in (7.15) does not hold. Using similar arguments to those above, one can further show that $r_{\max}(\mathcal{I}) < 2$ holds for all $c(\mathcal{I}) \leq m$, completing the proof of the converse direction of the above claim.

Hence, it follows directly from the above arguments that an algorithm for the PIMS problem can also be used to solve the X3C problem. Since X3C is NP-complete, we conclude that the PIMS problem is NP-hard. ■

Theorem 7.5.4 indicates that there is no polynomial-time algorithm that solves all instances of the PIMS problem optimally (if P $\neq$ NP). Moreover, we note from the formulation of the PIMS problem given by Problem 7.5.2 that the PIMS problem asks us to find a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ with minimum cost such that $r_{\max}(\mathcal{I}) = 2$. In other words, for a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$, one needs to check if $\max_{\Phi \in \tilde{\Phi}(\mathcal{I})} \operatorname{rank}(\Phi) = 2$ holds, *before* the corresponding measurements are collected. However, in general, it is not possible to calculate $\operatorname{rank}(\Phi)$ when no measurements are collected. In order to bypass these issues, we will explore additional properties of the PIMS problem in the following.

7.5.3 Solving the PIMS Problem

In this section, we will leverage properties of the PIMS problem and propose an approximation algorithm for the PIMS problem with performance guarantees. In particular, we will focus on measurement selection strategies that contain measurements corresponding to two equations in Eq. (7.4). Let us start with the following result whose proof can be found in Section 7.8.2.

Lemma 7.5.5 *Consider a discrete time SIR model given by Eq.* (7.1)*. Suppose Assumptions* 7.3.1-7.3.2 *hold. Then, the following results hold, where* $\Phi^x_{k_1,i_1} \in \mathbb{R}^{1\times 2}$ *and* $\Phi^r_{k_2,i_2} \in \mathbb{R}^{1\times 2}$ *are defined in* (7.11)*,* $\mathcal{S}'_I \triangleq \{i \in \mathcal{S}_I : a_{ii} > 0\}$*,* $\mathcal{S}' \triangleq \{i \in \mathcal{V} \setminus \mathcal{S}'_I : \mathcal{N}_i \neq \emptyset, \min\{d_j : j \in \mathcal{N}_i\} \neq \infty\}$*, and* $\mathcal{S}_I$ *and* d_i *are defined in Definition* 7.3.2 *for all* $i \in \mathcal{V}$*.* (a) *For any* $i_1 \in \mathcal{S}'_I$ *and for any* $i_2 \in \mathcal{V}$ *with* $d_{i_2} \neq \infty$*,* $\operatorname{rank}\Big(\begin{bmatrix} (\Phi^x_{k_1,i_1})^T & (\Phi^r_{k_2,i_2})^T \end{bmatrix} \Big) = 2$ *for all* $k_1 \geq 0$ *and for all* $k_2 \geq d_{i_2}$*, where* $k_1, k_2 \in \mathbb{Z}_{\geq 0}$*.*

(*b*) *For any* $i_1 \in \mathcal{S}'$ *and for any* $i_2 \in \mathcal{V}$ *with* $d_{i_2} \neq \infty$, *rank*$\Big(\begin{bmatrix} (\Phi^x_{k_1,i_1})^T & (\Phi^r_{k_2,i_2})^T \end{bmatrix} \Big) = 2$ *for all* $k_1 \geq \min\{d_j : j \in \mathcal{N}_{i_1}\}$, *and for all* $k_2 \geq d_{i_2}$, *where* $k_1, k_2 \in \mathbb{Z}_{\geq 0}$.

Thus, Lemma 7.5.5 leads to a sufficient condition on the measurements contained in a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ such that the rank condition $r_{\max}(\mathcal{I}) = 2$ in (7.15) holds. This sufficient condition indicates collecting the measurements corresponding to the coefficients in two equations from (7.4). Moreover, given a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$, we can check if the sufficient condition given in Lemma 7.5.5 holds before the corresponding measurements contained in $\mathcal{I}$ is collected. Therefore, we aim to find a measurement selection strategy $\mathcal{I} \subseteq \mathcal{I}_{t_1:t_2}$ that satisfies the sufficient condition given in Lemma 7.5.5, which will ensure that the rank condition $r_{\max}(\mathcal{I}) = 2$ holds. We formalize the analysis as follows.

Recalling the way we index the equations in Eq. (7.4) (see (7.7) and (7.10) for examples), we define the set that contains all the indices of the equations in Eq. (7.4) as

$$\mathcal{Q} \triangleq \{(k, i, \lambda) : k \in \{t_1, \ldots, t_2 - 1\}, i \in \mathcal{V}, \lambda \in \{x, r\}\}. \tag{7.24}$$

Following the arguments in Lemma 7.5.5, we denote

$$\begin{aligned} \mathcal{Q}_1 \triangleq \{(k, i, x) \in \mathcal{Q} : k \geq 0, i \in \mathcal{S}'_I\} \\ \cup \{(k, i, x) \in \mathcal{Q} : k \geq \min\{d_j : j \in \mathcal{N}_i\}, i \in \mathcal{S}'\}, \end{aligned} \tag{7.25}$$

and

$$\mathcal{Q}_2 \triangleq \{(k, i, r) \in \mathcal{Q} : k \geq d_i, i \in \mathcal{V}, d_i \neq \infty\}, \tag{7.26}$$

where $\mathcal{S}'_I$ and $\mathcal{S}'$ are defined in Lemma 7.5.5, and d_i is defined in Definition 7.3.2. Next, for all $(k, i, x) \in \mathcal{Q}$, we define the set of measurements that are needed to determine the coefficients in equation (k, i, x) (when no other equations are used) to be

$$\mathcal{I}(k, i, x) \triangleq \big(\{x_i[k+1], r_i[k]\} \cup \{x_j[k] : j \in \bar{\mathcal{N}}_i\}\big) \cap \mathcal{I}_{t_1:t_2}, \tag{7.27}$$

where $\mathcal{I}_{t_1:t_2}$ is defined in Eq. (7.3). Similarly, for all $(k, i, r) \in \mathcal{Q}$, we define

$$\mathcal{I}(k, i, r) \triangleq \big(\{r_i[k+1], r_i[k], x_i[k]\}\big) \cap \mathcal{I}_{t_1:t_2}. \tag{7.28}$$

Moreover, let us denote

$$\mathcal{I}((k_1,i_1,\lambda_1),(k_2,i_2,\lambda_2)) \triangleq \mathcal{I}(k_1,i_1,\lambda_1) \cup \mathcal{I}(k_2,i_2,\lambda_2) \tag{7.29}$$

for all $(k_1,i_1,\lambda_1),(k_2,i_2,\lambda_2) \in \mathcal{Q}$. Recall from Eq. (7.14) that we have

$$\begin{aligned} &c(\mathcal{I}((k_1,i_1,\lambda_1),(k_2,i_2,\lambda_2))) \\ &\qquad = \sum_{x_i[k]\in\mathcal{I}((k_1,i_1,\lambda_1),(k_2,i_2,\lambda_2))} c_{k,i} + \sum_{r_i[k]\in\mathcal{I}((k_1,i_1,\lambda_1),(k_2,i_2,\lambda_2))} b_{k,i}. \end{aligned} \tag{7.30}$$

Algorithm 7.5.1 Algorithm for PIMS

1: **Input**: An instance of PIMS

2: Find $(k_1,i_1,x) \in \mathcal{Q}_1$, $(k_2,i_2,r) \in \mathcal{Q}_2$ s.t. $c(\mathcal{I}((k_1,i_1,x),(k_2,i_2,r)))$ is minimized **return** $\mathcal{I}((k_1,i_1,x),(k_2,i_2,r))$

Based on the above arguments, we propose an algorithm defined in Algorithm 7.5.1 for the PIMS problem. Note that Algorithm 7.5.1 finds an equation from $\mathcal{Q}_1$ and an equation from $\mathcal{Q}_2$ such that the sum of the costs of the two equations is minimized, where $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are defined in Eq. (7.25) and Eq. (7.26), respectively. We have the following result for Algorithm 7.5.1

Proposition 7.5.1 *Consider an instance of the PIMS problem under Assumptions* 7.3.1-7.3.2. *Algorithm* 7.5.1 *returns a solution* $\mathcal{I}((k_1,i_1,x),(k_2,i_2,r))$ *to the PIMS problem that satisfies the constraint in* (7.15). *The solution returned by Algorithm* 7.5.1 *satisfies*

$$\frac{c(\mathcal{I}((k_1,i_1,x),(k_2,i_2,r)))}{c(\mathcal{I}^\star)} \leq \frac{\min_{(k,i,x)\in\mathcal{Q}_1}(b_{k+1,i}+b_{k,i}+c_{k+1,i}+\sum_{j\in\bar{\mathcal{N}}_i} c_{k,j})}{3c_{\min}}, \tag{7.31}$$

where $\mathcal{I}^\star$ *is an optimal solution to the PIMS problem,* $\mathcal{Q}_1$ *is defined in Eq.* (7.25)*, and* $c_{\min} \triangleq \min\{\min_{x_i[k]\in\mathcal{I}_{t_1:t_2}} c_{k,i}, \min_{r_i[k]\in\mathcal{I}_{t_1:t_2}} b_{k,i}\} > 0$ *with* $\mathcal{I}_{t_1:t_2}$ *is given by Eq.* (7.3).

Proof The feasibility of $\mathcal{I}((k_1,i_1,x),(k_2,i_2,r))$ follows directly from the definition of Algorithm 7.5.1 and Lemma 7.5.5. We now prove (7.31). Consider any equations $(k,i,x)\in\mathcal{Q}_1$ and $(k,i,r)\in\mathcal{Q}_2$. We have from Eq. (7.29) the following:

$$\begin{aligned}&\mathcal{I}((k,i,x),(k,i,r))\\&\qquad=\big(\{x_i[k+1],r_i[k]\}\cup\{x_j[k]:j\in\bar{\mathcal{N}}_i\}\cup\{r_i[k+1],r_i[k],x_i[k]\}\big)\cap\mathcal{I}_{t_1:t_2},\end{aligned}$$

which implies

$$c(\mathcal{I}((k_1,i_1,x),(k_2,i_2,r)))\leq\min_{(k,i,x)\in\mathcal{Q}_1}(b_{k+1,i}+b_{k,i}+c_{k+1,i}+\sum_{j\in\bar{\mathcal{N}}_i}c_{k,j}).$$

Next, since $\mathcal{I}^\star$ satisfies $r_{\max}(\mathcal{I}^\star)=2$, we recall from Remark 7.5.1 $|\bar{\mathcal{I}}^\star|\geq 2$, where

$$\begin{aligned}&\bar{\mathcal{I}}^\star=\{(k,i,x):x_i[k+1]\in\mathcal{I}^\star,x_i[k]=0\}\cup\{(k,i,x):\{x_i[k+1],x_i[k]\}\subseteq\mathcal{I}^\star\}\\&\cup\{(k,i,r):\{r_i[k+1],x_i[k]\}\subseteq\mathcal{I}^\star,r_i[k]=0\}\cup\{(k,i,r):\{r_i[k+1],r_i[k],x_i[k]\}\subseteq\mathcal{I}^\star\}\end{aligned}$$

which implies $|\mathcal{I}^\star|\geq 2$. In fact, suppose $\mathcal{I}^\star=\{x_i[k+1],x_j[k+1]\}$, where $i\in\mathcal{V}$ and $k\in\{t_1-1,\ldots,t_2-1\}$. Since the elements in $\Phi^x_{k,i}$ and $\Phi^x_{k,j}$ (defined in (7.11)) do not contain $x_i[0]$, $r_i[0]$ or $s_i[0]$ for any $i\in\mathcal{V}$, and cannot all be zero, we see that there exists $x_w[k]\in\mathcal{I}^\star$ (with $x_w[k]>0$), where $w\in\mathcal{V}$. This further implies $|\mathcal{I}^\star|\geq 3$. Using similar arguments, one can show that $|\mathcal{I}^\star|\geq 3$ holds in general, which implies $c(\mathcal{I}^\star)\geq 3c_{\min}$. Combining the above arguments leads to (7.31). ■

7.6 Measurement Selection Problem in Random Measurement Setting

In this section, we assume that the initial condition $l=[(s[0])^T\ (s[0])^T\ (r[0])^T]$ is known. Nevertheless, our analysis can potentially be extended to cases where the initial condition $l=[(s[0])^T\ (s[0])^T\ (r[0])^T]$ is given by a probability distribution.

7.6.1 Problem Formulation

Recall from Section 7.4 that we consider the scenario where the measurement of $x_i[k]$ (resp., $r_i[k]$), denoted as $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$), is given by the pmf $p(\hat{x}_i[k]|x_i[k])$

(resp., $p(\hat{r}_i[k]|r_i[k])$). Note that given the initial conditions $s[0]$, $x[0]$ and $r[0]$, and the parameters β, δ and h, one can express $x_i[k]$ in terms of $l = [(s[0])^T\ (x[0])^T\ (r[0])^T]^T$ and $\theta \triangleq [\beta\ \delta]^T$. Hence, given $l = [(s[0])^T\ (x[0])^T\ (r[0])^T]^T$ and $\theta = [\beta\ \delta]^T$, we can alternatively write $p(\hat{x}_i[k]|x_i[k])$ as $p(\hat{x}_i[k]|l, \theta)$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 1}$. Since the initial conditions are assumed to be known, we drop the dependency of $p(\hat{x}_i[k]|l, \theta)$ on l, and denote the pmf of $\hat{x}_i[k]$ as $p(\hat{x}_i[k]|\theta)$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 1}$. Similarly, given l and θ, we denote the pmf of $\hat{r}_i[k]$ as $p(\hat{r}_i[k]|\theta)$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 1}$. As we mentioned in Section 7.4, when collecting measurement $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$) under a limited budget, one possibility is to give virus (resp., antibody) tests to a group of randomly and uniformly sampled individuals of the population at node $i \in \mathcal{V}$ and at time $k \in \mathbb{Z}_{\geq 1}$. In other words, the obtained measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$, i.e., the corresponding pmfs $p(\hat{x}_i[k]|\theta)$ and $p(\hat{r}_i[k]|\theta)$, depend on the total number of conducted virus tests and antibody tests at node i and at time k, respectively.

Following the arguments in Section 7.5, we assume that collecting measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$ incurs certain costs. Specifically, consider any node $i \in \mathcal{V}$ and any time step $k \in \mathbb{Z}_{\geq 1}$, where the number of total population of i is denoted by $N_i \in \mathbb{Z}_{\geq 1}$ and is assumed to be fixed over time. Suppose we are also allowed to choose the number of virus (resp., antibody) tests that will be performed on the (randomly sampled) individuals at node $i \in \mathcal{V}$ and at time $k \in \mathbb{Z}_{\geq 1}$. Moreover, we assume that the cost of performing the virus (resp., antibody) tests is proportional to the number of the tests. For any $i \in \mathcal{V}$ and for any $k \in \{t_1, \ldots, t_2\}$, we let

$$\mathcal{C}_{k,i} \triangleq \{\zeta c_{k,i} : \zeta \in (\{0\} \cup [\zeta_i])\} \tag{7.32}$$

be the set of all possible costs that we can spend on collecting measurement $\hat{x}_i[k]$, where $c_{k,i} \in \mathbb{R}_{\geq 0}$ and $\zeta_i \in \mathbb{Z}_{\geq 1}$. Similarly, for any $i \in \mathcal{V}$ and for any $k \in \{t_1, \ldots, t_2\}$, we let

$$\mathcal{B}_{k,i} \triangleq \{\eta b_{k,i} : \eta \in (\{0\} \cup [\eta_i])\} \tag{7.33}$$

be the set of all possible costs that we can spend on collecting measurement $\hat{r}_i[k]$, where $b_{k,i} \in \mathbb{R}_{\geq 0}$ and $\eta_i \in \mathbb{Z}_{\geq 1}$. Therefore, spending a cost of $\zeta c_{k,i}$ (with $\zeta \in (\{0\} \cup [\zeta_i])$) on collecting measurement $\hat{x}_i[k]$ can represent testing 10ζ percentage of the population at node i and at time k, i.e., testing $\frac{\zeta}{10} N_i$ individuals at node i and at time k. Alternatively, $\zeta c_{k,i}$ can also be viewed as the cost of performing virus tests on ζN_i^x (randomly sampled) individuals in the population at node i, where $N_i^x \in \mathbb{Z}_{\geq 1}$ and $\zeta_i N_i^x \leq N_i$. To reflect the dependency of the pmf $p(\hat{x}_i[k]|\theta)$ (resp., $p(\hat{r}_i[k]|\theta)$) of measurement $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$) on the cost spent on collecting the measurement of $x_i[k]$ (resp., $r_i[k]$), we further denote the pmf of $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$) as $p(\hat{x}_i[k]|\theta, \varphi_{k,i})$ (resp., $p(\hat{r}_i[k]|\theta, \omega_{k,i})$), where $\varphi_{k,i} \in \mathcal{C}_{k,i}$ (resp., $\omega_{k,i} \in \mathcal{B}_{k,i}$) with $\mathcal{C}_{k.i}$ (resp., $\mathcal{B}_{k,i}$) given by Eq. (7.32) (resp., Eq. (7.33)). Note that $\varphi_{k,i}$ (resp., $\omega_{k,i}$) is the cost that we spend on collecting measurement $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$), and $\varphi_{k,i} = 0$ (resp., $\omega_{k,i} = 0$) indicates that measurement $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$) is not collected.

Recall in the measurement selection problem, the goal is to estimate the unknown parameters β and δ, i.e., the infection rate and the recovery rate of the disease in the SIR dynamics (given by Eq. (7.1)), using a limited number of measurements. In contrast with the exact measurement case studied in Section 7.5, it is not possible to uniquely identify β and δ using measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$ which are now random variables. Thus, we will consider estimators of β and δ based on the (random) measurements indicated by a measurement selection strategy. Similarly to Section 7.5, given time steps $t_1, t_2 \in \mathbb{Z}_{\geq 1}$ with $t_2 \geq t_1$, we first define the set of all candidate measurements as

$$\mathcal{U}_{t_1:t_2} \triangleq \{\hat{x}_i[k] : i \in \mathcal{V}, k \in \{t_1, \ldots, t_2\}\} \cup \{\hat{r}_i[k] : i \in \mathcal{V}, k \in \{t_1, \ldots, t_2\}\}. \tag{7.34}$$

Recalling $\mathcal{C}_{k,i}$ and $\mathcal{B}_{k,i}$ defined in Eq. (7.32) and Eq. (7.33), respectively, we let $\mu \in \mathbb{Z}_{\geq 0}^{\mathcal{U}_{t_1:t_2}}$ be a measurement selection that specifies the costs spent on collecting measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$ for all $i \in \mathcal{V}$ and for all $k \in \{t_1, \ldots, t_2\}$. Moreover, we define the set of all candidate measurement selections as

$$\mathcal{M} \triangleq \{\mu \in \mathbb{Z}_{\geq 0}^{\mathcal{U}_{t_1:t_2}} : \mu(\hat{x}_i[k]) \in (\{0\} \cup [\zeta_i]), \mu(\hat{r}_i[k]) \in (\{0\} \cup [\eta_i])\}, \tag{7.35}$$

where $\zeta_i, \eta_i \in \mathbb{Z}_{\geq 1}$ for all $i \in \mathcal{V}$. In other words, a measurement selection μ is defined over the integer lattice $\mathbb{Z}_{\geq 0}^{\mathcal{U}_{t_1:t_2}}$ so that μ is a vector of dimension $|\mathcal{U}_{t_1:t_2}|$, where each element of μ corresponds to an element in $\mathcal{U}_{t_1:t_2}$, and is denoted as $\mu(\hat{x}_i[k])$ (or $\mu(\hat{r}_i[k])$). The set $\mathcal{M}$ contains all $\mu \in \mathbb{Z}_{\geq 0}^{\mathcal{U}_{t_1:t_2}}$ such that $\mu(\hat{x}_i[k]) \in (\{0\} \cup [\zeta_i])$ and $\mu(\hat{r}_i[k]) \in (\{0\} \cup [\eta_i])$ for all $i \in \mathcal{V}$ and for all $k \in \{t_1, \ldots, t_2\}$. Thus, for any $\varphi_{k,i} \in \mathcal{C}_{k,i}$ and $\omega_{k,i} \in \mathcal{B}_{k,i}$, there exists $\mu \in \mathcal{M}_{\geq 0}^{\mathcal{U}_{t_1:t_2}}$ such that $\mu(\hat{x}_i[k])c_{k,i} = \varphi_{k,i}$ and $\mu(\hat{r}_i[k])b_{k,i} = \omega_{k,i}$. In other words, $\mu(\hat{x}_i[k])c_{k,i}$ (resp., $\mu(\hat{r}_i[k])b_{k,i}$) indicates the cost spent on collecting the measurement of $x_i[k]$ (resp., $r_i[k]$). Given a measurement selection $\mu \in \mathbb{Z}_{\geq 0}^{t_1:t_2}$, we can also denote the pmfs of $\hat{x}_i[k]$ and $\hat{r}_i[k]$ as $p(\hat{x}_i[k]|\theta, \mu(\hat{x}_i[k]))$ and $p(\hat{r}_i[k]|\theta, \mu(\hat{r}_i[k]))$, respectively, where we drop the dependencies of the pmfs on $c_{k,i}$ and $b_{k,i}$ for notational simplicity.

To proceed, we consider the scenario where measurements can only be collected under a budget constraint given by $B \in \mathbb{R}_{\geq 0}$. Using the above notations, the budget constraint can be expressed as

$$\sum_{\hat{x}_i[k] \in \mathcal{U}_{t_1:t_2}} c_{k,i}\mu(\hat{x}_i[k]) + \sum_{\hat{r}_i[k] \in \mathcal{U}_{t_1:t_2}} b_{k,i}\mu(\hat{r}_i[k]) \leq B. \tag{7.36}$$

We then consider estimators of $\theta = [\beta \ \delta]^T$ based on any given measurement selection $\mu \in \mathcal{M}$. Considering any $\mu \in \mathcal{M}$, we denote

$$\mathcal{U}_i^\lambda \triangleq \{k : \mu(\hat{\lambda}_i[k]) > 0, k \in \{t_1, \ldots, t_2\}\}, \tag{7.37}$$

for all $i \in \mathcal{V}$ and for all $\lambda \in \{x, r\}$. For all $i \in \mathcal{V}$ and for all $\lambda \in \{x, r\}$ with $\mathcal{U}_i^\lambda \neq \emptyset$, denote $y(\mathcal{U}_i^\lambda) \triangleq \left[\hat{\lambda}_i[k_1] \quad \cdots \hat{\lambda}_i[k_{|\mathcal{U}_i^\lambda|}]\right]^T$, where $\mathcal{U}_i^\lambda = \{k_1, \ldots, k_{|\mathcal{U}_i^\lambda|}\}$. Letting

$$\mathcal{U}_\lambda \triangleq \{i : \mathcal{U}_i^\lambda \neq \emptyset, i \in \mathcal{V}\}, \ \forall \lambda \in \{x, r\}, \tag{7.38}$$

we denote the measurement vector indicated by $\mu \in \mathcal{M}$ as

$$y(\mu) \triangleq \left[(y(\mathcal{U}_{i_1}^x))^T \quad \cdots \quad (y(\mathcal{U}_{i_{|\mathcal{U}_x|}}^x))^T \quad (y(\mathcal{U}_{j_1}^r))^T \quad \cdots \quad (y(\mathcal{U}_{j_{|\mathcal{U}_r|}}^r))^T\right]^T, \tag{7.39}$$

where $\mathcal{U}_x = \{i_1, \ldots, i_{|\mathcal{U}_x|}\}$ and $\mathcal{U}_r = \{j_1, \ldots, j_{|\mathcal{U}_r|}\}$. Note that $\hat{x}_i[k]$ and $\hat{r}_i[k]$ are (discrete) random variables with pmfs $p(\hat{x}_i[k]|\theta, \mu(\hat{x}_i[k]))$ and $p(\hat{r}_i[k]|\theta, \mu(\hat{r}_i[k]))$, respectively. We then see from Eq. (7.39) that $y(\mu)$ is a random vector whose pmf

is denoted as $p(y(\mu)|\theta,\mu)$. Similarly, the pmf of $y(\mathcal{U}_i^x)$ (resp., $y(\mathcal{U}_i^r)$) is denoted as $p(y(\mathcal{U}_i^x)|\theta,\mu)$ (resp., $p(y(\mathcal{U}_i^r)|\theta,\mu)$). Given $t_1,t_2 \in \mathbb{Z}_{\geq 1}$ with $t_2 \geq t_1$, we make the following assumption on measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$.

Assumption 7.6.1 *For any $i \in \mathcal{V}$ and for any $k_1,k_2 \in \{t_1,\ldots,t_2\}$ ($k_1 \neq k_2$), $\hat{x}_i[k_1]$, $\hat{x}_i[k_2]$, $\hat{r}_i[k_1]$ and $\hat{r}_i[k_2]$ are independent of each other. Moreover, for any $i,j \in \mathcal{V}$ ($i \neq j$) and for any $k_1,k_2 \in \{t_1,\ldots,t_2\}$, $\hat{x}_i[k_1]$ and $\hat{x}_j[k_2]$ are independent, and $\hat{x}_i[k_1]$ and $\hat{r}_j[k_2]$ are independent.*

The above assumption ensures that measurements from different nodes or from different time steps are independent, and the measurements of $x_i[k]$ and $r_i[k]$ are also independent. It then follows from Eq. (7.39) that the pmf of $y(\mu)$ can be written as

$$p(y(\mu)|\theta,\mu) = \prod_{i\in\mathcal{U}_x} p(y(\mathcal{U}_i^x)|\theta,\mu) \cdot \prod_{j\in\mathcal{U}_r} p(y(\mathcal{U}_j^r)|\theta,\mu), \tag{7.40}$$

where we can further write $p(y(\mathcal{U}_i^x)|\theta,\mu) = \prod_{k\in\mathcal{U}_i^x} p(\hat{x}_i[k]|\theta,\mu(\hat{x}_i[k]))$ for all $i \in \mathcal{U}_x$, and $p(y(\mathcal{U}_j^r)|\theta,\mu) = \prod_{k\in\mathcal{U}_j^r} p(\hat{r}_j[k]|\theta,\mu(\hat{r}_j[k]))$ for all $j \in \mathcal{U}_r$.

In order to quantify the performance (e.g., precision) of estimators of θ based on μ, we use the Cramer-Rao Lower Bound (CRLB) (e.g., [79]) associated with μ. In the following, we introduce the CRLB, and explain why we choose it as a performance metric for the problem considered in this section. First, given any measurement $\mu \in \mathcal{M}$, the corresponding CRLB, denoted as $C_\theta(\mu)$, is given by (e.g., [79])

$$C_\theta(\mu) \triangleq \big(F_\theta(\mu)\big)^{-1}, \tag{7.41}$$

where $F_\theta(\mu)$ is the corresponding Fisher information matrix defined as

$$F_\theta(\mu) \triangleq -\mathbb{E}\begin{bmatrix} \frac{\partial^2 \ln p(y(\mu)|\theta,\mu)}{\partial\beta^2} & \frac{\partial^2 \ln p(y(\mu)|\theta,\mu)}{\partial\beta\partial\delta} \\ \frac{\partial^2 \ln p(y(\mu)|\theta,\mu)}{\partial\delta\partial\beta} & \frac{\partial^2 \ln p(y(\mu)|\theta,\mu)}{\partial\delta^2} \end{bmatrix} \tag{7.42}$$

with the expectation $\mathbb{E}[\cdot]$ taken with respect to $p(y(\mu)|\theta,\mu)$. Under Assumption 7.6.1, one can use Eq. (7.40) to rewrite Eq. (7.42) as

$$F_\theta(\mu) = -\sum_{\lambda\in\{x,r\}} \sum_{i\in\mathcal{U}_\lambda} \sum_{k\in\mathcal{U}_i^\lambda} \mathbb{E}\begin{bmatrix} \frac{\partial^2 \ln p(\hat{\lambda}_i[k]|\theta,\mu(\hat{\lambda}_i[k]))}{\partial\beta^2} & \frac{\partial^2 \ln p(\hat{\lambda}_i[k]|\theta,\mu(\hat{\lambda}_i[k]))}{\partial\beta\partial\delta} \\ \frac{\partial^2 \ln p(\hat{\lambda}_i[k]|\theta,\mu(\hat{\lambda}_i[k]))}{\partial\delta\partial\beta} & \frac{\partial^2 \ln p(\hat{\lambda}_i[k]|\theta,\mu(\hat{\lambda}_i[k]))}{\partial\delta^2} \end{bmatrix}, \tag{7.43}$$

where $\mathcal{U}_\lambda$ is given in Eq. (7.38), $\mathcal{U}_i^\lambda$ is defined in Eq. (7.37) for all $i \in \mathcal{V}$, and each expectation $\mathbb{E}[\cdot]$ in the summation is taken with respect to $p(\hat{\lambda}_i[k]|\theta, \mu(\hat{\lambda}_i[k]))$. Under some regularity conditions on the pmfs of $\hat{x}_i[k]$ and $\hat{r}_i[k]$, Eq. (7.43) can be written as the following (e.g., [79]):

$$F_\theta(\mu) = \sum_{\lambda\in\{x,r\}} \sum_{i\in\mathcal{U}_\lambda} \sum_{k\in\mathcal{U}_i^\lambda} \mathbb{E}\Big[\frac{\partial \ln p(\hat{\lambda}_i[k]|\theta, \mu(\hat{\lambda}_i[k]))}{\partial\theta} \Big(\frac{\partial \ln p(\hat{\lambda}_i[k]|\theta, \mu(\hat{\lambda}_i[k]))}{\partial\theta}\Big)^T\Big]. \quad (7.44)$$

Denoting an (unbiased) estimator of $\theta = [\beta \ \delta]^T$ corresponding to a measurement selection μ as $\hat{\theta}(\mu) \in \mathbb{R}^2$, it is well-known that the following inequality holds under some regularity conditions on the pmfs of $\hat{x}_i[k]$ and $\hat{r}_i[k]$ (e.g., [79]):

$$R_{\hat{\theta}(\mu)} = \mathbb{E}[(\hat{\theta}(\mu) - \theta)(\hat{\theta}(\mu) - \theta)^T] \succeq C_\theta(\mu), \quad (7.45)$$

where $R_{\hat{\theta}(\mu)} \in \mathbb{R}^{2\times 2}$ is the covariance of the estimator $\hat{\theta}(\mu)$, the expectation $\mathbb{E}[\cdot]$ is taken with respect to $p(y(\mu)|\theta, \mu)$, and $C_\theta(\mu)$ is given by Eq. (7.41). In fact, there may exist estimators whose covariances achieve the lower bound in (7.45) (e.g., [79]). For instance, the covariance of the maximum likelihood estimator asymptotically achieves the bound in (7.45), as the number of measurement samples (corresponding to a single node $i \in \mathcal{V}$) indicated by μ tends to infinity (e.g., [79]).

Moreover, noting that when $\ln(p(y(\mu)|\theta, \mu))$ is a nonlinear function (i.e., a polynomial) in β and δ, it follows from Eq. (7.42) that $F_\theta(\mu)$, and thus $C_\theta(\mu)$, will potentially depend on the value of the unknown parameter θ. In other words, the bound in (7.45) is a local bound given any value of θ. However, our goal is to find a measurement selection $\mu \in \mathcal{M}$ such that certain performance metrics of a corresponding estimator of $\theta = [\beta \ \delta]^T$ is optimized, regardless of the true values of β and δ, while satisfying the budget constraint. Therefore, we desire to optimize a performance metric that does not depend on β and δ, which motivates us to further consider the Bayesian Cramer-Rao Lower Bound (BCRLB) (e.g., [117]) described as follows, which leverages prior knowledge about θ to yield a bound that does not depend on the true value of θ. Specifically, consider any estimator $\hat{\theta}(\mu)$ of θ based on a measurement se-

lection $\mu \in \mathcal{M}$, and consider a prior pdf of θ, denoted as $p(\theta)$. Under some regularity conditions on the pmfs of $\hat{x}_i[k]$ and $\hat{r}_i[k]$, and $p(\theta)$, we have (e.g., [117, 118]):

$$R_{\hat{\theta}(\mu)} = \mathbb{E}[(\hat{\theta}(\mu) - \theta)(\hat{\theta}(\mu) - \theta)^T] \succeq \bar{C}(\mu), \tag{7.46}$$

where $R_{\hat{\theta}(\mu)} \in \mathbb{R}^{2\times 2}$ is the error covariance of the estimator $\hat{\theta}(\mu)$, the expectation $\mathbb{E}[\cdot]$ is taken with respect to $p(y(\mu)|\theta, \mu)p(\theta)$, and $\bar{C}(\mu) \in \mathbb{R}^{2\times 2}$ is the BCRLB associated with the measurement selection μ. The BCRLB is defined as (e.g., [117, 118])

$$\bar{C}(\mu) \triangleq (\mathbb{E}_\theta[F_\theta(\mu)] + F_p)^{-1}, \tag{7.47}$$

where $\mathbb{E}_\theta[\cdot]$ denotes the expectation taken with respect to $p(\theta)$, $F_\theta(\mu)$ is given by Eq. (7.42), and $F_p \in \mathbb{R}^{2\times 2}$ encodes the prior knowledge of θ as

$$F_p = -\mathbb{E}_\theta \begin{bmatrix} \frac{\partial^2 \ln p(\theta)}{\partial \beta^2} & \frac{\partial^2 \ln p(\theta)}{\partial \beta \partial \delta} \\ \frac{\partial^2 \ln p(\theta)}{\partial \delta \partial \beta} & \frac{\partial^2 \ln p(\theta)}{\partial \delta^2} \end{bmatrix} = \mathbb{E}_\theta \Big[\frac{\partial \ln p(\theta)}{\partial \theta} \Big(\frac{\partial \ln p(\theta)}{\partial \theta} \Big)^T \Big] \succeq \mathbf{0}, \tag{7.48}$$

where the second equality holds under some regularity conditions on $p(\theta)$ [118]. Similar asymptotic analysis to that for the CRLB may be applied to the BCRLB (e.g., [118]).

Thus, the above arguments motivate us to consider (functions of) $\bar{C}(\cdot)$ as optimization metrics in the measurement selection problem studied in this section, in order to characterize the estimation performance corresponding to a measurement selection $\mu \in \mathcal{M}$. In particular, we will consider $\mathrm{tr}(\bar{C}(\cdot))$ and $\ln\det(\bar{C}(\cdot))$, which are widely used criteria in parameter estimation (e.g., [67]), and are also known as the Bayesian A-optimality and D-optimality criteria respectively in the context of experimental design (e.g., [119]). First, considering the optimization metric $\mathrm{tr}(\bar{C}(\cdot))$, we see from the above arguments that (7.46) directly implies $\mathrm{tr}(R_{\hat{\theta}(\mu)}) \geq \mathrm{tr}(\bar{C}(\mu))$ for all estimators $\hat{\theta}(\mu)$ of θ and for all $\mu \in \mathcal{M}$ [91]. Therefore, a measurement selection $\mu^\star$ that minimizes $\mathrm{tr}(\bar{C}(\mu))$ potentially yields a lower value of $\mathrm{tr}(R_{\hat{\theta}(\mu)})$ for an estimator $\hat{\theta}(\mu)$ of θ. Furthermore, there may exist an estimator $\hat{\theta}(\mu)$ that achieves the BCRLB (asymptotically), i.e., $\mathrm{tr}(\bar{C}(\mu))$ provides the minimum value of $\mathrm{tr}(R_{\hat{\theta}(\mu)})$ that can be possibly achieved by any estimator $\hat{\theta}(\mu)$ of θ, given a measurement selection

μ. Similar arguments hold for the optimization metric $\ln\det(\bar{C}(\cdot))$. Hence, the above arguments further justify using $\bar{C}(\cdot)$ as the optimization metric in the measurement selection problem considered in this section. Denoting

$$f_a(\mu) \triangleq \mathrm{tr}(\bar{C}(\mu)) \text{ and } f_d(\mu) \triangleq \ln\det(\bar{C}(\mu)) \ \forall \mu \in \mathcal{M}, \tag{7.49}$$

we now define the Parameter Estimation Measurement Selection (PEMS) problem.

Problem 7.6.2 *Consider a discrete-time SIR model given by Eq.* (7.1) *with a directed graph* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$*, a weight matrix* $A \in \mathbb{R}^{n\times n}$*, a sampling parameter* $h \in \mathbb{R}_{\geq 0}$*, and an initial condition* $l = [((s[0])^T \ (x[0])^T \ (r[0])^T]^T$*. Moreover, consider time steps* $t_1, t_2 \in \mathbb{Z}_{\geq 1}$ *with* $t_2 \geq t_1$*; a set* $\mathcal{C}_{k,i} = \{\zeta c_{k,i} : \zeta \in (\{0\} \cup [\zeta_i])\}$ *with* $c_{k,i} \in \mathbb{R}_{\geq 0}$ *and* $\zeta_i \in \mathbb{Z}_{\geq 1}$*, for all* $i \in \mathcal{V}$ *and for all* $k \in \{t_1, \ldots, t_2\}$*; a set* $\mathcal{B}_{k,i} = \{\eta b_{k,i} : \eta \in (\{0\} \cup [\eta_i])\}$ *with* $b_{k,i} \in \mathbb{R}_{\geq 0}$ *and* $\eta_i \in \mathbb{Z}_{\geq 1}$*, for all* $i \in \mathcal{V}$ *and for all* $k \in \{t_1, \ldots, t_2\}$*; a budget* $B \in \mathbb{R}_{\geq 0}$*; and a prior pdf* $p(\theta)$*. Suppose* $\hat{x}_i[k]$ *(resp.,* $\hat{r}_i[k]$*) is given by a pmf* $p(\hat{x}_i[k]|\theta, \varphi_{k,i})$ *(resp.,* $p(\hat{r}_i[k]|\theta, \omega_{k,i})$*), where* $\varphi_{k,i} \in \mathcal{C}_{k,i}$ *(resp.,* $\omega_{k,i} \in \mathcal{B}_{k,i}$*). The Parameter Estimation Measurement Selection (PEMS) problem is to find a measurement selection* μ *that solves*

$$\begin{aligned} &\min_{\mu\in\mathcal{M}} f(\mu) \\ &s.t. \sum_{\hat{x}_i[k]\in\mathcal{U}_{t_1:t_2}} c_{k,i}\mu(\hat{x}_i[k]) + \sum_{\hat{r}_i[k]\in\mathcal{U}_{t_1:t_2}} b_{k,i}\mu(\hat{r}_i[k]) \leq B, \end{aligned} \tag{7.50}$$

where $\mathcal{M}$ *is defined in Eq.* (7.35)*,* $f(\cdot) \in \{f_a(\cdot), f_d(\cdot)\}$ *with* $f_a(\cdot)$ *and* $f_d(\cdot)$ *defined in Eq.* (7.49)*,* $\mathcal{U}_{t_1:t_2}$ *is defined in Eq.* (7.34)*, and* $\bar{C}(\mu)$ *is given by Eq.* (7.47)*.*

Note that $F_p \succeq \mathbf{0}$ from (7.48), and $f_a(\mathbf{0}) = \mathrm{tr}(\bar{C}(\mathbf{0})) = \mathrm{tr}((F_p)^{-1})$ and $f_d(\mathbf{0}) = \ln\det(\bar{C}(\mathbf{0})) = \ln\det((F_p)^{-1})$ from Eq. (7.47). We further assume that $F_p \succ \mathbf{0}$ in the sequel, which implies $f(\mu) > 0$ for all $\mu \in \mathcal{M}$.

7.6.2 Solving the PEMS Problem

In this section, we restrict ourselves to a specific measurement model. Similar measurement models have also been considered in [116] and [111] for instance. Nonetheless, our analysis can potentially be extended to other measurement models.

Pmfs of Measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$

As mentioned in Section 7.4 and Section 7.6.1, measurement $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$) is obtained by performing virus (resp., antibody) tests in the population at node $i \in \mathcal{V}$ and at time $k \in \{t_1, \ldots, t_2\}$. Specifically, consider any node $i \in \mathcal{V}$ and any time $k \in \{t_1, \ldots, t_2\}$, where the total population of node i is assumed to be fixed over time and is denoted as $N_i \in \mathbb{Z}_{\geq 1}$. Given any measurement selection $\mu \in \mathcal{M}$ where $\mathcal{M}$ is defined in Eq. (7.35), we recall from Section 7.6.1 that $\mu(\hat{x}_i[k])c_{k,i}$ can be viewed as the cost of performing virus tests on $\mu(\hat{x}_i[k])N_i^x$ randomly and uniformly sampled individuals in the population of node $i \in \mathcal{V}$, where $\mu(\hat{x}_i[k]) \in (\{0\} \cup [\zeta_i])$ (with $\zeta_i \in \mathbb{Z}_{\geq 1}$), $c_{k,i} \in \mathbb{R}_{\geq 0}$ and $N_i^x \in \mathbb{Z}_{\geq 1}$ with $\zeta_i N_i^x \leq N_i$. Note that $x_i[k]$ is the proportion of population at node i and at time k that is infected, and $x_i[k] \in [0, 1)$ under Assumptions 7.3.1-7.3.2 as shown by Lemma 7.3.3. Thus, a randomly and uniformly sampled individual in the population at node i and at time k will be an infected individual (at time k) with probability $x_i[k]$, and will be a non-infected (i.e., susceptible or recovered) individual with probability $1 - x_i[k]$. Supposing the tests are accurate,[3] we see from the above arguments that the obtained number of individuals that are tested positive, i.e., $N_i\hat{x}_i[k]$, is a binomial random variable with parameters $N_i^x\mu(\hat{x}_i[k]) \in \mathbb{Z}_{\geq 1}$ and $x_i[k] \in [0, 1)$. Therefore, for any $i \in \mathcal{V}$ and for any $k \in \{t_1, \ldots, t_2\}$, the pmf of $\hat{x}_i[k]$ is given by

$$p(\hat{x}_i[k] = x|\theta, \mu(\hat{x}_i[k])) = \binom{N_i^x\mu(\hat{x}_i[k])}{N_i x}(x_i[k])^{N_i x}(1 - x_i[k])^{N_i^x\mu(\hat{x}_i[k]) - N_i x}, \quad (7.51)$$

where $x \in \{0, \frac{1}{N_i}, \frac{2}{N_i}, \ldots, \frac{N_i^x\mu(\hat{x}_i[k])}{N_i}\}$ with $x \in [0, 1]$ since $N_i^x\zeta_i \leq N_i$. Note that we do not define the pmf of measurement $\hat{x}_i[k]$ when $N_i^x\mu(\hat{x}_i[k]) = 0$, i.e., when $\mu(\hat{x}_i[k]) = 0$, since $\mu(\hat{x}_i[k]) = 0$ indicates no measurement is collected for state $x_i[k]$. Also note that when $x_i[k] = 0$, the pmf of $\hat{x}_i[k]$ given in Eq. (7.51) reduces to $p(\hat{x}_i[k] = 0|\theta, \mu(\hat{x}_i[k])) = 1$. Moreover, since the weight matrix $A \in \mathbb{R}^{n\times n}$ and the sampling parameter $h \in \mathbb{R}_{\geq 0}$ are assume to be given, we see that given $\theta = [\beta\ \delta]^T$ and initial

[3]Here, "accurate" means that an infected individual will be tested positive with probability one, and an individual that is not infected will be tested negative with probability one.

condition $l = [(s[0])^T\ (x[0])^T\ (r[0])^T]^T$, $x_i[k]$ can be obtained using Eq. (7.1b) for all $i \in \mathcal{V}$ and for all $k \in \{t_1, \ldots, t_2\}$, where we can view $x_i[k]$ as a function in the unknown parameter θ. In other words, given l, θ, $\mu(\hat{x}_i[k])$, N_i^x and N_i, one can obtain the right-hand side of Eq. (7.51). Again, we only explicitly express the dependency of the pmf of $\hat{x}_i[k]$ on θ and $\mu(\hat{x}_i[k])$ in Eq. (7.51). Following similar arguments to those above, we assume that for any $i \in \mathcal{V}$ and for any $k \in \{t_1, \ldots, t_2\}$, measurement $\hat{r}_i[k]$ has the following pmf:

$$p(\hat{r}_i[k] = r|\theta, \mu(\hat{r}_i[k])) = \binom{N_i^r \mu(\hat{r}_i[k])}{N_i r} (r_i[k])^{N_i r} (1 - r_i[k])^{N_i^r \mu(\hat{r}_i[k]) - N_i r}, \tag{7.52}$$

where $r \in \{0, \frac{1}{N_i}, \frac{2}{N_i}, \ldots, \frac{N_i^r \mu(\hat{r}_i[k])}{N_i}\}$ with $r \in [0, 1]$, $\mu(\hat{r}_i[k]) \in \{0, \ldots, \eta_i\}$, $N_i^r \in \mathbb{Z}_{\geq 1}$ and $N_i^r \mu(\hat{r}_i[k]) \leq N_i$. Similarly, we note that the pmf of $\hat{r}_i[k]$ given in Eq. (7.52) reduces to $p(\hat{r}_i[k] = 0|\theta, \mu(\hat{r}_i[k])) = 1$ when $r_i[k] = 0$. The following standard result for binomial random variables will be useful.

Lemma 7.6.3 *For a binomial random q with parameters $m \in \mathbb{Z}_{\geq 1}$ and $p_s \in (0, 1)$, the CRLB is given by $C_{p_s} \triangleq (F_{p_s})^{-1}$, where $F_{p_s} \triangleq -\mathbb{E}\Big[\frac{\partial^2 \ln p(q)}{\partial p_s^2}\Big] = \mathbb{E}\Big[\big(\frac{\partial \ln p(q)}{\partial p_s}\big)^2\Big] = \frac{m}{p_s(1-p_s)}$ with the expectation $\mathbb{E}[\cdot]$ taken with respect to the pmf $p(q)$.*

Considering any measurement selection $\mu \in \mathcal{M}$ and any measurement $\hat{\lambda}_i[k] \in \mathcal{U}_{t_1:t_2}$, where $\lambda \in \{x, r\}$ and $\mathcal{U}_{t_1:t_2}$ is defined in Eq. (7.34), we have the following:

$$\begin{aligned}
&\mathbb{E}\Big[\frac{\partial \ln p(\hat{\lambda}_i[k]|\theta, \mu(\hat{\lambda}_i[k]))}{\partial \theta}\big(\frac{\partial \ln p(\hat{\lambda}_i[k]|\theta, \mu(\lambda_i[k]))}{\partial \theta}\big)^T\Big] \\
=&\mathbb{E}\Big[\big(\frac{\partial \ln p(\hat{\lambda}_i[k]|\theta, \mu(\hat{\lambda}_i[k]))}{\partial \lambda_i[k]}\big)^2 \cdot \frac{\partial \lambda_i[k]}{\partial \theta}\big(\frac{\partial \lambda_i[k]}{\partial \theta}\big)^T\Big] &(7.53)\\
=&\frac{N_i^\lambda \mu(\hat{\lambda}_i[k])}{\lambda_i[k](1 - \lambda_i[k])} \cdot \frac{\partial \lambda_i[k]}{\partial \theta}\big(\frac{\partial \lambda_i[k]}{\partial \theta}\big)^T, &(7.54)
\end{aligned}$$

where the expectation $\mathbb{E}[\cdot]$ is taken with respect to $p(\hat{\lambda}_i[k]|\theta, \mu(\hat{\lambda}_i[k]))$, and $\lambda_i[k] \in [0, 1)$. To obtain (7.53), we note the form of $\ln p(\hat{\lambda}_i[k]|\theta, \mu(\hat{\lambda}_i[k]))$ in Eq. (7.51), and use the chain rule. Moreover, we obtain (7.54) from Lemma 7.6.3. Noting that the pmf of $\hat{\lambda}_i[k]$ reduces to $p(\hat{\lambda}_i[k] = 0|\theta, \mu(\hat{\lambda}_i[k])) = 1$ if $\lambda_i[k] = 0$ as argued above, we let the right-hand side of (7.54) be zero if $\lambda_i[k] = 0$.

Complexity of the PEMS Problem

Here, we will show that the PEMS problem is NP-hard, i.e., there exist instances of the PEMS problem that cannot be solved optimally by any polynomial-time algorithm (if P $\neq$ NP).

Theorem 7.6.4 *The PEMS problem is NP-hard.*

Proof We prove the NP-hardness of the PEMS problem via a (polynomial-time) reduction from the knapsack problem which is known to be NP-hard (e.g., [41]). An instance of the knapsack problem is given by a finite set $D = \{d_1, \ldots, d_\tau\}$, a size $s(d) \in \mathbb{Z}_{>0}$ and a value $v(d) \in \mathbb{Z}_{>0}$ for each $d \in D$, and $K \in \mathbb{Z}_{>0}$. The knapsack problem is to find $D' \subseteq D$ such that $\sum_{d\in D'} v(d)$ is maximized while satisfying $\sum_{d\in D'} s(d) \leq K$.

Given any knapsack instance, we construct an instance of the PEMS problem as follows. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph that contains a set of n isolated nodes with $n = \tau$ and $\mathcal{V} = [n]$. Set the weight matrix to be $A = \mathbf{0}_{n\times n}$, and set the sampling parameter as $h = 1$. The time steps t_1 and t_2 are set to be $t_1 = t_2 = 1$, i.e., only the measurements of $x_i[1]$ and $r_i[1]$ for all $i \in \mathcal{V}$ will be considered. The initial condition is set to satisfy $s_i[0] = 0.5$, $x_i[0] = 0.5$ and $r_i[0] = 0$ for all $i \in \mathcal{V}$. The budget constraint is set as $B = K$. Let $\mathcal{C}_{1,i} = \{0, B + 1\}$ and $\mathcal{B}_{1,i} = \{0, s(d_i)\}$ for all $i \in \mathcal{V}$. The pmfs of measurements $\hat{x}_i[1]$ and $\hat{r}_i[1]$ are given by Eqs. (7.51) and (7.52), respectively, with $N_i^x = N_i^r = v(d_i)$ and $N_i = \max_{i\in\mathcal{V}} v(d_i)$ for all $i \in \mathcal{V}$, where Assumption 7.6.1 is assumed to hold. Finally, let the prior pdf of $\beta \in (0, 1)$ be a Beta distribution with parameters $\alpha_1 = 3$ and $\alpha_2 = 3$, and let the prior pdf of $\delta \in (0, 1)$ also be a Beta distribution with parameters $\alpha_1 = 3$ and $\alpha_2 = 3$,[4] where we take β and δ to be independent. Noting that $\mathcal{C}_{1,i} = \{0, B + 1\}$ in the PEMS instance constructed above, i.e., $\hat{x}_i[k]$ incurs a cost of $B + 1 > B$, we only need to consider measurements $\hat{r}_i[1]$ for all $i \in \mathcal{V}$. Moreover, since $\mathcal{B}_{1,i} = \{0, s(d_i)\}$, a corresponding measurement selection is then given by $\mu \in \{0, 1\}^{\mathcal{V}}$. In other words, $\mu(i) = 1$ if measurement $\hat{r}_i[1]$

[4]The pdf of a Beta distribution with parameters $\alpha_1 \in \mathbb{R}_{>0}$ and $\alpha_2 \in \mathbb{R}_{>0}$ is given by $p(x|\alpha_1, \alpha_2) = \frac{x^{\alpha_1-1}(1-x)^{\alpha_2-1}}{\mathcal{B}(\alpha_1,\alpha_2)}$, where $x \in (0, 1)$ and $\mathcal{B}(\alpha_1, \alpha_2) = \int_0^1 y^{\alpha_1-1}(1 - y)^{\alpha_2-1}dy$. See [120] for more details.

is collected (with cost $s(d_i)$), and $\mu(i) = 0$ if measurement $\hat{r}_i[1]$ is not collected. We will see that there is a one to one correspondence between a measurement $\hat{r}_i[1]$ in the PEMS instance and an element $d_i \in D$ in the knapsack instance.

Given a measurement selection $\mu \in \{0,1\}^{\mathcal{V}}$, we have from Eq. (7.44) the following:

$$F_\theta(\mu) = \sum_{i \in \text{supp}(\mu)} \mathbb{E}\Big[\frac{\partial \ln p(\hat{r}_i[1]|\theta, \mu(i))}{\partial \theta}\Big(\frac{\partial \ln p(\hat{r}_i[1]|\theta, \mu(i))}{\partial \theta}\Big)^T\Big], \tag{7.55}$$

where each expectation $\mathbb{E}[\cdot]$ in the summation is taken with respect to $p(\hat{r}_i[1]|\theta, \mu(i))$, $\text{supp}(\mu) \triangleq \{i : \mu(i) \neq 0\}$ and $\theta = [\beta \ \delta]^T$. Moreover, we see from Eq. (7.54) that

$$\mathbb{E}\Big[\frac{\partial \ln p(\hat{r}_i[1]|\theta, \mu(i))}{\partial \theta}\Big(\frac{\partial \ln p(\hat{r}_i[1]|\theta, \mu(i))}{\partial \theta}\Big)^T\Big] = \frac{N_i^r \mu(i)}{r_i[1](1 - r_i[1])} \cdot \frac{\partial r_i[1]}{\partial \theta}\Big(\frac{\partial r_i[1]}{\partial \theta}\Big)^T \ \forall i \in \mathcal{V}. \tag{7.56}$$

Since $r_i[0] = 0$ and $x_i[0] = 0.5$ for all $i \in \mathcal{V}$, Eq. (7.1c) implies $r_i[1] = 0.5h\delta$ for all $i \in \mathcal{V}$, where $h = 1$. We then have from Eqs. (7.55) and (7.56) the following:

$$F_\theta(\mu) = \frac{1}{0.5\delta(1 - 0.5\delta)} \begin{bmatrix} 0 & 0 \\ 0 & 0.25 \end{bmatrix} \sum_{i \in \text{supp}(\mu)} N_i^r \mu(i). \tag{7.57}$$

Next, noting that β and δ are independent, one can show via Eq. (7.48) that

$$F_p = -\mathbb{E}_\theta \begin{bmatrix} \frac{\partial^2 \ln p(\beta)}{\partial \beta^2} & 0 \\ 0 & \frac{\partial^2 \ln p(\delta)}{\partial \delta^2} \end{bmatrix}, \tag{7.58}$$

where one can further show that $(F_p)_{11} = (F_p)_{22} > 0$ using the fact that the pdfs of β and δ are Beta distributions with parameters $\alpha_1 = 3$ and $\alpha_2 = 3$. Similarly, one can obtain $\mathbb{E}_\theta[1/0.5\delta(1 - 0.5\delta)] > 0$. It now follows from Eqs. (7.57) and (7.58) that

$$\mathbb{E}_\theta[F_\theta(\mu)] + F_p = \begin{bmatrix} z_1 & 0 \\ 0 & z_1 + z_2 \sum_{i \in \text{supp}(\mu)} N_i^r \mu(i) \end{bmatrix}, \tag{7.59}$$

where $z_1, z_2 \in \mathbb{R}_{>0}$ are some constants (independent of μ). Note that the objective in the PEMS instance is given by $\min_{\mu \in \{0,1\}^{\mathcal{V}}} f(\mu)$, where $f(\cdot) \in \{f_a(\cdot), f_d(\cdot)\}$. First, considering the objective function $f_a(\mu) = \text{tr}(\bar{C}(\mu))$, where $\bar{C}(\mu) = (\mathbb{E}_\theta[F_\theta(\mu)] + F_p)^{-1}$, we see from Eq. (7.59) that $\text{tr}(\bar{C}(\mu))$ is minimized (over $\mu \in \{0,1\}^{\mathcal{V}}$) if and only if

$\sum_{i\in\text{supp}(\mu)} N_i^r \mu(i)$ is maximized. Similarly, considering the objective function $f_d(\mu) = \ln\det(\bar{C}(\mu))$, we see from Eq. (7.59) that $\ln\det(\bar{C}(\mu))$ is minimized (over $\mu \in \{0,1\}^{\mathcal{V}}$) if and only if $\sum_{i\in\text{supp}(\mu)} N_i^r \mu(i)$ is maximized.

By the way we constructed the PEMS instance from the given knapsack instance, it follows directly from the above arguments that a measurement selection $\mu^\star \in \{0,1\}^{\mathcal{V}}$ is an optimal solution to the PEMS instance if and only if $D^\star \triangleq \{d_i : i \in \text{supp}(\mu^\star)\}$ is an optimal solution to the knapsack instance. Since the knapsack problem is NP-hard, we conclude that the PEMS problem is NP-hard. ■

Remark 7.6.5 *Theorem* 7.6.4 *shows that the PEMS problem is NP-hard even when only the measurements at time step $k = 1$ can be collected, and each measurement can either be collected with a certain cost or not collected.*

Algorithm for the PEMS Problem

Theorem 7.6.4 motivates us to consider approximation algorithms for solving the PEMS problem. To begin with, we note that the objective function in the PEMS problem can be viewed as a function defined over an integer lattice. We then have $f_a : \mathcal{M} \to \mathbb{R}_{\geq 0}$ and $f_d : \mathcal{M} \to \mathbb{R}_{\geq 0}$, where $\mathcal{M}$ is defined in Eq. (7.35). First, considering $f_a : \mathcal{M} \to \mathbb{R}_{\geq 0}$, we will define a set function $f_{Pa} : 2^{\bar{M}} \to \mathbb{R}_{\geq 0}$, where $\bar{\mathcal{M}}$ is a set constructed as

$$\bar{\mathcal{M}} \triangleq \{(\hat{x}_i[k], l_1) : i \in \mathcal{V}, k \in \{t_1, \ldots, t_2\}, l_1 \in [\zeta_i]\} \cup \{(\hat{r}_i[k], l_2) : i \in \mathcal{V}, k \in \{t_1, \ldots, t_2\}, l_2 \in [\eta_i]\}. \quad (7.60)$$

In other words, for any $i \in \mathcal{V}$ and for any $k \in \{t_1, \ldots, t_2\}$, we associate elements $(\hat{x}_i[k], 1), \ldots, (\hat{x}_i[k], l_1)$ (resp., $(\hat{r}_i[k], 1), \ldots, (\hat{r}_i[k], l_2)$) in set $\bar{\mathcal{M}}$ to measurement $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$). The set function $f_{Pa}(\cdot)$ is then defined as

$$f_{Pa}(\mathcal{Y}) \triangleq f_a(\mathbf{0}) - f_a(\mu_{\mathcal{Y}}) = \text{tr}(\bar{C}(\mathbf{0})) - \text{tr}(\bar{C}(\mu_{\mathcal{Y}})) \ \forall \mathcal{Y} \subseteq \bar{\mathcal{M}}, \quad (7.61)$$

where for any $\mathcal{Y} \subseteq \bar{\mathcal{M}}$, we define $\mu_{\mathcal{Y}} \in \mathcal{M}$ such that $\mu_{\mathcal{Y}}(\hat{x}_i[k]) = |\{(\hat{x}_i[k], l_1) : (\hat{x}_i[k], l_1) \in \mathcal{Y}\}|$ and $\mu_{\mathcal{Y}}(\hat{r}_i[k]) = |\{(\hat{r}_i[k], l_2) : (\hat{r}_i[k], l_2) \in \mathcal{Y}\}|$ for all $i \in \mathcal{V}$ and for all

$k \in \{t_1, \ldots, t_2\}$. In other words, $\mu_{\mathcal{Y}}(\hat{x}_i[k])$ (resp., $\mu_{\mathcal{Y}}(\hat{r}_i[k])$) is set to be the number of elements in $\mathcal{Y}$ that correspond to the measurement $\hat{x}_i[k]$ (resp., $\hat{r}_i[k]$). Also note that $f_{Pa}(\emptyset) = 0$. Following the arguments leading to (7.54), we define

$$H_y \triangleq \begin{cases} \mathbb{E}_\theta\big[\frac{N_i^x}{x_i[k](1-x_i[k])}\frac{\partial x_i[k]}{\partial \theta}\big(\frac{\partial x_i[k]}{\partial \theta}\big)^T\big] \text{ if } y = (\hat{x}_i[k], l_1) \\ \mathbb{E}_\theta\big[\frac{N_i^r}{r_i[k](1-r_i[k])}\frac{\partial r_i[k]}{\partial \theta}\big(\frac{\partial r_i[k]}{\partial \theta}\big)^T\big] \text{ if } y = (\hat{r}_i[k], l_2) \end{cases} \forall y \in \bar{\mathcal{M}}, \tag{7.62}$$

where $x_i[k], r_i[k] \in [0,1)$, $i \in \mathcal{V}$, $k \in \{t_1, \ldots . t_2\}$, $l_1 \in [\zeta_i]$, $l_2 \in [\eta_i]$, and the expectation $\mathbb{E}_\theta[\cdot]$ is taken with respect to the prior pdf $p(\theta)$. Given any $\theta = [\beta \ \delta]^T$, we see from the arguments for (7.54) that $\frac{N_i^x}{x_i[k](1-x_i[k])}\frac{\partial x_i[k]}{\partial \theta}\big(\frac{\partial x_i[k]}{\partial \theta}\big)^T \succeq \mathbf{0}$. Moreover, one can show that $\mathbb{E}_\theta\big[\frac{N_i^x}{x_i[k](1-x_i[k])}\frac{\partial x_i[k]}{\partial \theta}\big(\frac{\partial x_i[k]}{\partial \theta}\big)^T\big] \succeq \mathbf{0}$. Similarly, one can obtain $\mathbb{E}_\theta\big[\frac{N_i^r}{r_i[k](1-r_i[k])}\frac{\partial r_i[k]}{\partial \theta}\big(\frac{\partial r_i[k]}{\partial \theta}\big)^T\big] \succeq \mathbf{0}$, which implies $H_y \succeq \mathbf{0}$ for all $y \in \bar{\mathcal{M}}$. Now, suppose the pmfs of $\hat{x}_i[k]$ and $\hat{r}_i[k]$ are given by Eq. (7.51) and Eq. (7.52), respectively. Recall from Eq. (7.47) that $\mathrm{tr}(\bar{C}(\mu)) = \mathrm{tr}((\mathbb{E}_\theta[F_\theta(\mu)] + F_p)^{-1})$ for all $\mu \in \mathcal{M}$, where F_p and $F_\theta(\mu)$ are given by (7.48) and (7.44), respectively. Supposing Assumption 7.6.1 holds, for all $\mathcal{Y} \subseteq \bar{\mathcal{M}}$, one can first express $F_\theta(\mu_{\mathcal{Y}})$ using (7.54), and then use Eq. (7.62) to obtain $\mathbb{E}_\theta[F_\theta(\mu_{\mathcal{Y}})] = \sum_{y \in \mathcal{Y}} H_y \triangleq H(\mathcal{Y})$, where $\mu_{\mathcal{Y}}$ is defined above given $\mathcal{Y} \subseteq \bar{\mathcal{M}}$. Putting the above arguments together, we have from Eq. (7.61) the following:

$$f_{Pa}(\mathcal{Y}) = \mathrm{tr}\left((F_p)^{-1}\right) - \mathrm{tr}\left((F_p + H(\mathcal{Y}))^{-1}\right) \ \forall \mathcal{Y} \subseteq \bar{\mathcal{M}}. \tag{7.63}$$

We now associate costs to the elements in $\bar{\mathcal{M}}$. Specifically, let the cost of $(\hat{x}_i[k], l_1)$ be $c_{k,i}$, denoted as $c(\hat{x}_i[k], l_1)$, for all $(\hat{x}_i[k], l_1) \in \bar{\mathcal{M}}$, and let the cost of $(\hat{r}_i[k], l_2)$ be $b_{k,i}$, denoted as $c(\hat{r}_i[k], l_2)$, for all $(\hat{r}_i[k], l_2) \in \bar{\mathcal{M}}$, where $c_{k,i} \in \mathbb{R}_{>0}$ and $b_{k,i} \in \mathbb{R}_{>0}$ are given in the instance of the PEMS problem. Setting the cost structure of the elements in $\bar{\mathcal{M}}$ in this way, we establish an equivalence between the cost of a subset $\mathcal{Y} \subseteq \bar{\mathcal{M}}$ and the cost of $\mu_{\mathcal{Y}} \in \mathcal{M}$, where $\mu_{\mathcal{Y}}$ is defined above, i.e.,

$$\sum_{\lambda \in \{x,r\}} \sum_{(\hat{\lambda}_i[k], l) \in \mathcal{Y}} c(\hat{\lambda}_i[k], l) = \sum_{\hat{x}_i[k] \in \mathcal{U}_{t_1:t_2}} c_{k,i}\mu_Y(\hat{x}_i[k]) + \sum_{\hat{r}_i[k] \in \mathcal{U}_{t_1:t_2}} b_{k,i}\mu_{\mathcal{Y}}(\hat{r}_i[k]).$$

Similarly, considering the objective function $f_d : \mathcal{M} \to \mathbb{R}_{\geq 0}$ in the PEMS problem, we define a set function $f_{Pd} : 2^{\bar{\mathcal{M}}} \to \mathbb{R}_{\geq 0}$ as

$$f_{Pd}(\mathcal{Y}) \triangleq f_d(\mathbf{0}) - f_d(\mu_{\mathcal{Y}}) = \ln\det(F_p + H(\mathcal{Y})) - \ln\det(F_p) \ \forall \mathcal{Y} \subseteq \bar{\mathcal{M}}, \tag{7.64}$$

where we define $\mu_{\mathcal{Y}} \in \mathcal{M}$ such that $\mu_{\mathcal{Y}}(\hat{x}_i[k]) = |\{(\hat{x}_i[k], l_1) : (\hat{x}_i[k], l_1) \in \mathcal{Y}\}|$ and $\mu_{\mathcal{Y}}(\hat{r}_i[k]) = |\{(\hat{r}_i[k], l_2) : (\hat{r}_i[k], l_2) \in \mathcal{Y}\}|$ for all $i \in \mathcal{V}$ and for all $k \in \{t_1, \ldots, t_2\}$.

Note that given an instance of the PEMS problem in Problem 7.6.2, we can construct the set $\bar{\mathcal{M}}$ with the associated costs of the elements in $\bar{\mathcal{M}}$ in $O(n(\zeta + \eta))$ time, where n is the number of nodes in graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, and $\zeta, \eta \in \mathbb{Z}_{\geq 1}$ with $\zeta_i \leq \zeta$ and $\eta_i \leq \eta$ for all $i \in \mathcal{V}$. Assuming that ζ and η are constants (i.e., they are fixed), we can construct the set $\bar{\mathcal{M}}$ with the associated costs in $O(n)$ time which is a polynomial in the size of the given PEMS instance. Based on the above arguments, we further consider the following problem:

$$\begin{aligned} &\max_{\mathcal{Y} \subseteq \bar{\mathcal{M}}} f_P(\mathcal{Y}) \\ &s.t.\ c(\mathcal{Y}) \leq B, \end{aligned} \tag{P}$$

where $f_P(\cdot) \in \{f_{Pa}(\cdot), f_{Pd}(\cdot)\}$ with $f_{Pa}(\cdot)$ and $f_{Pd}(\cdot)$ given by in (7.63) and (7.64), respectively, and $c(\mathcal{Y}) \triangleq \sum_{y \in \mathcal{Y}} c(y)$ for all $\mathcal{Y} \subseteq \bar{\mathcal{M}}$. By the manner in which we construct $f_P(\cdot)$ and the costs of elements in $\bar{\mathcal{M}}$, one can verify that $\mathcal{Y}_a^\star \subseteq \bar{\mathcal{M}}$ (resp., $\mathcal{Y}_d^\star \subseteq \bar{\mathcal{M}}$) is an optimal solution to Problem (P) with $f_P(\cdot) = f_{Pa}(\cdot)$ (resp., $f_P(\cdot) = f_{Pd}(\cdot)$) if and only if $\mu_{\mathcal{Y}_a^\star}$ (resp., $\mu_{\mathcal{Y}_d^\star}$) defined above is an optimal solution to (7.50) in Problem 7.6.2 with $f(\cdot) = f_a(\cdot)$ (resp., $f(\cdot) = f_d(\cdot)$). Therefore, given a PEMS instance, we can first construct set $\bar{\mathcal{M}}$ with the associated cost for each element in $\bar{\mathcal{M}}$, and then solve Problem (P). Note that Problem (P) can be viewed as a problem of maximizing a set function subject to a knapsack constraint. In particular, a greedy algorithm (Algorithm 7.6.1) has been proposed to solve this problem with performance guarantees when the objective function is monotone nondecreasing and submodular[5] (e.g., [121] and [85]). Here, we note from the definition of Algorithm 7.6.1 that the number of evaluations of function $f_P(\cdot)$ required in the algorithm is $O(|\bar{\mathcal{M}}|^2)$. One can also observe that the objective function $f_{Pd}(\mathcal{Y}) = \ln\det(F_p + H(\mathcal{Y})) - \ln\det(F_p)$ in Problem (P) shares a similar form with that in [46]. Thus, using similar

[5] A set function $g : 2^{\mathcal{V}} \to \mathbb{R}$, where $\mathcal{V} = [n]$ is the ground set, is said to be monotone nondecreasing if $g(\mathcal{A}) \leq g(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$. $g(\cdot)$ is called submodular if $g(\{y\} \cup \mathcal{A}) - g(\mathcal{A}) \geq g(\{y\} \cup \mathcal{B}) - g(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and for all $y \in \mathcal{V} \setminus \mathcal{B}$.

arguments to those in [46], one can show that $f_{Pd}(\cdot)$ is monotone nondecreasing and submodular. Combining the above arguments together and noting that $f_{Pd}(\emptyset) = 0$, we then have the following result for the performance of the greedy algorithm defined in Algorithm 7.6.1 when applied to solving Problem (P) with $f_P(\cdot) = f_{Pd}(\cdot)$.

Algorithm 7.6.1 Greedy algorithm for PEMS

1: **Input**: An instance of PEMS transformed into the form in (P)
2: **Output**: $\mathcal{Y}_g$
3: Find $\mathcal{Y}_1 \triangleq \arg\max\{f_P(y) : y \in \bar{\mathcal{M}}\}$
4: Initialize $\mathcal{Y}_2 = \emptyset$ and $\mathcal{C} = \bar{\mathcal{M}}$
5: **while** $\mathcal{C} \neq \emptyset$ **do**
6: Find $y^\star \in \arg\max_{y\in\mathcal{C}} \frac{f_P(\{y\}\cup\mathcal{Y}_2)-f_P(\mathcal{Y}_2)}{c(y)}$
7: **if** $c(y^\star) + c(\mathcal{Y}_2) \leq B$ **then**
8: $\mathcal{Y}_2 = \{y^\star\} \cup \mathcal{Y}_2$
9: $\mathcal{C} = \mathcal{C} \setminus \{y^\star\}$
10: $\mathcal{Y}_g = \arg\max\{f_P(\mathcal{Y}_1), f_P(\mathcal{Y}_2)\}$

Theorem 7.6.6 *Consider Problem* (P) *with the objective function* $f_{Pd} : 2^{\bar{\mathcal{M}}} \to \mathbb{R}_{\geq 0}$ *given by* (7.64). *Then Algorithm* 7.6.1 *yields a solution, denoted as* $\mathcal{Y}_d^g$, *to Problem* (P) *that satisfies*

$$f_{Pd}(\mathcal{Y}_d^g) \geq \frac{1}{2}(1 - e^{-1}) f_{Pd}(\mathcal{Y}_d^\star), \tag{7.65}$$

where $\mathcal{Y}_d^\star \subseteq \bar{\mathcal{M}}$ *is an optimal solution to Problem* (P).

However, the objective function corresponding to the A-optimality criterion (i.e., $f_{Pa}(\cdot)$) is not submodular in general (e.g., [5]). In fact, one can construct examples where the objective function $f_{Pa}(\mathcal{Y}) = \mathrm{tr}((F_p)^{-1}) - \mathrm{tr}\left((F_p + H(\mathcal{Y}))^{-1}\right)$ in the PEMS problem is not submodular. Hence, in order to provide performance guarantees of the greedy algorithm when applied to Problem (P) with $f(\cdot) = f_{Pa}(\cdot)$, we will extend the analysis in [121] to nonsubmodular settings. To proceed, note that for all $\mathcal{A} \subseteq \mathcal{B} \subseteq$

$\bar{\mathcal{M}}$, we have $F_p + H(\mathcal{A}) \preceq F_p + H(\mathcal{B})$, which implies $(F_p + H(\mathcal{A}))^{-1} \succeq (F_p + H(\mathcal{B}))^{-1}$ and $\mathrm{tr}(F_p + H(\mathcal{A}))^{-1}) \geq \mathrm{tr}(F_p + H(\mathcal{B}))^{-1})$ [91]. Therefore, the objective function $f_{Pa}(\cdot)$ is also monotone nondecreasing with $f_{Pa}(\emptyset) = 0$. We then characterize how close $f_{Pa}(\cdot)$ is to being submodular by introducing the following definition.

Definition 7.6.1 *Consider Problem* (P) *with* $f_P(\cdot) = f_{Pa}(\cdot)$*, where* $f_{Pa} : 2^{\bar{\mathcal{M}}} \to \mathbb{R}_{\geq 0}$ *is defined in* (7.61)*. Suppose Algorithm* 7.6.1 *is applied to solve Problem* (P)*. For all* $j \in \{1, \ldots, |\mathcal{Y}_2|\}$*, let* $\mathcal{Y}_2^j = \{y_1, \ldots, y_j\}$ *denote the set that contains the first* j *elements added to set* $\mathcal{Y}_2$ *in Algorithm* 7.6.1*, and let* $\mathcal{Y}_2^0 = \emptyset$*. The type-1 greedy submodularity ratio of* $f_{Pa}(\cdot)$ *is defined to be the largest* $\gamma_1 \in \mathbb{R}$ *that satisfies*

$$\sum_{y \in \mathcal{A} \setminus \mathcal{Y}_2^j} \big(f_{Pa}(\{y\} \cup \mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j)\big) \geq \gamma_1 \big(f_{Pa}(\mathcal{A} \cup \mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j)\big), \tag{7.66}$$

for all $\mathcal{A} \subseteq \bar{\mathcal{M}}$ *and for all* $j \in \{0, \ldots, |\mathcal{Y}_2|\}$*. The type-2 greedy submodularity ratio of* $f_{Pa}(\cdot)$ *is defined to be the largest* $\gamma_2 \in \mathbb{R}$ *that satisfies*

$$f_{Pa}(\mathcal{Y}_1) - f_{Pa}(\emptyset) \geq \gamma_2 \big(f_{Pa}(\{y\} \cup \mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j)\big), \tag{7.67}$$

for all $j \in \{0, \ldots, |\mathcal{Y}_2|\}$ *and for all* $y \in \bar{\mathcal{M}} \setminus \mathcal{Y}_2^j$ *such that* $c(y) + c(\mathcal{Y}_2^j) > B$*, where* $\mathcal{Y}_1 = \arg\max\{f_{Pa}(y) : y \in \bar{\mathcal{M}}\}$*.*

Remark 7.6.7 *Note that* $f_{Pa}(\cdot)$ *is monotone nondecreasing as argued above. From the definition of* γ_1 *in* (7.66)*, one can then show that* $\gamma_1 \in [0, 1]$*; if* $f_{Pa}(\cdot)$ *is submodular,* $\gamma_1 = 1$*. Similarly, one can show that* $\gamma_2 \geq 0$*; if* $f_{Pa}(\cdot)$ *is submodular,* $\gamma_2 \geq 1$*.*

Based on Definition 7.6.1, the following result extends the analysis in [85,121], and characterizes the performance guarantees of the greedy algorithm (Algorithm 7.6.1) for solving Problem (P) with $f_P(\cdot) = f_{Pa}(\cdot)$.

Theorem 7.6.8 *Consider Problem* (P) *with the objective function* $f_{Pa} : 2^{\bar{\mathcal{M}}} \to \mathbb{R}_{\geq 0}$ *given by* (7.61)*. Then Algorithm* 7.6.1 *yields a solution, denoted as* $\mathcal{Y}_a^g$*, to Problem* (P) *that satisfies*

$$f_{Pa}(\mathcal{Y}_a^g) \geq \frac{\min\{\gamma_2, 1\}}{2}(1 - e^{-\gamma_1}) f_{Pa}(\mathcal{Y}_a^\star), \tag{7.68}$$

where $\mathcal{Y}_a^\star \subseteq \bar{\mathcal{M}}$ *is an optimal solution to Problem* (P)*, and* $\gamma_1 \in \mathbb{R}_{\geq 0}$ *and* $\gamma_2 \in \mathbb{R}_{\geq 0}$ *are defined in Definition* 7.6.1.

Proof Noting that (7.68) follows trivially if $\gamma_1 = 0$ or $\gamma_2 = 0$, we assume that $\gamma_1 > 0$ and $\gamma_2 > 0$. In this proof, we drop the subscript of $f_{Pa}(\cdot)$ and denote $f(\cdot)$ for notational simplicity. First, recall that for all $j \in \{1, \ldots, |\mathcal{Y}_2|\}$, we let $\mathcal{Y}_2^j = \{y_1, \ldots, y_j\}$ denote the set that contains the first j elements added to set $\mathcal{Y}_2$ in Algorithm 7.6.1, and let $\mathcal{Y}_2^0 = \emptyset$. Now, let j_l be the first index in $\{1, \ldots, |\mathcal{Y}_2|\}$ such that a candidate element $y^\star \in \arg\max_{y \in \mathcal{C}} \frac{f(\{y\} \cup \mathcal{Y}_2^{j_l}) - f(\mathcal{Y}_2^{j_l})}{c(y)}$ for $\mathcal{Y}_2$ (given in line 6 of Algorithm 7.6.1) cannot be added to $\mathcal{Y}_2$ due to $c(y^\star) + c(\mathcal{Y}_2^{j_l}) > B$. In other words, for all $j \in \{0, \ldots, j_l - 1\}$, any candidate element $y^\star \in \arg\max_{y \in \mathcal{C}} \frac{f(\{y\} \cup \mathcal{Y}_2^{j}) - f(\mathcal{Y}_2^{j})}{c(y)}$ for $\mathcal{Y}_2$ satisfies $c(y^\star) + c(\mathcal{Y}_2^j) \leq B$ and can be added to $\mathcal{Y}_2$. Considering any $j \in \{0, \ldots, j_l - 1\}$, we then have

$$f(\mathcal{Y}_a^\star \cup \mathcal{Y}_2^j) - f(\mathcal{Y}_2^j) \leq \frac{1}{\gamma_1} \sum_{y \in \mathcal{Y}_a^\star \setminus \mathcal{Y}_2^j} c(y) \cdot \frac{f(\{y\} \cup \mathcal{Y}_2^j) - f(\mathcal{Y}_2^j)}{c(y)} \tag{7.69}$$

$$\leq \frac{1}{\gamma_1} \sum_{y \in \mathcal{Y}_a^\star \setminus \mathcal{Y}_2^j} c(y) \cdot \frac{f(\mathcal{Y}_2^{j+1}) - f(\mathcal{Y}_2^j)}{c(y_{j+1})} \leq \frac{B}{\gamma_1} \cdot \frac{f(\mathcal{Y}_2^{j+1}) - f(\mathcal{Y}_2^j)}{c(y_{j+1})}, \tag{7.70}$$

where (7.69) follows from the definition of γ_1 in (7.66), and the first inequality in (7.70) follows from $\frac{f(\{y\} \cup \mathcal{Y}_2^j) - f(\mathcal{Y}_2^j)}{c(y)} \leq \frac{f(\{y_{j+1}\} \cup \mathcal{Y}_2^j) - f(\mathcal{Y}_2^j)}{c(y_{j+1})}$ $\forall y \in \mathcal{Y}_a^\star \setminus \mathcal{Y}_2^j$ by the greedy choice of Algorithm 7.6.1. To obtain the second inequality in (7.70), we use the fact $c(\mathcal{Y}_a^\star) \leq B$. Since $f(\cdot)$ is monotone nondecreasing, it then follows from (7.70) that

$$f(\mathcal{Y}_a^\star) \leq f(\mathcal{Y}_2^j) + \frac{B}{\gamma_1} \cdot \frac{f(\mathcal{Y}_2^{j+1}) - f(\mathcal{Y}_2^j)}{c(y_{j+1})}. \tag{7.71}$$

Moreover, let $y' \in \arg\max_{y \in \mathcal{C}} \frac{f(\{y\} \cup \mathcal{Y}_2^{j_l}) - f(\mathcal{Y}_2^{j_l})}{c(y)}$ be the (first) candidate element for $\mathcal{Y}_2$ that cannot be added to $\mathcal{Y}_2$ due to $c(y') + c(\mathcal{Y}_2^{j_l}) > B$, as we argued above. One can see that $\frac{f(\{y'\} \cup \mathcal{Y}_2^{j_l}) - f(\mathcal{Y}_2^{j_l})}{c(y')} \geq \frac{f(\{y\} \cup \mathcal{Y}_2^{j_l}) - f(\mathcal{Y}_2^{j_l})}{c(y)}$ also holds for for all $y \in \mathcal{Y}_a^\star \setminus \mathcal{Y}_2^{j_l}$. Letting $\bar{\mathcal{Y}}_2^{j_l+1} \triangleq \{y'\} \cup \mathcal{Y}_2^{j_l}$ and following similar arguments leading to (7.71), we have

$$f(\mathcal{Y}_a^\star) \leq f(\mathcal{Y}_2^{j_l}) + \frac{B}{\gamma_1} \cdot \frac{f(\bar{\mathcal{Y}}_2^{j_l+1}) - f(\mathcal{Y}_2^{j_l})}{c(y')}. \tag{7.72}$$

Letting $\Delta_j \triangleq f(\mathcal{Y}_a^\star) - f(\mathcal{Y}_2^j)$ for all $j \in \{0, \ldots, j_l\}$ and $\Delta_{j_l+1} \triangleq f(\mathcal{Y}_a^\star) - f(\bar{\mathcal{Y}}_2^{j_l+1})$, we obtain from (7.71) the following:

$$\begin{aligned} &\Delta_j \leq \Delta_{j-1}(1 - \frac{c(y_j)\gamma_1}{B}) \ \forall j \in \{0, \ldots, j_l + 1\}, \\ \Longrightarrow &\Delta_{j_l+1} \leq \Delta_0\big(\prod_{j=1}^{j_l}(1 - \frac{c(y_j)\gamma_1}{B})\big)(1 - \frac{c(y')\gamma_1}{B}). \end{aligned} \tag{7.73}$$

Moreover, one can show that $\big(\prod_{j=1}^{j_l}(1 - \frac{c(y_j)\gamma_1}{B})\big)(1 - \frac{c(y')\gamma_1}{B}) \leq \prod_{j=1}^{j_l+1}(1 - \frac{c(\bar{\mathcal{Y}}_2^{j_l+1})\gamma_1}{(j_l+1)B})$ (e.g., [31]). We then have from (7.73) and (7.72) the following:

$$\begin{aligned} &f(\mathcal{Y}_a^\star) - f(\bar{\mathcal{Y}}_2^{j_l+1}) \leq f(\mathcal{Y}_a^\star)(1 - \frac{c(\bar{\mathcal{Y}}_2^{j_l+1})\gamma_1}{(j_l+1)B})^{j_l+1} \leq f(\mathcal{Y}_a^\star)e^{-\gamma_1 \frac{c(\bar{\mathcal{Y}}_2^{j_l+1})}{B}} \\ \Longrightarrow &f(\bar{\mathcal{Y}}_2^{j_l+1}) \geq (1 - e^{-\gamma_1 \frac{c(\bar{\mathcal{Y}}_2^{j_l+1})}{B}})f(\mathcal{Y}_a^\star) \geq (1 - e^{-\gamma_1})f(\mathcal{Y}_a^\star), \end{aligned} \tag{7.74}$$

where the second inequality in (7.74) follows from $c(\bar{\mathcal{Y}}_2^{j_l+1}) > B$.

To proceed with the proof of the theorem, we note from the definition of γ_2 in Definition 7.6.1 that $f(\{y'\} \cup \mathcal{Y}_2^{j_l}) - f(\mathcal{Y}_2^{j_l}) \leq \frac{1}{\gamma_2} f(\mathcal{Y}_1)$ with $\gamma_2 > 0$, which together with (7.74) imply that $f(\mathcal{Y}_2^{j_l}) + \frac{1}{\gamma_2} f(\mathcal{Y}_1) \geq f(\bar{\mathcal{Y}}_2^{j_l+1}) \geq (1 - e^{\gamma_1})f(\mathcal{Y}_a^\star)$. Thus, we see that at least one of $f(\mathcal{Y}_2^{j_l}) \geq \frac{1}{2}(1 - e^{-\gamma_1})f(\mathcal{Y}_a^\star)$ and $f(\mathcal{Y}_1) \geq \frac{\gamma_2}{2}(1 - e^{-\gamma_1})f(\mathcal{Y}_a^\star)$ holds. Since $f(\mathcal{Y}_2) \geq f(\mathcal{Y}_2^{j_l})$ by the monotonicity of $f(\cdot)$ and $f(\mathcal{Y}_a^g) \geq \max\{f(\mathcal{Y}_1), f(\mathcal{Y}_2)\}$ by the definition of Algorithm 7.6.1, we obtain (7.68). ■

Remark 7.6.9 *Note that* (7.68) *becomes* $f_{Pa}(\mathcal{Y}_a^g) \geq \frac{1}{2}(1 - e^{-\gamma_1})f_{Pa}(\mathcal{Y}_a^\star)$ *if* $\gamma_2 \geq 1$. *Also note that* $\gamma_2 \geq 1$ *can hold when the objective function* $f_{Pa}(\cdot)$ *is not submodular, as we will see later in our numerical examples.*

Remark 7.6.10 *In [122], the authors also extended the analysis of Algorithm* 7.6.1 *to nonsubmodular settings, and obtained a performance guarantee for Algorithm* 7.6.1 *that depends on a submodularity ratio defined in a different manner. One can show that the submodularity ratios defined in Definition* 7.6.1 *are lower bounded by the one defined in [122], which further implies that the performance bound for Algorithm* 7.6.1 *given in Theorem* 7.6.8 *is tighter than that provided in [122].*

We see from Definition 7.6.1 that given $\mathcal{Y}_2^j$ for all $j \in \{0, \ldots, |\mathcal{Y}_2|\}$ from Algorithm 7.6.1, γ_2 can be obtained via $O(|\bar{\mathcal{M}}|^2)$ evaluations of $f_{Pa}(\cdot)$. However, finding γ_1 may require exponentially many evaluations of $f_{Pa}(\cdot)$. Thus, we provide a lower bound on γ_1 that can be computed in polynomial time (given H_y for all $y \in \bar{\mathcal{M}}$ defined in (7.62)). The lower bound on γ_1 and the value of γ_2 together with Theorem 7.6.8 will also provide performance guarantees for the greedy algorithm. We will use the following result.

Lemma 7.6.11 *([91]) For any positive semidefinite matrices $P, Q \in \mathbb{R}^{n\times n}$, $\lambda_1(P) \leq \lambda_1(P+Q) \leq \lambda_1(P) + \lambda_1(Q)$, and $\lambda_n(P+Q) \geq \lambda_n(P) + \lambda_n(Q)$.*

We then have the following result; the proof is included in Section 7.8.3.

Lemma 7.6.12 *Consider the set function $f_{Pa} : 2^{\bar{\mathcal{M}}} \to \mathbb{R}_{\geq 0}$ defined in (7.61). The type-1 greedy submodularity ratio of $f_{Pa}(\cdot)$ given by Definition 7.6.1 satisfies*

$$\gamma_1 \geq \min_{j\in\{0,\ldots,|\mathcal{Y}_2|\}} \frac{\lambda_2(F_p + H(\mathcal{Y}_2^j))\lambda_2(F_p + H(\{z_j\} \cup \mathcal{Y}_2^j))}{\lambda_1(F_p + H(\mathcal{Y}_2^j))\lambda_1(F_p + H(\{z_j\} \cup \mathcal{Y}_2^j))} \tag{7.75}$$

where $\mathcal{Y}_2^j$ contains the first j elements added to $\mathcal{Y}_2$ in Algorithm 7.6.1 $\forall j \in \{1, \ldots, |\mathcal{Y}_2|\}$ with $\mathcal{Y}_2^0 = \emptyset$, F_p is given by (7.48), $H(\mathcal{Y}) = \sum_{y\in\mathcal{Y}} H_y$ $\forall \mathcal{Y} \subseteq \bar{\mathcal{M}}$ with $H_y \succeq \mathbf{0}$ defined in (7.62), and $z_j \in \arg\min_{y\in\bar{\mathcal{M}}\setminus\mathcal{Y}_2^j} \frac{\lambda_2(F_p+H(\{y\}\cup\mathcal{Y}_2^j))}{\lambda_1(F_p+H(\{y\}\cup\mathcal{Y}_2^j))}$ $\forall j \in \{1, \ldots, |\mathcal{Y}_2|\}$.

Illustrations

Using Lemma 7.6.11, one can further obtain from (7.75) the following:

$$\gamma_1 \geq \min_{j\in\{0,\ldots,|\mathcal{Y}_2|\}} \frac{\lambda_2(F_p) + \lambda_2(H(\mathcal{Y}_2^j))}{\lambda_1(F_p) + \lambda_1(H(\mathcal{Y}_2^j))} \cdot \frac{\lambda_2(F_p) + \lambda_2(H(z_j)) + \lambda_2(H(\mathcal{Y}_2^j))}{\lambda_1(F_p) + \lambda_1(H(z_j)) + \lambda_1(H(\mathcal{Y}_2^j))}, \tag{7.76}$$

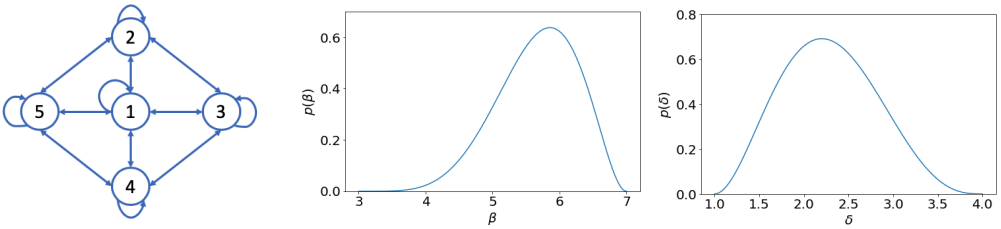
where $z_j \in \arg\min_{y\in\bar{\mathcal{M}}\setminus\mathcal{Y}_2^j} \frac{\lambda_2(F_p+H(\{y\}\cup\mathcal{Y}_2^j))}{\lambda_1(F_p+H(\{y\}\cup\mathcal{Y}_2^j))}$. Supposing F_p is fixed, we see from (7.76) that the lower bound on γ_1 would potentially increase if $\lambda_2(H(z_j))/\lambda_1(H(z_j))$ and $\lambda_2(H(\mathcal{Y}_2^j))/\lambda_1(H(\mathcal{Y}_2^j))$ increase. Recall that F_p given by (7.48) encodes the prior knowledge that we have about $\theta = [\beta\ \delta]^T$. Moreover, recall from (7.62) that $H(y)$ depends on the prior pdf $p(\theta)$ and the dynamics of the SIR model in (7.1). Therefore,

the lower bound given by Lemma 7.6.12 and thus the corresponding performance bound of Algorithm 7.6.1 given in Theorem 7.6.8 depend on the prior knowledge that we have on $\theta = [\beta \ \delta]^T$ and the dynamics of the SIR model. Also note that the performance bounds given in Theorem 7.6.8 are worst-case performance bounds for Algorithm 7.6.1. Thus, in practice the ratio between a solution returned by the algorithm and an optimal solution can be smaller than the ratio predicted by Theorem 7.6.8, as we will see in our simulations in next section. However, there may exist instances of the PEMS problem that let Algorithm 7.6.1 return a solution that meet the worst-case performance bound. Therefore, the performance bound provided in Theorem 7.6.8 indicates the worst performance bounds that Algorithm 7.6.1 can ever have when applied to any instance of the PEMS problem with objective function $f_{Pa}(\cdot)$. Moreover, instances with tighter performance bounds potentially imply better performance of the algorithm when applied to those instances. Similar arguments also hold for the performance bounds provided in Theorem 7.6.6.

Simulations

To further investigate the performance of Algorithm 7.6.1 in practice and validate the theoretical results in Theorems 7.6.6 and 7.6.8, and Lemma 7.6.12, we consider concrete PEMS instances. The directed network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is given by Fig. 7.2(a). According to the existing literature about the estimated infection and recovery rates for the COVID-19 pandemic (e.g., [123]), we assume that the infection rate β and the recovery rate δ lie in the intervals $[3, 7]$ and $[1, 4]$, respectively. Moreover, we let the prior pdf of β (resp., δ) be a (linearly transformed) Beta distribution with parameters $\alpha_1 = 6$ and $\alpha_2 = 3$ (resp., $\alpha_1 = 3$ and $\alpha_2 = 4$), where β and δ are also assumed to be independent. The prior pdfs of β and δ are then plotted in Fig. 7.2(b) and Fig. 7.2(c), respectively. The sampling parameter is set to be $h = 0.1$. We then randomly generate the weight matrix $A \in \mathbb{R}^{5\times 5}$ such that Assumptions 7.3.1-7.3.2 are satisfied, where each entry of A is drawn (independently) from certain uniform distributions. The

initial condition is set to be $s_1[0] = 0.95$, $x_1[0] = 0.05$ and $r_1[0] = 0$, and $s_i[0] = 0.99$, $x_i[0] = 0.01$ and $r_i[0] = 0$ for all $i \in \{2, \ldots, 5\}$. In the pmfs of measurements $\hat{x}_i[k]$ and $\hat{r}_i[k]$ given in Eq. (7.51) and Eq. (7.52), respectively, we set $N_i^x = N_i^r = 100$ and $N_i = 1000$ for all $i \in \mathcal{V}$.

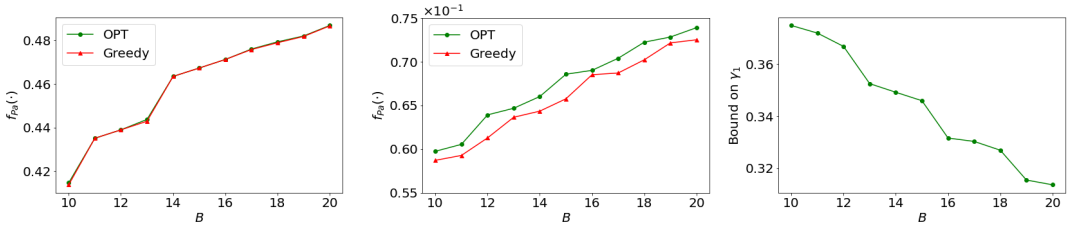


(a) Network (b) Prior pdf of β (c) Prior pdf of δ

Fig. 7.2. Network structure and prior pdfs of β and δ.

First, let us consider PEMS instances with a relatively smaller size. In such instances, we set the time steps $t_1 = t_2 = 5$, i.e., we only consider collection measurements at time step $t = 5$. In the sets $\mathcal{C}_{5,i} = \{\zeta c_{5,i} : \zeta \in (\{0\} \cup [\zeta_i])\}$ and $\mathcal{B}_{5,i} = \{\eta b_{5,i} : \eta \in (\{0\} \cup [\eta_i])\}$, we let $c_{5,i} = b_{5,i}$ and $\zeta_i = \eta_i = 2$ for all $i \in \mathcal{V}$, and draw $c_{5,i}$ and $b_{5,i}$ uniformly randomly from $\{1, 2, 3\}$. Here, we can choose to perform 0, 100, or 200 virus (or antibody) tests at a node $i \in \mathcal{V}$ and at $k = 5$. Since the set $\bar{\mathcal{M}}$ defined in Eq. (7.60) has size 20, it allows us to compare the performance of the greedy algorithm (Algorithm 7.6.1) to the optimal solution. In Fig. 7.3(a), we consider the objective function $f_{Pd}(\cdot)$, given by Eq. (7.64), in the PEMS instances constructed above, and plot the greedy solutions and the optimal solutions to the PEMS instances under different values of budget B. Note that for all the simulation results in this section, we obtain the averaged results from 50 randomly generated A matrices as described above, for each value of B. As shown in Theorem 7.6.6, the greedy algorithm yields a $\frac{1}{2}(1-e^{-1}) \approx 0.31$ approximation for $f_{Pd}(\cdot)$ (in the worst case), and the results in Fig. 7.3(a) show that the greedy algorithm performs near optimally for the PEMS instances generated above. Similarly, in Fig. 7.3(b), we plot the greedy solutions and the optimal solutions to the PEMS instances constructed above under different values of B, when the objective function is $f_{Pa}(\cdot)$ given in Eq. (7.61). Again, the results in

Fig. 7.3(b) show that the greedy algorithm performs well for the constructed PEMS instances. Moreover, according to Lemma 7.6.12, we plot the lower bound on the submodularity ratio γ_1 of $f_{Pa}(\cdot)$ in Fig. 7.3(c). Here, we note that the submodularity ratio γ_2 of $f_{Pa}(\cdot)$ is always greater than one in the PEMS instances constructed above. Hence, Theorem 7.6.8 yields a $\frac{1}{2}(1-e^{-\gamma_1})$ worst-case approximation guarantee for the greedy algorithm, where we note that $\frac{1}{2}(1-e^{-0.3}) \approx 0.13$.



(a) OPT vs. Greedy for $f_{Pd}(\cdot)$ (b) OPT vs. Greedy for $f_{Pa}(\cdot)$ (c) Bound on γ_1

Fig. 7.3. Results for PEMS instances of medium size.

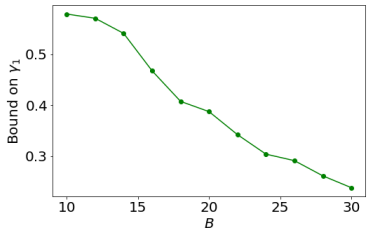


Fig. 7.4. Bound on γ_1 for PEMS instances of large size.

We then investigate the performance of the greedy algorithm for PEMS instances of a larger size. Since the optimal solution to the PEMS instances cannot be efficiently obtained when the sizes of the instances become large, we only obtain the lower bound on the submodularity ratio γ_1 of $f_{Pa}(\cdot)$ provided in Lemma 7.6.12, which can be computed in polynomial time. Different from the smaller instances constructed above, we set $t_1 = 1$ and $t_2 = 5$. We let $\zeta_i = \eta_i = 10$ for all $i \in \mathcal{V}$ in $\mathcal{C}_{k,i} = \{\zeta c_{k,i} : \zeta \in (\{0\} \cup [\zeta_i])\}$ and $\mathcal{B}_{k,i} = \{\eta b_{k,i} : \eta \in (\{0\} \cup [\eta_i])\}$, where we also set $c_{k,i} = b_{k,i}$ and draw $c_{k,i}$ and $b_{k,i}$ uniformly randomly from $\{1, 2, 3\}$, for all $k \in [5]$ and for all

$i \in \mathcal{V}$. Note that the size of the set $\bar{\mathcal{M}}$ defined in Eq. (7.60) is equal to 500 in these instances of the PEMS problem. Moreover, we modify the parameter of the Beta distribution corresponding to the pdf of β to be $\alpha_1 = 8$ and $\alpha_2 = 3$. The other constructions remain the same as the smaller PEMS instances described above. Here, we can choose to perform 0, 100, 200, ..., or 1000 virus (or antibody) tests at node $i \in \mathcal{V}$ and at $k \in [5]$. In Fig. 7.4, we plot the lower bound on γ_1 obtained from the PEMS instances constructed above. We note that the submodularity ratio γ_2 of $f_{Pa}(\cdot)$ is also always greater than one.

Overall, we see that the greedy algorithm for the PEMS problem performs well on the randomly generated instances. Moreover, the lower bounds on γ_1 plotted in Fig. 7.3(c) and Fig. 7.4 show that Lemma 7.6.12 together with Theorem 7.6.8 yield reasonably tight worst-case performance guarantees for the greedy algorithm when applied to PEMS instances with the objective function $f_{Pa}(\cdot)$.

7.7 Chapter Summary

In this chapter, we first considered the PIMS problem under the exact measurement setting, and showed that the problem is NP-hard. We then proposed an approximation algorithm that returns a solution to the PIMS problem that is within a certain factor of the optimal one. Next, we studied the PEMS problem under the noisy measurement setting. Again, we showed that the problem is NP-hard. We applied a greedy algorithm to solve the PEMS problem, and provided performance guarantees on the greedy algorithm. We presented numerical examples to validate the obtained performance bounds of the greedy algorithm, and showed that the greedy algorithm performs well in practice.

7.8 Proofs of Key Results

7.8.1 Proof of Lemma 7.3.3

We first prove part (a). Considering any $i \in \mathcal{V}$ and any $k \in \mathbb{Z}_{\geq 0}$, we note from Eq. (7.1a) that

$$s_i[k+1] = s_i[k](1 - h\beta \sum_{j \in \bar{\mathcal{N}}_i} a_{ij} x_j[k]). \tag{7.77}$$

Under Assumptions 7.3.1-7.3.2, we have $x_i[k] \in [0,1]$ for all $i \in \mathcal{V}$ as argued above, and $h\beta \sum_{j \in \bar{\mathcal{N}}_i} a_{ij} < 1$ for all $i \in \mathcal{V}$, which implies $1 - h\beta \sum_{j \in \bar{\mathcal{N}}_i} a_{ij} x_j[k] \geq 1 - h\beta \sum_{j \in \bar{\mathcal{N}}_i} a_{ij} > 0$. Supposing $s_i[k] > 0$, we have from Eq. (7.77) $s_i[k+1] > 0$. Combining the above arguments with the fact $s_i[0] \in (0,1]$ from Assumption 7.3.1, we see that $s_i[k] > 0$ for all $k \in \mathbb{Z}_{\geq 0}$. Noting that $s_i[k], x_i[k], r_i[k] \in [0,1]$ with $s_i[k] + x_i[k] + r_i[k] = 1$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$ as argued above and that $x_i[0] \in [0,1)$ and $r_i[0] = 0$ for all $i \in \mathcal{V}$, the result in part (a) also implies $x_i[k], r_i[k] \in [0,1)$ for all $i \in \mathcal{V}$ and for all $k \in \mathbb{Z}_{\geq 0}$.

One can then observe that in order to prove parts (b)-(d), it is sufficient to prove the following facts.

Fact 7.8.1 *Consider any $i \in \mathcal{V}$ and any $k_1 \in \mathbb{Z}_{\geq 0}$. If $x_i[k_1] > 0$, then $x_i[k_2] > 0$ for all $k_2 \in \mathbb{Z}_{\geq 0}$ with $k_2 \geq k_1$.*

Fact 7.8.2 *Consider any $i \in \mathcal{V}$ and any $k \in \mathbb{Z}_{\geq 0}$ such that $x_i[k] = 0$. If there exists $j \in \mathcal{N}_i$ such that $x_j[k] > 0$, then $x_i[k+1] > 0$. If $x_j[k] = 0$ for all $j \in \mathcal{N}_i$, then $x_i[k+1] = 0$.*

Fact 7.8.3 *Consider any $i \in \mathcal{V}$ and any $k_1 \in \mathbb{Z}_{\geq 0}$. If $x_i[k_1] > 0$, then $r_i[k_1+1] > 0$. If $x_i[k_1] = 0$, then $r_i[k_1+1] = 0$.*

Let us first prove Fact 7.8.1. Consider any $i \in \mathcal{V}$ and any $k \in \mathbb{Z}_{\geq 0}$. Supposing $x_i[k] > 0$, we have from Eq. (7.1)

$$x_i[k+1] = (1 - h\delta) x_i[k] + h s_i[k] \beta \sum_{j \in \mathcal{N}_i} a_{ij} x_j[k], \tag{7.78}$$

where the first term on the right-hand side of the above equation is positive, since $1 - h\delta > 0$ from Assumption 7.3.2, and the second term on the right-hand side of the above equation is nonnegative. It then follows that $x_i[k+1] > 0$. Repeating the above argument proves Fact 7.8.1.

We next prove Fact 7.8.2. Considering any $i \in \mathcal{V}$ and any $k \in \mathbb{Z}_{\geq 0}$ such that $x_i[k] = 0$, we note from Eq. (7.1) that

$$x_i[k+1] = hs_i[k]\beta \sum_{j \in \mathcal{N}_i} a_{ij} x_j[k], \tag{7.79}$$

where $s_i[k] > 0$ as shown in part (a). Suppose there exists $j \in \mathcal{N}_i$ such that $x_j[k] > 0$. Since $h, \beta \in \mathbb{R}_{>0}$ and $a_{ij} > 0$ for all $j \in \mathcal{N}_i$ from Assumption 7.3.2, we have from Eq. (7.79) $x_i[k+1] > 0$. Next, supposing $x_j[k] = 0$ for all $j \in \mathcal{N}_i$, we obtain from Eq. (7.79) $x_i[k+1] = 0$. This proves Fact 7.8.2.

Finally, we prove Fact 7.8.3. Let us consider any $i \in \mathcal{V}$ and any $k_1 \in \mathbb{Z}_{\geq 0}$. Suppose $x_i[k_1] > 0$. Since $h, \delta \in \mathbb{R}_{>0}$ from Assumption 7.3.2, we have from Eq. (7.1c) $r_i[k_1+1] = r_i[k] + h\delta x_i[k_1] > 0$. Next, supposing $x_i[k_1] = 0$, we note from Fact 7.8.1 that $x_i[k_1'] = 0$ for all $k_1' \leq k_1$. It then follows from Eq. (7.1c) and Assumption 7.3.1 that $r_i[k_1+1] = r_i[k_1] = \cdots = r_i[0] = 0$, completing the proof of Fact 7.8.3. ■

7.8.2 Proof of Lemma 7.5.5

Noting from (7.11), we have

$$\begin{bmatrix} \Phi^x_{k_1,i_1} \\ \Phi^r_{k_2,i_2} \end{bmatrix} = \begin{bmatrix} s_{i_1}[k_1] \sum_{j \in \bar{\mathcal{N}}_{i_1}} a_{i_1 j} x_j[k_1] & -x_{i_1}[k_1] \\ 0 & x_{i_2}[k_2] \end{bmatrix}. \tag{7.80}$$

To prove part (a), consider any $i_1 \in \mathcal{S}'_I$ and any $i_2 \in \mathcal{V}$ with $d_{i_2} \neq \infty$, where we note $x_{i_1}[0] > 0$ and $a_{i_1 i_1} > 0$ from the definition of $\mathcal{S}'_I$. We then see from Lemma 7.3.3(a)-(b) that $s_{i_1}[k_1] > 0$ and $x_{i_1}[k_1] > 0$ for all $k_1 \geq 0$. It follows that $s_{i_1}[k_1] \sum_{j \in \bar{\mathcal{N}}_{i_1}} a_{i_1 j} x_j[k_1] > 0$ for all $k_1 \geq 0$. Also, we obtain from Lemma 7.3.3(b) $x_{i_2}[k_2] > 0$ for all $k_2 \geq d_{i_2}$. This proves part (a).

We then prove part (b). Considering any $i_1 \in \mathcal{S}'$ and any $i_2 \in \mathcal{V}$ with $d_2 \neq \infty$, we see from the definition of $\mathcal{S}'$ that $\mathcal{N}_{i_1} \neq \emptyset$ and there exists $j \in \mathcal{N}_{i_1}$ such that

$d_j \neq \infty$. Letting j_1 be a node in $\mathcal{N}_{i_1}$ such that $d_{j_1} = \min\{d_j : j \in \mathcal{N}_{i_1}\} \neq \infty$, we note from Lemma 7.3.3(a) that $x_{j_1}[k_1] > 0$ for all $k_1 \geq \min\{d_j : j \in \mathcal{N}_{i_1}\}$. Also note that $a_{i_1 j_1} > 0$ from Assumption 7.3.2. The rest of the proof of part (b) is then identical to that of part (a). ■

7.8.3 Proof of Lemma 7.6.12

Noting the definition of γ_1 in Definition 7.6.1, we will provide lower bound on $\frac{\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j}(f_{Pa}(\{y\}\cup\mathcal{Y}_2^j)-f_{Pa}(\mathcal{Y}_2^j))}{f_{Pa}(\mathcal{A}\cup\mathcal{Y}_2^j)-f_{Pa}(\mathcal{Y}_2^j)}$ for all $\mathcal{A} \subseteq \bar{\mathcal{M}}$ and for all $\mathcal{Y}_2^j$, where we will assume that $\mathcal{A} \setminus \mathcal{Y}_2^j \neq \emptyset$, otherwise (7.66) would be satisfied for all $\gamma_1 \in \mathbb{R}$. We begin by lower bounding $\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j} \left(f_{Pa}(\{y\}\cup\mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j)\right)$ in the following:

$$
\begin{aligned}
&\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j} (f_{Pa}(\{y\}\cup\mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j)) \\
=&\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j} \left(\operatorname{tr}\left((F_p + H(\mathcal{Y}_2^j))^{-1}\right) - \operatorname{tr}\left((F_p + H(\{y\}\cup\mathcal{Y}_2^j))^{-1}\right)\right) \\
=&\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j} \sum_{i=1}^{2} \left(\frac{1}{\lambda_i(F_p + H(\mathcal{Y}_2^j))} - \frac{1}{\lambda_i(F_p + H(\{y\}\cup\mathcal{Y}_2^j))}\right) \\
=&\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j} \sum_{i=1}^{2} \frac{\lambda_i(F_p + H(\{y\}\cup\mathcal{Y}_2^j)) - \lambda_i(F_p + H(\mathcal{Y}_2^j))}{\lambda_i(F_p + H(\mathcal{Y}_2^j))\lambda_i(F_p + H(\{y\}\cup\mathcal{Y}_2^j))} \\
\geq&\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j} \frac{\sum_{i=1}^{2}(\lambda_i(F_p + H(\{y\}\cup\mathcal{Y}_2^j)) - \lambda_i(F_p + H(\mathcal{Y}_2^j)))}{\lambda_1(F_p + H(\mathcal{Y}_2^j))\lambda_1(F_p + H(\{z'\}\cup\mathcal{Y}_2^j))} \qquad (7.81) \\
=&\frac{\sum_{y\in\mathcal{A}\setminus\mathcal{Y}_2^j} \operatorname{tr}(H_y)}{\lambda_1(F_p + H(\mathcal{Y}_2^j))\lambda_1(F_p + H(\{z'\}\cup\mathcal{Y}_2^j))}. \qquad (7.82)
\end{aligned}
$$

To obtain (7.81), we let $z' \in \arg\max_{y \in \mathcal{A} \setminus \mathcal{Y}_2^j} \lambda_1(F_p + H(\{y\} \cup \mathcal{Y}_2^j))$ and note that $\lambda_1(F_p + H(\{z'\} \cup \mathcal{Y}_2^j)) \geq \lambda_i(F_p + H(\{y\} \cup \mathcal{Y}_2^j))$ for all $i \in \{1, 2\}$ and for all $y \in \mathcal{A} \setminus \mathcal{Y}_2^j$. Next, we upper bound $f_{Pa}(\mathcal{A} \cup \mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j)$ in the following:

$$\begin{aligned}
f_{Pa}(\mathcal{A} \cup \mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j) &= \operatorname{tr}\left((F_p + H(\mathcal{Y}_2^j))^{-1}\right) - \operatorname{tr}\left((F_p + H(\mathcal{A} \cup \mathcal{Y}_2^j))^{-1}\right) \\
&= \sum_{i=1}^{2} \Big(\frac{1}{\lambda_i(F_p + H(\mathcal{Y}_2^j))} - \frac{1}{\lambda_i(F_p + H(\mathcal{A} \cup \mathcal{Y}_2^j))} \Big) \\
&= \sum_{i=1}^{2} \frac{\lambda_i(F_p + H(\mathcal{A} \cup \mathcal{Y}_2^j)) - \lambda_i(F_p + H(\mathcal{Y}_2^j))}{\lambda_i(F_p + H(\mathcal{Y}_2^j))\lambda_i(F_p + H(\mathcal{A} \cup \mathcal{Y}_2^j))} \\
&\leq \frac{\sum_{i=1}^{2} \left(\lambda_i(F_p + H(\mathcal{A} \cup \mathcal{Y}_2^j)) - \lambda_i(F_p + H(\mathcal{Y}_2^j)\right)}{\lambda_2(F_p + H(\mathcal{Y}_2^j))\lambda_2(F_p + H(\{z'\} \cup \mathcal{Y}_2^j))} \qquad (7.83) \\
&= \frac{\sum_{y \in \mathcal{A} \setminus \mathcal{Y}_2^j} \operatorname{tr}(H_y)}{\lambda_2(F_p + H(\mathcal{Y}_2^j))\lambda_2(F_p + H(\{z'\} \cup \mathcal{Y}_2^j))}. \qquad (7.84)
\end{aligned}$$

To obtain (7.83), we note that $\lambda_i(F_p + H(\mathcal{A} \cup \mathcal{Y}_2^j)) \geq \lambda_2(F_p + H(\mathcal{A} \cup \mathcal{Y}_2^j)) \geq \lambda_2(F_p + \{z'\} \cup \mathcal{Y}_2^j)$ for all $i \in \{1, 2\}$, where the second inequality follows from Lemma 7.6.11 with the fact $H(\mathcal{A} \cup \mathcal{Y}_2^j) - H(\{z'\} \cup \mathcal{Y}_2^j) \succeq \mathbf{0}$, and z' is defined above. Combining (7.82) and (7.84), we have

$$\begin{aligned}
\frac{\sum_{y \in \mathcal{A} \setminus \mathcal{Y}_2^j} (f_{Pa}(\{y\} \cup \mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j))}{f_{Pa}(\mathcal{A} \cup \mathcal{Y}_2^j) - f_{Pa}(\mathcal{Y}_2^j)} &\geq \frac{\lambda_2(F_p + H(\mathcal{Y}_2^j))\lambda_2(F_p + H(\{z'\} \cup \mathcal{Y}_2^j))}{\lambda_1(F_p + H(\mathcal{Y}_2^j))\lambda_1(F_p + H(\{z'\} \cup \mathcal{Y}_2^j))} \\
&\geq \frac{\lambda_2(F_p + H(\mathcal{Y}_2^j))\lambda_2(F_p + H(\{z_j\} \cup \mathcal{Y}_2^j))}{\lambda_1(F_p + H(\mathcal{Y}_2^j))\lambda_1(F_p + H(\{z_j\} \cup \mathcal{Y}_2^j))}, \qquad (7.85)
\end{aligned}$$

where $z_j \in \arg\min_{y \in \bar{\mathcal{M}} \setminus \mathcal{Y}_2^j} \frac{\lambda_2(F_p + H(\{y\} \cup \mathcal{Y}_2^j))}{\lambda_1(F_p + H(\{y\} \cup \mathcal{Y}_2^j))}$. Since (7.85) holds for all $\mathcal{Y}_2^j$ with $j \in \{0, \ldots, |\mathcal{Y}_2|\}$ and for all $\mathcal{A} \subseteq \bar{\mathcal{M}}$, we obtain (7.75). ■

8. SUMMARY AND FUTURE WORK

8.1 Summary

In this thesis, we studied the sensor selection problem in large-scale systems using algorithmic and graph-theoretic approaches, under a variety of settings. We summarize our main results below.

1. In Chapter 3, we considered the sensor selection and attack problems for Kalman filtering. We showed that the sensor selection and attack problems for Kalman filtering are NP-hard and cannot be approximated within any constant factor in polynomial time for general systems.

2. In Chapter 4, we studied a class of the sensor selection and attack problems for Kalman filtering for networked systems where there is a single node in the network that has a stochastic input. We showed that polynomial-time algorithms exist for this class of the sensor selection and attack problems for Kalman filtering, respectively. We further showed that the resilient sensor selection problem for Kalman filtering under the networked system setting is NP-hard, but admits a pseudo-polynomial-time algorithm.

3. In Chapter 5, we considered sensor selection problems for hypothesis testing in signal detection based on the Neyman-Pearson detector and Bayesian detector. We showed that the sensor selection problem for the Neyman-Pearson (resp., Bayesian) detector is NP-hard when we considered the miss probability of the Neyman-Pearson detector (resp., error probability of the Bayesian detector) as the optimization objective (in the hypothesis testing sensor selection problem). While considering optimization metrics based on the Kullback-Leibler distance,

J-Divergence, and Bhattacharyya distance, respectively, we provided theoretical performance guarantees on greedy algorithms when applied to this problem.

4. In Chapter 6, we considered the data source selection problem for Bayesian learning. We showed that the data source selection problem is NP-hard, and provided a standard greedy algorithm to solve it with performance guarantees. We further proposed a fast greedy algorithm to solve the problem that improves the running times of the standard greedy algorithm, and achieves performance guarantees that are comparable to those of the standard greedy algorithm.

5. In Chapter 7, we studied the measurement selection problem for parameter estimation in epidemic spread networks. We considered settings with exact measurements and settings with stochastic measurements. We showed that the measurement selection problems under these two settings are NP-hard. We then provided approximation algorithms to solve the problems with performance guarantees.

8.2 Future Work

Let us now outline some ongoing and future work related to the problems considered in this thesis.

1. Similarly to the sensor attack and resilient sensor selection problems for Kalman filtering that we studied in Chapter 3 and Chapter 4, one can extend our analysis in Chapter 6, and formulate the data source attack problem and the resilient data source selection problem for Bayesian learning.

2. The discussions in this work only consider solving the sensor selection problem in a centralized manner, where a single system designer is involved in the sensor selection task. This requires the system designer to have complete knowledge of the system and communicate with each (selected) sensor directly, which is unrealistic in large-scale systems or hostile environments. Therefore, one can

aim to extend our analysis to the case when there are multiple designers who can select sensors on a targeted system simultaneously. This corresponds to a scenario where multiple agents want to build a sensor network together to monitor (or estimate) states of a system of common interest. Moreover, each designer has her own utility function in terms of estimation performance and a sensor selection budget. Due to the nature of this setting where multiple designers are making decisions in a potentially selfish and conflict manner, one can apply techniques from game theory to analyze this scenario, e.g., characterizing the existence of a Nash equilibrium and the efficiency of it. This scenario is related to a sensor coverage game as described in [124] and [125], where the goal of each system designer is to allocate sensors across a given mission space such that the probability of detecting a particular event is maximized.

3. In the problem formulations in this thesis, we did not consider the communication issue in gathering the measurements from the sensors. In practice, the sensors may be distributed in the environment, and gathering the sensor measurements may require communications between the sensors and a fusion center using communication channels. Under this scenario, communication losses (i.e., drops) need to be taken into consideration. For instance, the Kalman filter with intermittent measurements has been studied in literature (e.g., [126]). Therefore, when considering the sensor selection problem, it would be of practical interest to consider communication losses when gathering the measurements from remote sensors. Along with this direction, one can also consider communication constraints when collecting the measurements from the selected sensors, which serve as additional constraints in the sensor selection problem formulation. For instance, one can have a constraint on the number of bits that can be used when the remote sensors send their measurements to the fusion center (e.g., [127]).

REFERENCES

REFERENCES

[1] Y. Mo, R. Ambrosino, and B. Sinopoli, "Sensor selection strategies for state estimation in energy constrained wireless sensor networks," *Automatica*, vol. 47, no. 7, pp. 1330–1338, 2011.

[2] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202–210, 2017.

[3] D. Bajovic, B. Sinopoli, and J. Xavier, "Sensor selection for event detection in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4938–4953, 2011.

[4] S. P. Chepuri and G. Leus, "Sparse sensing for distributed detection," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1446–1460, 2016.

[5] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[6] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2618–2624, 2016.

[7] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *computer*, vol. 35, no. 10, pp. 54–62, 2002.

[8] A. Laszka, Y. Vorobeychik, and X. Koutsoukos, "Resilient observation selection in adversarial settings," in *Proc. IEEE Confernece on Decision and Control*, 2015, pp. 7416–7421.

[9] V. Tzoumas, K. Gatsis, A. Jadbabaie, and G. J. Pappas, "Resilient monotone submodular function maximization," in *Proc. IEEE Conference on Decision and Control*, 2018, pp. 1362–1367.

[10] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo, "An ADMM algorithm for optimal sensor and actuator selection," in *Proc. IEEE Conference on Decision and Control*, 2014, pp. 4039–4044.

[11] C. Yang, J. Wu, X. Ren, W. Yang, H. Shi, and L. Shi, "Deterministic sensor selection for centralized state estimation under limited communication resource," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2336–2348, 2015.

[12] L. F. Chamon, G. J. Pappas, and A. Ribeiro, "The mean square error in Kalman filtering sensor selection is approximately supermodular," in *Proc. IEEE Conference on Decision and Control*, 2017, pp. 343–350.

[13] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.

[14] F. Lin, M. Fardad, and M. R. Jovanović, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1789–1802, 2014.

[15] A. Clark, L. Bushnell, and R. Poovendran, "A supermodular optimization framework for leader selection under link noise in linear multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 283–296, 2014.

[16] K. Fitch and N. E. Leonard, "Joint centrality distinguishes optimal leaders in noisy networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 4, pp. 366–378, 2016.

[17] M. Pirani, E. M. Shahrivar, B. Fidan, and S. Sundaram, "Robustness of leader-follower networked dynamical systems," *IEEE Transactions on Control of Network Systems*, 2017.

[18] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. International Conference on Knowledge Discovery and Data mining*, 2003, pp. 137–146.

[19] A. V. Goltsev, S. N. Dorogovtsev, J. G. Oliveira, and J. F. Mendes, "Localization and spreading of diseases in complex networks," *Physical Review Letters*, vol. 109, no. 12, p. 128702, 2012.

[20] S. Roy and R. Dhal, "Situational awareness for dynamical network processes using incidental measurements," *IEEE Journal of Selected topics in Signal Processing*, vol. 9, no. 2, pp. 304–316, 2015.

[21] M. Coutino, S. P. Chepuri, and G. Leus, "Submodular sparse sensing for Gaussian detection with correlated observations," *IEEE Transactions on Signal Processing*, vol. 66, no. 15, pp. 4025–4039, 2018.

[22] H. V. Poor, *An introduction to signal detection and estimation*. Springer Science & Business Media, 2013.

[23] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statistical Science*, pp. 273–304, 1995.

[24] I. Kononenko, "Inductive and Bayesian learning in medical diagnosis," *Applied Artificial Intelligence an International Journal*, vol. 7, no. 4, pp. 317–337, 1993.

[25] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[26] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Reviews of Modern Physics*, vol. 87, no. 3, p. 925, 2015.

[27] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, "Epidemic thresholds in real networks," *ACM Transactions on Information and System Security*, vol. 10, no. 4, pp. 1–26, 2008.

[28] H. J. Ahn and B. Hassibi, "Global dynamics of epidemic spread over complex networks," in *Proc. IEEE Conference on Decision and Control*, 2013, pp. 4579–4585.

[29] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. J. Pappas, "Optimal resource allocation for network protection against spreading processes," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 99–108, 2014.

[30] M. E. Newman, "Spread of epidemic disease on networks," *Physical Review E*, vol. 66, no. 1, p. 016128, 2002.

[31] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[32] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[33] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.

[34] ——, "On (non) supermodularity of average control energy," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1177–1181, 2018.

[35] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On efficient sensor scheduling for linear dynamical systems," *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.

[36] M. F. Huber, "Optimal pruning for multi-step sensor scheduling," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1338–1343, 2012.

[37] S. T. Jawaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.

[38] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[39] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proc. ACM symposium on Theory of computing*, 2008, pp. 45–54.

[40] ——, "Approximate submodularity and its applications: subset selection, sparse approximation and dictionary selection," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 74–107, 2018.

[41] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-Completeness*. Freeman, 1979.

[42] M. Van De Wal and B. De Jager, "A review of methods for input/output selection," *Automatica*, vol. 37, no. 4, pp. 487–510, 2001.

[43] X. Liu, B. M. Chen, and Z. Lin, "On the problem of general structural assignments of linear systems through sensor/actuator selection," *Automatica*, vol. 39, no. 2, pp. 233–241, 2003.

[44] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[45] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 67–78, 2016.

[46] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, 2015.

[47] F. Lin, M. Fardad, and M. R. Jovanović, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, 2013.

[48] T.-S. Yoo and S. Lafortune, "NP-completeness of sensor selection problems arising in partially observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1495–1499, 2002.

[49] L. Ye, S. Roy, and S. Sundaram, "Optimal sensor placement for Kalman filtering in stochastically forced consensus network," in *Proc. IEEE Conference on Decision and Control*, 2018, pp. 6686–6691.

[50] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *Proc. IEEE Conference on Decision and Control*, 2010, pp. 2572–2577.

[51] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions – I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[52] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, p. 13, 2011.

[53] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 717–729, 2014.

[54] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, 2006.

[55] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal DoS attack scheduling in wireless networked control system," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 843–852, 2016.

[56] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a denial of service attack on TCP," in *Proc. IEEE Symposium on Security and Privacy*, 1997, pp. 208–223.

[57] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms," in *Proc. American Control Conference*, 2016, pp. 191–196.

[58] J. Lu and R. Niu, "Sparse attacking strategies in multi-sensor dynamic systems maximizing state estimation errors," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 3151–3155.

[59] L. Ye, S. Roy, and S. Sundaram, "On the complexity and approximability of optimal sensor selection for Kalman filtering," in *Proc. American Control Conference*, 2018, pp. 5049–5054.

[60] L. Ye, N. Woodford, S. Roy, and S. Sundaram, "On the complexity and approximability of optimal sensor selection and attack for Kalman filtering," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020, doi: 10.1109/TAC.2020.3007383.

[61] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Dover Books, 1979.

[62] D. E. Catlin, *Estimation, control, and the discrete Kalman filter*. Springer, 1989.

[63] B. D. Anderson and J. B. Moore, *Optimal control: Linear quadratic methods*. Courier Corporation, 2007.

[64] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[65] R. Horn and C. Johnson, *Matrix analysis*. Cambridge University Press, 1985.

[66] S. Liu, S. P. Chepuri, M. Fardad, E. Maşazade, G. Leus, and P. K. Varshney, "Sensor selection for estimation with correlated measurement noise," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3509–3522, 2016.

[67] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2008.

[68] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[69] S. Roy, M. Xue, and S. Sundaram, "Graph-theoretic analysis of estimators for stochastically-driven diffusive network processes," in *Proc. American Control Conference*, 2018, pp. 1796–1801.

[70] L. Ye, S. Roy, and S. Sundaram, "Resilient sensor placement for Kalman filtering in networked systems: Complexity and algorithms," *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2020, doi: 10.1109/TCNS.2020.3006271.

[71] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[72] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2012.

[73] C.-T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[74] J.-M. Dion, C. Commault, and J. Van Der Woude, "Generic properties and control of linear structured systems: A survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[75] J. Van der Woude, "A graph-theoretic characterization for the rank of the transfer matrix of a structured system," *Mathematics of Control, Signals and Systems*, vol. 4, no. 1, pp. 33–40, 1991.

[76] J. van der Woude, "The generic number of invariant zeros of a structured linear system," *SIAM Journal on Control and Optimization*, vol. 38, no. 1, pp. 1–21, 1999.

[77] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[78] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer Science & Business Media, 2004.

[79] S. M. Kay, *Fundamentals of statistical signal processing: Estimation theory*. Prentice Hall PTR, 1993.

[80] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio: State-of-the-art and recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 101–116, 2012.

[81] C.-T. Yu and P. K. Varshney, "Sampling design for gaussian detection problems," *IEEE Transactions on Signal Processing*, vol. 45, no. 9, pp. 2328–2337, 1997.

[82] R. K. Bahr and J. A. Bucklew, "Optimal sampling schemes for the gaussian hypothesis testing problem," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1677–1686, 1990.

[83] A. Krause and V. Cevher, "Submodular dictionary selection for sparse representation," in *Proc. International Conference on Machine Learning*, 2010, pp. 567–574.

[84] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *Proc. International Conference on Machine Learning*, 2017, pp. 498–507.

[85] M. Streeter and D. Golovin, "An online algorithm for maximizing submodular functions," in *Advances in Neural Information Processing Systems*, 2009, pp. 1577–1584.

[86] L. Ye and S. Sundaram, "Sensor selection for hypothesis testing: Complexity and greedy algorithms," in *Proc. IEEE Conference on Decision and Control*, 2019, pp. 7844–7849.

[87] H. V. Poor, *An introduction to signal detection and estimation*. Springer, 1994.

[88] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, 1967.

[89] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[90] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[91] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.

[92] S.-D. Wang, T.-S. Kuo, and C.-F. Hsu, "Trace bounds on the solution of the algebraic matrix riccati and lyapunov equation," *IEEE Transactions on Automatic Control*, vol. 31, no. 7, pp. 654–656, 1986.

[93] S. Dasgupta, "Analysis of a greedy active learning strategy," in *Advances in Neural Information Processing Systems*, 2005, pp. 337–344.

[94] D. Golovin, A. Krause, and D. Ray, "Near-optimal Bayesian active learning with noisy observations," in *Advances in Neural Information Processing Systems*, 2010, pp. 766–774.

[95] J. Palmer, B. D. Rao, and D. P. Wipf, "Perspectives on sparse Bayesian learning," in *Advances in Neural Information Processing Systems*, 2004, pp. 249–256.

[96] L. Ye, A. Mitra, and S. Sundaram, "Near-optimal data source selection for Bayesian learning," *arXiv preprint arXiv:2011.10712*, 2020.

[97] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-Bayesian social learning," *Games and Economic Behavior*, vol. 76, no. 1, pp. 210–225, 2012.

[98] Q. Liu, A. Fang, L. Wang, and X. Wang, "Social learning with time-varying weights," *Journal of Systems Science and Complexity*, vol. 27, no. 3, pp. 581–593, 2014.

[99] A. Lalitha, A. Sarwate, and T. Javidi, "Social learning and distributed hypothesis testing," in *Proc. IEEE International Symposium on Information Theory*, 2014, pp. 551–555.

[100] A. Nedić, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-Bayesian learning," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5538–5553, 2017.

[101] A. Mitra, J. A. Richards, and S. Sundaram, "A new approach to distributed hypothesis testing and non-Bayesian learning: Improved learning rate and Byzantine-resilience," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020, doi: 10.1109/TAC.2020.3033126.

[102] A. Jadbabaie, P. Molavi, and A. Tahbaz-Salehi, "Information heterogeneity and the speed of learning in social networks," *Columbia Business School Research Paper*, no. 13-28, 2013.

[103] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Science & Business Media, 2013, vol. 31.

[104] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[105] A. Badanidiyuru and J. Vondrák, "Fast algorithms for maximizing submodular functions," in *Proc. ACM-SIAM Symposium on Discrete algorithms*, 2014, pp. 1497–1514.

[106] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Control Systems Magazine*, vol. 36, no. 1, pp. 26–46, 2016.

[107] W. Mei, S. Mohagheghi, S. Zampieri, and F. Bullo, "On the dynamics of deterministic epidemic propagation over networks," *Annual Reviews in Control*, vol. 44, pp. 116–128, 2017.

[108] P. E. Paré, J. Liu, C. L. Beck, B. E. Kirwan, and T. Başar, "Analysis, estimation, and validation of discrete-time epidemic processes," *IEEE Transactions on Control Systems Technology*, 2018.

[109] D. Vrabac, P. E. Paré, H. Sandberg, and K. H. Johansson, "Overcoming challenges for estimating virus spread dynamics from data," in *Proc. IEEE Conference on Information Sciences and Systems*, 2020, pp. 1–6.

[110] M. Pezzutto, N. B. Rossello, L. Schenato, and E. Garone, "Smart testing and selective quarantine for the control of epidemics," *arXiv preprint arXiv:2007.15412*, 2020.

[111] A. R. Hota, J. Godbole, P. Bhariya, and P. E. Paré, "A closed-loop framework for inference, prediction and control of SIR epidemics on networks," *arXiv preprint arXiv:2006.16185*, 2020.

[112] "Centers for Disease Control and Prevention Test for Current Infection," https://www.cdc.gov/coronavirus/2019-ncov/testing/diagnostic-testing.html, 2020.

[113] C. F. Manski and F. Molinari, "Estimating the COVID-19 infection rate: Anatomy of an inference problem," *Journal of Econometrics*, 2020.

[114] J. Watson, P. F. Whiting, and J. E. Brush, "Interpreting a COVID-19 test result," *BMJ*, vol. 369, 2020.

[115] "Mayo Clinic How do COVID-19 antibody tests differ from diagnostic tests?" https://www.mayoclinic.org/diseases-conditions/coronavirus/expert-answers/covid-antibody-tests/faq-20484429, 2020.

[116] E. Bendavid, B. Mulaney, N. Sood, S. Shah, E. Ling, R. Bromley-Dulfano, C. Lai, Z. Weissberg, R. Saavedra, J. Tedrow *et al.*, "COVID-19 antibody seroprevalence in Santa Clara County, California," *MedRxiv*, 2020.

[117] H. L. Van Trees, *Detection, estimation, and modulation theory, part IV: Optimum array processing*. John Wiley & Sons, 2004.

[118] ——, *Detection, estimation, and modulation theory, part I*. John Wiley & Sons, 2004.

[119] F. Pukelsheim, *Optimal design of experiments*. SIAM, 2006.

[120] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions*. John Wiley & Sons, Ltd, 1995.

[121] A. Krause and C. Guestrin, *A note on the budgeted maximization of submodular functions*. Carnegie Mellon University. Center for Automated Learning and Discovery, 2005.

[122] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "LQG control and sensing co-design," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[123] K. Prem, Y. Liu, T. W. Russell, A. J. Kucharski, R. M. Eggo, N. Davies, S. Flasche, S. Clifford, C. A. Pearson, J. D. Munday *et al.*, "The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: A modelling study," *The Lancet Public Health*, 2020.

[124] J. R. Marden and A. Wierman, "Distributed welfare games," *Operations Research*, vol. 61, no. 1, pp. 155–168, 2013.

[125] P. G. Sessa, M. Kamgarpour, and A. Krause, "Bounding inefficiency of equilibria in continuous actions games using submodularity and curvature," *arXiv preprint arXiv:1903.00950*, 2019.

[126] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[127] Y. Zhang, J. Duchi, M. I. Jordan, and M. J. Wainwright, "Information-theoretic lower bounds for distributed statistical estimation with communication constraints," in *Advances in Neural Information Processing Systems*, 2013, pp. 2328–2336.

VITA

VITA

Lintao Ye was born on January 26th, 1993, in Wuhan, China. He received his B.E. degree in Material Science and Engineering from Huazhong Unversity of Science and Technology, China, in 2015. He received his M.S. degree in Mechanical Engineering from Purdue University, West Lafayette, USA, in 2017. He is currently pursuing a Ph.D. at the school of Electrical and Computer Engineering at Purdue University. His research interests include combinatorial optimization, control systems, signal processing, and machine learning.