Accountable Data Fusion and Privacy Preservation Techniques in Cyber-Physical Systems

Ruoxi Jia

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2018-135
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-135.html

October 10, 2018

Accountable Data Fusion and Privacy Preservation Techniques in Cyber-Physical Systems

by

Ruoxi Jia

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Costas Spanos, Chair
Professor Dawn Song
Professor Kameshwar Poolla
Professor Deirdre Mulligan

Fall 2018

Accountable Data Fusion and Privacy Preservation Techniques in Cyber-Physical Systems

Abstract

Accountable Data Fusion and Privacy Preservation Techniques in Cyber-Physical Systems

by

Ruoxi Jia

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Costas Spanos, Chair

With the deployment of large sensor-actuator networks, Cyber-Physical Systems (CPSs), such as smart buildings, smart grids, and transportation systems, are producing massive amounts of data often in different forms and quality. These data are in turn being used collectively to inform decision-making of the entities that engage with the CPSs. The impact of these systems on people's lives has led to a strong call for accountability of system decisions made based upon various data sources. The collection, analysis, and dissemination of these data also present a privacy risk that needs to be addressed.

The first part of this dissertation focuses on accountable data fusion. We develop an online prediction framework that integrates dynamic sensor measurements with prior knowledge. The proposed framework facilitates reasoning about prediction confidence, which is crucial to making dependable decisions. We also move beyond predictive modeling to interpretable analytics by evaluating the influence of each data instance on the algorithmic outcome. We formalize the notion of "data value," and provide efficient algorithms to compute it. This value notion not only enables us to better understand black-box predictions through the lens of training data, but allows for fair allocation of the profit generated from a prediction model that is built with data from cooperative entities. We further use the proposed value notion to develop an effective data sanitization mechanism, which screens off low-quality or even adversarial data instances from the training set.

In the second part, we address the problem of incorporating privacy as an active engineering constraint into the CPS design and operation. We discuss a privacy metric inspired by information theory and provide algorithms to optimize the privacy mechanism for a given system or co-design the privacy mechanism and system control. In order to avoid unnecessary privacy-utility tradeoffs, we develop a framework to identify redundant data for specific decision-making processes. Furthermore, we present a privacy-preserving data publishing system, which can achieve improved data utility by optimizing the privacy mechanism according to the use of published data. While the algorithms and techniques introduced can be applied to many CPSs, we will mainly focus on the implications for smart buildings.

To my family

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my extreme gratitude to my advisor Costas Spanos for his mentorship, generosity with his time, inspiration, and immense support that I got from him during my doctoral study. Every word in a conversation with him is a life-long advice. He helped me better understand academic life, taught me what a good research is, and enlightened me to become an independent thinker. Most importantly, he helped me believe in my potential and make most of it.

I am also grateful to the faculty and researchers who helped shape my research path and brought multidisciplinary perspectives to my work: Dawn Song for introducing me to the fantastic world of security research; Deirdre Mulligan for sharing her deep insights into legal and policy perspective on privacy challenges; Kameshwar Poolla for the fruitful and constructive discussions on control theory; Tianzhen Hong for giving me opportunities to work on practical problems with regard to buildings; Ce Zhang for being an invaluable source of experience, a very helpful critic, and teaching me everything about fonts.

Many of my recent work has been in close collaboration with Bo Li. Over the short time that Bo has been at Berkeley, she has given me a unique viewpoint of research and shared invaluable experience of academic life. Her energy, passion, and caring have constantly inspired me to move forward. I cannot go without acknowledging Roy Dong, a great mentor who provided me with invaluable advice and helped me navigate through graduate school and research.

I would like to thank all of the wonderful people I have had the pleasure to discuss research, without whom this work would not have been possible: Han Zou for being a vast knowledge base about wireless communication and a great teammate, Ioannis Konstantakopoulos for the insights into game theory and lightening my days with a great sense of humor, Yuxun Zhou for refining my research with his great mathematical sharpness, being a sounding board for concerns, and providing me with needed advice. I also would like to thank Gerald Friedland, Mikkel Baun Kjrgaard, Fisayo Caleb Sangogboye, Zhaoyi Kang, Chaowei Xiao, Joe Near, Roel Dobbe, Kaiyu Sun, Om Thakkar, Yigitcan Yesilata, David Dao, Boxin Wang, Zhuolin Yang, Frances Ann Hubis, Nick Hynes, Prashanth Ganesh, Baihong Jin, Hari Prasanna Das, Lucas Spangher for fruitful discussions.

I want to thank Claire Baek, Pengpeng Lu and Ying Qiao for being caring and supportive sisters and helping me manage the various stresses of the graduate school life. And a special callout to my friends, Yuting Wei and Qian Zhong, who have listened to my problems and lent their moral support over the course of my PhD.

I want to thank my friends in the CREST center: Jason Poon, Linda Lee, Daniel Gerber, Yongjun Li, Kelly Fernandez for a unparalleled and immensely pleasant work environment. Many thanks to special people of CREST, who have made everything run smoothly: Yovana Gomez, Judy Huang, Chris Hsu.

My final thanks go to my family. I want to thank my parents who supported their only child to move all the way across Pacific Ocean so that she can pursue her passion and dreams. I am grateful for their love, sacrifices, and support—I will try my utmost to make it worth

their while. I feel so fortunate to forge through the graduate life with my husband, Ming Jin, who supported me in every capacity. My life with you is more colorful and enjoyable than it have ever been — thank you, Ming.

Chapter 1

Introduction

Seamless integration of computation, networking, and the physical world is featured in a multitude of engineering systems such as buildings, energy grids, transportation, and healthcare. These systems are commonly termed Cyber-Physical Systems (CPSs) due to the tight coupling between cyber and physical processes thereof. Indeed, new CPS technologies are being deployed to create sensor and actuator networks that produce massive amounts of data, opening up opportunities to improve efficiency, resilience, and sustainability of CPSs. For example, various sensors are deployed in smart buildings to collect data about indoor environments [164, 86] and people's activities [89]; these data can be in turn used to understand people's comfort preferences [96, 87], analyze buildings' energy consumption [107, 85], and further smartly control lighting and air conditioning systems for better comfort, productivity, and energy efficiency. In the transportation sector, drivers communicate real-time locations to companies like Google and Uber, which aggregate data to provide a range of useful services, including real-time traffic maps, travel routing, dispatch of public transportation resources, and accident management. In the energy sector, real-time and granular metering of energy supply and demand helps create a new energy market that can incorporate unpredictable renewable energy sources and demand response [3] while ensuring grid stability and reliability.

With the dense instrumentation of sensors and actuators, it has become increasingly common to gather data from many heterogeneous sources, in different forms, content, and quality. There often exists a vast knowledge base about the principle of how the CPSs function. As a result, modeling and decision-making in these systems are neither completely data-driven nor completely derived from expert knowledge. Because of the distributed nature of data gathering, the modeling processes are susceptible to data poisoning attacks [16, 84], wherein malicious users inject well-crafted data aimed at misleading models to make arbitrarily incorrect predictions. Moreover, many time-sensitive and safety-critical decisions in the CPSs are made based upon the data. The heterogeneity of data, combined with its impact on the CPS operation, poses unique challenges to data integration and analysis techniques: How do we encode prior knowledge? How do we fuse data streams from heterogeneous sources and prior knowledge for real-time prediction? How do we model the uncertainty of a model prediction? How do we interpret model predictions? How can we impute the utility of

predictions to each data instance used for modeling? How do we identify low-quality data instances and even the ones that have been manipulated by a malicious attacker?

The ubiquitous monitoring in these CPSs also triggers people's concern about privacy. For instance, smart home data, such as occupancy, energy consumption, reveals information about people's schedules, habits, etc. Without a framework to protect the privacy of people who interact with the CPSs, the public may be very conservative about sharing their data, preventing adding new intelligence to the CPSs. Over the past decade, there have been breakthroughs in understanding privacy from a scientific point of view. Differential privacy [49] has emerged as a very strong notion of privacy that protects private records in a database against adversaries with any side information. Differential privacy has been actively used by Google, Apple, and Uber for analyzing users' data. Information-theoretic measures have also been employed to measure the amount of information leakage [129, 77, 78]. These quantifiable perspectives of privacy provide unprecedented opportunities to understand the tradeoff between privacy and data utility.

In the context of CPSs, new opportunities and challenges for addressing privacy issues emerge. A unique characteristic of the data generated from sensing and actuation networks is that they are streaming. Unlike the static data (e.g., medical records), streaming data are temporally rich and often correlated, which makes privacy over streaming data far more difficult. On the other hand, the data sources are often distributed and have a significant computational capability. This is a different setup from the conventional central server or cloud-based architecture considered for privacy study. In addition, these data are often used for complex, and oftentimes pre-determined, tasks such as real-time control, planning, and learning. The computational capability at the sensor level and the prior knowledge about data use bring us unique opportunities to optimize the privacy mechanism and deploy it locally. However, there is still a set of fundamental problems we need to study for privacy in CPSs: How do we develop meaningful privacy metrics for streaming data? How do we model the complex use of data? What kind of mathematical models are suitable for formal analysis of the privacy-utility tradeoff? How do we identify useless data and avoid unnecessary tradeoffs? How do we incorporate privacy into the design and operation of CPSs in a principled manner? How do we publish the dataset collected from CPSs in a privacy-preserving way in order to promote the development of more advanced data analytics?

1.1 Thesis Approach

In order to address the aforementioned questions, we divide this dissertation into two parts, first focusing on accountable data fusion, and then discussing privacy protection techniques.

Our approach in accountable data fusion is to use Bayesian hierarchical modeling as a unifying framework to model the interdependence and uncertainty of sensor measurements and prior knowledge. We develop efficient sequential Monte Carlo methods to make real-time predictions given the measurements obtained on the fly. Using the Bayesian modeling approaches, each model prediction is associated with a posterior distribution that characterizes

the confidence of the prediction. We further develop a framework for interpreting a model's predictions by tracing them back to the training data, which the model parameters are ultimately derived from. Drawing ideas from cooperative game theory, we formulate the modeling process as a coalitional game among training data instances and apply the Shapley value to assess the contribution of each training data to the model predictions.

The second part of the thesis is aimed at providing privacy protection to the entities that engage with the CPSs while maintaining the quality of services provided. One of the key aspects for achieving the goal is to model privacy loss and utility as a function of data noise. The relationship between utility and data noise is usually overlooked by assuming that data with large volume and high accuracy always leads to better data-dependent decisions. Due to the asymmetry between extremely high dimensional data measurements and relative small dimension of decision variables, the intuition is that much of the data is redundant for the decision-making and control underlying the system operation. We formalize this intuition by using multiparametric programming to study the effect of data noising on an optimization-based decision-making process, and further proposing concepts to characterize the data redundancy. In summary, our approaches to reasoning about the privacy-utility trade-offs include

1. using a rigorous quantifiable definition of privacy inspired by information theory;
2. incorporating privacy loss into the objective function for CPS operation;
3. characterizing the optimal performance of a CPS that takes into account privacy objectives, and
4. learning the data utility function when it is not known a priori.

1.2 Contributions

> *The goal of my dissertation is to develop an accountable and privacy-preserving data analysis framework for Cyber-Physical Systems.*

This dissertation makes the following contributions.

Fusing Sensor Data and Prior Knowledge for Online Prediction

We propose a data fusion framework for inferring unobserved states of a dynamical system from sensor measurements and prior knowledge. The framework is based on hierarchical Bayesian modeling, which leads to quantifiable measures of prediction confidence. We propose a sequential Monte Carlo method to efficiently estimate unobserved states in real-time. The proposed data fusion framework is applied to indoor localization, in which the goal is to track people's location based on measurements from WiFi access points, ultrasonic sensors, and the knowledge about floormap. We test the information fusion-based indoor localization

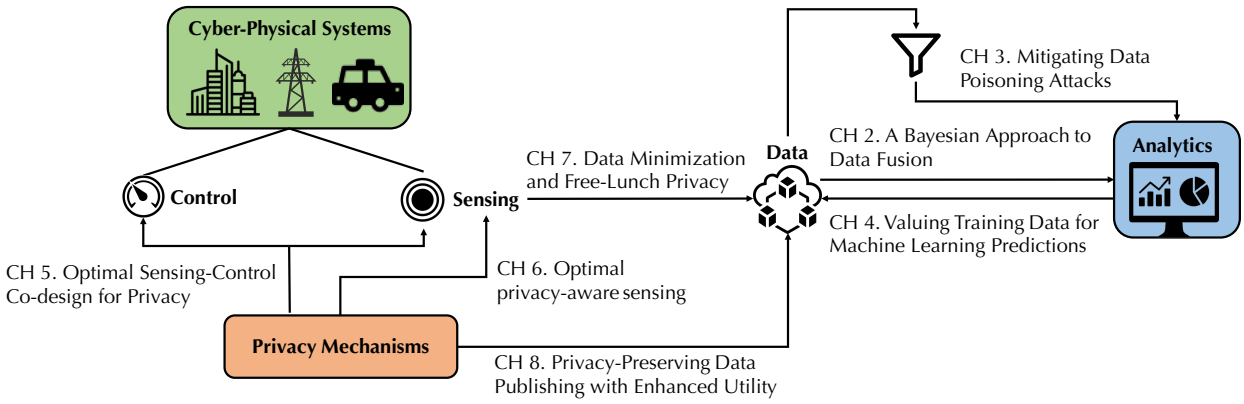


Figure 1.1: Thesis overview.

system in a real-world office environment and demonstrate the improvements in localization accuracy (Chapter 2).

Understanding Model Predictions via Valuing Training Data

We propose to interpret model predictions by tracing a model's prediction back to each training data point from which the model parameters are learned. To formalize the impact of a training point on the prediction, we first model the prediction process as a coalitional game. The Shapley value in cooperative game theory assigns a unique distribution of a total surplus generated by the coalition of all "players." Our game-theoretic formulation of the prediction process allows us to use the Shapley value to determine the importance of each training data for a given model prediction. However, calculating the Shapley value requires exponential time. Therefore, we propose efficient algorithms for approximating the Shapley value with provable error bounds. Particularly, when data is used for training K-nearest neighbor classifiers, we develop an exact analytical formula for the Shapley value, enabling the valuation algorithm to scale to millions of data points. The data valuation algorithms proposed can also be used in the data marketplace to divide the revenue generated by a data-driven model into each data contributor's fair share (Chapter 3).

Mitigating Data Poisoning Attacks

Models built with distributed data sources are highly susceptible to data poisoning attacks, wherein malicious users inject well-crafted training data aimed at misleading models to make arbitrarily incorrect predictions. We propose a data sanitization mechanism that employs the Shapley value to assess the contribution of each training data to the model predictions and filters out the training points with low Shapley values. The mechanism is computationally efficient and agnostic to the actual learning algorithms and attack methods. We provide theoretical insights into the robustness enabled by the mechanism. We compare our proposed

mechanism with existing defense methods through extensive experiments, showing that our mechanism can more effectively defend against multiple state-of-the-art poisoning attacks (Chapter 4).

Optimal Sensing-Control Co-design for Privacy

The ability of CPSs to accommodate privacy in tandem with system performance relies on an integrative and design-thinking perspective. It is crucial to incorporate privacy as an active engineering constraint throughout the system design and operation. We provide a new approach for the rigorous analysis of the interplay between privacy and control in CPSs. We characterize the privacy loss from streaming sensor measurements under inference attacks using information theoretic measures. Using the proposed privacy loss measure, we simultaneously design the privacy mechanism that modifies the noise level of the observer of private information and the controller that controls the environment based on the privatized measurements output by the noisy observer. The design problem can be converted into a partially observable Markov decision process, with the private information as part of hidden states. We investigate the optimal joint design of the privacy mechanism and the controller and propose efficient algorithms to achieve optimal (or near-optimal) design. We apply our algorithm to the co-design of an occupancy-based smart home temperature controller and the privacy mechanism implemented in the occupancy sensor (Chapter 5).

Privacy-Aware Sensing for Model Predictive Control

Although the co-design approach offers new opportunities to improve system efficiency and privacy, under many circumstances, we are faced with an existing system with a fixed control policy which is not likely to be re-designed. We focus on Model Predictive Control (MPC), a widely used controller in CPSs, and propose an optimal scheme to add noise into sensor measurements while ensuring that the control performance of the MPC is acceptable. We formalize the design of the optimal noising scheme as a constrained optimization, where we seek for a random data perturbation mechanism that minimizes privacy loss subject to the allowable controller performance sacrifice specified by the system operator. We apply the proposed scheme to enhance privacy in occupancy-adaptive smart building control, and demonstrate that our noising scheme generates data measurements that can hide occupants movement patterns while still satisfying the utility requirements desired by the building manager (Chapter 6).

Data Minimization and Free-Lunch Privacy

We formalize the CPS operation as an optimization problem, where data affects the operation by entering the optimization as a parameter. This formalization can incorporate a wide range of applications including model predictive control, optimal control, planning, etc. We build upon the multi-parametric optimization theory which studies the effect of parameter

variations on the solution of the optimization problem, and develop various theoretical results about the sensitivity of system operation to data measurements. Particularly, we characterize the conditions under which there is a direct tradeoff between privacy and utility, and the situations where free-lunch privacy exists, i.e., the data measurement can be concealed or falsified without harming the optimality of decision-making. Further, we propose a pragmatic privacy mechanism that provides provable guarantees of optimal usage of data while exploiting free-lunch privacy whenever it exists. This framework allows system operators to better understand the quantity and accuracy of data needed for efficient system operation, and to further minimize data collection in order to respect people's privacy. We illustrate the framework by exmamining the redundancy of occupancy data collected for smart home control (Chapter 7).

Privacy-Preserving Data Publishing with Enhanced Utility

Spurred on by the benefits mutual to the public, system operators and research communities, there is a continually rising demand for the publication of datasets collected in various CPSs. Data published in the original form comes with the risk of privacy loss. On the other hand, high-quality published data is vital to enable robust data-driven models. We propose a privacy-preserving data publication system, which customizes the privacy mechanism to the data use so that the published data can retain more utility. However, due to the diversity of potential data uses, it will be cumbersome to enumerate and hard-code every possible data use and design the corresponding privatization process. We propose a unified protocol to comprehend users' diverse interests by learning from their interactions with the data publishing system. In the proposed protocol, data users are first provided with some data that does not involve privacy risks such as public datasets, and then label the similarity of these data points according to the features of particular interest to them. We introduce an algorithm to learn their intended data use from the similarity labels and optimize the privatization processes accordingly. We demonstrate the efficacy of our data publishing system through various real-world datasets and show the published data can adapt to the idiosyncrasies of different data uses and better retain relevant information (Chapter 8).

Part I

Accountable Data Fusion

Chapter 2

A Bayesian Approach to Data Fusion

2.1 Motivating Application: Indoor Localization

The indoor location sensing technology has emerged as an inherent part of the "smart buildings" as it provides great potential for building operation improvement and energy saving. For instance, an on-demand ventilation or lighting control policy must know the usage of the building spaces, which may involve when building occupants enter or exit the building, where they inhabit, what time they occupy the spaces, the duration of occupancy, *etc.* Such applications require the location sensing systems to provide real-time estimate of occupants' locations, which is also termed "indoor tracking", in order to realize fine-grained, responsive building operations.

Most indoor tracking systems necessitate each occupant to carry or wear a powered device such as an infrared [162], ultrasonic [112, 130, 70], or Radio Frequency transceiver [165, 10, 136]. Even if the transceiver is miniaturized into a convenient form, occupants are not willing or likely to carry it at all times. Another subset of tracking systems alleviate the need for carrying specialized devices by using the inertial sensors on smartphones to perform dead reckoning [27, 9, 158]. However, specialized programs are required to be installed on smartphones to continuously collect inertial sensing data, and thereby the associated energy issues or occupants' engagement become the main impediment.

On the contrary, we enable non-intrusive indoor tracking by developing an information fusion system that takes advantage of noisy measurements from various sensors, namely, WiFi access points and ultrasonic sensors. WiFi access points are beneficial for wide spatial coverage while WiFi signals transmitted in the indoor environments suffer from large variations [20]; ultrasonic sensors are able to accurately locate the occupants in their detection zones which are nevertheless limited spatially. Our vision is of occupants carrying some device with WiFi module, which can be smartphones, tablets, wearable devices, *etc.*, in the indoor space where ultrasonic sensors can provide opportunistic calibration of the location estimation. The location sensing system is operating in a passive way, *i.e.*, there is no need for specialized devices or programs for location inference.

In addition to the sensor measurements, another key input for our system is the floormap of the indoor space of interest. Floormap information has been used to refine walking trajectory estimates by eliminating wall-crossings or unfeasible locations [54, 160, 53]. There has also been efforts to use the floormap to reduce the complexity of the tracking task by properly quantizing the indoor space [71, 101, 151, 73]. In effect, we can also acquire some prior knowledge of occupants' dynamic motion from the floormap. The indoor space comprises several typical components, such as cubicles, offices, corridors, open areas, *etc.*, where occupants' motion exhibit distinctive patterns. For example, when located at his/her office or cubicle, the occupant is very likely to keep static; the occupant walking on a particular corridor tends to continue the motion constrained along the corridor, while an occupant in an open space is free to move in any direction. Such information of space use is useful to track occupants' movement, notwithstanding it is less considered in previous work. Gusenbauer *et al.* [68] exploited different types of movements to improve the tracking model. This was done by introducing an activity recognition algorithm based on accelerometer data to model pedestrians' steps more reliably. Park [124] proposed incorporating the floormap information by "path compatibility", where occupants' motion sequences and motion-related information (e.g., duration and speed) are first estimated based on mobile sensing data, and then localization is achieved via matching occupants' motion sequences and the hypothetical trajectories provided by the floormap. Kaiser *et al.* [90] proposed a motion model based on the floormap, which weights the possible headings of the pedestrian as a function of the local environment. Our work differs from [68] and [124] in that our work does not rely on the inertial measurements to recognize the motion. Instead, the motion information is extracted from the floormap. We exploit the prior knowledge that the floormap endows us about the occupants' typical movement and activity, not merely the possible headings at each point of the floormap as in [90]. It is, therefore, the objective of this chapter to propose MapSentinel, a non-intrusive location sensing system via information fusion, which combines the various sensor measurements with the floormap information, not only as a sanity check of estimating trajectories but as an input for occupants' kinematic models.

The main contributions of this chapter are as follows:

- We build a non-intrusive location sensing network consisting of modified WiFi access points and ultrasonic calibration stations, which does not require the occupants to install any specialized programs on their smartphones and prevents the energy and occupant engagement issues.

- We propose an information fusion framework for indoor tracking, which theoretically formalizes the fusion of the floormap information and the noisy sensor data using Factor Graph. The Context-Augmented Particle Filtering algorithm is developed to efficiently solve the walking trajectories in real time. The fusion framework can flexibly graft floormap information onto other types of tracking systems, not limited to the WiFi tracking schemes that we will demonstrate herein.

- We evaluate our system in a large typical office environment, and our tracking system can achieve significant tracking accuracy improvement over the purely WiFi-based tracking systems.

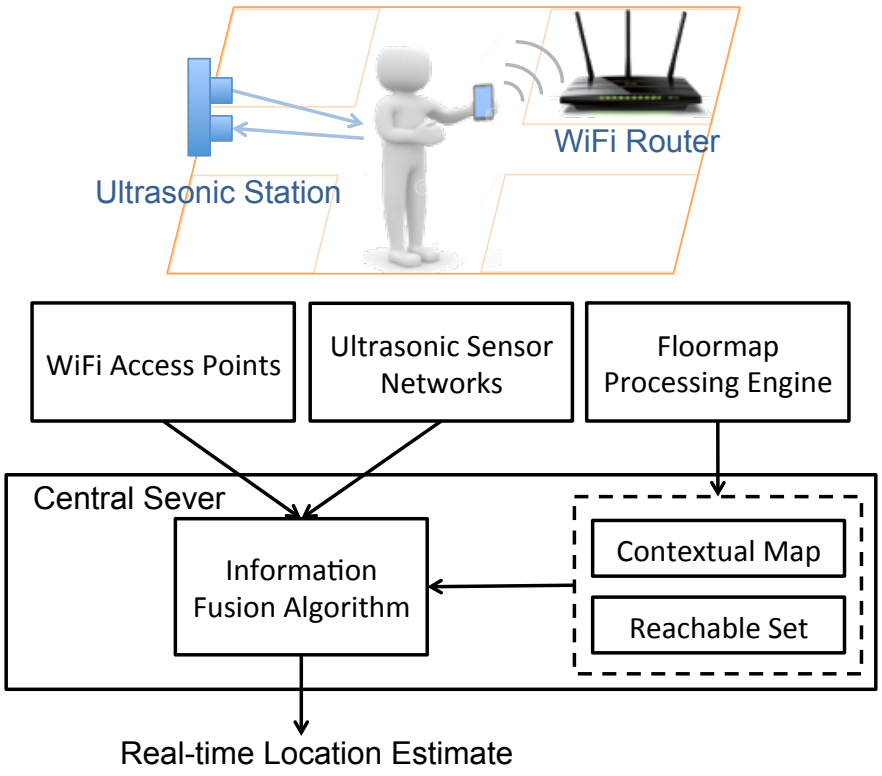


Figure 2.1: MapSentinel architecture—WiFi APs keep tracking occupants' locations, and the estimation is periodically refined using the ultrasonic stations deployed in the environment. Furthermore, the sensor measurements and the floormap information are combined via the information fusion algorithm to estimate location in real-time. The floormap processing engine helps transform the floormap to the information accessible to the fusion algorithm.

2.2 System Architecture

Figure 2.1 presents the overall architecture of the proposed indoor positioning system, which we call MapSetinel. There are three key components in MapSentinel: the non-intrusive sensing networks, the floormap processing engine, and the information fusion algorithm. The non-intrusive sensing networks, as the name suggests, generate location-related measurements without the need for computation on the smartphone end. Our sensing networks consist of WiFi access points (APs) and ultrasonic calibration stations, which track locations by relating

the WiFi signal strength or the sound time-of-flight to the distance. The floormap processing engine converts the pictorial floormap to the information that can be directly combined with the sensor measurements in the fusion algorithm. The output of the floormap processing engine represents the prior knowledge obtained from the map, and can be computed in the offline phase. We will present the details of the main components of MapSentinel in this section.

WiFi Access Points

IEEE 802.11 (WiFi) is the most commonly used wireless networking technology with widely available infrastructure in large numbers of commercial and residential buildings. Nearly every existing commercial mobile device is WiFi enabled. The common method to utilize WiFi for indoor location sensing is to enable the mobile device to collect WiFi Received Signal Strengths (RSS) of nearby WiFi APs by installing an application on the mobile devices. Our system, on the contrary, leverages WiFi in a non-intrusive manner. Rather than modifying the hardware or software of occupants' mobile devices, we upgrade the software of the existing commercial WiFi APs to allow them to detect the RSS of each mobile device, while providing basic internet service to occupants as well. The RSS and media access control (MAC) address of each mobile device will be forwarded to the server and the occupant can be identified through the unique MAC address of the mobile device.

Ultrasonic Calibration Stations

Ultrasonic sensors measure the distance to the obstacle in the front to accurately position the object in its detecting range, which works by detecting the time of return, t, and the distance is given by:

$$d = \frac{v_{\text{sound}} \times t}{2} \tag{2.1}$$

where $v_{\text{sound}} \approx 340$ m/s is the velocity of sound in the air. The advantages include centimeter-resolution distance measurements and limited span of detection angles, which make it suitable for online calibration of indoor positioning systems. Figure 2.2 demonstrates typical traces of the ultrasonic sensor readings when the occupant moves across the detection zones. By properly thresholding the distance measurements, the ultrasonic sensor can be used as an indicator of occupant presence inside its detection zone.

The network consists of deployed ultrasonic stations and data collection center, which communicate with XBee radio modules operating the IEEE 802.15.4 standard, more specifically, the ZigBee protocols, as shown in Figure 2.3. The radios are low-power and can operate reliably in the indoor space, where the network can be automatically established by the coordinator, in our case, the data collection center. The data center controlled by Arduino enquires about the ultrasonic station for measurements periodically, so that the measurement frequency is 1 Hz, and transfers the data to the computer connected by serial ports. Each ultrasonic station is equipped with three ultrasonic sensors, whose directions are offset by

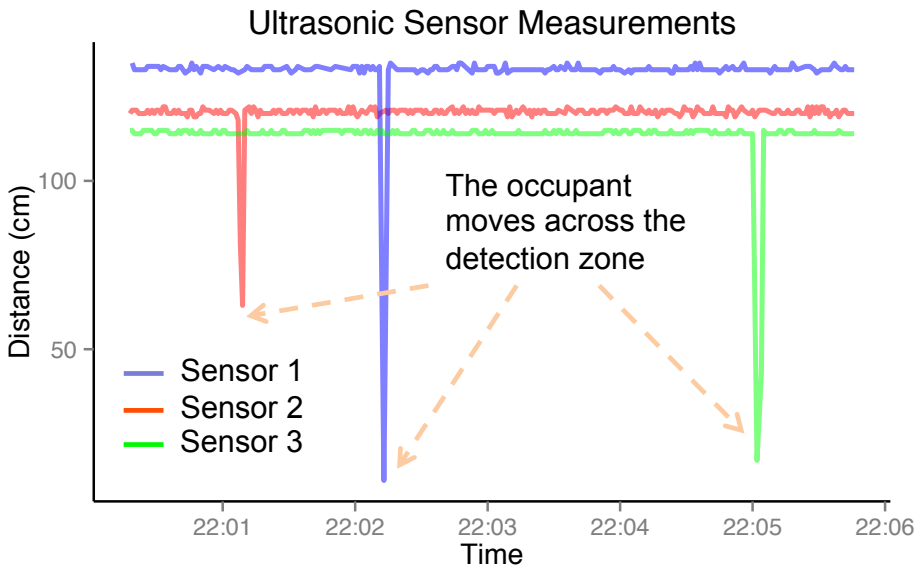


Figure 2.2: The measurements of ultrasonic stations deployed in the space. When the occupant is within the detection zone of the ultrasonic station, the sensor reading exhibits a smaller value.

15°. As the measurement range spans 15° for each ultrasound, this covers an area of 45° in the front of the station, which is sufficient for indoor area localization.

Floormap Processing Engine

The indoor space is well structured and typically organized into corridors, open areas, walls, rooms, *etc.* Depending on the occupant's present location, the motion is constrained by these external factors. For instance, an occupant on a particular corridor has high probability continuing its motion constrained along the corridor—or an occupant walking in the open area is free to move in any direction. Likewise, an occupant in his/her cubicle area is more likely to stay static. Based on different motion capabilities, we categorize the indoor space into several contexts, namely, open space, constrained space and static space. In addition, the floormap processing engine is designed to convert the original floormap into the *contextual floormap* that indicates the context of each point in the original floormap. The details of each component of contextual floormap is provided in Table 2.1. We use the word "canonical direction" to refer to the direction of constrained space along which the movement has more freedom.

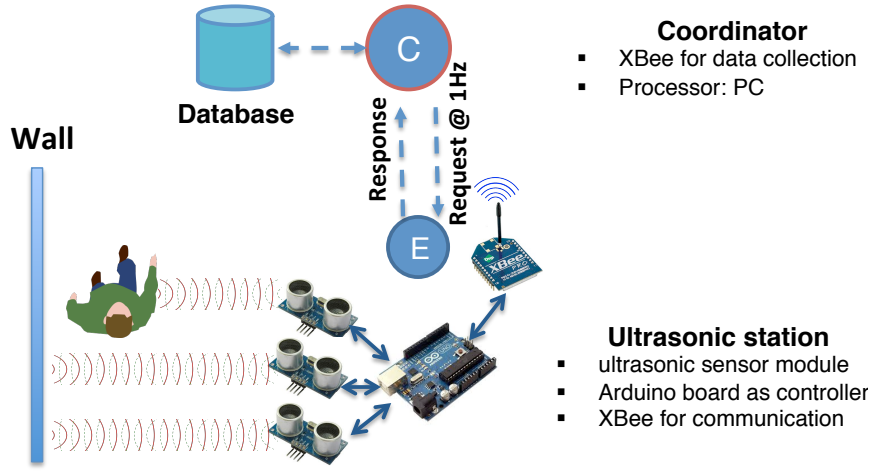


Figure 2.3: Illustration of the configuration of the ultrasonic calibration station. The coordinator requests measurements at 1 Hz frequency through the IEEE 802.15.4 protocol, and deposits collected data to the local database. The ultrasonic station takes three independent measures from its sensor points to detect occupant presence in the vicinity.

Table 2.1: Components of contextual floormap.

Context	Symbols	Motion Characteristics
Free Space	FS	Move freely, e.g., rooms
Constrained Space	CS	Move along canonical direction, e.g., corridors
Static Space	SS	Stay static, e.g., cubicles

In addition, the occupant motion is also constrained by speed restrictions. Another function of the floormap processing engine is to compute the reachable set containing all the points visited with admissible speed from a given starting point. In the indoor space, the geographical distance between two positions in a floormap does not necessarily equal to the walking distance between them due to the block of walls and other obstacles. Hence, the physical features of the indoor environments would be ignored if the reachable set is confined within a fixed radius centered around the given starting point. The floormap processing engine addresses this problem by converting the floormap to a graph where all the non-barricade nodes connect to their neighboring non-barricade nodes and the barricade nodes do not have connections to any other nodes. In this way, the reachable set of a given node can be computed through finding the nodes within the maximum depth from the root node, which can be efficiently solved by breadth-first search algorithm [146].

2.3 Information Fusion Framework

Consider that the indoor space of interest is composed of M contexts, in each of which occupants exhibit a particular sort of kinematic patterns. Denote the context at time k as m_k where $m_k \in \{FS, CS_1, \cdots, CS_R, SS\}$. The subscript of CS represents the index of the certain direction of constrained space and R is the total number of different directions. Let the state $\mathbf{x}_k = (\mathbf{z}_k, m_k)$ consist of the position and velocity components of the occupant in the Cartesian coordinates $\mathbf{z}_k = (x_k, y_k, \dot{x}_k, \dot{y}_k)$, as well as the context m_k. If the position is known, the context can be uniquely determined by the contextual floormap. We characterize this correspondence via a function $\mathcal{M} : \mathbb{R}^4 \to \mathbb{R}$ which assigns a specific context m_k for $\mathbf{z}_k$. The tracking problem can be viewed as a statistical filtering problem where $\mathbf{z}_k$ is to be estimated based on a set of noisy measurements $y_{1:k} = \{y_1, \cdots, y_k\}$ up to time k. Specifically, y_k is the measurements available at time k, and, in our case, it includes measurements from multiple sensors, $\{y_k^n\}_{n=1}^{N_s}$ where N_s is total number of sensors deployed in the space of interest. We model the uncertainty about the observations and the states by treating them as random variables and assigning certain probability distribution to each random variable. In this setting, we want to compute the posterior distribution of the state given the measurements up to time k, i.e., $p(\mathbf{z}_k|y_{1:k})$.

The impact of introducing context as an auxiliary state variable is manifold. Firstly, the transition of contexts m_{k-1} to m_k determines the type of motion executed during the time interval $(k-1, k]$. For instance, if the context remains the same, then the occupant should follow the motion type defined by the two identical contexts; on the contrary, if the context varies during $(k-1, k]$, then the occupant would execute the motion that is defined by neither of the contexts. For simplicity, we will assume a free motion. That is, the position/velocity state at time k, $\mathbf{z}_k$, depends on not only the past state $\mathbf{z}_{k-1}$ and m_{k-1}, but also the current context m_k stochastically. Moreover, there is a deterministic mapping between $\mathbf{z}_k$ and m_k as is specified by the contextual map. In order to facilitate visualization and analysis of the complex dependencies among the variables, we use a factor graph to represent the states, observations and the functions bridging these variables, as illustrated in Figure 2.4.

A factor graph has two types of nodes, *variable node* for each variable and *function node* for each local function, which are indicated by circles and squares, respectively. The edges in the graph represents the "is an argument of" relation between variables and local functions. For example, the function $\mathcal{T}_k$ has four arguments, $\mathbf{z}_k$, $\mathbf{z}_{k-1}$, m_{k-1} and m_k. Three types of local functions are involved in our model:

• $\mathcal{T}_k(\mathbf{z}_k, \mathbf{z}_{k-1}, m_k, m_{k-1}) = p(\mathbf{z}_k|\mathbf{z}_{k-1}, m_k, m_{k-1})$: transition model, or the prior information on the state evolution over time. Inspired by Variable Structure Multiple Model Estimator in [7], we propose CDKM to capture the context-dependent characteristics of occupants' motion in the indoor space.

• $\mathcal{O}_k(\mathbf{z}_k, y_k) = p(y_k|\mathbf{z}_k)$: observation model, or how the unknown states and sensor observations relate. We will introduce PSMM where the relationship between locations and sensor observations is characterized by certain conditional probabilities and multiple sensor observations are combined via Bayes' theorem.

• $\mathcal{C}_k(\mathbf{z}_k, m_k)$: characteristic function that checks the validity of the correspondence between $\mathbf{z}_k$ and m_k using the contextual floormap.

Note that the prior knowledge abstracted from the floormap is inherently accommodated to this problem by defining characteristic function and parameterizing the transition model as will be elaborated in the following section.

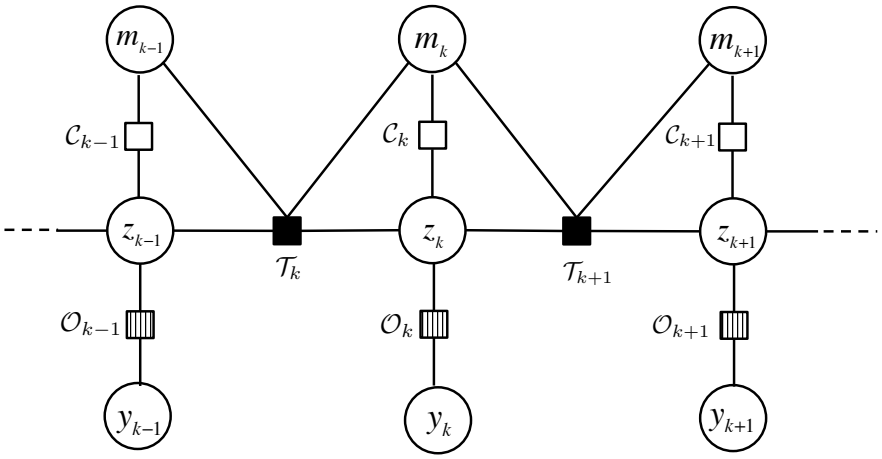


Figure 2.4: A factor graph model representation of the dependencies among location, velocity, context and observation.

Context-Dependent Kinematic Model

We assume that given $\mathbf{z}_{k-1}$, m_{k-1} and m_k, the current position/velocity $\mathbf{z}_k$ follows a Gaussian distribution, of which the mean and covariance matrix are specified as

$$p(\mathbf{z}_k|\mathbf{z}_{k-1}, m_k, m_{k-1}) \sim \mathcal{N}(F(m_{k-1}, m_k)\mathbf{z}_{k-1}, GQ(m_{k-1}, m_k)G') \tag{2.2}$$

The equivalent state space model of Equation (2.2) is given by

$$\begin{aligned} &\mathbf{z}_k = F(m_{k-1}, m_k)\mathbf{z}_{k-1} + Gv(m_{k-1}, m_k) \\ &v(m_{k-1}, m_k) \sim \mathcal{N}(0, Q(m_{k-1}, m_k)) \end{aligned} \tag{2.3}$$

where $F(m_{k-1}, m_k) \in \mathbb{R}^{4\times4}$ determines the mean of the distribution of the next state. Let a denote the acceleration, we have the following kinematic equations,

$$x_k = x_{k-1} + \dot{x}_{k-1}T + \frac{1}{2}aT^2 \tag{2.4}$$

$$\dot{x}_k = \dot{x}_{k-1} + aT \tag{2.5}$$

where T is the sampling period. We will assume constant velocity, and model a as a Gaussian noise term. If we manipulate Equations (2.4) and (2.5) into matrix forms, then it can be

identified that $F(m_{k-1}, m_k)$ has two possible values corresponding to moving or remaining static,

$$F_0 = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.6}$$

F_1 imposes the velocity component of the state $\mathbf{z}_k$ to be zero and $F = F_1$ when the context remains to be static space, *i.e.*, $m_{k-1} = m_k = SS$; otherwise, $F = F_0$.

The matrix G is given by

$$G = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix} \tag{2.7}$$

$Q(m_{k-1}, m_k)$ stands for the process noise and, as the notation indicates, it is also a function of the context transition from $k-1$ to k. We will adopt the concept of directional noise to handle the constraints imposed by the contextual map. To see this, note that occupants in the free space ($m_{k-1} = m_k = FS$) can move in any direction with equal probability, therefore using equal process noise variance in both x and y direction, *i.e.*,

$$Q_0 = \begin{bmatrix} \sigma_f^2 & 0 \\ 0 & \sigma_f^2 \end{bmatrix} \tag{2.8}$$

For occupants moving on the constrained space ($m_{k-1} = m_k = CS_i, \forall i = 1, \cdots, R$) such as corridors, more uncertainty exists along than orthogonal to the corridor. Denote the variances along and orthogonal to the corridor by σ_a^2 and σ_o^2 ($\sigma_a^2 > \sigma_o^2$), respectively, and the canonical direction of the constrained space CS_i is specified by the angle ϕ_i (measured clockwise from y-axis). Then the process noise covariance matrix corresponding to the motion in the constrained space is given by

$$Q_i = \begin{bmatrix} -\cos\phi_i & \sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \begin{bmatrix} \sigma_o^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} \begin{bmatrix} -\cos\phi_i & \sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \tag{2.9}$$

The preceding model specification incorporates the scenarios where the context remains the same during the time interval $[k-1, k]$ and the occupant will keep the motion type defined by the two identical contexts. On the contrary, if the context switches during the time interval $[k-1, k]$, we will assume a free motion pattern, *i.e.*, $F = F_0$, $Q = Q_0$. Table 2.2 summarizes our model given all possible context transitions.

Probabilistic Sensor Measurement Model

We construct probabilistic models for each sensor and multisensor fusion can be performed via Bayes' rule. Assuming that N_s different sensors function independently, then the observation

Table 2.2: Context-dependent kinematic models.

Context Transition	**Model Specification**	
	$F(m_{k-1}, m_k)$	$Q(m_{k-1}, m_k)$
$m_{k-1} = m_k = FS$	F_0	Q_0
$m_{k-1} = m_k = CS_i$	F_0	Q_i
$m_{k-1} = m_k = SS$	F_1	Q_0
$m_{k-1} \neq m_k$	F_0	Q_0

model $p(y_k|\mathbf{z}_k)$ can be factored as

$$p(y_k|\mathbf{z}_k) = \prod_{n=1}^{N_s} p(y_k^n|\mathbf{z}_k) \tag{2.10}$$

This actually forms a convenient and unified interface to combine distinctive sensor data by projecting the heterogeneous measurements (y^n) to the probability space via *likelihood function*, $p(y^n|\mathbf{z})$. If one more sensor is added into the system, then the observation model can be simply updated by multiplying the corresponding likelihood. Different likelihood functions requires being trained for different types of sensors.

WiFi Measurement. In the free space, the WiFi signal strength is a log linear function of the distance between the transmitter and receiver. However, due to the multipath effect caused by obstacles and moving objects in the indoor environments, the log linear relationship no longer holds. Previous work has proposed to adding a Gaussian noise term to account for the variations arising from the multipath effect; however, the simple model-based method can hardly guarantee a reasonable performance in practice. Another popular way is to construct a WiFi database comprising WiFi measurements at known locations to fingerprint the space of interest, but it requires onerous calibration to ensure the accuracy. We propose a novel WiFi modeling method based on a relatively small WiFi training set to accommodate for the complex variations of WiFi signals in the indoor space. The key insight is to use Gaussian process (GP) to model the WiFi signal where the simple model-based method provides a prior over the function space of GP.

We collect WiFi signal strength data at N_w reference points over the space and let $\{l^j, y_w^j\}_{j=1}^{N_w}$ denote the training dataset, where l^j is a vector containing the distances of jth reference point to each of the WiFi APs deployed in the field and y_w^j is the observed WiFi signal strengths. Assume the WiFi observations are drawn from the GP,

$$y_w \sim \mathcal{GP}\big(\mu(l), k(l, l')\big) \tag{2.11}$$

where the mean function $\mu(\cdot)$ is imposed to be a linear model with the parameters adapted to the training samples. The covariance function $k(\cdot, \cdot)$ takes the squared exponential form,

$$k(l, l') = \sigma_f^2 \exp(-\frac{1}{2r^2}(l - l')^2) + \sigma_n^2 \tag{2.12}$$

where σ_n^2 stands for the variance of the additive Gaussian noise term in the observation process, and σ_f^2 and r are the hyperparameters of the GP. These parameters can be tweaked according to the training data, and we set $\sigma_n = 4$, $\sigma_f = 2$, $r = 5$ in our experiments. At an arbitrary point l_* in the space of interest, the posterior mean and variance of the WiFi signal y_* are

$$\bar{y}_* = \mu(l_*) + K(l_*, \mathbf{L})[K(\mathbf{L}, \mathbf{L}) + \sigma_n^2 I]^{-1}\mathbf{y}_w \tag{2.13}$$

$$cov(y_*) = K(l_*, l_*) - K(l_*, \mathbf{L})[K(\mathbf{L}, \mathbf{L}) + \sigma_n^2 I]^{-1}K(\mathbf{L}, l_*) \tag{2.14}$$

where $\mathbf{L}$ and $\mathbf{y}_w$ are the vectors concatenated by $\{l^j\}_{j=1}^{N_w}$ and $\{y_w^j\}_{j=1}^{N_w}$, respectively. $K(l_*, \mathbf{L})$ denotes the $1 \times N_w$ matrix of the covariances evaluated at all pairs of training and testing points, and similarly for the other entries $K(\mathbf{L}, \mathbf{L})$ and $K(\mathbf{L}, l_*)$. In previous work using GP to model the WiFi signal strength [58], the WiFi signal is assumed to follow the Gaussian distribution with the mean and variance given by Equations (2.13) and (2.14), respectively. However, the posterior variance derived from GP is a indicator of estimation confidence. It depends largely on the density of training samples in the vicinity of the evaluated position. That is, if the evaluated point l_* happens to fall into the area that is densely calibrated, then the posterior variance will be relatively small. The posterior variance derived from GP cannot truly reflect the variations of WiFi signals over time. Therefore, instead of using the posterior variance (2.14) in classical predictive equations, we model the likelihood as

$$y_* \sim \mathcal{N}(\bar{y}_*, \sigma_n^2) \tag{2.15}$$

Ultrasonic Measurement. Essentially, each of the ultrasonic sensors in the ultrasonic station can output the distance to the occupant passing in front of it. However, due to the missing data and measurement noise, the distance measurement is not always steady. Here, we will consider the ultrasonic station to be a binary sensor to indicate the occupancy in its detection zone. To be specific, the likelihood function is modeled as

$$p(y_k < \eta | \mathbf{z}_k \text{ in the detection zone}) = 1 \tag{2.16}$$

where η is the threshold for ultrasonic measurements.

Characteristic Function

The characteristic function imposes constraints on the correspondence between the position and the context, and embodies the prior knowledge available from the floormap. In the preceding section, we have defined a function $\mathcal{M}$ that sets up the relationship between the context and the position/velocity, *i.e.*, $m_k = \mathcal{M}(\mathbf{z}_k)$, and $\mathcal{M}$ can be readily read out from the contextual map. We thereby define the characteristic function to be

$$\mathcal{C}_k(\mathbf{z}_k, m_k) = \mathcal{I}[\mathcal{M}(\mathbf{z}_k) - m_k = 0] \tag{2.17}$$

where $\mathcal{I}[\cdot]$ is an indicator function. In other words, the characteristic function enforces the local correspondence defined by $\mathcal{M}$.

2.4 Context-Augmented Particle Filtering

In this section, we will discuss how to perform inference on the underlying factor graph of the tracking problem we formulated previously. The particle filter is a technique for implementing a recursive Bayesian filter by Monte-Carlo simulations [6]. The key idea of particle filter is to represent the required posterior density function by a set of random samples or "particles" associated with discrete probability mass, and compute the state estimate based on these "particles". The original particle filter proposed by Gordon *et al.* [62] was designed for a simple hidden Markov chain, which is also a cycle-free factor graph, using the Sampling Importance Resampling (SIR) algorithm to propagate and update the particles. However, the factor graph in our problem, as illustrated in Section 2.4, does have cycles due to the introduction of the context variable, and only approximate inference algorithms exist. We present a recursive approximate inference method for the cyclic factor graph by extending the particle filter and the resulting algorithm is termed *Context-Augmented Particle Filter* (CAPF).

To see the operation of the CAPF, consider a set of particles $\{\mathbf{z}_{k-1}^i, m_{k-1}^i\}_{i=1}^N$ that represents the posterior distribution $p(\mathbf{z}_{k-1}, m_{k-1}|y_{1:k-1})$ of the state. Note that m_{k-1}^i can be uniquely determined by $\mathbf{z}_{k-1}^i$ via the characteristic function. At time k, we have some new measurement y_k. It is required to construct a new set of particles $\{\mathbf{z}_k^i, m_k^i\}_{i=1}^N$ which characterizes the posterior distribution $p(\mathbf{z}_k, m_k|y_{1:k})$. Now, suppose we have an "*oracle*" that is capable of providing the context value m_k^i of the corresponding $\mathbf{z}_k^i$ even before we generate $\mathbf{z}_k^i$'s, then our task is equivalent to draw samples from the distribution

$$p(\mathbf{z}_k|m_k, y_{1:k}) \tag{2.18}$$

This can be carried out in two steps: First, the historical density $p(\mathbf{z}_{k-1}, m_{k-1}|y_{1:k-1})$ is propagated via the transition model $p(\mathbf{z}_k|\mathbf{z}_{k-1}, m_k, m_{k-1})$ to produce the prediction density

$$p(\mathbf{z}_k|m_k, y_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{z}_{k-1}, m_k)p(\mathbf{z}_{k-1}|y_{1:k-1})d\mathbf{z}_{k-1} \tag{2.19}$$

where $p(\mathbf{z}_k|\mathbf{z}_{k-1}, m_k) = p(\mathbf{z}_k|\mathbf{z}_{k-1}, m_k, m_{k-1})$ since m_{k-1} is completely determined conditioning on $\mathbf{z}_{k-1}$. Second, our interested density $p(\mathbf{z}_k|m_k, y_{1:k})$ can be updated from the prediction density using Bayes' theorem,

$$p(\mathbf{z}_k|m_k, y_{1:k}) = \frac{p(y_k|\mathbf{z}_k)p(\mathbf{z}_k|m_k, y_{1:k-1})}{p(y_k|y_{1:k-1}, m_k)} \tag{2.20}$$

$$= \gamma p(y_k|\mathbf{z}_k)p(\mathbf{z}_k|m_k, y_{1:k-1}) \tag{2.21}$$

where γ is a normalization constant. Thus, Equations (2.19) and (2.20) form a recursive solution to Equation (2.18). In particle filter framework, the aforementioned prediction and update steps are performed by propagating and weighting the random samples.

Prediction Step. In the prediction phase, we generate the predicted particles by

$$\widetilde{\mathbf{z}}_k^i \sim p(\mathbf{z}_k|\mathbf{z}_{k-1}^i, \widetilde{m}_k^i, m_{k-1}^i) \tag{2.22}$$

where $\{\widetilde{m}_k^i\}_{i=1}^N$ is a set of particles representing the estimates of m_k produced by the "oracle". Given the different possible values of m_{k-1}^i and $\widetilde{m}_k^i$, $\widetilde{\mathbf{z}}_k^i$ will be sampled from different models, detailed in Table 2.2. We will then perform sanity check on newly generated particles, where the particles $\widetilde{\mathbf{z}}_k^i$ absent from the reachable set of $\mathbf{z}_{k-1}^i$ will be eliminated.

Update Step. To update, each predicted particle $\widetilde{\mathbf{z}}_k^i$ is assigned with a weight proportional to its likelihood.

$$\widetilde{w}_k^i = p(y_k|\widetilde{\mathbf{z}}_k^i) \tag{2.23}$$

The weight is then normalized by

$$w_k^i = \frac{\widetilde{w}_k^i}{\sum_{i=1}^N \widetilde{w}_k^i} \tag{2.24}$$

We resample N times with replacement from the set $\{\widetilde{\mathbf{z}}_k^i\}_{i=1}^N$ using weights $\{w_k^i\}_{i=1}^N$ to obtain a new set of samples $\{\mathbf{z}_k^i\}_{i=1}^N$ such that $p(\mathbf{z}_k^i = \widetilde{\mathbf{z}}_k^i) = w_k^i$. Correspondingly, the contexts m_k^i's are obtained through the characteristic function, *i.e.*,

$$m_k^i = \mathcal{M}(\mathbf{z}_k^i) \tag{2.25}$$

"Oracle" Design. The oracle is supposed to be able to answer the query about the next possible contexts m_k, based upon which the position/velocity component of the state can be properly propagated according to different transition models. For computational efficiency, we adopt a simple discriminative model to produce $\widetilde{m}_k$'s. Given a small database of WiFi fingerprints, we apply the K-Nearest Neighbors (K-NN) algorithm and a modified distance weighted rule to generate an empirical distribution of the context. To be specific, let the WiFi database be denoted by $\{(m^j, y_w^j)\}_{j=1}^{N_w}$, and N_w is the number of WiFi fingerprints. When the new WiFi observation y_k is querying the possible contexts, the K nearest neighbors of y_k are found among the given training set. Let these K nearest neighbors of y_k, with their associated context, be given by $\{(m^{j'}, y_w^{j'})\}_{j'=1}^K$. In addition, let the corresponding distances of these neighbors from y_k be given by $d^{j'}$, $j' = 1, \cdots, K$. The weight attributed to the j'th nearest neighbor is then defined as

$$\widetilde{q}^{j'} = \frac{d^K - d^{j'}}{d^K - d^1}, \quad j' = 1, \cdots, K \tag{2.26}$$

We then normalize the weights, $q^{j'} = \frac{\widetilde{q}^{j'}}{\sum_{j'=1}^K \widetilde{q}^{j'}}$, and sample the context according to the following discrete probability distribution,

$$P(\widetilde{m}_k = m^{j'}) = \begin{cases} q^{j'}(1-\alpha) + \alpha, & m^{j'} = m_{k-1} \\ q^{j'}(1-\alpha), & m^{j'} \neq m_{k-1} \end{cases} \tag{2.27}$$

where α is a context resilience factor and $\alpha \in [0, 1]$. We incorporate α to accommodate for the prior knowledge that the context will not change too often and to make the "oracle" more robust to the observation noise. Moreover, for the particles on the boundary of distinctive contexts, $\widetilde{m}_k$ is equally probable to be these contexts. The pseudo-code of the CAPF algorithm is provided in Algorithm 1.

Algorithm 1 Context-Augmented Particle Filter

function CAPF($y_{1:T}, wifi\_database, reachable\_set$)
  Initialization:
  Uniformly generate N samples $\{\mathbf{z}_0^i\}_{i=1}^N$
  Set $m_0^i = \mathcal{M}(\mathbf{z}_0^i)$, $w_0^i = N^{-1}$, $i = 1, \cdots, N$
  for $k = 1, \cdots, T$ **do**
    for $i = 1 : N$ **do**
      Context Estimate:
      if $\mathbf{z}_{k-1}^i$ on the boundary of $\{m_b\}_{b=1}^B$ **then**
        Uniformly sample $\widetilde{m}_k^i$ from $\{m_b\}_{b=1}^B$
      else
        Sample $\widetilde{m}_k^i$ from Equation (2.27)
      end if
      Prediction Step:
      $\widetilde{\mathbf{z}}_k^i \sim p(\mathbf{z}_k | \mathbf{z}_{k-1}^i, \widetilde{m}_k^i, m_{k-1}^i)$
      Discard particles $\widetilde{\mathbf{z}}_k^i \notin reachable\_set(\mathbf{z}_{k-1}^i)$
      Update Step:
      Compute weight $\widetilde{w}_k^i = p(y_k | \widetilde{\mathbf{z}}_k^i)$
    end for
    Normalize weights: $w_k^i = \frac{\widetilde{w}_k^i}{\sum_{i=1}^N \widetilde{w}_k^i}$
    Resampling:
    Select N particle indices $i' \in \{1, \cdots, N\}$ according to weights $\{w_k^i\}_{i=1}^N$
    Set $\mathbf{z}_k^i = \widetilde{\mathbf{z}}_k^{i'}$ and $w_k^i = N^{-1}$
    Set $m_k^i = \mathcal{M}(\mathbf{z}_k^i)$
    Estimate:
    $\hat{\mathbf{z}}_k = \sum_{i=1}^N w_k^i \mathbf{z}_k^i$
  end for
  return $\hat{\mathbf{z}}_{1:T}$
end function

2.5 Performance Evaluation

Our experiment was carried out in the Singapore–Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) located in CREATE Tower at the National University

of Singapore campus, which is a typical office environment consisting of cubicles, individual offices, corridors and obstacles like walls, desks, *etc.* The total area of the testbed is around 1000 m^2. There are 10 WiFi routers and four ultrasonic stations deployed in the testbed in total. We utilize TP-LINK TL-WDR4300 Wireless N750 Dual Band Routers as WiFi APs and HC-SR04 Ultrasonic Sensors as the components of ultrasonic stations. The floormap and the corresponding contextual map are shown in Figure 2.5. Different contexts are colored differently in the contextual map. The static space contains the seating areas in the cubicles and offices, where occupants hardly move. The corridors of horizontal and vertical directions are considered to be two types of constrained spaces (HCS and VCS, respectively). The free space includes the open areas where occupants can freely move. We seek to answer the questions including how well MapSentinel is able to track the occupant, and whether the map information exploited by way of MapSentinel can bring additional benefits to the tracking performance.

Experimental methodology. In a real-world setting, we expect the occupant to carry the smartphone as they walk through various sections of an indoor space. Moreover, occupants are unlikely to walk continuously; they would walk between locations of special interest and dwell at certain locations for a significant length of time. Our experiment aims at emulating these practical scenarios in an office environment and incorporating all the contexts defined in our model. Therefore, the following routes were designed as the ground truth for evaluation: (1) A enters the office from the front gate and walks through the corridors to find her colleague (different CSs are included); (2) B enters the office from the side door, walks to her own seat, stays there for a while and exits the office from the front gate (CSs, SS are included); (3) C enters the office from the front gate, walks through corridors, takes some time at her office and goes to the open area (CSs, SS, FS are included). We asked the experimenter to behave as usual when walking in the space. At the same time, the WiFi APs and ultrasonic stations constantly collect the measurements and send them to the central server. To obtain the ground truth at the sampling time of the tracking system, we mark the ground with a 1 m grid on the pre-specified route and ask the experimenter to create lap times with a stopwatch when happening to be on the grid. By recording the starting time of the experiment, we can obtain the time stamp of each grid and then interpolate the ground truth at the sampling time.

Does the "oracle" work? The current context estimation done by the "oracle" is critical to the CAPF algorithm, as the tuple of the current and previous context jointly steer the states in our model. Here, we would like to evaluate the context prediction performance of the "oracle" we constructed in light of several design rules presented in the Section 2.4. Figure 2.6 illustrates the result of the context estimation for different walks. Since the context estimates are represented by a set of particles in the algorithm, we visualize the context estimate by the purple lines centered at the possible contexts, and the lengths of the purple lines are scaled by the proportions of the particles of different contexts. Ideally, the purple

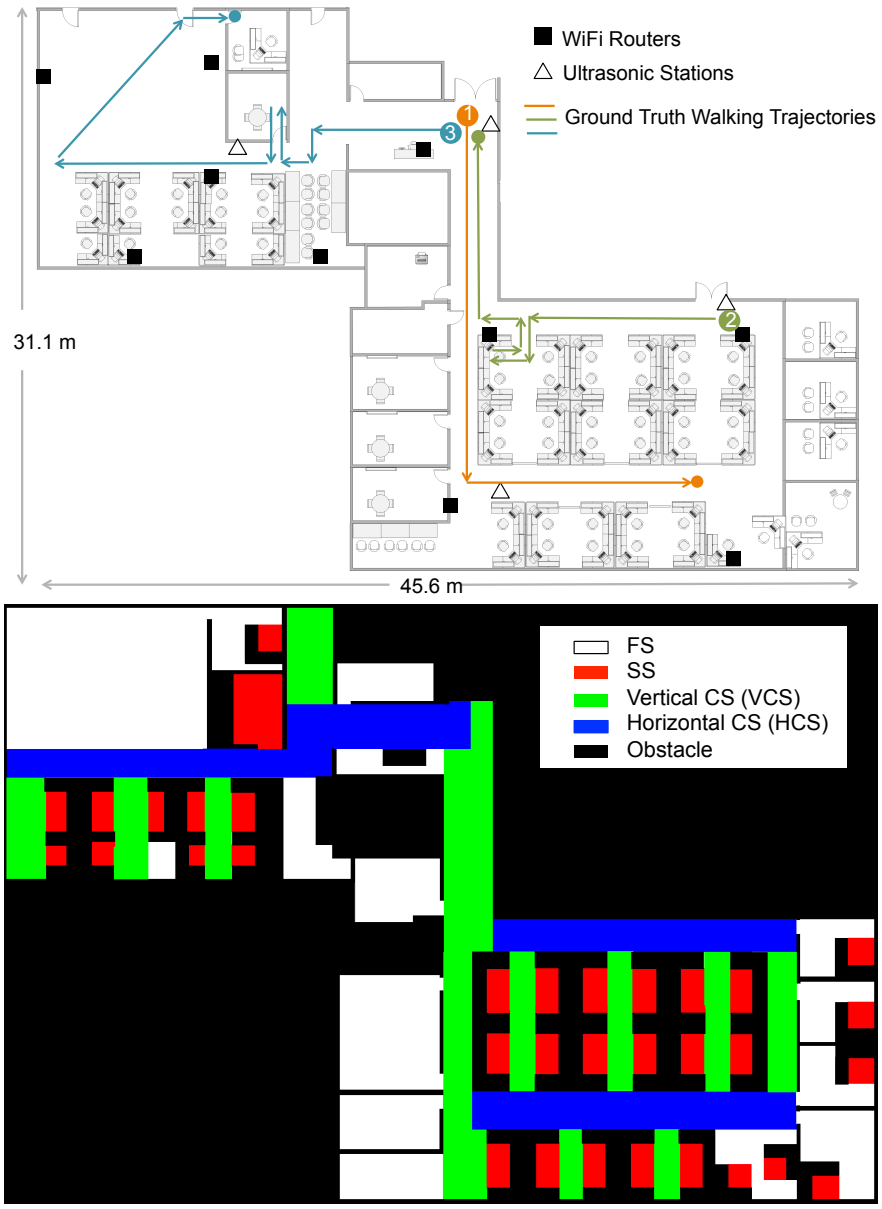


Figure 2.5: The floormap (**top**) and corresponding contextual map (**bottom**) of the testbed. Four different contexts (FS, SS, VCS, HCS) are defined and color coded as illustrated in the legend.

cloud should scatter around the ground truth context. Figure 2.6 suggests that the estimates given by the "oracle" can generally capture the ground truth. Evidently, the context estimate is not perfect, especially for the static space (SS). However, these approximate "ground truths" essentially present other possibilities of the current context and avoids particles trapping in the static space. We define the context estimation accuracy to be the ratio of the number

of particles with correct context estimate to the total number of particles. The context estimation accuracy is calculated for each time step of the experiments, and the empirical distribution of the context estimation accuracy is illustrated in Figure 2.7, where the mean accuracy is 52.41%. With this noisy "oracle", the system can achieve median tracking error of 1.96 m, while the tracking error would be 1.84 m if a perfect "oracle" was utilized. Therefore, our work has the potential to be further improved with a more advanced "oracle" design.

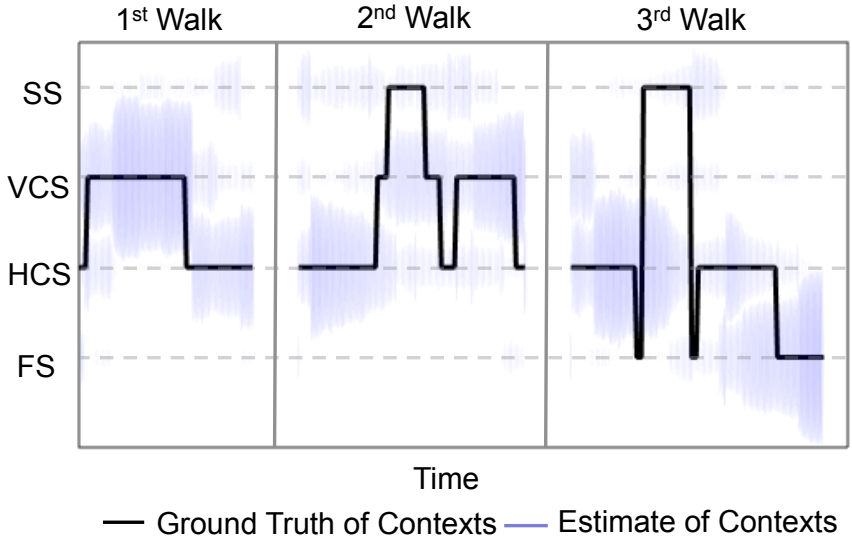


Figure 2.6: The context estimate produced by the "oracle" *versus* the ground truth context. The radius of the purple cloud is proportional to the number of particles of the estimated context which the cloud is centered around.

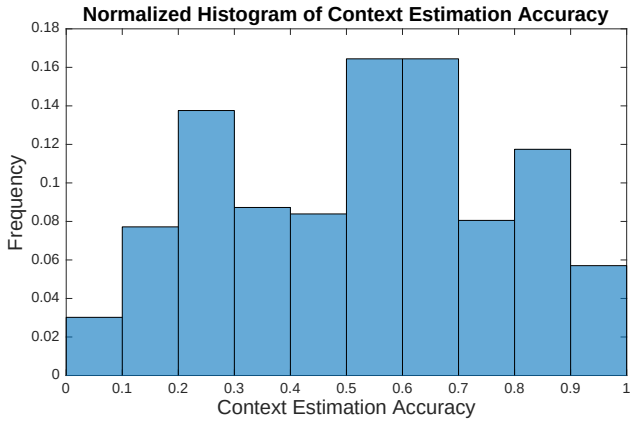
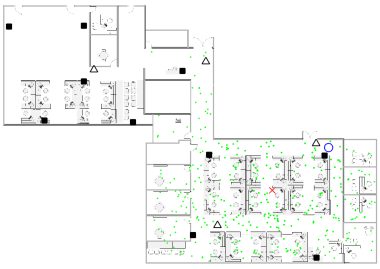


Figure 2.7: Normalized histogram of context estimation accuracy of the "oracle". The mean accuracy is 52.41%.

Figure 2.8 demonstrates some snapshots of the CAPF algorithm in progress. At the beginning, the particles are initialized to be uniformly distributed in the space. In addition, the spread of the particles shrinks as the new WiFi observations come. When the ultrasonic station reports a detection, the particles are concentrated in the corresponding detection zone. As the occupant exits the detection zone, the particles spread out along the direction of the corridor. When the occupant sits in the cubicle, the particles distribute over the seating area as well as some possible routes through which the occupant might leave the seating area. The particles distribute evenly along different directions when the occupant is moving in the free space, in which case our model is identical to the traditional constant velocity dynamic model for the particle filter.

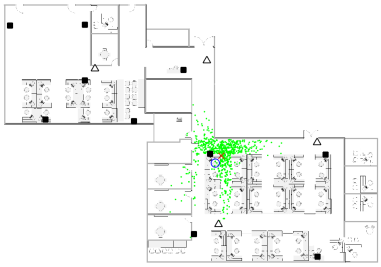


(a) $t = 0$. Initialization.

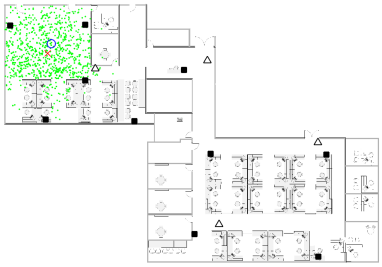
(b) $t = 3$. Particles shrink.

(c) $t = 5$. Passing the ultrasonic station.

(d) $t = 15$. Moving in the constrained space.



(e) $t = 53$. Seating in the static space.



(f) $t = 205$. Moving in the free space.

Figure 2.8: The snapshots of the intermediate steps of the CAPF algorithm visualized. The location estimate, ground truth location, particles are presented by the red cross, blue circle, green dots, respectively. As before, the black square and white triangles give the positions of WiFi routers and ultrasonic stations.

MapSentinel's tracking performance. We aggregate the data from different walks and compare the performance of MapSentinel against the fusion system of WiFi and ultrasonic station without leveraging the floormap information, as well as the purely WiFi-based tracking system. The tracking error distributions are depicted in Figure 2.9. As can be seen, the MapSentinel achieves an essential performance improvement, 31.3% over the WiFi tracking

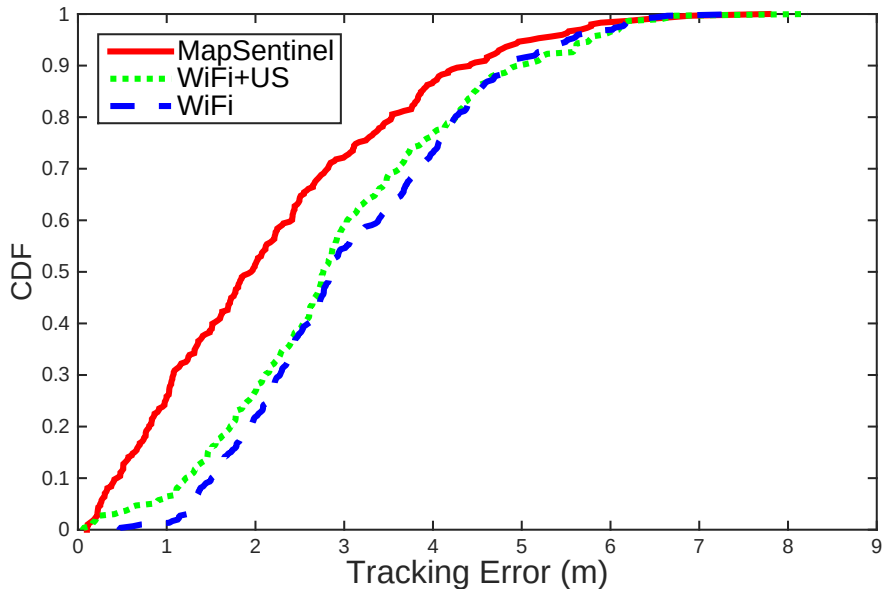


Figure 2.9: Tracking performance of MapSentinel, the fusion system of WiFi and ultrasound sensor, the pure WiFi system. The median tracking accuracy of the MapSentinel is 1.96 m, MapSentinel can achieve the performance improvement of 31.3% over the purely WiFi-based tracking system, 29.1% over the fusion system.

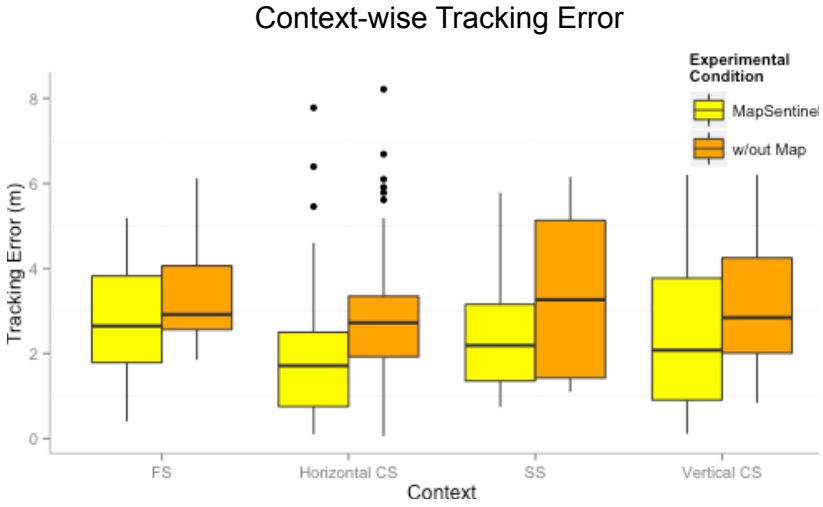


Figure 2.10: Tracking error in different contexts for the MapSentinel and the WiFi+Ultrasound system.

system and 29.1% over the fusion scheme. Note that adding the ultrasonic calibration into the WiFi system is able to realize a small amount of accuracy increment. Due to the high degree of uncertainty of WiFi signals, the effect of ultrasonic calibration will not last for

long. The map information elongates the effect of the ultrasonic calibration via imposing additional constraints to the motion, and that is why MapSentinel greatly enhances the tracking performance compared with the purely WiFi-based system. We also evaluate the tracking performance in different contexts, and the result is shown by boxplots in Figure 2.10. Here, "without map" means using the WiFi and ultrasonic sensing systems without taking into account the reachable set as well as the context-dependent kinematic model. A unified dynamical model, the free space model, is applied in this case, and a traditional particle filter is implemented to estimate the location. As can be readily read from the figure, the MapSentinel performs better in all contexts. More significant increase is achieved in constrained spaces and static spaces, as expected.

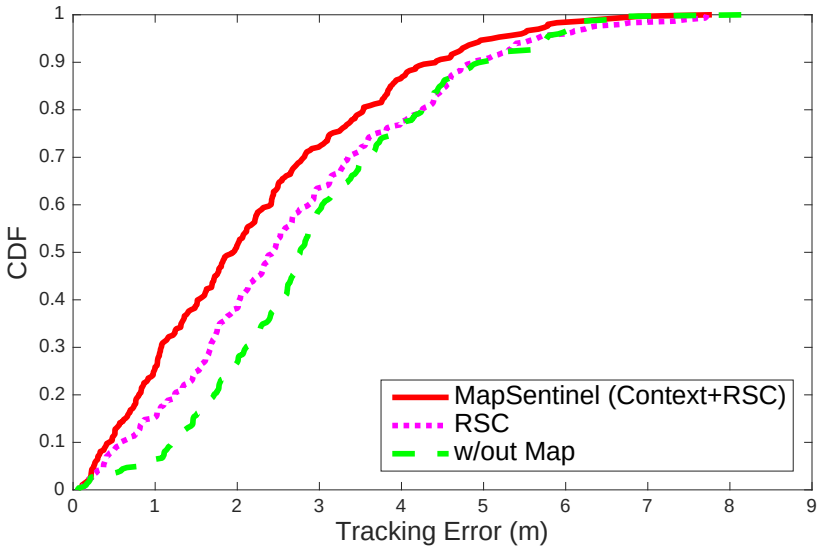


Figure 2.11: Tracking performance of different usage of floormap information. "RSC" stands for reachable set check. MapSentinel extracts the context information from the floormap, and simultaneously eliminates the particles falling outside the reachable set. MapSentinel is compared with the tracking system without using context information (*i.e.*, only performing RSC) and the one without using the map information at all. The median tracking errors of MapSentinel, the system only performing RSC, and the one without exploiting the floormap information are 1.96 m, 2.44 m and 2.77 m, respectively.

Figure 2.11 compares the performance of tracking systems with distinctive floormap usage. MapSentinel exploits the floormap information in two folds: first, MapSentinel integrates the context information into the kinematic model, and the movement patterns of people on different locations of the map are better captured. Secondly, MapSentinel takes into account the speed restrictions as well as physical obstacles in the indoor space by checking if the particles fall inside the reachable set at each time step. The second fold of the floormap information has been widely utilized in the previous work, while the context information is less explored. We therefore compare the tracking error of our system with the one that

merely uses the reachable conditions. Figure 2.11 shows that incorporating information about physical constraints, as the previous work did, is surely beneficial to the tracking system. Particularly, the performance can be further improved by 19.8% by introducing the context information into the tracking system.

To better understand how the map helps improve the location estimation, we demonstrate the velocity estimation of different tracking schemes in Figure 2.12. Typically, the occupants will not perform complex motions in the indoor space due to the constraints of the wall and other barricades. The more the velocity estimate deviates from the canonical directions defined by the indoor environment, the worse the tracking performance can be. Using the fusion schemes of WiFi and ultrasonic calibration, only the location is the observable state. The velocity estimates depend largely on the location estimate and it has little effect in smoothing out the location estimate. Hence, extensive research has been focusing on using inertial measurements to perform dead reckoning, which makes the velocity observable. Analogously, the MapSentinel creates a *virtual* inertial sensor for the occupant, which mimics the actual inertial sensor to provide the possible walking speed and directions. As is shown in Figure 2.12, the velocity estimation without map information tends to point to any direction while the MapSentinel constrains the velocity via the context-dependent kinematic model.

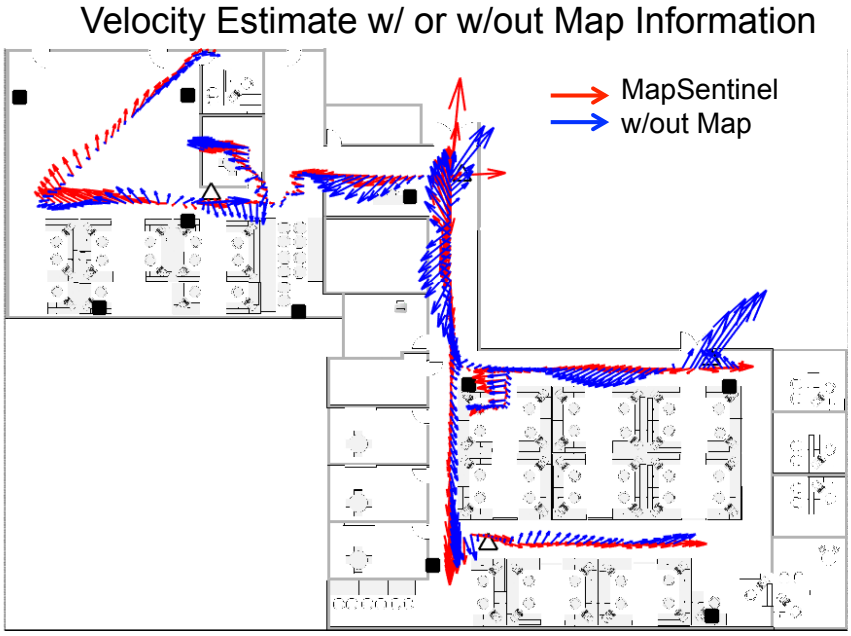


Figure 2.12: The velocity estimation for the MapSentinel and the WiFi+Ultrasound system. The vector indicates the speed and direction of the estimated motion.

2.6 Chapter Summary

In this chapter, we present a Bayesian approach to data fusion and its application to indoor tracking. We demonstrate MapSentinel, a system for real-time location tracking that emphasizes both non-intrusiveness and accuracy. The MapSentinel makes novel attempts to exploit the floormap information by categorizing the indoor space to different contexts to capture the diversity of typical motion characteristics. This mimics having an inertial sensor attached to the occupant to obtain the knowledge of velocity. We formalize the fusion of floormap information as well as the noisy sensor readings using the Factor Graph, and develop the Context-Augmented Particle Filtering algorithm to efficiently solve real-time walking trajectories. Our evaluation in the large typical office environment shows that MapSentinel can achieve the performance improvement of 31.3% over the purely WiFi-based tracking system. MapSentinel is among the early attempts to obviate the need for the inertial sensors in indoor tracking and our results are promising.

For future work, we would like to explore multiple occupants tracking. The ultrasonic sensor is essentially anonymous which cannot identify the occupant entering its detection zone. The WiFi access points are able to identify the occupant from the MAC address of the mobile device and can approximately tell which occupant is approaching the ultrasonic station. Further work to improve the reliability of the identification is necessary. Moreover, we would like to integrate our tracking method to the control of lighting and ventilation systems to improve energy efficiency of buildings. Last but not least, it is crucial to reduce the effort to collect training data [173] and optimize the placement of sensors to improve the tracking accuracy [82].

Chapter 3

Valuing Training Data for Machine Learning Predictions

3.1 Background

As data becomes increasingly commoditized, a natural question is, *how can data contributors who provide different data be remunerated accordingly?* The objective of this chapter is to devise a reasonable and practical way to distribute the total utility generated by a dataset to its contributors. We focus on the Shapley value, a popular definition of value that has been extensively studied since it was first proposed in 1953 [142]. Because the Shapley value is the only definition of value that satisfies *efficiency*, *fairness*, and *additivity* (see Section 3.3), it has been used in applications ranging from economics [66], counter-terrorism [115, 103], environmental science [126] to measuring the importance of features in machine learning [30].

We adopt the Shapley value as our definition of "data value," and study two questions: (1) Does the Shapley value intuitively reflect the value of data? and, (2) If the dataset is used for training a machine learning model, how can we take advantage of the properties of the underlying machine learning algorithm to estimate and calculate the Shapley value efficiently over millions of data points?

The question about how to validate that the Shapley value reflects the value of data looks non-technical; however, as one of the very first works on data valuation, we believe it is important to analyze the Shapley value with several experiments rather than credulously assume that this formalism coincides with intuition. As the first contribution of this paper, we examine various scenarios where we know the relative importance of different data points *a priori* and empirically demonstrate that the Shapley value is consistent with our perception of data value.

Using the Shapley value in the context of data valuation brings in two other challenges and one opportunity. Computing the Shapley value is known to be expensive as it requires evaluating the utility function exponentially many times in N, the number of data points. The technical challenge is how to calculate, or approximate, Shapley values over millions

Table 3.1: Summary of Technical Results. N is the number of data points and C is the number of clusters.

	Assumption	Technique	Complexity	Approximation
Existing	Bounded utility	Sampling	$\mathcal{O}(N \log(N))$ incremental model evals $\times$ computation per incremental model eval	(ϵ, δ)
Novel & practical	KNN classifier	Dynamic prog.	$\mathcal{O}(N \log(N))$ computation	Exact
	Smooth utility	Influence function	$\mathcal{O}(N)$ opt. routines $\times$ computation per opt. routine	Heuristic
Novel & theoretically interesting & less practical	Bounded utility	Group testing	$\mathcal{O}(\log(N)^2)$ model evals $\times$ computation per model eval	(ϵ, δ)
	Stable algorithm	Uniform division	$\mathcal{O}(1)$ computation	ϵ
	Lipschitz Shapley value	Clustering	$\mathcal{O}(C \log(C))$ model evals $\times$ computation per model eval	ϵ

or even billions of data points, a scale that is rare in previous applications of the Shapley value, but not uncommon for real-world data valuation tasks. On the other hand, the data valuation application brings in a special opportunity to accommodate the scalability challenge as the underlying machine learning algorithm could potentially provide useful information for efficient Shapley value calculation.

Table 3.1 summarizes the technical results of this chapter, which can be organized into two categories: those that are practical but assume certain properties on the utility or machine learning algorithm, and those that are highly general, theoretically interesting, but relatively slow.

- **Exact Calculation for K-Nearest Neighbor.** Our most surprising result is that if the data is used for training a KNN classifier and the utility function is the probability of the correct test label assignment, the Shapley value can be—exactly—calculated in $\mathcal{O}(N \log N)$ time. The underlying computation is no more expensive than a single pass of sorting training data points according to their distance with testing data points. In practice, this allows us to calculate the exact Shapley values on one million data points in seconds on a single machine.

- **Probabilistic Approximation for Bounded Utilities.** In the most general case we only make a bounded utility assumption. Given N data points, we develop a novel algorithm based on group testing [46] that only requires $\mathcal{O}(\log(N)^2)$ utility evaluations. The idea is to share the information we get from a single utility evaluation across all data points, as opposed to treating different data points independently as in existing approaches. When the utility function cannot be efficiently incrementally evaluated for a given a new data point and the number of training data points is large, our group testing-based approach requires significantly fewer model evaluations than the state-of-the-art sampling-based approach [109].

- **Using the Influence Function Heuristic.** For a smooth utility function one can use influence functions [95] as a proxy to the Shapley value by assuming (1) the value of a data point depends only on its marginal contribution to the data subset that contains all other points, and (2) the influence function is a good *local* approximation to the change of the utility caused by adding one more training data point. Whether this heuristic is acceptable is application-specific. Influence functions can be very efficient, although utilizing this heuristic violates property of the Shapley value.

- **Uniform Value Division for Stable Algorithms.** For stable learning algorithms, such as many norm regularized models [21], the model only changes slightly when the training data is changed slightly. Intuitively, stable algorithms are not sensitive to individual training points and therefore all training points should have very similar values. Our theoretical results show that uniform data value division is a fairly good approximation to the Shapley value for stable learning algorithms.

- **Clustering for Lipschitz Shapley Values.** If the Shapley value is Lipschitz continuous, i.e., close data points have close Shapley values, then we can cluster data points and only compute the values for cluster representatives. We show a class of machine learning models that lead to Lipschitz Shapley values and can exploit clustering to improve the efficiency for computing Shapley values.

Each of these techniques is novel to the best of our knowledge. The applicability of these techniques is context-dependent. When the underlying machine learning model is a KNN classifier, we provide an extremely efficient way of computing the exact Shapley value. As we will validate empirically, even when the machine learning model is not a KNN classifier, the KNN Shapley value can still correlate with the real Shapley value, so much so that it can serve as a default estimate for real-world applications. Separately, the group testing-based approach can be used either to compute the ground truth for data valuation benchmarks or in scenarios where respecting the exact utility function is crucial.

3.2 Related Work

Originated from game theory, the Shapley value, in its most general form, can be #P-complete to compute [39]. Efficiently estimating Shapley value has been studied extensively for decades. For bounded utility functions, Maleki et al. [109] described a sampling-based approach that requires $O(N \log N)$ samples to achieve a desired approximation error. By taking into account special properties of the utility function, one can derive more efficient approximation algorithms. For instance, Fatima et al. [56] proposed a probabilistic approximation algorithm with $O(N)$ complexity for weighted voting games. The structure of utility in data valuation is different from these applications. In this paper we focus on developing novel and more efficient algorithms for data valuation; notably, the group testing-based algorithm can also be applied to general games.

Using the Shapley value in the context of machine learning is not new. For instance, the Shapley value has been applied to feature selection [30, 150, 117, 140, 106]. While their contributions have inspired this paper, many assumptions made for feature "valuation" do not hold in the case of data valuation. As we will see, by studying the Shapley value tailored to *data valuation*, we can develop novel algorithms that are more efficient than the previous approaches [109].

Despite not being used for data valuation, ranking the importance of training data points has been used for understanding model behaviors, detecting dataset errors, reducing training complexity, etc. Existing methods include using the influence function [95] for smooth parametric models and a variant [143] for non-parametric ones. Ogawa et al. [122] proposed rules to identify and remove the least influential data when training support vector machines (SVM) to reduce the computation cost. One can also construct coresets, weighted data subsets, such that models trained on these coresets are provably competitive with models trained on the full dataset [37]. These approaches could potentially be used for data valuation; however, it is not clear whether they satisfy the efficiency, fairness, and additivity properties of the Shapely value. We leave it as future work to understand these distinct approaches for data valuation.

3.3 Problem Formulation

We focus on the scenario in which each user contributes to a single data instance to the dataset. Consider a dataset $D = \{z_i\}_{i=1}^N$ containing data from N users. Let $U(S)$ be the utility function, representing the value created by the additive aggregation of $\{z_i\}_{i\in S}$ and $S \subseteq I = \{1, \cdots, N\}$. Without loss of generality, we assume throughout that $U(\emptyset) = 0$. Our goal is to partition $U_{\text{tot}} \triangleq U(I)$, the value created by the entire dataset, to the individual users. Data value attribution can be formally defined as a function that assigns to user i for a given utility U, a number $s(U, i)$. We suppress the dependency on U when the utility is self-evident and use s_i to represent the value allocated to user i.

The Shapley value [142] is a popular definition of "value". Given a utility function $U(\cdot)$, the Shapley value for user i is defined as the average marginal contribution of z_i to all possible subsets of $D = \{z_i\}_{i\in I}$ formed by other users:

$$s_i = \sum_{S\subseteq I\setminus\{i\}} \frac{1}{N\binom{N-1}{|S|}} \big[U(S\cup\{i\}) - U(S)\big] \tag{3.1}$$

The above definition can be stated in the equivalent form:

$$s_i = \frac{1}{N!} \sum_{\pi\in\Pi(D)} \big[U(P_i^\pi \cup \{i\}) - U(P_i^\pi)\big] \tag{3.2}$$

where $\pi \in \Pi(D)$ is a permutation of users and P_i^π is the set of users which precede user i in π. Intuitively, imagine all users' data are to be collected in a random order, and that every

user i receives his or her data's marginal contribution that would bring to those whose data are already collected. If we average these contributions over all the possible orders of users, we obtain s_i. The Shapley value *uniquely* satisfies the following properties which remain sensible in the context of data valuation.

1. **Efficiency**: The value of the entire dataset is completely distributed among all users, i.e., $U(I) = \sum_{i \in I} s_i$.

2. **Fairness**: (1) Two users who are identical with respect to what they contribute to a dataset's utility should have the same value. That is, if user i and j are equivalent in the sense that $U(S \cup \{i\}) = U(S \cup \{j\}), \forall S \subseteq I \setminus \{i, j\}$, then $s_i = s_j$. (2) Users with zero marginal contributions to all subsets of the dataset receive zero payoff, i.e., $s_i = 0$ if $U(S \cup \{i\}) = 0$ for all $S \subseteq I \setminus \{i\}$.

3. **Additivity**: The values under multiple utilities sum up to the value under a utility that is the sum of all these utilities: $s(U, i) + s(V, i) = s(U + V, i)$ for $i \in I$.

We propose to use the Shapley value for data value attribution, not only because it uniquely possesses these desirable properties, but also because of its high degree of flexibility. Depending on the particular application, different utility functions can be developed and the Shapley value then serves as a unified scheme to attribute value.

The challenge in adopting Shapley value lies in its computational cost. Evaluating the exact Shapley value using (3.1) involves computing the marginal utility of every user to every subset, which is $\mathcal{O}(2^N)$. Even worse, in many machine learning tasks, evaluating utility such as testing accuracy *per se* is computationally expensive as it requires training numerous machine learning models.

Baseline: Permutation Sampling. As a baseline solution, we describe an existing sampling algorithm [108] that can produce an (ϵ, δ)-approximation for any bounded utility functions. We say that $\hat{s} \in \mathbb{R}^N$ is a (ϵ, δ)-approximation to the true Shapley value $s = [s_1, \ldots, s_N]^T \in \mathbb{R}^N$ if $P[\max_i |\hat{s}_i - s_i| \leq \epsilon] \geq 1 - \delta$. This baseline approach samples user permutations to calculate the marginal contribution to the permutation-specific subset for every user. The Shapley value is then approximated as the average marginal contribution over all sampled permutations.

According to the definition of Shapley value in (3.1) and the law of large numbers, the average marginal contribution of each user will converge to the true Shapley value if the number of permutations is large enough. An application of Hoeffding's bound indicates that the number of permutations needed to achieve an (ϵ, δ)-approximation is $m_{\text{s,b}} = (2r^2/\epsilon^2) \log(2N/\delta)$, where r is the range of the utility function. For each permutation the utility function is evaluated N times in order to compute the marginal contribution for all N users; therefore, the number of utility evaluations involved in the baseline approach is $m_{\text{u,b}} \triangleq N m_{\text{s,b}} \sim \mathcal{O}(N \log N)$.

3.4 Calculating Exact Shapley Values for KNN

In this section, we show that if the utility function is the probability of a correct label assignment for a KNN classifier (for 1-NN it is equivalent to the test accuracy), we can calculate the exact Shapley value with a complexity of only $\mathcal{O}(N \log N)$.

KNN is a simple, popular non-parametric classifier and can achieve the state-of-the-art performance using deep features [29]. Given a single testing point x_{test} with the label y_{test}, the simplest, unweighted version of a KNN classifier first finds the top-K training points $(x_{\alpha_1}, \cdots, x_{\alpha_K})$ that are most similar to x_{test} and outputs the probability of x_{test} taking the label y_{test} as $P[x_{\text{test}} \to y_{\text{test}}] = \frac{1}{K}\sum_{i=1}^{K} \mathbb{I}[y_{\alpha_i} = y_{\text{test}}]$. We assume that the confidence of predicting the right label is used as the utility function, i.e., $U(S) = \frac{1}{K}\sum_{\alpha_i \in S, i=1\cdots,K} \mathbb{I}[y_{\alpha_i} = y_{\text{test}}]$. Using this utility, we have a simple but novel algorithm that returns the *exact* Shapley value.

Theorem 1. *Let x_{α_i}, $i = 1, \ldots, N$, be the training point that is ith closest to a given test point x_{test}. If KNN ($K < N$) is used as the classifier and the utility is*

$$U(S) = \frac{1}{K} \sum_{\alpha_i \in S, i=1,\cdots,\min\{|S|,K\}} \mathbb{I}[y_{\alpha_i} = y_{test}], \tag{3.3}$$

then the Shapley value of each training point can be calculated recursively as follows:

$$s_{\alpha_N} = \frac{\mathbb{I}[y_{\alpha_N} = y_{test}]}{N} \tag{3.4}$$

$$s_{\alpha_i} = s_{\alpha_{i+1}} + \frac{\mathbb{I}[y_{\alpha_i} = y_{test}] - \mathbb{I}[y_{\alpha_{i+1}} = y_{test}]}{K} \frac{(\min(K-1, i-1)+1)}{i}. \tag{3.5}$$

Intuition of Proof for $K = 1$. Given an ordered list of training data points and one test data point, we analyze the difference in utility between two *neighboring* training data points (Figure 3.1). Consider Case 1, where two data points i and j have different labels $y_i \neq y_j$. If there is no data point in $S \subseteq I \setminus \{i, j\}$ that ranks higher (is more similar) than i and j (Case 1.1) then the utility difference between i and j will be ± 1, because i or j will be used by the 1-NN algorithm to make a prediction. However, if the subset S contains a point l ranked lower than i or j (Case 1.2) then the utility difference will be 0, because neither i nor j, but only l, will be used for prediction, i.e., $U(S \cup \{i\}) = U(S) = U(S \cup \{j\})$. As we can see, if i and j have identical labels (Case 2), the utility difference is 0 for any S. Therefore, we can simply calculate the Shapley value difference between i and j by counting how many subsets S fall into Case 1.1.

We now have an extremely efficient, but exact way of computing the Shapley value when a KNN classifier is used as the underlying machine learning model. The computational complexity is only $\mathcal{O}(N \log(N) N_t)$ for N training data points and N_t test data points–this is simply to sort N_t arrays of N numbers! The corresponding pseudo-code is provided in Algorithm 2. Note that the KNN Shapley value result for a single test point utility in

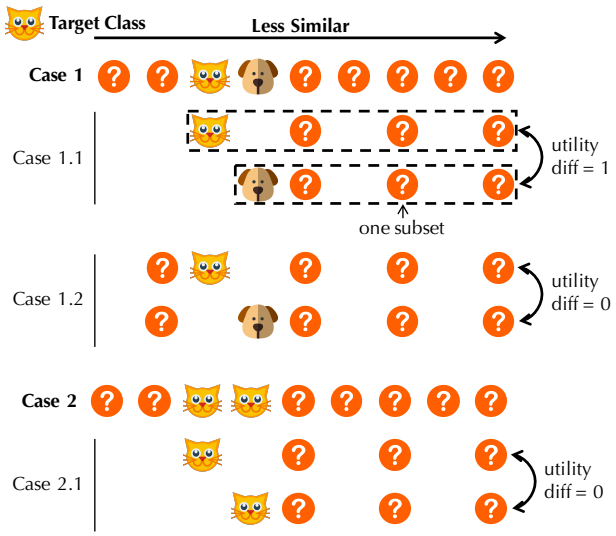


Figure 3.1: Illustration of the proof idea for 1NN.

Theorem 1 can be readily extended to the multiple testing data utility by averaging the Shapley value obtained from the utility for each test point.

Algorithm 2 KNN-based Shapley value calculation

Input: $D = \{(x_i, y_i)\}_{i=1}^N$ - training set, $(x_{\text{test}}, y_{\text{test}})$ - test data, $d(\cdot, \cdot)$ - distance metric
Output: $\{s_i\}_{i=1}^N$

$(\alpha_1, ..., \alpha_N) \leftarrow$ Indices of data in an ascending order using $d(\cdot, x_{\text{test}})$
$s_{\alpha_N} \leftarrow \frac{\mathbb{I}[y_{\alpha_N} = y_{\text{test}}]}{N}$
for $k = N - 1$ to 1 **do**
    $s_{\alpha_k} \leftarrow s_{\alpha_{k+1}} + \frac{\mathbb{I}[y_{\alpha_k} = y_{\text{test}}] - \mathbb{I}[y_{\alpha_{k+1}} = y_{\text{test}}]}{K} \frac{\min(K-1, k-1)+1}{k}$
end for

Use the KNN Shapley Value for Other Classifiers. The above algorithm is possible only because of the property of KNN classifiers. However, given many previous empirical results showing that a KNN classifier can often achieve a classification accuracy that is comparable with classifiers such as SVMs and logistic regression given sufficient memory, we wonder: *Can we use the KNN Shapley value as a proxy for other classifiers?* As we will see in an upcoming experiment involving the `iris` dataset, the Shapley value with a KNN classifier is correlated with the Shapley value with a logistic regression classifier. The only caveat is that KNN Shapley value does not distinguish between neighboring data points that have the same label. If this caveat is acceptable, we believe that the KNN Shapley value provides an efficient and default way of data valuation.

Use Case of KNN Shapley Value. The most restrictive assumption we made behind the KNN Shapley value is that the features associated with each data point are already informative enough to allow a KNN classifier to perform well. This assumption does not appear to be restrictive with the prevalence of deep neural networks and representation learning. In this paper, when we need to extract features for images, we use a deep learning model pre-trained on ImageNet as the default feature extractor, a strategy that has shown to be effective across a range of natural image classification tasks [60, 135]. We leave the study of the interaction between data valuation and representation learning as future work.

3.5 Efficiently Approximating the Shapley Value

In this section, we introduce an influence function-based heuristic, which can be used to efficiently evaluate the marginal contribution of a training point for differentiable learning loss functions, and the group testing-based approach, which is designed for approximating the Shapley value for more general utility functions and requires at most $\mathcal{O}((\log N)^2)$ utility evaluations to achieve a desired approximation error. Further, we show that for stable learning algorithms, the values of different training points are quite similar and uniform division of the total utility produces a good approximation to the true Shapley value. We close the section by discussing how to leverage the continuity assumptions on the the Shapley value for more efficient estimation.

Influence Function and Stratified Sampling

Computing the Shapley value involves evaluating the change in utility of all possible sets of data points after adding one more point. A plain way to evaluate the difference requires training a large number of models on different subsets of data. Koh et al. [95] show that influence functions can be used as an efficient approximation of parameter changes after adding or removing one point. Therefore, the need for re-training models is circumvented. Assume that model parameters are obtained by solving an empirical risk minimization problem $\hat{\theta}^m = \text{argmin}_\theta \frac{1}{m}\sum_{i=1}^m l(z_i, \theta)$. Applying the result in [95], we can approximate the parameters learned after adding z by using the relation $\hat{\theta}_z^{m+1} = \hat{\theta}^m - \frac{1}{m} H_{\hat{\theta}^m}^{-1} \nabla_\theta L(z, \hat{\theta}^m)$ where $H_{\hat{\theta}^m} = \frac{1}{m}\sum_{i=1}^m \nabla_\theta^2 L(z_i, \hat{\theta}^m)$ is the Hessian. The parameter change after removing z can be approximated similarly, except replacing the $-$ by $+$ in the above formula. The efficiency of the baseline permutation sampling method can be significantly improved by combining it with influence functions. Moreover, we can employ a more sophisticated sampling scheme to reduce the variance of the result. Indeed, we can re-write the Shapley value as $s_i = \frac{1}{N}\sum_{k=1}^N \mathbb{E}[X_i^k]$, where $X_i^k = U(S \cup \{i\}) - U(S)$ is the marginal contribution of user i to a size k subset that is randomly selected with probability $1/\binom{N-1}{k}$. This suggests that stratified sampling can be used to approximate the Shapley value, which customizes the number of samples for estimating each expectation term according to the variance of X_i^k.

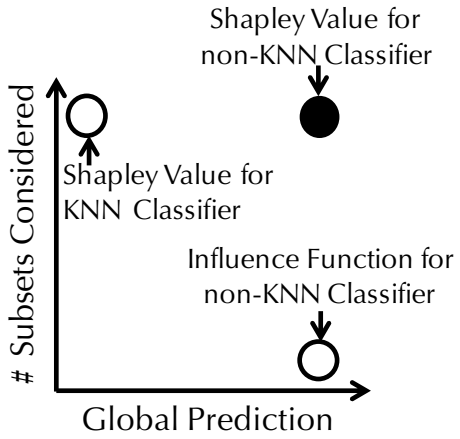


Figure 3.2: Comparison of the KNN Shapley value and largest-S influence.

Largest-S Approximation. One practical heuristic of using influence functions is to consider a single subset S for computing s_i, namely, $I \setminus \{i\}$. With this heuristic, we can simply take a trained model on the whole dataset, and calculate the influence function for each data point. For logistic regression models, the first and second derivations enjoy closed-form expressions and the change in parameters after removing one point $z = (x, y)$ can be approximated by $-\big(\sum_{i=1}^{N} \sigma(x_i^T \hat{\theta}^N)\sigma(-x_i^T \hat{\theta}^N) x_i x_i^T\big)^{-1} \sigma(-y x_i^T \hat{\theta}^N) yx$ where $\sigma(u) = 1/(1+\exp(-u))$ and $y \in \{-1, 1\}$. In the experimental section, we empirically validate this approach. On the `iris` dataset we show that this heuristic can produce a value that is correlated with the true Shapley value.

KNN Shapley Value vs. Largest-S Influence on a non-KNN Classifier. When applied to a non-KNN Classifier, e.g., logistic regression, both KNN Shapley value and Largest-S Influence can be seen as approximations to the true Shapley value, in two orthogonal dimensions: By using a KNN as the underlying classifier, prediction needs to be *local*, using only the top-K data points. By using largest-S influence, it only considers it uses only a single subset rather than exponentially many subsets of the whole dataset. For logistic regression, the prediction is more *global* than the KNN and computing the true Shapley value requires the consideration of exponentially many subsets. Moreover, the fact that largest-S influence only considers a single subset makes it impossible to satisfy the *efficiency* and *additivity* properties simultaneously.

Theorem 2. *Consider the value attribution scheme that assign the value $\hat{s}(U, i) = C_U[U(S \cup \{i\}) - U(S)]$ to user i where $|S| = N - 1$ and C_U is a constant such that $\sum_{i=1}^{N} \hat{s}(U, i) = U(I)$. Consider two utility functions $U(\cdot)$ and $V(\cdot)$. Then, $\hat{s}(U + V, i) \neq \hat{s}(U, i) + \hat{s}(V, i)$ unless $V(I)[\sum_{i=1}^{N} U(S \cup \{i\}) - U(S)] = U(I)[\sum_{i=1}^{N} V(S \cup \{i\}) - V(S)]$.*

As a result, when dealing with non-KNN classifiers, we believe that the decision of whether we should use KNN Shapley value or largest-S influence depends on application's requirements.

Group Testing-based approach

The KNN-based approach assumes particular property of the utility function. For general utility functions, an exact computation of the Shapley value is unavoidably slower. We now describe an algorithm that only assumes a bounded utility function and can significantly outperform the baseline in terms of the number of utility evaluations needed.

One can derive the following formula of Shapley value difference between any pair of points from the definition of Shapley value.

Lemma 3. *For any $i, j \in I$ and $i \neq j$, the difference in Shapley values between i and j is*

$$s_i - s_j = \frac{1}{N-1} \sum_{S \subseteq I \setminus \{i,j\}} \frac{1}{\binom{N-2}{|S|}} \big[U(S \cup \{i\}) - U(S \cup \{j\})\big] \tag{3.6}$$

Our proposed approximation algorithm is inspired by the theory of group testing. Recall the group testing is a combinatorial search paradigm [46], in which one wants to determine whether each item in a set is "good" or "defective" by performing a sequence of tests. The result of a test may be positive, indicating that at least one of the items of that subset is defective, or negative, indicating that all items in that subset are good. Each test is performed on a pool of different items and the number of tests can be made significantly smaller than the number of items by smartly distributing items into pools. Hence, group testing is particularly useful when testing an individual item's quality is expensive. Analogously, we can think of Shapley value calculation as a group testing problem with continuous quality measure. Each user's data is an "item" and Shapley value corresponds to the quality of items. Evaluating Shapley value of each user's data is expensive; nevertheless, we hope to recover it from a small amount of customized tests.

Each "test" in our scenario corresponds to evaluating the utility of a subset of users. Let T be the total number of tests. At test t, a random set S_t of indices is drawn from I and we evaluate the utility of the selected set of data, i.e., $B_t = U(S_t)$. Let $B = (B_1, \cdots, B_T) \in \mathbb{R}^T$ We encode the collection of T test with a binary matrix $A = (a_{ti}) \in \mathbb{R}^{T \times N}$, where $a_{ti} = 1$ indicates that user i's data is used in test t. Let a_i denote the ith column of A.

If we model the appearance of user i and j's data in a test as Boolean random variables β_i and β_j, respectively, then the difference the utility of user i and that of user j is

$$\mathbb{E}[(\beta_i - \beta_j) U(\beta_1, \cdots, \beta_N)] \tag{3.7}$$

where $U(\beta_1, \cdots, \beta_N)$ is the utility evaluated on the data with the corresponding Boolean appearance random variable equal to 1. The key idea of the proposed algorithm is to smartly design the sampling distribution of $\beta_1, \cdots, \beta_N$ such that (3.7) tallies with the Shapley

difference in (3.6); as a result, we can calculate Shapely differences from the test results with high-probability bound on the error. Take any user i and estimate the corresponding Shapley value using the baseline sampling approach. Then, the Shapley value of all other $N-1$ users can be estimated using the difference of the Shapley value with user i. Algorithm 3 presents the pseudo-code for the group testing-based algorithm.

Algorithm 3 Group-Testing-Based Shapley Value approximation

Input: $D = \{z_i\}_{i=1}^N$ - training dataset, $U(\cdot) \in [0, r]$ - utility function, ϵ, δ - approximation error parameters, T - number of tests

Output: $(\hat{s}_1, ...\hat{s}_N)$ - estimated Shapley values

Variables: Z - Normalization constant, ϵ - error bound, $A \in \{0,1\}^{T\times N}$ - activation matrix, $B \in [0, r]^T$ - utility matrix, C_{ij} - estimated Shapley value difference between i and j where $i, j \in \{1, \cdots, N\}$

$U_{tot} \leftarrow U(D)$
$Z \leftarrow 2\sum_{k=1}^{N-1} \frac{1}{k}$
$q(k) \leftarrow \frac{1}{k}(\frac{1}{k} + \frac{1}{N-k})$ for $k = 1, \cdots, N-1$
$q_{tot} \leftarrow \frac{N-2}{N} q(1) + \sum_{k=2}^{N-1} q(k)\big[1 + \frac{2k(k-N)}{N(N-1)}\big]$
$T \leftarrow \frac{4}{(1-q_{tot}^2)h\big(\frac{2\epsilon}{ZrC_\epsilon(1-q_{tot}^2)}\big)} \log\frac{C_\delta(N-1)}{2\delta}$
Initialize $(a)_{ti} \leftarrow 0$, $t = 1, ..., T, i = 1, ..., N$
for $t = 1$ to T **do**
    Draw $k_t \sim q(k)$
    for $j = 1$ to k_t **do**
        Uniformly sample a length-k_t sequence I_j from $\{1, \cdots, N\}$
        $a_{ti} \leftarrow 1$ for all $i \in I_j$
    end for
    $B_t \leftarrow U(\{i : a_{ti} = 1\})$
end for
$C_{ij} \leftarrow \frac{Z}{T}\sum_{t=1}^{T} B_t(A_{ti} - A_{tj})$ for $i = 1, .., N$, $j = 1, ..., N$ and $j \geq i$
Recovering the Shapley value from the estimated Shapley differences C_{ij}

The following theorem provides the lower bound on the number of tests T needed to achieve an (ϵ, δ)-approximation.

Theorem 4. *The group testing-based approach returns an (ϵ, δ)-approximation to the Shapley value if the number of tests T satisfies $T \geq 4\log\frac{C_\delta(N-1)}{2\delta}/\big((1-q_{tot}^2)h\big(\frac{2\epsilon}{ZrC_\epsilon(1-q_{tot}^2)}\big)\big)$, where $C_\epsilon, C_\delta > 1$, $q_{tot} = \frac{N-2}{N}q(1) + \sum_{k=2}^{N-1} q(k)[1 + \frac{2k(k-N)}{N(N-1)}]$, $h(u) = (1+u)\log(1+u) - u$, $Z = 2\sum_{k=1}^{N-1}\frac{1}{k}$, and r is the range of the utility function.*

To recover the Shapley value of individual data points from the estimated difference between two data points, we first take an arbitrary data point x_*, which we call a "baseline

point", and estimate its Shapley value s_* up to $(\frac{(C_\epsilon-1)\epsilon}{C_\epsilon}, \frac{(C_\delta-1)\delta}{C_\delta})$ (where $C_\epsilon, C_\delta > 1$) using the standard permutation sampling approach. Then, for each data point $x_i \in D \setminus \{x_*\}$, estimate the Shapley difference $(s_i - s_*)$ up to $(\frac{\epsilon}{C_\epsilon}, \frac{\delta}{C_\delta})$. Then, the Shapley value of data point x_i, i.e., s_i, can be estimated as $s_* + (s_i - s_*)$. With a $(\frac{(C_\epsilon-1)\epsilon}{C_\epsilon}, \frac{(C_\delta-1)\delta}{C_\delta})$-approximation for s_* and a $(\frac{\epsilon}{C_\epsilon}, \frac{\delta}{C_\delta})$-approximation for $(s_i - s_*)$, we achieve an (ϵ, δ)-approximation for s_i. The number of model evaluations needed for estimating $N-1$ Shapley differences up to $M_1 = (\frac{\epsilon}{C_\epsilon}, \frac{\delta}{C_\delta})$ is $4/(1-q_{tot}^2)/h\big(\frac{2\epsilon}{ZrC_\epsilon(1-q_{tot}^2)}\big) \log \frac{C_\delta(N-1)}{2\delta}$. The number of model evaluations for estimating the baseline point up to $(\frac{(C_\epsilon-1)\epsilon}{C_\epsilon}, \frac{(C_\delta-1)\delta}{C_\delta})$ is $M_2 = \frac{4r^2C_\epsilon^2}{(C_\epsilon-1)^2\epsilon^2} \log \frac{2C_\delta}{(C_\delta-1)\delta}$. C_ϵ, C_δ are chosen so that $M_1 + M_2$ are minimized.

Comparison with Baseline. Note that $Z = 2\sum_{k=1}^{N-1} \frac{1}{k} \leq 2(\log(N-1)+1)$ and $1/h(1/Z) \leq 1/\log(1+1/Z) \leq Z+1$. Since only one utility evaluation is required for a single test, the number of utility evaluations is at most $\mathcal{O}((\log N)^2)$. On the other hand, in the baseline approach, the number of utility evaluations is $\mathcal{O}(N \log N)$. Therefore, when the utility function is a complete *black box*, group testing-based algorithm requires significantly fewer utility evaluations. One caveat in the comparison is when the utility function can be *incrementally* evaluated: After training a machine learning model on S, training it on $S \cup \{i\}$ might be cheaper. In this scenario, the baseline approach could be faster. As a rule-of-thumb, if the utility function cannot be maintained at the cost of $\mathcal{O}(\log N)$, the group testing-based algorithm outperforms the baseline approach.

Remark. *Is the above algorithm practical?* The answer to this question depends on the application, the time budget, and how important it is to respect the exact non-KNN utility. The more general algorithm is significantly more expensive than the KNN-based approach. Fiven *a fixed time budget*, it is possible that the KNN-based approach can achieve a better result even for a non-KNN classifier. Nevertheless, the above algorithm is novel to our best knowledge and we believe that it is interesting from a theoretical perspective and is potentially valuable for applications beyond data valuation.

Stable Learning Algorithms

A learning algorithm is *stable* if the model learned by the algorithm is insensitive to the removal of an arbitrary point in the training dataset [21]. More specifically, an algorithm G has uniform stability γ with respect to the loss function l if $\|l(G(S), \cdot) - l(G(S^{\setminus i}), \cdot)\|_\infty \leq \gamma$ for all $i \in \{1, \cdots, |S|\}$, where S denotes the training set and $S^{\setminus i}$ denotes the one by removing ith element of S. Indeed, a broad variety of learning algorithms are stable, including all learning algorithms with Tikhonov regularization. Stable learning algorithms are appealing as they enjoy provable generalization error bounds [21]. Assume that the model is trained via a stable learning algorithm and training data's utility is measured in terms of the testing loss. Due to the inherent insensitivity of a stable learning algorithm to the training data, we

expect that the Shapley value of each training point is similar to one another. The following theorem confirms our intuition and provides an upper bound on the Shapley value difference between any pair of training data points.

Theorem 5. *For a learning algorithm $G(\cdot)$ with uniform stability $\beta = \frac{C}{N}$, where N is the number of data points and C is some constant. Let the utility of D be $U(D) = M - L_e(G(D), D_{test})$ where $L_e(G(D), D_{test}) = \frac{1}{N}\sum_{i=1}^{N} l(G(D), z_{test,i})$ and $0 \leq l(\cdot,\cdot) \leq M$. Then, $s_i - s_j \leq (2C + M)\frac{1+\log(N-1)}{N-1}$ and the Shapley difference vanishes as $N \to \infty$.*

Therefore, if $(2C+M)\frac{1+\log(N-1)}{N-1}$ is less than the desirable approximation error ϵ, uniformly assigning $\frac{U_{\text{tot}}}{N}$ to each data contributor provides an ϵ-approximation to the Shapley value.

Continuity Assumptions and Clustering

If similar data points have similar Shapley values, we could cluster data points and only compute the values for cluster representatives. The following theorem shows that the clustering technique can be applied for a well-conditioned regularized (generalized) linear classification models.

Theorem 6. *Let $J(\theta, D) = \frac{1}{N}\sum_{i=1}^{N} l(\theta^T x_i, y_i) + \Lambda R(\theta)$. Let D and D^* be two data sets that differ by one point such that $D^* \setminus \{(x^*, y^*)\} = D \setminus \{(x, y)\}$, $\theta = \operatorname{argmin}_\theta J(\theta, D)$ and $\theta^* = \operatorname{argmin}_\theta J(\theta, D^*)$ be the corresponding ERM predictions. If $R(\cdot)$ is differentiable and 1-strongly convex, and l is convex and differentiable with the property that $|l'(z)| \leq 1$ and $l'(z)$ has the same sign for all z. Assume that the utility is $U(D) = M - \frac{1}{N_{test}}\sum_{i=1}^{N_{test}} l(\theta^T x_{test,i}, y_{test,i})$ and $l(\cdot,\cdot) \leq M$, then $s(z_i) - s(z_i^*) \leq \frac{1}{N_{test}\Lambda}\sum_{i=1}^{N_{test}} \|x_{test,i}\|\|x_i - x_i^*\|\frac{1+\log(N-1)}{N-1}$.*

As a result, if we are interested in grouping data points whose Shapley value differences are bounded by ϵ, then it suffices to group the points that are at most

$$\frac{\Lambda}{\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}} \|x_{\text{test},i}\|}\frac{N-1}{1+\log(N-1)}\epsilon \tag{3.8}$$

away from each other in the original data space. As long as we can compute the Shapley value of the cluster representative in each cluster accurately, we could obtain the Shapley value of other points in the same cluster by assigning the same value.

Algorithm 4 is inspired by the above theoretical results and combines the clustering and the permutation sampling method. The algorithm can achieve $(2\epsilon, \delta)$-approximation to the Shapley value.

The `PairwiseClustering` function in Algorithm 4 clusters all training data points in D such that pairwise distances within each cluster are at most d_{max}.

Algorithm 4 Clustering-Based Shapley Value Approximation

Input: $U(\cdot) \in [0, r]$ - utility function, ϵ, δ - approximation parameters, Λ - regularizer of empirical loss (see Theorem 6), $D = \{(x_i, y_i)\}_{i=1}^N$ - training dataset, $D_{\text{test}} = \{(x_{\text{test},i}, y_{\text{test},i})\}_{i=1}^N$ - testing dataset

Output: $\{\hat{s}_i\}_{i=1,\ldots,N}$ - estimated Shapley values, $\{\mathcal{C}_c\}_{c=1,\ldots,C}$ - clusters

$\hat{s}_i \leftarrow 0$ for $i = 1, \cdots, N$
$d_{\max} \leftarrow \Lambda \frac{\Lambda}{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \|x_{\text{test},i}\|} \frac{N-1}{1+\log(N-1)} \epsilon$
$\{\mathcal{C}_c\}_{c=1,\ldots,C} \leftarrow \text{PairwiseClustering}(D, d_{\max})$
$M \leftarrow \frac{2r^2}{\epsilon^2} \log(\frac{2C}{\delta})$
$\{\pi_m\}_{m=1}^M \leftarrow \text{GenerateUniformRandomPermutation}(\{(x_i, y_i)\}_{i=1}^N)$
for $c = 1$ to C **do**
    Draw $j \sim \frac{1}{|\mathcal{C}_c|}$, $j \in \mathcal{C}_c$
    for $m = 1$ to M **do**
        $P_j^{\pi_m} \leftarrow \text{CalculateSetOfPrecedingUsers}(\pi_m, j)$
        $\hat{s}_j \leftarrow \hat{s}_j + \frac{1}{M} \left(U(P_j^{\pi_m} \cup \{j\}) - U(P_j^{\pi_m})\right)$
    end for
    $\hat{s}_i \leftarrow \hat{s}_j \quad \forall i \in \mathcal{C}_c$
end for

3.6 Experimental Results

In this section, we demonstrate the Shapley value of training points in various datasets and compare different techniques for computing the Shapley value.

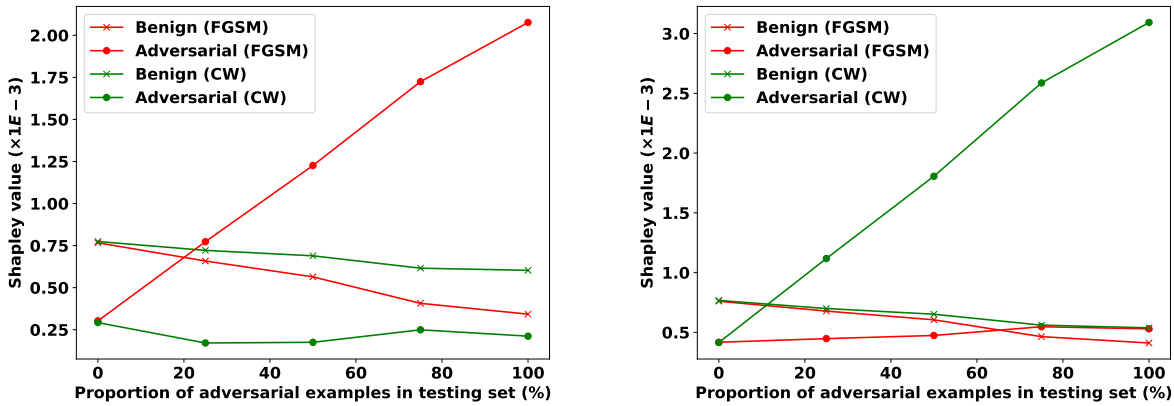
Does the Shapley Value Reflect the Value of Data?

We validate the hypothesis that the Shapley value produces data values that make intuitive sense.

Data Value for Adversarial Examples. Mixing adversarial examples with benign examples in the training dataset, or adversarial training, is an effective method to improve the adversarial robustness of a model. In practice, we measure the robustness in terms of the testing accuracy on a dataset containing adversarial examples. We expect that the adversarial examples in the training dataset become more valuable as more adversarial examples are added into the testing dataset. Based on MNIST, we construct a training dataset that contains both benign and adversarial examples and synthesize testing datasets with different adversarial-benign mixing ratios. Two popular attack algorithms, namely, Fast Gradient Sign Method (FGSM) [61] and the Carlini and Wagner (CW) attack [23] are used to generate adversarial examples. Figure 3.4(a, b) compares the average Shapley value for adversarial examples and for benign examples in the training dataset. The negative testing loss for

logistic regression is used as the utility function. We see that the Shapley value of adversarial examples increases as the testing data becomes more adversarial and contrariwise for benign examples. This is consistent with our expectation. In addition, the adversarial examples in the training set are more valuable if they are generated from the same attack algorithm for testing adversarial examples.

If the KNN Shapley value, which has very efficient algorithms to calculate, is correlated with the true Shapley value using the utility function based on logistic regression, we would expect that similar trend should also appear for data values calculated with KNN. Figure 3.3 repeats this experiment by using KNN Shapley value as the y-axis. We see that the same trend holds with more adversarial examples added to the test set.



(a) Training on benign + FGSM adversarial examples (b) Training on benign + CW adversarial examples

Figure 3.3: Comparison of KNN Shapley value of benign and adversarial examples. FGSM and CW in the legends indicate the attack algorithms used for generating adversarial examples in the testing dataset

Data Value for Privacy-Preserving Data. Current and future data markets value privacy. Differential privacy [49] has emerged as a standard privacy notation and is often achieved by adding noise that has a magnitude proportional to the desired privacy level. On the other hand, noise diminishes the usefulness of data and thereby degrades the value of data. We divide the training dataset into two halves, one half containing normal images and the other half containing noisy ones. The testing accuracy on normal images is used as the utility function. Figure 3.4(c) illustrates a clear tradeoff between privacy and data value - the Shapley value decreases as data becomes noisier.

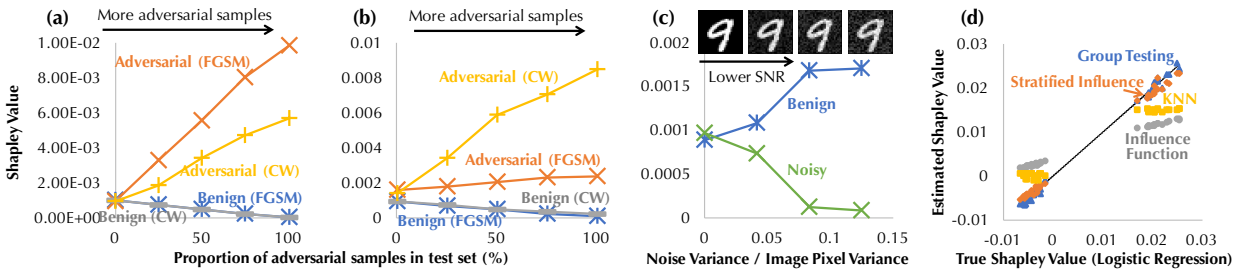


Figure 3.4: (a, b) Comparison of Shapley value of benign and adversarial examples. FGSM and CW are different attack algorithms used for generating adversarial examples in the testing dataset: (a) (resp. (b)) is trained on Benign + FGSM (resp. CW) adversarial examples. (c) Tradeoff between data value and privacy. (d) Comparison of data values produced by different methods for training a logistic regression model.

Comparison of Different Approaches for Data Valuation

We first validate that different valuation methods lead to results that are correlated, including (a) the permutation sampling, (b) the group testing-based method, (c) the combination of influence function and stratified sampling (d) the KNN-based approach and (e) largest-S influence function. (a)-(c) produces an approximation to Shapley value. (d) outputs the exact Shapley value for a KNN classifier. (e) approximates the effect of testing loss after removing a training point, which can also be considered a data value measure. We use a small-scale dataset, `iris`, and use (a) to estimate the true Shapley value for a regularized logistic regression up to $\epsilon = 1/N$. Figure 3.4(d) shows the result. While the outputs of all methods are correlated, the results (a)-(c) are closest to each other, as (a)-(c) share the same utility function (i.e., logistic testing loss). Due to the discrepancy of the underlying utility function, KNN attributes values in a way quite different from (a)-(c). In addition, the KNN value demonstrates small variations within the group of high-value points and that of low-value points. This can be explained by the recursion formula in (3.4), which does not differentiate the value of points if they all have the same label.

We implement the Shapley value calculation techniques on a machine with 16 cores (Intel Xeon CPU E5-2620 v4 @ 2.10GHz) and compare the runtime of different techniques on the Imagenet dataset. For each training data point, we first pre-compute the 2048-dimensional inception features and then train a logistic regression using the stochastic gradient descent for 150 epochs. The utility function is the negative testing loss of the logistic regression model. For largest-S influence and stratified influence sampling, we use the method in [95] to compute the influence function. The runtime of different techniques in logarithmic scale is displayed in Figure 3.5. We can see that the group testing-based method outperforms the permutation sampling baseline by several orders of magnitude for a large number of data points. By combining influence functions and stratified sampling, the computational costs can be further reduced. Due to the fact that the largest-S influence heuristic only focuses

the marginal contribution of each training data point to a single subset, it is much more efficient than the permutation sampling, group testing and combination of influence functions and stratified sampling, which compute the marginal contributions to a large number of subsets for approximating the Shapley value. The runtime of the KNN Shapley value grows linear with the number of training points and is therefore most practical to valuate enormous amounts of data.

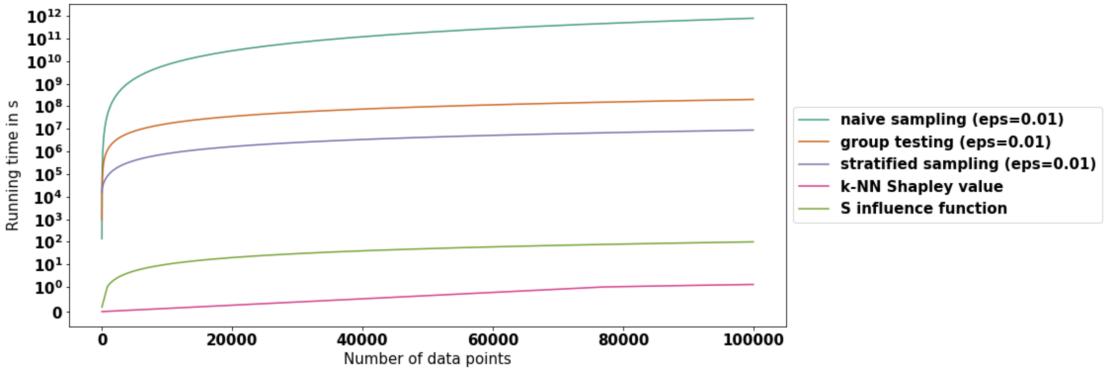


Figure 3.5: Run time comparison (in log scale) of our proposed methods. Each data point has a dim 2048.

Figure 3.6 illustrates the result of a large-scale experiment using the KNN Shapley value. We take *all* 1.5 million images with pre-calculated features and labels from Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset. We see that, the KNN Shapley value makes intuitive sense–the top valued images are semantically correlated with the corresponding testing image. This experiment takes only seconds per test image on a single CPU.

KNN Shapley Value vs. Largest-S Influence

We propose two efficient ways of computing the Shapley value — the KNN-based approach and the Largest-S Influence function. When the utility function is not a KNN classifier, both approaches are approximations to the true Shapley value. In the following experiment, we investigate the correlation between the KNN Shapley value and the Largest-S Influence on a middle-sized dataset.

Protocol We construct the DogFish that contains 900 dog and 900 fish images from ImageNet to form the training set and 300 dog and 300 fish images to form the testing set. We use logistic regression as the underlying classifier and the accuracy as the utility. We use a pre-trained InceptionV3 as the feature extractor. Figure 3.7(a) illustrates the top data points obtained using KNN Shapley value and Largest-S Influence for a test set that only contains a single image. We see that all approaches return images that make intuitive sense.

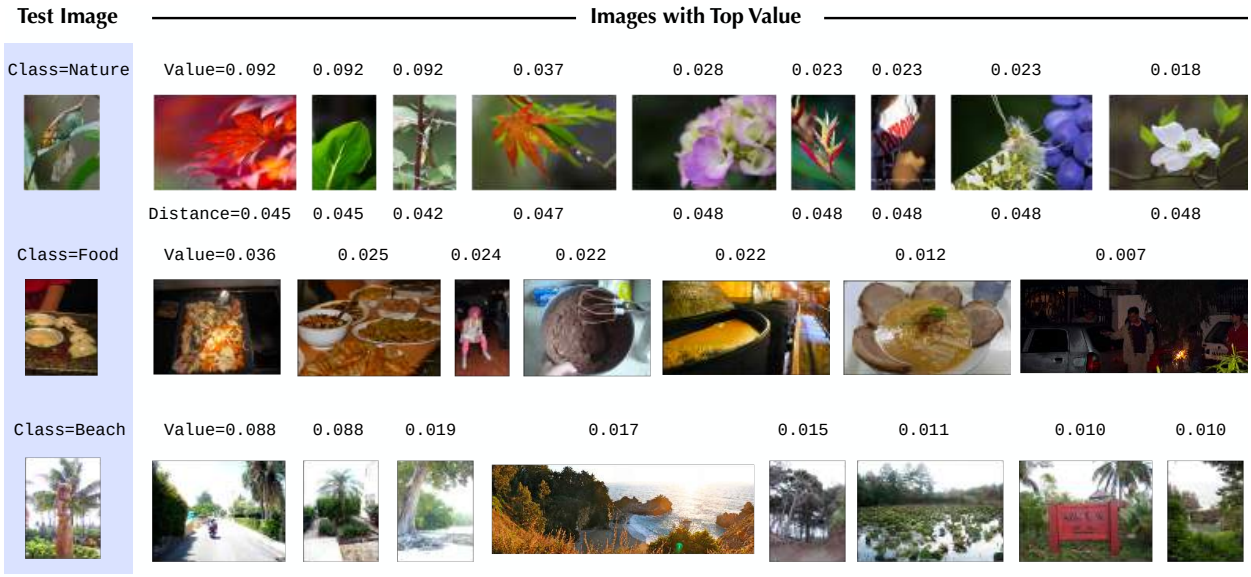


Figure 3.6: Data valuation using KNN classifiers ($K = 10$) on 1.5M images (all images with pre-calculated features in the Yahoo100M data set). The utility is the probability of correct classification of a single test image.

Experiment 1: Single Image. We compare the difference between KNN Shapley value and Largest-S Influence on a test set with a single image. Figure 3.7(b) illustrates the correlation between the data values calculated using these two approaches. We see that, for the given fish image, both KNN and Influence function assigns higher value for fish images and lower value for dog images. On this single image, the few data points with very high (resp. very low) Largest-S influence also has high (resp. low) KNN Shapley value. However, there are more data points that have the highest KNN Shapley value. The reason is that KNN Shapley value does not distinguish between all top-ranked data points with the same label. In this data set, the feature is reasonably good and therefore most top ranked images for the test image belong to the right class. In this case, all these top ranked images have the same KNN Shapley value.

Experiment 2: Multiple Images. We compare the difference of KNN Shapley value and Largest-S Influence on a test set containing 600 images, half of which are dogs and half of which are fish. In expectation, we expect that there would be valuable data points with labels of both fish and dog as the test set is uniformly mixed between these two classes. As shown in Figure 3.7(c), influence function returns a pretty uniform distribution between fish and dog; however, KNN assigns more values to dog images than fish. We investigate the reason: Figure 3.7(d) plots the distribution of the number test examples with respect to how many of their top-10 neighbors are with a wrong label. We see that dog images on average have more nearest neighbors with wrong labels than fish images. In other words, the dog

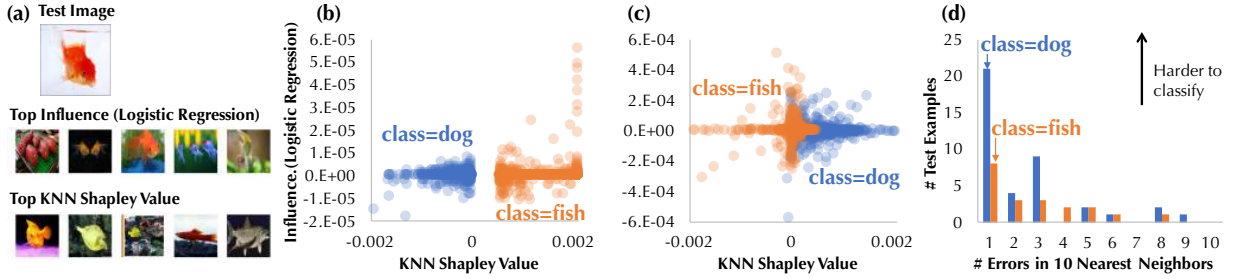


Figure 3.7: Data valuation on DogFish dataset. (a) top valued data points; (b, c) KNN Shapley value vs. Influence Function on (b) a single test image and (c) the whole test set; (d) Per-class mis-classifications in top-10 neighbors.

class is more difficult to be classified correctly. This intuitively explains the reason that KNN assigns higher values to dog images.

Group Testing vs. Permutation Sampling

Group testing provides an efficient way of sharing information across different samples to estimate the Shapley value. When the utility function is a black box that cannot be incrementally maintained efficiently, the number of model evaluations of the group testing-based approach is less sensitive to the number of data points in the dataset compared with the permutation sampling-based approach—$O((\log N)^2)$ vs. $O(N \log N)$. Figure 3.8 illustrates this effect. We see that, when the number of data points in the dataset is small, group testing actually requires *more* model evaluations–this is because group testing has a larger constant in the bound. However, when the number of samples grows, group testing becomes orders of magnitude more efficient in terms of the number of model evaluations it requires. For large-scale datasets (e.g., each model evaluation involves an ImageNet-scale dataset and a InceptionV3-level neural network) this could still be expensive; nevertheless, we believe that group testing provides an interesting theoretical angle to the problem of estimating Shapley value for general utility functions.

Convergence Results. Both theoretical bounds for group testing and permutation sampling-based approach provide approximation error guarantees in the worst case. In practice, the convergence of the estimation can be much faster than what the theoretically lower bounds indicate. Therefore, by looking at the convergence graph, it is possible to make some early-stopping decisions. Figure 3.10 and Figure 3.9 illustrate two example convergence graphs on the `iris` dataset. The x-axis is the number of samples and the y-axis is the Shapley value estimates. Each curve corresponds to the Shapley value for a single data point. We see that, for group testing, the theory predicts a sample size around 4×10^5, however, the process converges much faster–in fact, with 2×10^5 samples, the high-value group and low-value

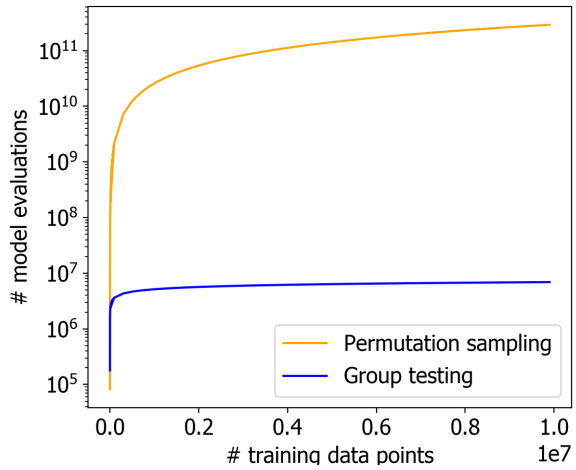


Figure 3.8: Comparison of the number of model evaluations for permutation sampling and group testing. The underlying machine learning model is regularized logistic regression. The utility is the negative loss on the testing dataset.

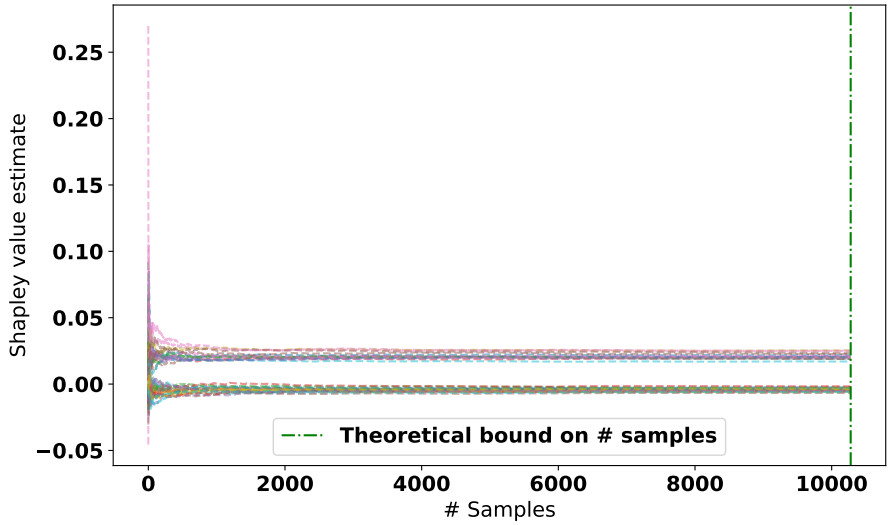


Figure 3.9: Convergence of the permutation sampling-based method on `iris`. The theoretical bound on the number of samples is the green vertical line.

group are already clearly distinguished. We observe a similar phenomenon for the baseline permutation sampling-based approach.

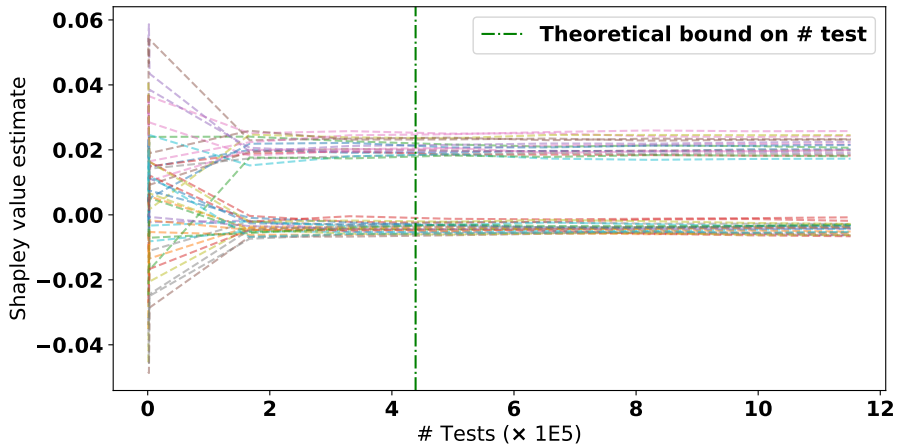


Figure 3.10: Convergence of the group testing-based method on `iris`. The bound on the number of tests in Theorem 4 is the green vertical line.

3.7 Chapter Summary

Machine learning has opened up exciting opportunities to tackle a wide variety of problems; nevertheless, very few works have attempted to understand the value of data used for training models. A principled way of data valuation is the key to stimulate data exchange, enabling the development of more sophisticated and robust machine learning models. We adopt the Shapley value, a classic concept from cooperative game theory, for data valuation. The Shapley value has many unique properties (e.g., efficiency, fairness, and additivity) that are appealing for data valuation. However, the lack of efficient methods to compute the Shapley value has prevented it from being adopted in the past. We develop a repertoire of techniques for computing or approximating the Shapley value for different scenarios. In particular, the K-NN-based approach allows us to compute the exact Shapley value for millions of data points.

For future work, we would like to study the implication of data poisoning in future data markets and define proper values to characterize malicious data points. We also intend to explore efficient data valuation methods for the case where a user contributes more than one data. This will be particularly interesting to appraise the data exchange between organizations. We wish to continue exploring the connection between machine learning and game theory and develop efficient valuation methods for other classifiers. It is also critical to understand other concepts from cooperative game theory (e.g., stable coalition) in the context of data valuation. Last but not least, we hope to apply the techniques to real-world applications and revolutionize the way of data collection and dissemination.

3.8 Proof of Main Results

This section contains the proofs for the theoretical results presented in this section.

Proof of Theorem 1

We introduce two basic lemmas and proceed to the proof of our main result, the recursive K-NN Shapley value computation described in Theorem 1.

Lemma 1.

$$\sum_{i=0}^{min(a,N)} \sum_{j=0}^{M} \frac{\binom{N}{i}\binom{M}{j}}{\binom{N+M}{i+j}} = (min(a,N)+1)\frac{M+N+1}{N+1}. \tag{3.9}$$

Proof of Lemma 1. Remark that Equation 3.9 can be re-written as

$$\sum_{i=0}^{min(a,N)} \sum_{j=0}^{M} \frac{\binom{N}{i}\binom{M}{j}}{\binom{N+M}{i+j}} = \frac{1}{\binom{M+N}{M}} \sum_{i=0}^{min(a,N)} \sum_{j=0}^{M} \binom{i+j}{i}\binom{M+N-i-j}{N-i}. \tag{3.10}$$

Consider the inner summation $\sum_{j=0}^{M} \binom{i+j}{i}\binom{M+N-i-j}{N-i}$. First, we show that

$$\binom{M+N+1}{N+1} = \sum_{j=0}^{M} \binom{i+j}{i}\binom{M+N-i-j}{N-i}.$$

A direct application of Hockey-Stick Identity to $\binom{M+N+1}{N+1}$ implies

$$\binom{M+N+1}{N+1} = \sum_{i=N}^{M+N} \binom{i}{N}. \tag{3.11}$$

Let $k \coloneqq M+N-i$. Equation 3.11 then becomes

$$\binom{M+N+1}{N+1} = \sum_{k=0}^{M} \binom{M+N-k}{N}. \tag{3.12}$$

Next, we apply the above expansion recursively $i+1$ times, that is,

$$\binom{M+N+1}{N+1} = \sum_{\substack{k_l,\ l\in\{1,2,\cdots,i+1\} \\ s.t.\ 0\le k_l\le M-\sum_{m=1}^{l-1} k_m}} \binom{M+N-i-\sum_{l=1}^{i+1} k_l}{N-i}. \tag{3.13}$$

Let $j \coloneqq \sum_{l=1}^{i+1} k_l$. We have

$$\sum_{\substack{k_l,\ l\in\{1,2,\cdots,i+1\} \\ s.t.\ 0\le k_l\le M-\sum_{m=1}^{l-1} k_m}} \binom{M+N-i-\sum_{l=1}^{i+1} k_l}{N-i} = \sum_{j=0}^{M} c_j \binom{M+N-i-j}{N-i}. \tag{3.14}$$

The coefficient c_j of each term of the form $\binom{M+N-i-j}{N-i}$ with $j = \{0, 1, \ldots, M\}$ is induced by all different combinations of k_l's, $l \in \{1, 2, \cdots i+1\}$ such that $\sum_{l=1}^{i+1} k_l = j$. Evidently, this is a problem of distributing a total of j indistinguishable numbers to $i+1$ distinguishable summations. Distinguishability is due to the relation that $k_l \le M - \sum_{m=1}^{l-1} k_m$. A direct application of balls-and-urns (also known as stars-and-bars) implies that the total number of $\binom{M+N-i-j}{N-i}$ terms coming from all different combinations of k_l's with $\sum_{l=1}^{i+1} k_l = j$ is given by $\binom{i+j}{j}$. Hence

$$\binom{M+N+1}{N+1} = \sum_{j=0}^{M} \binom{i+j}{j} \binom{M+N-i-j}{N-i}. \tag{3.15}$$

□

The following lemma says that the difference in the utility gain induced by either point i or point j translates linearly to the difference in the respective Shapley values, which can be calculated simply by consideration of all possible subsets excluding these points.

Lemma 2. *For any $i, j \in I$ and $i \neq j$, the difference in Shapley values between i and j is*

$$s_i - s_j = \frac{1}{N-1} \sum_{S\subseteq I\setminus\{i,j\}} \frac{1}{\binom{N-2}{|S|}} \big[U(S\cup\{i\}) - U(S\cup\{j\})\big] \tag{3.16}$$

Proof.

$$
\begin{aligned}
s_i - s_j = & \sum_{S \subseteq I \setminus \{i\}} \frac{|S|!(N-|S|-1)!}{N!} \big[U(S \cup \{i\}) - U(S)\big] \\
& - \sum_{S \subseteq I \setminus \{j\}} \frac{|S|!(N-|S|-1)!}{N!} \big[U(S \cup \{j\}) - U(S)\big] \qquad (3.17) \\
= & \sum_{S \subseteq I \setminus \{i,j\}} \frac{|S|!(N-|S|-1)!}{N!} \big[U(S \cup \{i\}) - U(S \cup \{j\})\big] \\
& + \sum_{S \in \{T | T \subseteq I, i \notin T, j \in T\}} \frac{|S|!(N-|S|-1)!}{N!} \big[U(S \cup \{i\}) - U(S)\big] \\
& - \sum_{S \in \{T | T \subseteq I, i \in T, j \notin T\}} \frac{|S|!(N-|S|-1)!}{N!} \big[U(S \cup \{j\}) - U(S)\big] \qquad (3.18) \\
= & \sum_{S \subseteq I \setminus \{i,j\}} \frac{|S|!(N-|S|-1)!}{N!} \big[U(S \cup \{i\}) - U(S \cup \{j\})\big] \\
& + \sum_{S' \subseteq I \setminus \{i,j\}} \frac{(|S'|+1)!(N-|S'|-2)!}{N!} \big[U(S' \cup \{i\}) - U(S' \cup \{j\})\big] \qquad (3.19) \\
= & \sum_{S \subseteq I \setminus \{i,j\}} \Big(\frac{|S|!(N-|S|-1)!}{N!} + \frac{(|S|+1)!(N-|S|-2)!}{N!}\Big) \\
& \cdot \big[U(S \cup \{i\}) - U(S \cup \{j\})\big] \qquad (3.20) \\
= & \frac{1}{N-1} \sum_{S \subseteq I \setminus \{i,j\}} \frac{1}{C_{N-2}^{|S|}} \big[U(S \cup \{i\}) - U(S \cup \{j\})\big] . \qquad (3.21)
\end{aligned}
$$

□

Loosely speaking, the proof distinguishes subsets S which include neither i nor j (such that the subset utility $U(S)$ of the marginal contribution directly cancels) and subsets including either i or j. In the latter case, S can be partitioned to a mock subset S' by excluding the respective point from S such that a common sum over S' again eliminates all terms other than $U(S' \cup \{i\}) - U(S' \cup \{j\})$.

In the following we restate, comment and prove Theorem 1.

Theorem 1. *Let x_{α_i}, $i = 1, \ldots, N$, be the training point that is ith closest to a given test point x_{test}. If KNN ($K < N$) is used as the classifier and the utility is $U(S) = \frac{1}{K} \sum_{\alpha_i \in S, i=1,\cdots,\min\{|S|,K\}} \mathbb{I}[y_{\alpha_i} = y_{test}]$, then the Shapley value of each training point can be*

calculated recursively as follows:

$$s_{\alpha_N} = \frac{\mathbb{I}[y_{\alpha_N} = y_{test}]}{N} \tag{3.22}$$

$$s_{\alpha_i} = s_{\alpha_{i+1}} + \frac{\mathbb{I}[y_{\alpha_i} = y_{test}] - \mathbb{I}[y_{\alpha_{i+1}} = y_{test}]}{K} \frac{(\min(K-1, i-1)+1)}{i}. \tag{3.23}$$

The proof is motivated by the following insight: Recall that the probability $P[x_{\text{test}} \to y_{\text{test}}|S]$ of a correct test label assignment is calculated from the labels of the K points $\alpha_1, .., \alpha_K$ closest to x_{test}, measured by a given distance $d(\cdot, x_{\text{test}})$. Recall also that this defines the subset utility $U(S) := \frac{1}{K}\sum_{\alpha_i \in S, i=1,\cdots,K} \mathbb{I}[y_{\alpha_i} = y_{\text{test}}]$, from which we can directly see that $U(S)$ depends only on the K points $\{x_{\alpha_1}, ..., x_{\alpha_K}\} \subseteq S$ nearest to the test point x_{test}.

If we consider the utility $U(S \cup \{i\})$ of the union of S with any point i, the K nearest points $\{\alpha_1, ..., \alpha_K\} \subseteq S$ within S may not be identical to the K nearest points of the union, that is, $U(S) \neq U(S \cup \{i\})$. Adding a point i to a subset changes the utility if and only if the ith point is closer to the test point than point α_K, or, equivalently

$$U(S) = U(S \cup \{i\}) \iff d(x_{\alpha_K}, x_{\text{test}}) < d(x_i, x_{\text{test}})\,.$$

This motivates a case distinction if we are interested in the difference in utility $\Delta U(S, i, j) = U(S \cup \{i\}) - U(S \cup \{j\})$. Following the intuition outlined in Section 3.4 we now provide a more technical case distinction for the general K-NN-based approach to computing the Shapley value.

Proof of Theorem 1. Let $x_{\text{test}} \in \mathbb{R}^d$ be the testing point which defines the utility function $U(S) := \frac{1}{K}\sum_{\alpha_i \in S, i=1,\cdots,K} \mathbb{I}[y_{\alpha_i} = y_{\text{test}}]$ for a K-NN classifier with training points $\{x_{\alpha_i}\}_{i=1}^N$ and labels $\{y_{\alpha_i}\}_{i=1}^N$. Let $d(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be the distance measure according to which $\{\alpha_i\}_{i=1}^N$ is determined.

W.l.o.g., we assume that $x_1, \ldots, x_n$ are sorted according to their similarity $d(\cdot, x_{\text{test}})$ with x_{test}, that is, $x_i = x_{\alpha_i}$ (x_1 is most similar to x_{test}). We further assume that there are no ties in terms of similarity. This procedure can be finished by $Nd + N \log N$ operations where d is the dimension of features.

We now choose to split a subset $S \subseteq I \setminus \{i, i+1\}$ of size k into two disjoint sets S_1 and S_2 with $S = S_1 \cup S_2$ and $|S_1| + |S_2| = |S| = k$. Given two neighboring points with indices $i, i+1 \in I$ we constrain S_1 and S_2 to $S_1 \subseteq \{1, ..., i-1\}$ and $S_2 \subseteq \{i+2, ..., N\}$.

Let s_i be the Shapley value of data point x_i. Recall from Lemma 2 that for any $i, j \in I$ and $i \neq j$, the difference of Shapley values between i and j is

$$s_i - s_j = \frac{1}{N-1} \sum_{S \subseteq I \setminus \{i,j\}} \frac{1}{\binom{N-2}{|S|}} \big[U(S \cup \{i\}) - U(S \cup \{j\})\big] \tag{3.24}$$

We can now draw conclusions about the utility function $U(S \cup \{i\})$ of any subset $S \subseteq I \setminus \{i, i+1\}$ and the resulting Shapley value by considering the following cases:

Case 1.1 Consider the case $|S_1| < K$ with $y_i \neq y_{i+1}$. We then know that $i < K$ and therefore $U(S \cup \{i\}) \neq U(S)$ if $y_i \neq y_{i+1}$. However, the difference of including a point i affects only one term in the sum $U(S) := \frac{1}{K} \sum_{k \in S}^{K} \mathbb{I}[y_{\alpha_k} = y_{\text{test}}]$ over all K nearest neighbors, i.e.

$$U(S \cup \{i\}) - U(S) = \begin{cases} \frac{1}{K}(\mathbb{I}[y_i = y_{\text{test}}] - \mathbb{I}[y_K = y_{\text{test}}]) & \text{when } |S| \geq K \\ \frac{1}{K}\mathbb{I}[y_i = y_{\text{test}}] & \text{when } |S| < K \end{cases} \tag{3.25}$$

The same holds for the inclusion of point $i+1$. Therefore,

$$U(S \cup \{i\}) - U(S \cup \{i+1\}) = \frac{\mathbb{I}[y_i = y_{\text{test}}] - \mathbb{I}[y_{i+1} = y_{\text{test}}]}{K}$$

Using (3.24) we have

$$s_i - s_{i+1} = \frac{1}{N-1} \sum_{k=0}^{N-2} \frac{1}{\binom{N-2}{k}} \sum_{\substack{S_1 \subseteq \{1,\ldots,i-1\}, \\ S_2 \subseteq \{i+2,\ldots,N\}: \\ |S_1|+|S_2|=k, |S_1|<K}} \frac{\mathbb{I}[y_i = y_{\text{test}}] - \mathbb{I}[y_{i+1} = y_{\text{test}}]}{K} \,. \tag{3.26}$$

Case 1.2 Consider the case $|S_1| \geq K$ with $y_i \neq y_{i+1}$. We then know that $i > K$ and therefore $U(S \cup \{i\}) = U(S)$, hence the term $U(S \cup \{i\}) - U(S)$ is zero. Since the same holds for point $i+1$ ($i+1 \geq K$ and $U(S \cup \{i+1\}) = U(S)$) and by (3.24), it follows that

$$s_i - s_{i+1} = \frac{1}{N-1} \sum_{k=0}^{N-2} \frac{1}{\binom{N-2}{k}} \sum_{\substack{S_1 \subseteq \{1,\ldots,i-1\}, \\ S_2 \subseteq \{i+2,\ldots,N\}: \\ |S_1|+|S_2|=k, |S_1|\geq K}} \Big[U(S \cup \{i\}) - U(S \cup \{i+1\}) \Big] = 0 \,. \tag{3.27}$$

Case 2 Consider the case of neighboring training points with identical labels $y_i = y_{i+1}$. From $\mathbb{I}[y_i = y_{\text{test}}] = \mathbb{I}[y_{i+1} = y_{\text{test}}]$ and $U(S) = \frac{1}{K} \sum_{k \in S}^{K} \mathbb{I}[y_{\alpha_k} = y_{\text{test}}]$ we directly conclude $U(S \cup \{i\}) = U(S \cup \{i+1\})$ for any $S \subseteq I \setminus \{i, i+1\}$ and therefore $s_i - s_{i+1} = 0$.

Rewriting the Non-Trivial Shapley Difference In the following we consider only the non-trivial case (Case 1.2). Note that $\frac{\mathbb{I}[y_i = y_{\text{test}}] - \mathbb{I}[y_{i+1} = y_{\text{test}}]}{K}$ does not depend on the summation at all:

$$\begin{aligned} s_i - s_{i+1} &= \frac{\mathbb{I}[y_i = y_{\text{test}}] - \mathbb{I}[y_{i+1} = y_{\text{test}}]}{K} \times \frac{1}{N-1} \sum_{k=0}^{N-2} \frac{1}{\binom{N-2}{k}} \sum_{\substack{S_1 \subseteq \{1,\ldots,i-1\}, \\ S_2 \subseteq \{i+2,\ldots,N\}: \\ |S_1|+|S_2|=k, |S_1|<K}} 1 \\ &= \frac{\mathbb{I}[y_i = y_{\text{test}}] - \mathbb{I}[y_{i+1} = y_{\text{test}}]}{K} \times \frac{1}{N-1} \sum_{k=0}^{N-2} \frac{1}{\binom{N-2}{k}} \sum_{m=0}^{\min(K-1,k)} \binom{i-1}{m} \binom{N-i-1}{k-m} \end{aligned} \tag{3.28}$$

Then, we can use Lemma 1 to analyze the following term:

$$\sum_{k=0}^{N-2} \frac{1}{\binom{N-2}{k}} \sum_{m=0}^{\min(K-1,k)} \binom{i-1}{m}\binom{N-i-1}{k-m} = \sum_{m=0}^{\min(K-1,i-1)} \sum_{k'=0}^{N-i-1} \frac{\binom{i-1}{m}\binom{N-i-1}{k'}}{\binom{N-2}{m+k'}} \tag{3.29}$$

Take $i-1=N$ and $N-i-1=M$ in Lemma 1 to get

$$\sum_{k=0}^{N-2} \frac{1}{\binom{N-2}{k}} \sum_{m=0}^{\min(K-1,k)} \binom{i-1}{m}\binom{N-i-1}{k-m} = \frac{(\min(K-1,i-1)+1)}{i}(N-1) \tag{3.30}$$

Therefore, we have the following recursion

$$s_i - s_{i+1} = \frac{\mathbb{I}[y_i = y_{\text{test}}] - \mathbb{I}[y_{i+1} = y_{\text{test}}]}{K} \frac{(\min(K-1,i-1)+1)}{i} \tag{3.31}$$

Now, we analyze the formula for s_N, the starting point of the recursion. Since x_N is farthest to x_{test} among all training points, x_N results in non-zero marginal utility only when it is added to a set of size smaller than K. Hence, s_N can be written as

$$s_N = \frac{1}{N}\sum_{k=0}^{K-1} \frac{1}{\binom{N-1}{k}} \sum_{|S|=k, S\subseteq I\setminus\{N\}} U(S\cup N) - U(S) \tag{3.32}$$

$$= \frac{1}{N}\sum_{k=0}^{K-1} \frac{1}{\binom{N-1}{k}} \sum_{|S|=k, S\subseteq I\setminus\{N\}} \frac{\mathbb{I}[y_N = y_{\text{test}}]}{K} \tag{3.33}$$

$$= \frac{\mathbb{I}[y_N = y_{\text{test}}]}{N} \tag{3.34}$$

□

Proof of Theorem 4

We prove Theorem 4 in Section 3.5, which specifies a lower bound on the number of tests needed for achieving a certain approximation error. Before delving into the proof, we first present a lemma that is useful for establishing the bound in Theorem 4.

Lemma 3 (Bennett's inequality [12])**.** *Given independent zero-mean random variables $X_1, \cdots, X_n$ satisfying the condition $|X_i| \le a$, let $\sigma^2 = \sum_{i=1}^n \sigma_i^2$ be the total variance. Then for any $t \ge 0$,*

$$P[S_n > t] \le \exp(-\frac{\sigma^2}{a^2} h(\frac{at}{\sigma^2})) \tag{3.35}$$

where $h(u) = (1+u)\log(1+u) - u$.

We now restate Theorem 4 and proceed to the main proof.

Theorem 4. *The group testing-based approach returns an (ϵ, δ)-approximation to the Shapley value if the number of tests T satisfies $T \geq \frac{4}{(1-q_{tot}^2)h\left(\frac{2\epsilon}{Zr C_\epsilon (1-q_{tot}^2)}\right)} \log \frac{C_\delta (N-1)}{2\delta}$ where $C_\epsilon, C_\delta > 1$, $q_{tot} = \frac{N-2}{N} q(1) + \sum_{k=2}^{N-1} q(k)[1 + \frac{2k(k-N)}{N(N-1)}]$, $h(u) = (1+u)\log(1+u) - u$, $Z = 2\sum_{k=1}^{N-1} \frac{1}{k}$, and r is the range of utility function.*

Proof of Theorem 4. By Lemma 3, the difference in Shapley values between points i and j is given as

$$s_i - s_j = \frac{1}{N-1} \sum_{S \subseteq I \setminus \{i,j\}} \frac{1}{C_{N-2}^{|S|}} \Big[U(S \cup \{i\}) - U(S \cup \{j\}) \Big] \tag{3.36}$$

$$= \frac{1}{N-1} \sum_{k=0}^{N-2} \frac{1}{C_{N-2}^{k}} \sum_{S \subseteq I \setminus \{i,j\}, |S|=k} \Big[U(S \cup \{i\}) - U(S \cup \{j\}) \Big]. \tag{3.37}$$

Let $\beta_1, \cdots, \beta_N$ denote N Boolean random variables drawn with the following sampler:

1. Sample the "length of the sequence" $\sum_{i=1}^{N} \beta_i = k \in \{1, 2, \cdots, N-1\}$, with probability $q(k)$.
2. Uniformly sample a length-k sequence from $\binom{N}{k}$ all possible length-k sequences

Then the probability of any given sequence $\beta_1, \cdots, \beta_N$ is

$$P[\beta_1, \cdots, \beta_N] = \frac{q(\sum_{i=1}^{N} \beta_i)}{C_N^{\sum_{i=1}^{N} \beta_i}}. \tag{3.38}$$

Now, we consider any two data points x_i and x_j where $i, j \in I = \{1, \cdots, N\}$ and their associated Boolean variables β_i and β_j, and analyze

$$\Delta = \beta_i U(\beta_1, \cdots, \beta_N) - \beta_j U(\beta_1, \cdots, \beta_N) \tag{3.39}$$

The expectation of Δ is given by

$$\mathbb{E}[\Delta] = \sum_{k=0}^{N-2} \frac{q(k+1)}{C_N^{k+1}} \sum_{S \subseteq I \setminus \{i,j\}, |S|=k} \big[U(\beta_1, \cdots, \beta_{i-1}, 1, \beta_{i+1}, \cdots, \beta_{j-1}, 0, \beta_{j+1}, \cdots, \beta_N)$$
$$- U(\beta_1, \cdots, \beta_{i-1}, 0, \beta_{i+1}, \cdots, \beta_{j-1}, 1, \beta_{j+1}, \cdots, \beta_N) \big] \tag{3.40}$$

$$= \sum_{k=0}^{N-2} \frac{q(k+1)}{C_N^{k+1}} \sum_{S \subseteq I \setminus \{i,j\}, |S|=k} \big[U(S \cup \{i\}) - U(S \cup \{j\}) \big] \tag{3.41}$$

We would like to have $Z\mathbb{E}[\Delta] = s_i - s_j$

$$Z\frac{q(k+1)}{C_N^{k+1}} = \frac{1}{(N-1)C_{N-2}^k} \tag{3.42}$$

which yields

$$q(k+1) = \frac{N}{Z(k+1)(N-k-1)} = \frac{1}{Z}(\frac{1}{k+1} + \frac{1}{N-k-1}) \tag{3.43}$$

for $k = 0, \cdots, N-2$. Equivalently,

$$q(k) = \frac{1}{Z}(\frac{1}{k} + \frac{1}{N-k}) \tag{3.44}$$

for $k = 1, \cdots, N-1$. The value of Z is given by

$$Z = \sum_{k=1}^{N-1}(\frac{1}{k} + \frac{1}{N-k}) = 2\sum_{k=1}^{N-1}\frac{1}{k} \leq 2(\log(N-1)+1) \tag{3.45}$$

Now, $\mathbb{E}[Z\Delta] = s_i - s_j$. Assume that the utility function ranges from $[0, r]$; then, we know from (3.6) that $Z\Delta$ is random variable ranges in $[-Zr, Zr]$.

Consider

$$\Delta := \beta_i U(\beta_1, \cdots, \beta_N) - \beta_j U(\beta_1, \cdots, \beta_N) \tag{3.46}$$

Note that $\Delta = 0$ when $\beta_i = \beta_j$. If $P[\beta_i = \beta_j]$ is large, then the variance of Δ will be much smaller than its range.

$$P[\beta_i = \beta_j] = P[\beta_i = 1, \beta_j = 1] + P[\beta_i = 0, \beta_j = 0] \tag{3.47}$$

$$= \left[\sum_{k=2}^{N-1}\frac{q(k)}{C_N^k}C_{N-2}^{k-2}\right] + \left[q(1) + \sum_{k=2}^{N-1}\frac{q(k)}{C_N^k}C_{N-2}^k\right] \tag{3.48}$$

$$= \frac{N-2}{N}q(1) + \sum_{k=2}^{N-1}q(k)\left[1 + \frac{2k(k-N)}{N(N-1)}\right] \equiv q_{tot} \tag{3.49}$$

Let $W = \mathbb{1}[\Delta \neq 0]$ be an indicator of whether or not $\Delta = 0$. Then, $P[W=0] = q_{tot}$ and $P[W=1] = 1 - q_{tot}$.

Now, we analyze the variance of Δ. By the law of total variance,

$$\texttt{Var}[\Delta] = \mathbb{E}[\texttt{Var}[\Delta|W]] + \texttt{Var}[\mathbb{E}[\Delta|W]] \tag{3.50}$$

Recall $\Delta \in [-r, r]$. Then, the first term can be bounded by

$$\begin{aligned}
\mathbb{E}[\texttt{Var}[\Delta|W]] &= P[W=0]\texttt{Var}[\Delta|W=0] + P[W=1]\texttt{Var}[\Delta|W=1] && (3.51)\\
&= q_{tot}\texttt{Var}[\Delta|\Delta=0] + (1-q_{tot})\texttt{Var}[\Delta|\Delta\neq 0] && (3.52)\\
&= (1-q_{tot})\texttt{Var}[\Delta|\Delta\neq 0] && (3.53)\\
&\leq (1-q_{tot})r^2 && (3.54)
\end{aligned}$$

where the last inequality follows from the fact that if a random variable is in the range $[m, M]$, then its variance is bounded by $\frac{(M-m)^2}{4}$.

The second term can be expressed as

$$\begin{aligned}
\texttt{Var}[\mathbb{E}[\Delta|W]] &= \mathbb{E}_W[(\mathbb{E}[\Delta|W] - \mathbb{E}[\Delta])^2] && (3.55)\\
&= P[W=0](\mathbb{E}[\Delta|W=0] - \mathbb{E}[\Delta])^2 + P[W=1](\mathbb{E}[\Delta|W=1] - \mathbb{E}[\Delta])^2 && (3.56)\\
&= q_{tot}(\mathbb{E}[\Delta|\Delta=0] - \mathbb{E}[\Delta])^2 + (1-q_{tot})(\mathbb{E}[\Delta|\Delta\neq 0] - \mathbb{E}[\Delta])^2 && (3.57)\\
&= q_{tot}(\mathbb{E}[\Delta])^2 + (1-q_{tot})(\mathbb{E}[\Delta|\Delta\neq 0] - \mathbb{E}[\Delta])^2 && (3.58)
\end{aligned}$$

Note that

$$\begin{aligned}
\mathbb{E}[\Delta] &= P[W=0]\mathbb{E}[\Delta|\Delta=0] + P[W=1]\mathbb{E}[\Delta|\Delta\neq 0] && (3.59)\\
&= (1-q_{tot})\mathbb{E}[\Delta|\Delta\neq 0] && (3.60)
\end{aligned}$$

Plugging (3.60) into (3.55), we obtain

$$\texttt{Var}[\mathbb{E}[\Delta|W]] = (q_{tot}(1-q_{tot})^2 + q_{tot}^2(1-q_{tot}))(\mathbb{E}[\Delta|\Delta\neq 0])^2 \quad (3.61)$$

Since $|\Delta| \leq r$, $(\mathbb{E}[\Delta|\Delta\neq 0])^2 \leq r^2$. Therefore,

$$\texttt{Var}[\mathbb{E}[\Delta|W]] \leq q_{tot}(1-q_{tot})r^2 \quad (3.62)$$

It follows that

$$\texttt{Var}[\Delta] \leq (1-q_{tot}^2)r^2 \quad (3.63)$$

Given T samples, the application of Bennett's inequality in Lemma 3 yields

$$P\Big[\sum_{t=1}^{T}(Z\Delta_t - \mathbb{E}[Z\Delta_t]) > \epsilon'\Big] \leq \exp\Big(-\frac{T(1-q_{tot}^2)}{4}h\big(\frac{2\epsilon'}{TZr(1-q_{tot}^2)}\big)\Big) \quad (3.64)$$

By letting $\epsilon = \epsilon'/T$

$$P\big[(Z\bar{\Delta} - \mathbb{E}[Z\Delta]) > \epsilon\big] \leq \exp\Big(-\frac{T(1-q_{tot}^2)}{4}h\big(\frac{2\epsilon}{Zr(1-q_{tot}^2)}\big)\Big) \quad (3.65)$$

Therefore, the number of tests T we need in order to get an (ϵ, δ)-approximation to the difference of two Shapley values for a single pair of data points is

$$T \geq \frac{4}{(1-q_{tot}^2)h(\frac{2\epsilon}{Zr(1-q_{tot}^2)})} \log \frac{1}{\delta} \tag{3.66}$$

By union bound, the number of tests T for achieving $(\frac{\epsilon}{C_\epsilon}, \frac{\delta}{C_\delta})$-approximation to the difference of the Shapley values for $N-1$ pairs of data points is

$$T \geq \frac{4}{(1-q_{tot}^2)h(\frac{2\epsilon}{ZrC_\epsilon(1-q_{tot}^2)})} \log \frac{C_\delta(N-1)}{2\delta} \tag{3.67}$$

□

Proof of Theorem 5

For the proof of Theorem 5 we need the following definition of a *stable utility function.*

Definition 1. *A utility function $U(\cdot)$ is called λ-stable if*

$$\max_{i,j \in I, S \subseteq I \setminus \{i,j\}} |U(S \cup \{i\}) - U(S \cup \{j\})| \leq \frac{\lambda}{|S|+1} \tag{3.68}$$

Then, Shapley values calculated from λ-stable utility functions have the following property.

Proposition 7. *If $U(\cdot)$ is λ-stable, then for all $i, j \in I$ and $i \neq j$*

$$s_i - s_j \leq \frac{\lambda(1+\log(N-1))}{N-1} \tag{3.69}$$

Proof of Proposition 7. By Lemma 3, we have

$$s_i - s_j \leq \frac{1}{N-1} \sum_{S \subseteq I \setminus \{i,j\}} \frac{1}{C_{N-2}^{|S|}} \frac{\lambda}{|S|+1} = \frac{1}{N-1} \sum_{|S|=0}^{N-2} \frac{\lambda}{|S|+1} \tag{3.70}$$

Recall the bound on the harmonic sequences

$$\sum_{k=1}^{N} \frac{1}{k} \leq 1 + \log(N) \tag{3.71}$$

which gives us

$$s_i - s_j \leq \frac{\lambda(1+\log(N-1))}{N-1} \tag{3.72}$$

□

Then, we can prove Theorem 5.

Theorem 5. *For a learning algorithm $G(\cdot)$ with uniform stability $\beta = \frac{C}{N}$, where N is the number of data points and C is some constant. Let the utility of D be $U(D) = M - L_e(G(D), D_{test})$ where $L_e(G(D), D_{test}) = \frac{1}{N}\sum_{i=1}^{N} l(G(D), z_{test,i})$ and $0 \leq l(\cdot,\cdot) \leq M$. Then, $s_i - s_j \leq (2C + M)\frac{1+\log(N-1)}{N-1}$ and the Shapley difference vanishes as $N \to \infty$.*

Proof of Theorem 5. For any $i, j \in I$ and $i \neq j$,

$$|U(S \cup \{i\}) - U_A(S \cup \{j\})| \tag{3.73}$$

$$\leq \frac{1}{|S|+1}\sum_{k \neq i,j} |l(A(S \cup \{i\}), z_k) - l(A(S \cup \{j\}), z_k)|$$

$$+ \frac{1}{|S|+1}|l(A(S \cup \{j\}), z_j) - l(A(S \cup \{i\}), z_i)| \tag{3.74}$$

$$\leq \frac{1}{|S|+1}\sum_{k \neq i,j} \big(|l(A(S \cup \{i\}), z_k) - l(A(S, z_k)|$$

$$+ |l(A(S), z_k) - l(A(S \cup \{j\}), z_k)|\big) + \frac{M}{|S|+1} \tag{3.75}$$

$$\leq \frac{2C|S|}{(|S|+1)^2} + \frac{M}{|S|+1} \tag{3.76}$$

$$\leq \frac{2C+M}{|S|+1} \tag{3.77}$$

Combining the above inequality with Proposition 7 proves the theorem. □

Proof of Theorem 6

In the following we present the theoretical foundations used in the proof of Theorem 6 and comment on examples of related loss functions.

Suppose that the aggregate data is used for regularized empirical risks minimization (ERM), where we choose a prediction θ that minimizes the regularized empirical loss:

$$J(\theta, D) = \frac{1}{N}\sum_{i=1}^{N} l(\theta^T x_i, y_i) + \Lambda R(\theta) \tag{3.78}$$

We assume the norm regularizer $R(\cdot)$ and loss function $l(\cdot,\cdot)$ to be differentiable functions of θ. Moreover, we assume that the loss function has the form $l(\theta^T x_i, y_i) = l(y_i \theta^T x_i)$, the examples of which include hinge loss, squared loss and logistic loss. Further, we assume that the input space $\mathcal{X} := \{x_i : \|x_i\| \leq 1\}$.

The main ingredient of the proof of Theorem 6 is a result about the sensitivity of regularized ERM, which is provided below.

Lemma 4 (Chaudhuri et al., [24])**.** *Let $G(\theta)$ and $g(\theta)$ be two vector-valued functions, which are continuous and differentiable at all points. Moreover, let $G(\theta)$ and $G(\theta)+g(\theta)$ be γ-strongly convex. If $\theta_1 = \text{argmin}_\theta\, G(\theta)$ and $\theta_2 = \text{argmin}_\theta\, G(\theta) + g(\theta)$, then*

$$\|\theta_1 - \theta_2\| \leq \frac{1}{\gamma} \max_\theta \|\nabla g(\theta)\| \tag{3.79}$$

We can then present a theorem that bounds the change of learned parameters when altering the value of a single point in the training dataset.

Theorem 8. *Let D and D^* be two data sets that differ in one individual such that $D^* \setminus \{(x^*, y^*)\} = D \setminus \{(x, y)\}$, $\theta = \text{argmin}_\theta\, J(\theta, D)$ and $\theta^* = \text{argmin}_\theta\, J(\theta, D^*)$ be the corresponding ERM predictions. If $R(\cdot)$ is differentiable and 1-strongly convex, and l is convex and differentiable with the property that $|l'(z)| \leq 1$ and $l'(z)$ has the same sign for all z, then*

$$\|\theta - \theta^*\| \leq \frac{1}{\Lambda N}\|x - x^*\| \tag{3.80}$$

if $y = y^*$.

Proof. We let $G(\theta) = J(\theta, D)$ and $g(\theta) = J(\theta, D^*) - J(\theta, D) = \frac{1}{N}(l(y^*\theta^T x^*) - l(y\theta^T x))$. Due to the convexity of l and 1-strong convexity of $R(\cdot)$, $G(\theta) = J(\theta, D)$ is Λ-strongly convex. Moreover, $G(\theta) + g(\theta) = J(\theta, D^*)$ is also Λ-strongly convex. Finally, due to the differentiability of $R(\cdot)$ and l, $G(\theta)$ and $g(\theta)$ are both differentiable at all points.

$$\|\nabla_\theta g(\theta)\| = \frac{1}{N}\|y^* l'(y^*\theta^T x^*)x^* - y l'(y\theta^T x)x\| \tag{3.81}$$

Given $y = y^*$, $l'(z)$ have the same sign, $|l'(z)| \leq 1$ for all z, and $\|x\| \leq 1$, for any θ, $\|\nabla g(\theta)\| \leq \frac{1}{N}\|x^* - x\|$. Applying Lemma 4 gives us

$$\|\theta - \theta^*\| \leq \frac{1}{N\Lambda}\|x^* - x\| \tag{3.82}$$

□

Remark 9. *The examples of classification loss functions that satisfy "l is convex and differentiable, $l'(z) \leq 1$ and $l'(z)$ has the same sign for all z" include:*

- *Logistic loss: $l(z) = \log(1 + e^{-z})$ where $z = y\theta^T x$. $l'(z) = -\frac{1}{1+e^z}$ and $l''(z) = \frac{1}{(1+e^{-z})(1+e^z)}$. So $|l'(z)| \leq 1$ and $l'(z) < 0$ for all z.*
- *Hinge loss: $l(z) = \max(0, 1 - z)$, which is used in support vector machines. However, it is not differentiable. We can approximate the hinge loss by a different loss function, which is doubly differentiable:*

$$l_s(z) = \begin{cases} 0 \text{ if } z > 1 + h \\ -\frac{(1-z)^4}{16h^3} + \frac{3(1-z)^2}{8h} + \frac{1-z}{2} + \frac{3h}{16} \text{ if } |1 - z| \leq h \\ 1 - z \text{ if } z < 1 - h \end{cases} \tag{3.83}$$

and

$$l_s'(z) = \begin{cases} 0 \text{ if } z > 1+h \\ \frac{(1-z)^3}{4h^3} - \frac{3(1-z)}{4h} - \frac{1}{2} \text{ if } |1-z| \leq h \\ -1 \text{ if } z < 1-h \end{cases} \tag{3.84}$$

This approximate hinge loss also satisfies the desired properties.

- *Huber loss:*

$$l(z) = \begin{cases} 0 \text{ if } z > 1+h \\ \frac{1}{4h}(1+h-z)^2 \text{ if } |1-z| \leq h \\ 1-z \text{ if } z < 1-h \end{cases} \tag{3.85}$$

and

$$l'(z) = \begin{cases} 0 \text{ if } z > 1+h \\ -\frac{1}{2h}(1+h-z) \text{ if } |1-z| \leq h \\ -1 \text{ if } z < 1-h \end{cases} \tag{3.86}$$

which also satisfies the desired properties.

Corollary 10. *Consider the definitions and assumptions specified in Theorem 8. Let the testing dataset of the ERM prediction be denoted by D_{test} and $|D_{test}| = N_{test}$. Define the testing loss associated with θ as $J(\theta, D_{test}) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} l(\theta, D_{test})$ and the testing loss $J(\theta^*, D_{test})$ associated with θ^* is similarly defined.*

$$J(\theta, D_{test}) - J(\theta^*, D_{test}) \leq \frac{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \|x_{test,i}\|}{N\Lambda} \|x - x^*\| \tag{3.87}$$

Proof. Since l is convex and $|l'(z)| \leq 1$ for all z, it follows that

$$l(\theta, D_{\text{test},i}) - l(\theta^*, D_{\text{test},i}) \leq \|y_{\text{test},i} l'(y_{\text{test},i} \theta^T x_{\text{test},i}) x_{\text{test},i}\| \cdot \|\theta - \theta^*\| \tag{3.88}$$

Moreover, since

$$J(\theta, D_{\text{test}}) - J(\theta^*, D_{\text{test}}) \leq \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (l(\theta, D_{\text{test},i}) - l(\theta^*, D_{\text{test},i})) \tag{3.89}$$

$$\leq \left(\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \|y_{\text{test},i} l'(y_{\text{test},i} \theta^T x_{\text{test},i}) x_{\text{test},i}\| \right) \|\theta - \theta^*\| \tag{3.90}$$

$$\leq \left(\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \|x_{\text{test},i}\| \right) \|\theta - \theta^*\| \tag{3.91}$$

the proof now follows by an application of Theorem 8. □

Assume the utility function to be some offset minus the testing loss, i.e., $U_B(D) = M - J(\theta(D), D_{\text{test}})$, where we use $\theta(D)$ to make the dependence of the trained parameter on the training data D explicit. Corollary 10 indicates that for differential and convex loss functions, two close data points have similar utility values. However, we are interested in clustering the data points based on their Shapley values. The next lemma says that the closeness of the utility implies the closeness of the Shapley value.

Lemma 5. *If* $U(S \cup \{z_i\}) - U(S \cup \{z_j\}) \leq \frac{\alpha}{|S|+1}\|x_i - x_j\|$, *then*

$$s_i - s_j \leq \alpha\|x_i - x_j\|\frac{S_{N-2}}{N-1} \tag{3.92}$$

where $S_{N-2} = \sum_{k=0}^{N-2} \frac{1}{k+1}$.

Proof of Lemma 5. By Lemma 3, we have

$$s_i - s_j \leq \frac{1}{N-1}\sum_{S\subseteq I\setminus\{i,j\}} \frac{1}{C_{N-2}^{|S|}}\frac{\alpha}{|S|+1}\|x_i - x_j\| \tag{3.93}$$

$$= \frac{1}{N-1}\sum_{k=0}^{N-2}\frac{\alpha}{k+1}\|x_i - x_j\| \tag{3.94}$$

$$\leq \alpha\frac{S_{N-2}}{N-1}\|x_i - x_j\| \tag{3.95}$$

□

We can now restate and prove the main theorem of this section:

Theorem 6. *Let* $J(\theta, D) = \frac{1}{N}\sum_{i=1}^{N} l(\theta^T x_i, y_i) + \Lambda R(\theta)$. *Let* D *and* D^* *be two data sets that differ by one point such that* $D^* \setminus \{(x^*, y^*)\} = D \setminus \{(x, y)\}$, $\theta = \operatorname{argmin}_\theta J(\theta, D)$ *and* $\theta^* = \operatorname{argmin}_\theta J(\theta, D^*)$ *be the corresponding ERM predictions. If* $R(\cdot)$ *is differentiable and* 1*-strongly convex, and* l *is convex and differentiable with the property that* $|l'(z)| \leq 1$ *and* $l'(z)$ *has the same sign for all* z. *Assume that the utility is* $U(D) = M - \frac{1}{N_{test}}\sum_{i=1}^{N_{test}} l(\theta^T x_{test,i}, y_{test,i})$ *and* $l(\cdot,\cdot) \leq M$, *then* $s(z_i) - s(z_i^*) \leq \frac{1}{N_{test}\Lambda}\sum_{i=1}^{N_{test}} \|x_{test,i}\|\|x_i - x_i^*\|\frac{1+\log(N-1)}{N-1}$.

Proof of Theorem 6. Set $\alpha = \frac{\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}}\|x_{\text{test},i}\|}{\Lambda}$. An application of Lemma 5 gives us

$$s(z_i) - s(z_i^*) \leq \frac{\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}}\|x_{\text{test},i}\|}{\Lambda}\|x_i - x_i^*\|\frac{S_{N-2}}{N-1}\,, \tag{3.96}$$

Using $S_{N-2} = \sum_{k=0}^{N-2}\frac{1}{k+1} \leq 1 + log(N-1)$ we retrieve the final form $s(z_i) - s(z_i^*) \leq \frac{\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}}\|x_{\text{test},i}\|}{\Lambda}\|x_i - x_i^*\|\frac{1+\log(N-1)}{N-1}$ of Theorem 6.

□

Proof of Theorem 2

Theorem 2. *Consider the value attribution scheme that assign the value* $\hat{s}(U,i) = C_U[U(S \cup \{i\}) - U(S)]$ *to user* i *where* $|S| = N-1$ *and* C_U *is a constant such that* $\sum_{i=1}^N \hat{s}(U,i) = U(I)$. *Consider two utility functions* $U(\cdot)$ *and* $V(\cdot)$. *Then,* $\hat{s}(U+V,i) \neq \hat{s}(U,i) + \hat{s}(V,i)$ *unless* $V(I)[\sum_{i=1}^N U(S \cup \{i\}) - U(S)] = U(I)[\sum_{i=1}^N V(S \cup \{i\}) - V(S)]$.

Proof. Consider two utility functions $U(\cdot)$ and $V(\cdot)$. The values attributed to user i under these two utility functions are given by

$$\hat{s}(U,i) = C_U[U(S \cup \{i\}) - U(S)] \tag{3.97}$$

and

$$\hat{s}(V,i) = C_V[V(S \cup \{i\}) - V(S)] \tag{3.98}$$

where C_U and C_V are constants such that $\sum_{i=1}^N \hat{s}(U,i) = U(I)$ and $\sum_{i=1}^N \hat{s}(V,i) = V(I)$. Now, we consider the value under the utility function $W(S) = U(S) + V(S)$:

$$\hat{s}(U+V,i) = C_W[U(S \cup \{i\}) - U(S) + V(S \cup \{i\}) - V(S)] \tag{3.99}$$

where

$$C_W = \frac{U(I) + V(I)}{\sum_{i=1}^N [U(S \cup \{i\}) - U(S) + V(S \cup \{i\}) - V(S)]} \tag{3.100}$$

Then, $\hat{s}(U+V,i) = \hat{s}(U,i) + \hat{s}(V,i)$ if and only if $C_U = C_V = C_W$, which is equivalent to

$$V(I)[\sum_{i=1}^N U(S \cup \{i\}) - U(S)] = U(I)[\sum_{i=1}^N V(S \cup \{i\}) - V(S)] \tag{3.101}$$

□

Proof of Sample Complexity for Permutation Sampling-Based Approximation

Let π_m be a random permutation of $D = \{z_i\}_{i=1}^N$ and each permutation has a probability of $\frac{1}{N!}$. Let $\hat{s}_{i,m} = U(P_i^{\pi_m} \cup \{i\}) - U(P_i^{\pi_m})$, we consider the following estimator of s_i:

$$\hat{s}_i = \frac{1}{M} \sum_{m=1}^M \hat{s}_{i,m} \tag{3.102}$$

Theorem 11. *Given the range of the utility function r, an error bound ϵ, and a confidence $1-\delta$, the sample size required such that*

$$P[\max_{i=1,\cdots,N} |\hat{s}_i - s_i| \geq \epsilon] \leq \delta \tag{3.103}$$

is

$$M \geq \frac{2r^2}{\epsilon^2} \log \frac{2N}{\delta} \tag{3.104}$$

Proof.

$$P[\max_{i=1,\cdots,N} |\hat{s}_i - s_i| \geq \epsilon] = P[\cup_{i=1,\cdots,N} \{|\hat{s}_i - s_i| \geq \epsilon\}] \tag{3.105}$$

$$\leq \sum_{i=1}^{N} P[|\hat{s}_i - s_i| \geq \epsilon] \tag{3.106}$$

$$\leq 2N \exp\left(-\frac{2M\epsilon^2}{4r^2}\right) \tag{3.107}$$

The first inequality follows from the union bound and the second one is due to Hoeffding's inequality.

Setting $2N \exp(-\frac{M\epsilon^2}{2r^2}) \leq \delta$ yields

$$m \geq \frac{2r^2}{\epsilon^2} \log \frac{2N}{\delta} \tag{3.108}$$

□

Algorithm 5 Baseline: Permutation Sampling-Based Approach

Input: $D = \{z_i\}_{i=1}^{N}$ - data, $U(\cdot)$ - utility function with range r, ϵ, δ - approximation error parameters

Output: $(\hat{s}_1, ...\hat{s}_N)$ - estimated Shapley values

```
M ← (2r^2/ε^2) log(2N/δ)
Initialize ŝ_i ← 0 for i = 1, .., N
for m = 1 to M do
    π_m ← GenerateUniformRandomPermutation(D)
    for i = 1 to N do
        P_i^{π_m} ← CalculateSetOfPrecedingUsers(π_m, i)
        ŝ_i ← ŝ_i + (1/M) (U(P_i^{π_m} ∪ {i}) − U(P_i^{π_m}))
    end for
end for
```

The permutation sampling-based method used as baseline in the experimental part of this work was adapted from Maleki et al. [109] and is presented in Algorithm 5.

Chapter 4

Mitigating Data Poisoning Attacks

4.1 Background

Machine learning (ML) models have been widely deployed in a multitude of applications, including image classification [72], speech recognition [35], etc. The advances of ML are mainly enabled by the availability of large and high-quality datasets. In practice, one often relies on public crowdsourcing services, such as Amazon Mechanical Turk, or private teams to collect training data. In both scenarios, an attacker can launch *data poisoning attacks*, in which malicious data are injected to the training dataset with the aim to degrade the performance of a model. Rather than being a hypothetical concern, such attacks have been reported in the wild against spam filters [121], face recognition systems [25], autonomous bots[1], among others. This requires us to re-think about the pipeline of constructing ML models and develop proper countermeasures that can mitigate various data poisoning attacks [95, 141].

The crux of defending against data poisoning attacks is to identify and characterize how poisoned examples are different from normal ones. Intuitively, this can be achieved by measuring the impact of each training instance on the performance of a trained model. In [121], the impact is assessed by testing the performance difference with and without a training instance. Although it is effective to detect poisoned instances, such way of assessing the impact of each instance is computationally intensive and often impractical, as it requires re-training for every instance in the training set. Various heuristics have been proposed to measure the influence of a training instance more efficiently. For example, one could argue that poisoned instances are artificial and thereby could be distant from the rest of instances in the training set. Other similar distance-based heuristics can be found in [125, 149]. While these heuristics induce minimal overhead to training processes, they often have limited capability of resolving the robustness issue, as we will show in the experiment section of the chapter. Rather than plugging a pre-processing step into the regular learning process to differentiate the poisoned instances from the normal ones, some existing works focused on developing models with "inborn" robustness [105, 57] and their robustness guarantees can

[1] https://www.wired.com/2017/02/keep-ai-turning-racist-monster/

often be formally established under certain assumptions. To implement the defenses in these works, it is, however, required to modify the training process, e.g., changing the loss function and the corresponding optimization procedure. In addition, these defenses are often designed for simple models, such as linear regression and logistic regression, and their use in more sophisticated models, such as deep neural networks, have not been investigated.

In this chapter, we propose an effective and computationally efficient method for mitigating data poisoning attacks based on a data sanitization mechanism, which screens off the susceptible instances prior to training. This data sanitization mechanism is agnostic to the actual learning algorithms and attack methods. The proposed mechanism is enabled by a game-theoretic formulation of a ML problem. We further leverage the Shapley value, a concept originated from cooperative game theory, to measure the impact of each training point on the model performance. Since the Shapley value is generally computationally expensive, except for a KNN classifier, we propose to use the KNN Shapley value as a proxy to distinguish poisoned instances for general ML models.

The contributions of this chapter can be summarized as follows.

- We propose a data sanitization mechanism by removing potential poisoned training instances based on the Shapley value.
- We provide theoretical justification of the data sanitization mechanism, relating the KNN Shapley value to the distribution of training examples in the geometric space and showing that for 1NN the data sanitization mechanism can achieve provable robustness against data poisoning attacks.
- We perform extensive experiments to show that the proposed data sanitation mechanism outperforms other state-of-the-art defense strategies against various data poisoning attacks.

4.2 Related Work

Poisoning Attacks

Depending on whether the attack aims at degrading the test accuracy indiscriminately or pertaining to specific examples, data poisoning attacks can be categorized in to untargeted vs. targeted ones. Untargeted poisoning attacks have been studied for various types of machine learning models, such as support vector machines [14], Bayes classifiers [121], collaborative filtering [100], and deep neural networks [119]. Since untargeted attacks only affect the test performance on a small set of examples but do not render the entire machine learning system useless, they are less detectable and thus arguably more dangerous than targeted ones. Effective strategies have been demonstrated in [25, 65] to cause a model to fail for special test examples; however, they make the assumption that the attacker can control the labeling process for instances in the training set, which excludes some real-world scenarios

where the training set is audited by human reviewers. Recent work [95, 141] has proposed clean label attacks, which generate poisoned instances that appear to be labeled correctly according to an expert observer and thus do not require the control over the labeling process.

Defense Against Poisoning Attacks

The study of model robustness against corrupted examples can be traced back to robust statistics [75], which provides a repertoire of concepts, such as breakdown points and influence functions, to characterize the robustness in a rigorous manner, as well as methods to robustify simple estimators, e.g., location, scale, etc.

Several ideas to defend against poisoning attacks for more complex models have emerged in recent work. [15] used ensemble methods to diminish the influence of poisoned data. [149] considered defenses that remove training points distant to the class centroids. While these methods are applicable to different types of models and attacks, they often demonstrate limited capability of defending against powerful attacks. There are studies related to defenses against specific models, such as linear regression [105, 26], logistic regression [57], and support vector machines [97]; nevertheless, defenses for deep neural networks have been rarely studied. [149] proposed a theoretical framework to evaluate the performance bound of a given defense method when assuming that the dataset is large and outliers in the clean data are harmless.

4.3 Attack Mitigation using the Shapley Value

Figure 4.1 presents an overview of the proposed framework to mitigate data poisoning attacks by making use of the Shapley value to identify the training instances that have a detrimental effect on a model's test performance.

Hereinafter, we will refer to the Shapley value computed via (3.4) and (3.5) as the KNN Shapley value. We assume the attacker has full knowledge of the clean training data and of the training algorithm and attempts to craft adversarial training instances that can flip a model's prediction on targeted test data. The data sanitization mechanism calculates the KNN Shapley value of each data instance in the poisoned training set and the utility function pertaining to the KNN Shapley value is defined with respect to all validation data. The mechanism then filters out the training data instances that have lowest Shapley values. The detailed implementation of the data sanitization mechanism is provided in Algorithm 6.

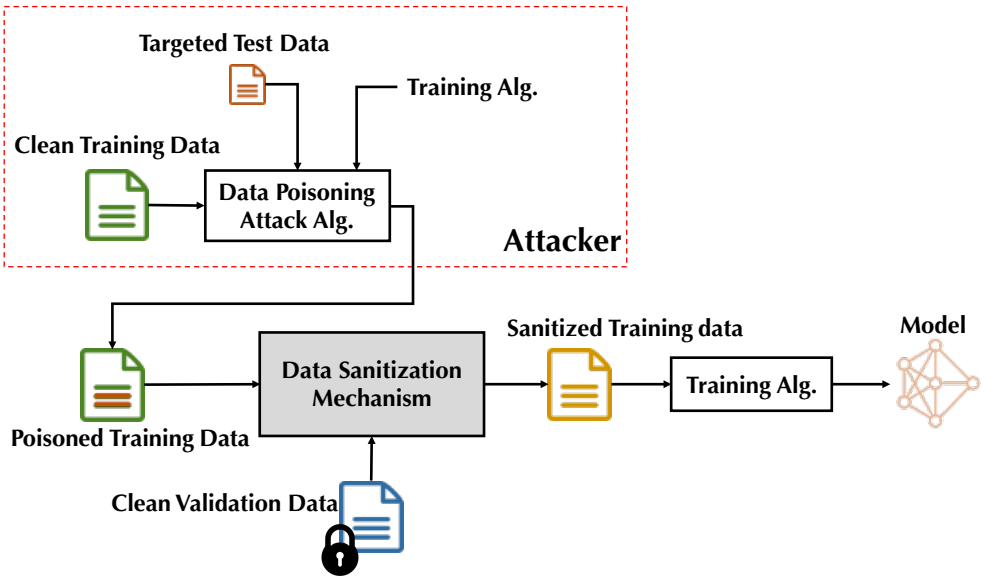


Figure 4.1: Overview of the proposed framework for mitigating data poisoning attacks.

Algorithm 6 Data Sanitization Mechanism based on the KNN Shapley value

Input: Poisoning training data $D_{\text{trn}} = \{(x_i, y_i)\}_{i=1}^{N}$, clean validation data $D_{\text{val}} = \{(x_{\text{val},i}, y_{\text{val},i})\}_{i=1}^{N_{\text{val}}}$, removal ratio $\gamma \in (0, 1)$

Output: Sanitized training data D_{san}

for $j \leftarrow 1$ to N_{val} **do**
  $(\alpha_1, ..., \alpha_N) \leftarrow$ Indices of training data in an ascending order using $d(\cdot, x_{\text{val}})$
  $s_{j,\alpha_N} \leftarrow \frac{\mathbb{I}[y_{\alpha_N} = y_{\text{val}}]}{N}$
  for $i \leftarrow N-1$ to 1 **do**
    $s_{j,\alpha_i} \leftarrow s_{j,\alpha_{i+1}} + \frac{\mathbb{1}[y_{\alpha_i} = y_{\text{val},j}] - \mathbb{1}[y_{\alpha_{i+1}} = y_{\text{val},j}]}{K} \frac{\min(K,i)}{i}$
  end for
end for
for $i \leftarrow 1$ to N **do**
  $s_i^{\text{avg}} \leftarrow \frac{1}{N_{\text{val}}} \sum_{j=1}^{N_{\text{val}}} s_{j,i}$
end for
$(\beta_1, \ldots, \beta_N) \leftarrow$ Indices of training data sorted in a descending order using s_i^{avg}, $i = 1, \ldots, N$
$D_{\text{san}} \leftarrow \{(x_{\beta_i}, y_{\beta_i})\}_{i=1}^{[N(1-\gamma)]}$

Theoretical Justification

The following theorem states that removing low KNN Shapley value training points are equivalent to removing the training points that are closest to the validation point but labeled

differently from it.

Theorem 12. *Let $s(z_i)$ be the Shapley value of the training point $z_i = (x_i, y_i)$ $(i = 1, \cdots, N)$ under the KNN utility defined in (3.3) for a given validation point $z_{val} = (x_{val}, y_{val})$. Let $s_{(i)}$ denote the ith-smallest Shapley value among all training points. Let D be the set of training points whose labels disagree with the label of the validation point and $z^D_{(i)}$ be the training point that is ith closest to x_{val} in D. Then, $s_{(i)} = s(z^D_{(i)})$.*

Proof. Without loss of generality, assume that x_i's are sorted in an ascending order according to the distance to x_{val}. We examine the change in the Shapley value with respect to the distance to the validation point. Note that two points with adjacent ranks have the same Shapley values if their labels are the same. Therefore, to trace the change in the Shapley value, we only need to consider x_i and x_{i+1} which have different labels. The Shapley value can either increase or decrease as a result of the label disagreement: (1) When $y_{i+1} = y_{\text{val}}$ and $y_i \neq y_{\text{val}}$, then $s_i - s_{i+1} = -\frac{\min(K,i)}{Ki} < 0$; (2) $y_{i+1} \neq y_{\text{val}}$ and $y_i = y_{\text{val}}$, then $s_i - s_{i+1} = \frac{\min(K,i)}{Ki} > 0$.

Now, we prove that the training points in D that are closer to the validation point have lower Shapley values. Consider two pairs of training points with adjacent ranks, (x_{i_1}, x_{i_1+1}) and (x_{i_2}, x_{i_2+1}), where $i_1 < i_2$, $y_{i_1+1} = y_{\text{val}} \neq y_{i_1}$, and $y_{i_2+1} = y_{\text{val}} \neq y_{i_2}$. Further, consider that for all adjacent pairs with indices between i_1 and i_2, there exists just one pair denoted by (x_{i_3}, x_{i_3+1}) such that $y_{i_1+1} \neq y_{\text{val}} = y_{i_1}$. Therefore, we have

$$s(x_{i_1}) = s(x_{i_1+1}) - \frac{\min(K, i_1)}{Ki_1} \tag{4.1}$$

$$s(x_{i_3}) = s(x_{i_3+1}) + \frac{\min(K, i_3)}{Ki_3} \tag{4.2}$$

$$s(x_{i_1+1}) = s(x_{i_3}) \tag{4.3}$$

$$s(x_{i_3+1}) = s(x_{i_2}) \tag{4.4}$$

Combining (4.1)-(4.4) yields

$$s(x_{i_1}) = s(x_{i_2}) - \frac{\min(K, i_1)}{Ki_1} + \frac{\min(K, i_3)}{Ki_3} \tag{4.5}$$

Since $i_3 > i_1$, $s(x_{i_1}) \leq s(x_{i_2})$. Further, due to (4.2)-(4.4), $s(x_j) \geq s(x_{i_1})$ for all $i_1 < j \leq i_2$. That is, in the KNN Shapley recursion from $i = N$ to $i = 1$, whenever there is a label disagreement between $i+1$ and i such that $y_{i+1} = y_{\text{val}} \neq y_i$, the Shapley value will be reduced to the lowest among all $\{x_j\}_{j=i}^N$. Therefore, ith-smallest Shapley value will be achieved by points that are i-th closest to the validation point among all the training points that have different labels from the validation point. □

Theorem 12 indicates that when the Shapley value is computed with respect to a single validation point, the corresponding data cleaning mechanism essentially removes the training points with a different label from its neighborhood. For the proposed mechanism, which

utilizes the Shapley value computed with multiple validation points, it can be expected that the removed training points will be the ones whose labels are mostly inconsistent with the nearby validation points.

For a 1NN classifier, we show that filtering out low KNN Shapley values leads to direct improvement of the model robustness.

Theorem 13. *Suppose that we remove the training points with low KNN Shapley values with respect to the validation point x_{val} and that the test point x_{test} falls into the radius r ball centered at x_{val}. Let x_{low} denote the point with lowest Shapley value among the remaining training points and x_{cl} denote the training point closest to x_{test} among the remaining training points. If $d(x_{low}, x_{val}) \geq 2r + d(x_{cl}, x_{val})$ and $y_{cl} = y_{val}$, then the 1NN prediction at x_{test} is y_{val}.*

Proof. Let $D = \{x_i^D\}$ be the set of remaining training points whose labels are different from the label of x_{val}. By Theorem 12, $d(x_{\text{low}}, x_{\text{val}}) \leq d(x_i^D, x_{\text{val}})$ for all $x_i^D \in D$. Hence, we have $d(x_i^D, x_{\text{val}}) \geq 2r + d(x_{\text{cl}}, x_{\text{val}})$. Let $\Delta = d(x_{\text{cl}}, x_{\text{val}})$. Since $d(x_{\text{test}}, x_{\text{val}}) \leq r$, by triangle inequality we have $d(x_{\text{test}}, x_{\text{cl}}) \leq d(x_{\text{test}}, x_{\text{val}}) + d(x_{\text{cl}}, x_{\text{val}}) \leq r + \Delta \leq d(x_{\text{val}}, x_i^D) - d(x_{\text{val}}, x_{\text{test}}) \leq d(x_{\text{test}}, x_i^D)$. As a result, the prediction at x_{test} is the y_{val}. □

Note that we can always ensure $d(x_{\text{low}}, x_{\text{val}}) \geq 2r + d(x_{\text{cl}}, x_{\text{val}})$ and $y_{\text{cl}} = y_{\text{val}}$ in Theorem 13 by removing a large enough number of low KNN Shapley value training points. It is plausible to assume that two points that are very close in the data (or feature) space have the same label. Therefore, when the data sanitization mechanism removes a "right" amount of training data such that $d(x_{\text{low}}, x_{\text{val}}) \geq 2r + d(x_{\text{cl}}, x_{\text{val}})$ is guaranteed for a small r, say, $r < r_0$ where r_0 is the maximum radius within which data points share the same label, Theorem 13 implies that the test point is always correctly predicted by the 1NN classifier built with the sanitized training points. On the other hand, if too many training points are removed, then $d(x_{\text{low}}, x_{\text{val}})$ will be large, which, in turn, makes $d(x_{\text{low}}, x_{\text{val}}) \geq 2r + d(x_{\text{cl}}, x_{\text{val}})$ hold for a relatively large r. When $r > r_0$, some test points around x_{val} will be misclassifed. As a result, it is important to choose a proper removal ratio in the data sanitization mechanism. In the next section, we will provide some guidance in selecting the removal ratio by drawing insights from empirical studies on various datasets.

4.4 Evaluation

In this section we will compare the proposed Shapley value based data sanitization mechanism with the state-of-the-art defenses against various attacks.

Attack methods. We focus on the targeted attacks and construct data poisoning instances using two state-of-the-art attack methods against deep neural networks, presented in [95] and [141]. We will refer to these methods as the influence function based (IFB) method and the poisoning frog (PF) method, respectively. The IFB method uses influence

functions, a tool originated from robust statistics for measuring the change in the learning loss as a result of the change in a training data, and crafts poisoning instances by adding visually imperceptible noise to clean training data. The PF method generates poisoned instances by moving some normal instances in the targeted class toward the targeted test point; consequently, during training, the decision boundary is rotated to include the poisoned instances, which may inadvertently misclassify the nearby targeted test point.

Defense baselines. We compare the proposed data sanitization mechanism with three baselines, including *random removal*, which randomly selects and removes a subset of training points, and two more sophisticated defense methods proposed in [149], namely, the *sphere defense*, which removes points outside a spherical radius of the centroid of each class, and the *slab defense*, which first projects points onto the line between the centroids and then discards points that are too far on this line.

Datasets, Model and Protocol. We perform experiments on the same neural network and datasets as those of [95] and [141]. Our neural network uses the pretrained Inception-V3 as the feature extractor, which produces a deep feature representation with 2048 dimensions. We train the neural network and evaluate the performance of the defense strategies on three datasets with different sizes: ImageNet (dog vs fish), which is also used in [95] and [141], and two new ones including MNIST (1 vs 7), CIFAR-10 (birds vs airplane). Our experiment setup is also similar to that of [95] and [141]: We use 900 images per class for training and 600 images per class for testing. In order to construct a clean validation data needed for the data sanitization mechanism, we randomly select another 300 images per class from the training set and ensure that the validation set is balanced. We choose $K = 6$ for computing the KNN Shapley value.

Results and Discussion

Comparison with Existing Defense Strategies. We will refer to our proposed defense strategy as the *Shapley Value enabled (SVE)* defense. Table 4.1 compares our defense to various baseline methods against the IFB attacks. Without defense, the IFB attack can flip the prediction of $56\%, 75\%, 100\%$ of the total 588 images which are classified correctly in the test set by injecting only 1, 2, 10 training data, respectively. Table 4.1 shows that our defense significantly outperforms the baseline methods. Table 4.2 demonstrates the performance of different defenses under "severe" poisoning, where 30% of the training set is poisoned. In that case, the labels of all test images can be manipulated. Our method is proved to be effective under this "severe" poisoning and reduces the attack rate to lower than 7% for MNIST and ImageNet and 32% for CIFAR, while others methods fail to provide acceptable defenses. The PF attack only adds one poisoned instance to the training set. Without defenses, this attack can manipulate the predictions at all training points with only one poisoned instance for all datasets considered in our paper. The comparison of our method

with other defenses is presented in Table 4.3. We can see that our method mitigates the PF attack more effectively, compared with the rest. Figure 4.2 illustrates a training image before and after being poisoned, as well as the change in the prediction accuracy for a test image. We can see that the poisoned image is visually indistinguishable from a normal one, but can significantly change the classifier's prediction at the given test image.

Table 4.1: Comparison of various defense strategies against the IFB attack method with $\gamma = 1/6$.

Datasets	#/Poisoned Instances	Defense Strategies			
		Random	Sphere	Slab	SVE
MNIST	1	11%	8%	6%	**1.8%**
	2	25%	21%	10%	**1.8%**
	10	22%	22%	11%	**2.2%**
CIFAR	1	40%	35%	33%	**9%**
	2	49%	42%	40%	**9.4%**
	10	56%	53%	52%	**9.6%**
ImageNet	1	26%	20%	13%	**1%**
	2	30%	25%	15%	**1.2%**
	10	34%	28%	18%	**1.2%**

Choosing γ. We study the attack success rate under different values of removal ratio γ and showcase the results in Figure 4.3. Here, the unattacked model can successfully classify 588/600 test images in ImageNet dataset, 598/600 in MNIST and 499/600 in CIFAR10. Our defense can greatly mitigate the attacks for a large range of γ. As γ increases, the defense is less effective due to the fact that more normal data are removed simultaneously. Despite the dependence of the defense effectiveness on the removal ratio, the defense can always reduce the attack success rate for all $\gamma \in (0, 0.5]$. In Figure 4.3 (d), we exclude $\gamma = 0$ from the result to zoom into the region with low attack success rates. The figure shows that $1/20 \sim 1/3$ is a robust choice for γ across all datasets considered herein.

Separability of normal vs. poisoned Shapley values. Figure 4.4 illustrates the distributions of the Shapley values of normal instances and those of poisoned ones. We also indicate the value used for differentiating poisoned examples from normal ones in the data sanitization mechanism using a vertical line. In general, poisoned data have lower Shapley values than normal ones. Since normal and poisoned instances are not perfectly separable with the Shapley value, the threshold filters out poisoned data in tandem with almost the same amount of normal data.

Effect of removing normal data. To examine the impact of removing normal data on the model performance, we conduct experiments that remove the training instances with low

Table 4.2: Comparison of various defenses against "severe" IFB attacks which poison 30% of the training instances.

Dataset	Random	Sphere	Slab	SVE
ImageNet	60%	59%	58%	**6.2%**
CIFAR10	96%	90%	86%	**32%**
MNIST	43%	40%	38%	**4.8%**

Table 4.3: Comparison of various defense strategies against the PF attack method with $\gamma = 1/6$.

Datasets	Random	Sphere	Slab	SVE
MNIST	16%	15%	10%	**1.4%**
CIFAR	32%	31%	23%	**8.5%**
ImageNet	28%	26%	21%	**2.4%**

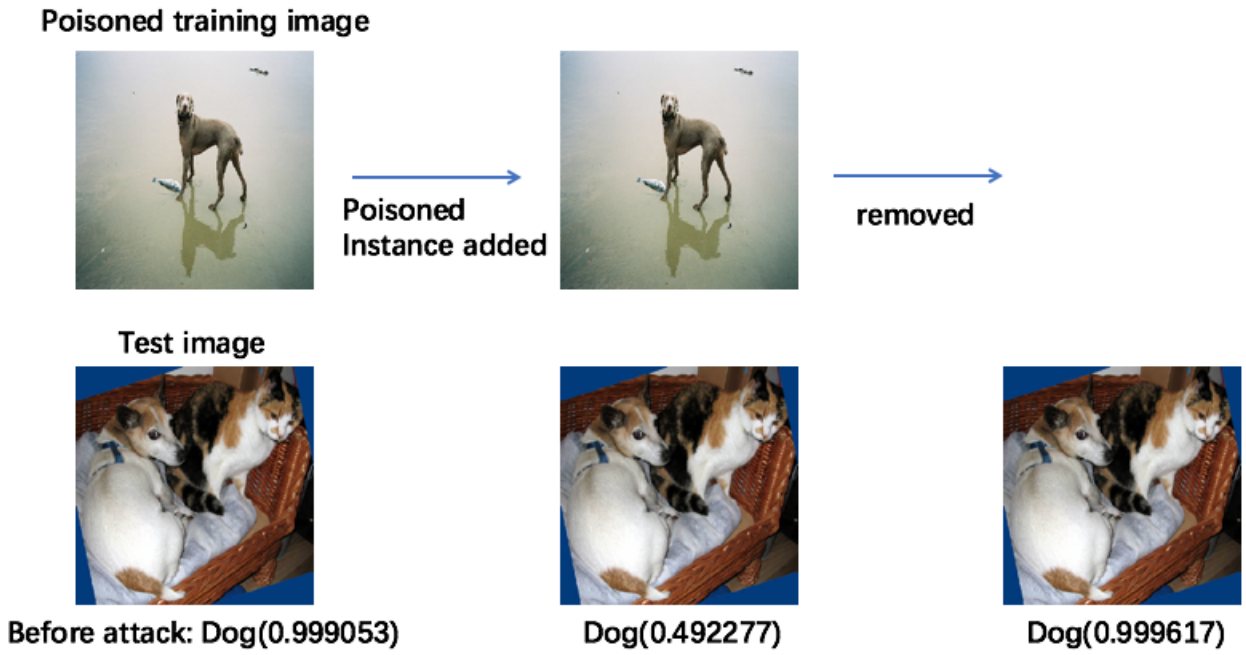


Figure 4.2: Illustration of a training image before and after being poisnoned, as well as the change in the prediction accuracy for a test image.

Shapley values from a normal dataset and use the rest for training. Figure 4.5 demonstrates the model performance when different number of training instances are removed and compares

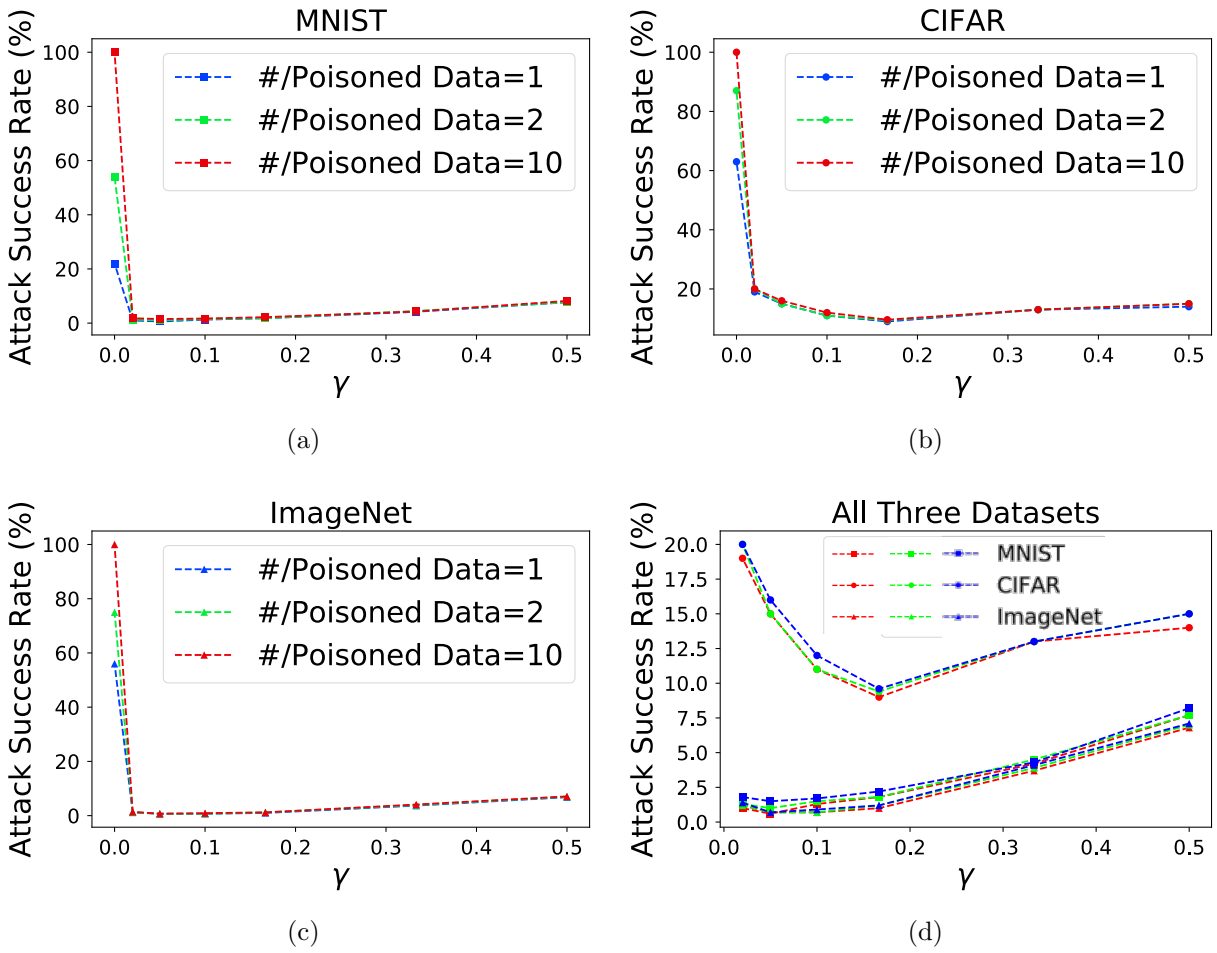


Figure 4.3: The effect of γ on the attack success rate, demonstrated on (a) MNIST, (b) CIFAR and (c) ImageNet Dataset. In (d), we zoom in to the range where $\gamma > 0$ in order to examine the optimal choice of γ.

it with a random removal baseline. It is shown that removing a moderate amount of normal training data (e.g., $\gamma = 0.29$) can even help train a model with better generalization capability.

4.5 Chapter Summary

This chapter proposed a data sanitization mechanism that can help mitigate data poisoning attacks. Our mechanism is able to be plugged into different ML systems with little overhead and does not require modification of the original learning algorithm used in the ML system. Through experiments on various datasets, we show that the mechanism can effectively distill out useful training instances from a poisoned training set.

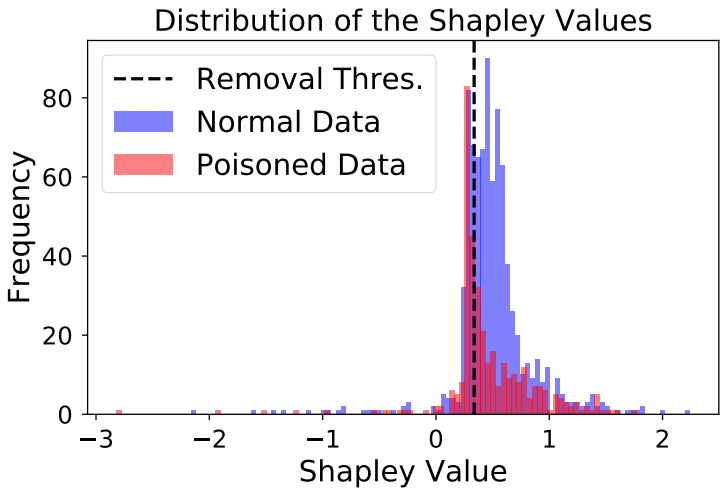
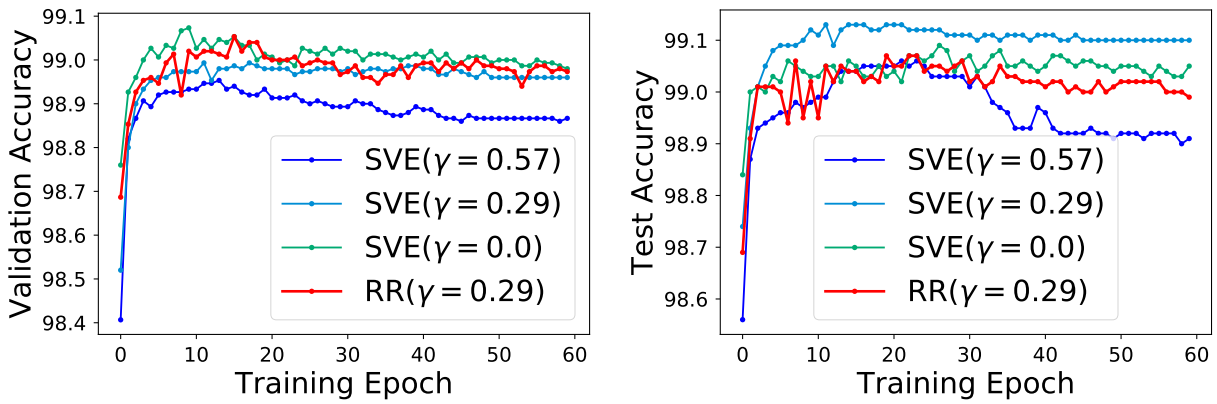


Figure 4.4: Distribution of the Shapley value of a poisoned dataset with 30% of poisoned instances. The dashed line illustrates the threshold corresponding to $\gamma = 0.3$.



(a) (b)

Figure 4.5: Model performance in terms of (a) validation accuracy and (b) test accuracy when removing different proportions $\gamma = 0, 0.29, 0.57$ of normal training data with low Shapley values. The performance is compared with a baseline (RR) that randomly removes $\gamma = 0.29$ of the training data.

Note that the proposed mechanism requires the knowledge of the parameters of a trained feature extractor, although this does not seem a stringent requirement due to the existence of myriad pre-trained feature extractors. For future work, we are interested in exploring defense strategies that no longer enforce such requirements, i.e., "black-box" defenses. These defenses might be of particular interest to the contexts where the owner of the model is worried about

leaking intellectual property information to a third party security company that offers the defense service.

Part II

Privacy Protection in CPS

Chapter 5

Optimal Sensing-Control Co-design for Privacy

5.1 Background

Occupancy sensing forms a core aspect of the fabric of modern thermostats. Nest [1], as a popular example, exploits occupancy information derived from the built-in motion sensor or tracking users' smartphone location to automatically switch thermostat behaviors for increased energy saving and better thermal comfort. The collection of users' activity data naturally causes the concern about privacy. Currently, the thermostat companies rely on the encryption of data transmission to guard against malicious intercept of private data, and the institution of privacy policies to provide users with "notice and choice" [34]. However, neither of these measures will prevent users' private data from being eavesdropped by an insider who has legitimate access to the data.

In recent years, privacy-differentiated goods have surfaced as a way to address the tension between privacy requirements and data utility expectation of users. For instance, AT&T has announced a price model that allows users to pay for an opt-out from the default setting where their web browsing activities are wiretapped and utilized for targeted advertisement; Telematics offers a more cost-effective car insurance plan to users who are willing to share their driving data. Inspired by these applications, we prototype a privacy-differentiated occupancy sensing module that adjusts the precision of occupancy data revealed to the controller in response to users' priority of energy saving, thermal comfort, and privacy.

Building upon the example of occupancy-based thermostat control, we consider a more general formulation which is referred to as private-input-driven system shown in Fig. 5.1. The system contains a local plant whose dynamics are influenced by a private input process. In the thermostat control example, the private input process may represent users' presence. The control of the plant is accomplished by a controller held by an honest but curious third party. The controller acquires the knowledge about the plant state and issues the control demands based on the noisy observations from one sensor that measures the plant state and

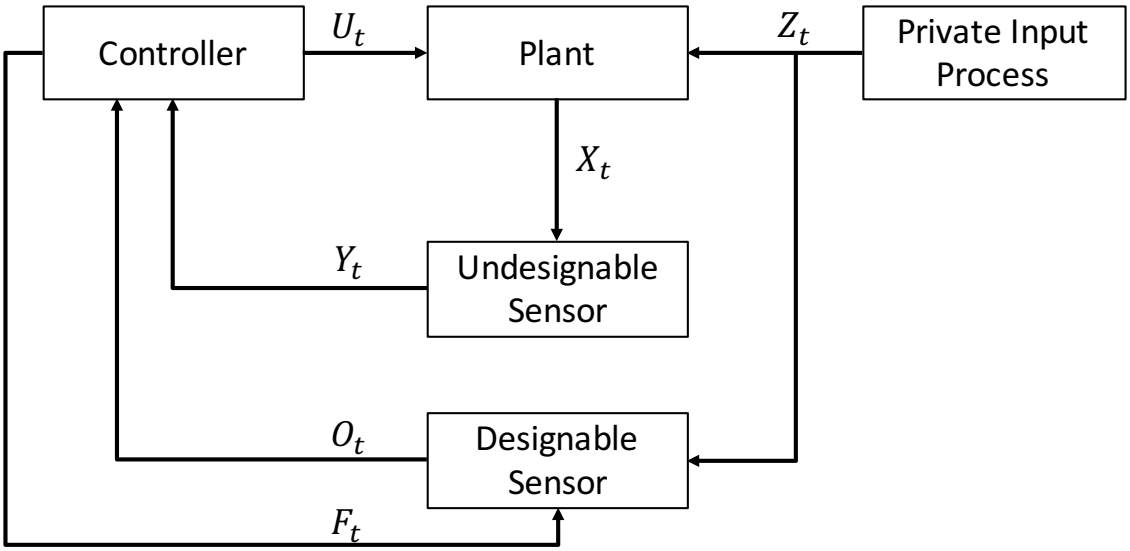


Figure 5.1: Private-input-driven system diagram.

another that measures the private input process. To accommodate the privacy needs, the controller supplies another control signal to the sensor that directly monitors the private input process, and tailors the sensor configuration according to users' privacy preference.

A quantification scheme of privacy leakage is essential to modeling and analysis of privacy-preserving systems. Two popular privacy metrics have been explored in the context of dynamical systems. One is differential privacy [98], which characterizes the change in the probability distribution of observables of a system by adding any single user's data; the other is information theoretic privacy [129, 79], which captures the reduction of adversary's uncertainty about private data after observing publicly available measurements.

Differential privacy relies on hiding an individual's private parameters in an aggregate of several different users, often known as "hiding in the crowd" [48]. It also has a one-size-fits-all privacy model, where the overall system provides the same level of privacy to all users. As we are interested in customizing the privacy loss for individual users instead of database-level privacy, information theoretic privacy is more suitable to our context. However, current information theoretic measures of time series data [51, 139] are not amendable to the optimal control framework where the instantaneous privacy leakage from sequentially generated data needs to be properly characterized. Mutual information or directed information between private time series and public measurements are used in [51, 139] as privacy metrics; however, these metrics essentially compute the expected privacy leakage over publicly measured random variables, and do not take into account the fact that at time t the public measurements before t have already been realized rather than remaining random.

The contributions of this chapter are three-fold. Firstly, we propose a measure for instantaneous privacy loss that captures the privacy leakage of time series data on-the-fly.

An analytic form of the privacy loss is derived for a linear Gaussian (LG) systems. We also develop an approximation framework for computing the privacy loss for more general dynamical systems. Secondly, we formulate the problem of finding optimal plant and measurement control policies for the private-input-driven systems using the framework of Partially Observed Markov Decision Processes (POMDPs), and give the optimal policies via dynamical programming. We further prove the separation of measurement control, plant control, and state estimation for LG systems with quadratic plant control cost. Thirdly, we leverage the techniques developed to investigate the trade-off between privacy, comfort and energy performance for occupancy-based thermostat control.

The problem studied in this chapter has a close connection to some research work conducted in the control and information theory communities. A popular topic in control studies the case where the information used for controller's decision making consumes resources that need to be explicitly dealt with, such as bandwidth, power, or delay incurred by potential network traffic. A line of research [154] focuses on control under channel capacity constrained channels, which also results in an information theoretic constraint similar to the privacy loss proposed herein. However, it is often assumed that the sensor model and/or channel model is given a priori. This is different from our objective which is to jointly design the controller and sensor/channel. Another line of research, so-called "optimal sensing", coincides with our objective. The seminal work in [113] considers an adaptive measurement system where control is available for both the plant and measurement subsystem, and proves that the optimization of plant control can be carried out independently of the measurement control optimization for linear Gaussian system with quadratic costs. More recent work has focused on optimal measurement control [168] in which there are a set of sensors with different levels of precision and operation costs and the controller can access one sensor at a time to receive observation. The main results provided in [113, 168] assume the cost is only a function of the state and control actions. However, in our case, the privacy cost depends also on the sensor observations. [153] adopts a similar information constraint to the proposed privacy loss in this chapter, and studies the problem of joint sensor and controller design. Our work differs from [153] in that we only consider a partial state to be private which is modeled as private input process. In addition, previous work has largely focused on LGQ control problems as it leads to separation of control and estimation and thereby an analytic solution. The chapter tries to tackle the general dynamical systems and provides an approximation framework to compute the optimal control policy. The chapter is also partially inspired by [157] which considers the state of a system as being private and studies the privacy-aware control in a POMDP setting. However, the privacy is protected by randomizing the control action instead of adding random disturbance to sensory data as in this chapter.

5.2 Problem Formulation

Notation. Throughout the chapter, we will use capital letters to indicate random variables and lower case ones to refer to the realizations of the corresponding random

variables. $X_{0:t}$ is used as a shorthand for $\{X_0, \cdots, X_t\}$.

Consider the discrete-time control problem depicted in Fig. 5.1. The dynamics of the plant are described by

$$X_{t+1} = f_t^X(X_t, Z_t, U_t, W_t^X) \tag{5.1}$$

where f_t^X is some function of possibly nonlinear form, $\{X_t\}$ is a plant state process, $\{U_t\}$ is a plant control process, $\{W_t^X\}$ is an independent and identically distributed (i.i.d.) disturbance. Additionally, the dynamics of the plant are steered by an exogenous input process $\{Z_t\}$, which represents personal attributes or behaviors and is thus *privacy-sensitive*. $\{Z_t\}$ is assumed to evolve according to the known dynamics given by

$$Z_{t+1} = f_t^Z(Z_t, W_t^Z) \tag{5.2}$$

where W_t^Z are i.i.d. random disturbance of private input process $\{Z_t\}$. The state of the plant is observed via an *undesignable* noisy sensor

$$Y_t = h_t^X(X_t, V_t^X) \tag{5.3}$$

where h_t^X represents the measurement model for the plant, and $\{V_t^X\}$ is a i.i.d measurement noise process. To cater for the needs of privacy, the measurement of the private input process $\{Z_t\}$ is regulated by a *designable* sensor. The control over the designable sensor is denoted by $\{F_t\}$, and the noisy observation for private input is thus given by

$$O_t = h_t^Z(Z_t, F_{t-1}, V_t^Z) \tag{5.4}$$

where h_t^Z represents the measurement model for the private input, and $\{V_t^Z\}$ is the i.i.d. measurement noise.

The controller can get access to noisy measurements for the plant and private input and its past control actions. We denote the information available to the controller at time t by $I_t = (Y_{0:t}, O_{0:t}, U_{0:t-1}, F_{0:t-1})$. The controls are assumed to be a deterministic function of the available information. Let the control policies for the plant and designable measurement system be μ and α, respectively. Then we have $U_k = \mu(I_k)$ and $F_k = \alpha(I_k)$.

Our aim is to find the optimal balance between the privacy loss due to measurements and the savings in plant operation costs made possible by the measurements. Suppose the control horizon is T, the plant control cost associated with policy $\eta = (\mu, \alpha)$ is

$$C^\eta = \sum_{t=0}^{T-1} c_t(X_t^\eta, Z_t^\eta, U_t^\eta) + c_T(X_T^\eta, Z_T^\eta) \tag{5.5}$$

where the superscript η indicates that the corresponding random processes are induced by policy η. c_t represents the instantaneous cost at time t. Let the total privacy loss during horizon T be denoted by G^η, our objective is to solve

$$\inf_\eta \mathbb{E}[(1-\gamma)C^\eta + \gamma G^\eta] \tag{5.6}$$

for a desired weight γ. As γ is increased from 0 to 1, the controller gradually changes from being completely utility driven to one prioritizing privacy.

We note the closed-loop system under control law $\eta = (\mu, \alpha)$ determines the information available I_t^η as well as the dynamics and costs. In the following sections, when we are analyzing the performance of a fixed controller η, or considering the effects of particular control actions (u_t, f_t), we will suppress the η or (u_t, f_t) dependence for notational cleanliness when the context is clear.

5.3 Inferential Privacy

In this section, we present a statistical inference view to capture the privacy threat incurred by releasing streaming noisy measurements to achieve certain control objectives.

Threat model

We start by defining the threat model. An adversary observes I_T by constantly eavesdropping the communication link between the controller and the local infrastructure including the plant and sensors. The adversary is interested in inferring the private input process $Z_{0:T}$ from I_T. To formalize the privacy threat model, we assume the standard statistical inference threat model in [137].

Definition 2. *(Inference attack). An inference attack on $Z_{0:T}$ given the observation I_T takes as input the joint distribution $p(Z_{0:T}, I_T)$ and the observation $I_T = i_T$, and outputs a probability distribution q^* defined over the domain of $Z_{0:T}$, as the solution to the minimization*

$$q^* = \arg\min_q \sum_{z_{0:T}} p(z_{0:T}|i_T)\Psi(z_{0:T}, q) \tag{5.7}$$

for some cost function $\Psi(z_{0:T}, q)$.

Remark 14. *$\Psi(z_{0:T}, q)$ is a generic notation for inference loss function. E.g., $\Psi(z_{0:T}, q) = -\log q(z_{0:T}|i_T)$ if the logarithm loss is used. The inference attack generates a belief distribution q over the private process $Z_{0:T}$ given the observation I_T by minimizing the expected inference loss.*

We proceed to define the privacy leakage as follows: Without observing i_T, the adversary's belief about the private process can be captured by the solution of the minimization

$$\Psi_0^* = \min_q \sum_{z_{0:T}} p(z_{0:T})\Psi(z_{0:T}, q) \tag{5.8}$$

After observing i_T, the adversary would update the belief q such that it minimizes

$$\Psi_{i_T}^* = \min_q \sum_{z_{0:T}} p(z_{0:T}|i_T)\Psi(z_{0:T}, q) \tag{5.9}$$

Note that $\Psi^*_{i_T}$ is determined by the realization i_T, whereas we wish to consider the average privacy loss across the distribution of I_T. In order to quantify how much an adversary gains in terms of inference of the private process $Z_{0:T}$ by virtue of observing I_T, we consider the average cost gain

$$\Delta\Psi = \Psi^*_0 - \sum_{i_T} p(i_T)\Psi^*_{i_T} \tag{5.10}$$

Definition 3. *(Expected total privacy loss). The expected total privacy loss of $Z_{0:T}$ from the observation I_T is given by $\Delta\Psi$.*

The log-loss is used as the loss function in recent work, e.g., [129], to compute the privacy loss, as it results in a natural measure of relevance - mutual information.

Proposition 15. *If an adversary uses the log-loss $\Psi(Z_{0:T}, q) = -\log q(Z_{0:T})$, the total expected privacy loss of $Z_{0:T}$ from I_T is equivalent to the mutual information between $Z_{0:T}$ and I_T, i.e.,*

$$\Delta\Psi = \mathbb{I}(Z_{0:T}; I_T) \tag{5.11}$$

and $\mathbb{I}(Z_{0:T}; I_T) \triangleq \mathbb{H}(Z_{0:T}) - \mathbb{H}(Z_{0:T}|I_T)$ represents the mutual information between the two sequences $Z_{0:T}$ and I_T.

In this chapter, we will also use mutual information as a measure of privacy loss. Mutual information is shown to be the *only* measure that satisfies the data processing axiom which is needed to properly define the benefit of side information in an inference problem [81]. In addition, mutual information is closely related to the probability of success of an inference algorithm used by the adversary via Fano's Inequality. Let $\hat{Z}_{0:T}$ be the adversary's inference based on the observation I_T. Assuming Z_t has a finite alphabet size denoted by $|Z|$, Fano's Inequality states a lower bound on the probability of inference error

$$P(Z_{0:T} \neq \hat{Z}_{0:T}) \geq \frac{\mathbb{H}(Z_{0:T}) - \mathbb{I}(Z_{0:T}; I_T) - 1}{T \log |Z|} \tag{5.12}$$

It is clear from (5.12) that the bound on the probability of error is maximized when $\mathbb{I}(Z_{0:T}; I_T) = 0$, i.e., the two sequences are independent.

Instantaneous privacy loss

Solutions to classical POMDPs reduce the full horizon policy optimization to the Bellman equation that finds the optimal control at each single time step iteratively. A key element in the reduction is the additive form of the classical cost function definitions. To benefit from the Bellman-type reduction, the following lemma presents a stepwise decomposition of total expected privacy loss.

Lemma 6. *For the dynamical model in (5.1) and (5.2) and the measurement model in (5.3) and (5.4), assuming η is deterministic, $\mathbb{I}(Z_{0:T}; I_T)$ can be decomposed into the sum of*

$$\mathbb{I}(Z_{0:T}; I_T) = \sum_{t=0}^{T} \mathbb{I}(Z_{0:t}; Y_t, O_t | I_{t-1}) \tag{5.13}$$

Proof. Note that $I_T = (I_{T-1}, Y_T, O_T, U_{T-1}, F_{T-1})$, hence

$$\begin{aligned}
\mathbb{I}(Z_{0:T}; I_T) &= \mathbb{I}(Z_{0:T}; I_{T-1}, Y_T, O_T, U_{T-1}, F_{T-1}) && (5.14)\\
&= \mathbb{I}(Z_{0:T-1}; I_{T-1}) + \mathbb{I}(Z_T; I_{T-1} | Z_{0:T-1}) \\
&\quad + \mathbb{I}(Z_{0:T}; Y_T, O_T, U_{T-1}, F_{T-1} | I_{T-1}) && (5.15)\\
&= \mathbb{I}(Z_{0:T-1}; I_{T-1}) \\
&\quad + \mathbb{I}(Z_{0:T}; Y_T, O_T, U_{T-1}, F_{T-1} | I_{T-1}) && (5.16)\\
&= \mathbb{I}(Z_{0:T-1}; I_{T-1}) + \mathbb{I}(Z_{0:T}; Y_T, O_T | I_{T-1}) && (5.17)
\end{aligned}$$

where (5.15) is by applying the chain rule for mutual information, (5.16) follows from the fact that Z_T is conditionally independent of I_{T-1} given $Z_{0:T-1}$, and (5.17) follows from the presumption that U_{T-1} and F_{T-1} are a deterministic function of I_{T-1}. We can recursively break down $\mathbb{I}(Z_{0:T-1}; I_{T-1})$ in a similar fashion to obtain (5.13). □

Definition 4. *(Instantaneous privacy loss). The instantaneous privacy loss at time t due to the observation $I_t = i_t$ is given by*

$$g_t(i_t, f_t, u_t) = \mathbb{I}(Z_{0:t+1}; Y_{t+1}, O_{t+1} | I_t = i_t) \tag{5.18}$$

Remark 16. *It is obvious that the RHS of the equation above is a function of i_t. The reason the RHS also depends on f_t and u_t is that the distribution of Y_{t+1} and O_{t+1} hinges on the control actions exerted from time 0 to t, while the dependence on $\{u_i, f_i\}_{i=0}^{t-1}$ are implicitly encapsulated into i_t. G^η in (5.6) is thus defined as $\sum_{t=0}^{T-1} g_t(i_t, f_t, u_t)$.*

We can represent the instantaneous privacy loss in the form of entropy difference:

$$\mathbb{H}(Z_{0:t+1} | I_t = i_t) - \mathbb{H}(Z_{0:t+1} | Y_{t+1}, O_{t+1}, I_t = i_t) \tag{5.19}$$

from which we can obtain an intuitive interpretation of the instantaneous privacy loss at time t - it indicates the anticipated reduction in adversary's uncertainty about the private process up to time $t+1$ due to the upcoming measurements given all the information received up to time t.

Special Case: Linear Gaussian System

For a general dynamical system stated in Section 5.2, the instantaneous privacy loss is a function of the information available to the controller as well as the plant and measurement

control policy. However, if the plant and the measurement systems are linear, and the disturbance and measurement noise are Gaussian, then the instantaneous privacy loss at each time step is independent of plant control policy.

Suppose $X_t \in \mathbb{R}^{n_X}$, $Z_t \in \mathbb{R}^{n_Z}$, $U_t \in \mathbb{R}^{n_U}$, $Y_t \in \mathbb{R}^{n_Y}$, $O_t \in \mathbb{R}^{n_O}$, and the LG system is described by the following equations:

$$X_{t+1} = A_t^X X_t + A_t^{XZ} Z_t + B_t^X U_t + W_t^X \tag{5.20}$$
$$Z_{t+1} = A_t^Z Z_t + W_t^Z \tag{5.21}$$
$$Y_t = C_t^X X_t + V_t^X \tag{5.22}$$
$$O_t = C_t^Z Z_t + V_t^Z \tag{5.23}$$

where $W_t^X \in \mathbb{R}^{n_X}$, $W_t^Z \in \mathbb{R}^{n_Z}$, $V_t^X \in \mathbb{R}^{n_Y}$ and $V_t^Z \in \mathbb{R}^{n_O}$ are zero-mean Gaussian random variables with covariance M_t^X, M_t^Z, N_t^X and N_t^Z, respectively. N_t^Z and C_t^Z are determined by $F_{t-1} \in \mathbb{R}^{n_F}$.

We introduce the following notations. Let

$$A_t = \begin{bmatrix} A_t^X & A_t^{XZ} \\ \mathbf{0}_{n_Z \times n_X} & A_t^Z \end{bmatrix} B_t = \begin{bmatrix} B_t^X \\ \mathbf{0}_{n_Z \times n_U} \end{bmatrix} \tag{5.24}$$
$$C_t = \begin{bmatrix} C_t^X & C_t^Z \end{bmatrix} \tag{5.25}$$
$$M_t = \begin{bmatrix} M_t^X & \mathbf{0}_{n_X \times n_X} \\ \mathbf{0}_{n_Z \times n_Z} & M_t^Z \end{bmatrix} N_t = \begin{bmatrix} N_t^X & \mathbf{0}_{n_Y \times n_Y} \\ \mathbf{0}_{n_O \times n_O} & N_t^Z \end{bmatrix} \tag{5.26}$$

where $\mathbf{0}_{\cdot \times \cdot}$ stands for a zero matrix and the associated subscripts represent its dimension.

Proposition 17. *The instantaneous privacy loss induced by measurement control policy α at time t of the LQ system presented in (5.20)-(5.23) is given by*

$$g_{t,lin} = \frac{1}{2} \log \frac{|\Sigma_{t+1|t,sub}|}{|\Sigma_{t+1|t+1,sub}|} \tag{5.27}$$

where $\Sigma_{t+1|t,sub}, \Sigma_{t+1|t+1,sub} \in \mathbb{R}^{m \times m}$ are the lower-right submatrices of $\Sigma_{t+1|t}$ and $\Sigma_{t+1|t+1}$, respectively, which are the error covariance matrices in Kalman filter and can be iteratively computed from

$$\Sigma_{t+1|t} = A_t \Sigma_{t|t} A_t' + M_t \tag{5.28}$$
$$\Sigma_{t+1|t+1} = \Sigma_{t+1|t} - \Sigma_{t+1|t} C_{t+1}' (C_{t+1} \Sigma_{t+1|t} C_{t+1}' + N_{t+1})^{-1} C_{t+1} \Sigma_{t+1|t} \tag{5.29}$$

Proof. Since

$$Cov(X_{t+1}, Z_{t+1} | I_t = i_t) = \Sigma_{t+1|t} \tag{5.30}$$
$$Cov(X_{t+1}, Z_{t+1} | I_{t+1} = i_{t+1}) = \Sigma_{t+1|t+1} \tag{5.31}$$

and the fact that $p(X_{t+1}, Z_{t+1}|I_t = i_t)$ and $p(X_{t+1}, Z_{t+1}|I_{t+1} = i_{t+1})$ are Gaussian distributions, we have

$$Cov(Z_{t+1}|I_t = i_t) = \Sigma_{t+1|t,sub} \tag{5.32}$$
$$Cov(Z_{t+1}|I_{t+1} = i_{t+1}) = \Sigma_{t+1|t+1,sub} \tag{5.33}$$

It follows that

$$H(Z_{t+1}|I_t = i_t) = \frac{1}{2}\log(2\pi e)^{n_Z}|\Sigma_{t+1|t,sub}| \tag{5.34}$$
$$\begin{aligned} &H(Z_{t+1}|Y_{t+1}, O_{t+1}, I_t = i_t) \qquad (5.35)\\ &= \sum_{y_{t+1},o_{t+1}} p(y_{t+1}, o_{t+1}|i_t)H(Z_{t+1}|i_{t+1}) \\ &= H(Z_{t+1}|i_{t+1}) \\ &= \frac{1}{2}\log(2\pi e)^{n_Z}|\Sigma_{t+1|t+1,sub}| \end{aligned}$$

where the second equality in (5.35) is because $\Sigma_{t+1|t+1,sub}$ and thereby $H(Z_{t+1}|i_{t+1})$ are not functions of the observations received, i.e., y_{t+1} and o_{t+1}. Then by (5.19) and simple algebraic manipulation, we complete the proof. □

Remark 18. *From (5.27) we can see that $g_{t,lin}$ only depends on the measurement control signal $(f_0, \cdots, f_t)$. This is because the recursive structure of $\Sigma_{t+1|t+1}$ makes it a function of $N_0, \cdots, N_{t+1}$ and $C_0, \cdots, C_{t+1}$, which, in turn, hinge on $(f_0, \cdots, f_t)$. An important consequence of this result, as we will prove in the later section, is that the measurement control policy can be designed separately from the plant control policy.*

Privacy Loss Approximation for General Control Systems

Unlike LG systems, a general dynamical system does not enjoy an analytic form of privacy loss. Direct evaluation of the exact privacy loss is computationally intractable as it involves calculating the joint probabilities of a sequence of random variables whose size grows exponentially with the control horizon. Rather than computing the exact privacy loss, we propose to approximate it using Gibbs sampling and "plug-in" estimators, and thereby avoid operating on exponentially many state patterns at some cost in accuracy.

More specifically, given the information i_t at time t, we first estimate the following probabilities distributions: $P(Z_{0:t+1}|i_t)$, $P(Z_{0:t+1}|Y_{t+1}, O_{t+1}, i_t)$, and $P(Y_{t+1}, O_{t+1}|i_t)$ by sampling, and then plug the estimates into (5.19) to obtain a "plug-in" estimator of $g_t(i_t, u_t, a_t)$. To acquire the samples from the aforementioned probabilities, we consider two probabilities $P(X_{0:t}, Z_{0:t+1}|i_t)$ and $P(X_{0:t+1}, Z_{0:t+1}, Y_{t+1}, O_{t+1}|i_t)$, which can be efficiently sampled via Gibbs sampling due to the Markovian structure of the problem. For detailed implementation procedure of Gibbs sampling, we refer the readers to [17].

Note that in contrast to the second probability, the first one does not depend on the control action (u_t, f_t). So the samples from the first one can be used to approximate $P(Z_{0:t+1}|i_t)$, which is also not dependent on the current control action, by simply considering the samples of $Z_{0:t+1}$ and ignoring the rest. Similarly, $P(Z_{0:t+1}|y_{t+1}, o_{t+1}, i_t)$, and $P(y_{t+1}, o_{t+1}|i_t)$ can be estimated by the samples from $P(X_{0:t+1}, Z_{0:t+1}, Y_{t+1}, O_{t+1}|i_t)$.

5.4 Characterization of optimal policy

The optimal policy for the general problem formulation stated in Section 5.2 with the proposed instantaneous privacy loss is presented as follows.

Proposition 19. *If (5.18) is used as the privacy loss at time t, then the optimal plant and measurement control policy η in (5.6) for the system described by Equation (5.1)-(5.4) is obtained by minimizing the right-hand side of the following Bellman equations (if exists):*

$$J_t(i_t) = \min_{u_t, f_t} g_t(i_t, u_t, f_t) + \mathbb{E}\Big[c_t(X_t, Z_t, U_t) + J_{t+1}(i_t, Y_{t+1}, O_{t+1}, u_t, f_t)|i_t, u_t, f_t\Big] \tag{5.36}$$

for $t = 1, \cdots, T-1$, and

$$J_T(i_T) = \mathbb{E}\Big[c_T(X_T, Z_T)|i_T\Big] \tag{5.37}$$

Proof. This can be shown by treating (u_t, f_t) as a combined control action of the dynamical system with the state (X_t, Z_t) and then applying dynamic programming. □

Proposition 19 indicates that the solutions of optimal plant control and measurement control are tightly coupled for a general dynamical system as the privacy loss is determined by both plant and measurement controls. The coupling of solutions also arises from the interaction between control and estimation. The measurement control signal affects knowledge of states through noisy observations, which, in turn, affect control actions exerted to the plant.

By combining the result that the instantaneous privacy loss is only a function of measurement control signals shown in Proposition 17 and the famous control-estimation separation theorem for LG systems with quadratic costs, we conjecture that the plant and measurement control policies can be solved separately in LGQ systems. The following proposition states and proves the conjecture above in a rigorous manner. In addition, it shows that the optimal measurement control signal can be determined a priori, regardless of the measurement received on-the-fly.

Proposition 20. *Consider the LQ system described by (5.20)-(5.23). Defining $S_t = [X_t; Z_t]$, the optimal plant and measurement controls u_t^* and f_t^* that minimize the expected value of the sum of the quadratic operation cost and the privacy loss*

$$\mathbb{E}[S_N' Q_N S_N + \sum_{t=0}^{T-1} (S_t' Q_t S_t + U_t' R_t U_t + \gamma g_{t,lin})] \tag{5.38}$$

are given by

$$u_t^* = L_t \mathbb{E}[S_t | i_t], t = 0, \cdots, T-1 \tag{5.39}$$

where L_t is defined by

$$L_t = -(R_t + B_t' K_{t+1} B_t)^{-1} B_t' K_{t+1} A_t \tag{5.40}$$

with the matrices K_t given recersively by the Riccati equation

$$K_T = Q_T \tag{5.41}$$
$$K_t = Q_t + A_t' K_{t+1} A_t - P_t \tag{5.42}$$
$$P_t = A_t' K_{t+1} B_t (R_t + B_t' K_{t+1} B_t)^{-1} B_t' K_{t+1} A_t \tag{5.43}$$

and f_t^* ($t = 0, \cdots, T-1$) that solve the following deterministic optimization problem

$$\min_{\substack{f_t \\ t=1,\cdots,T-1}} \sum_{t=1}^{T-2} Tr\big(P_{t+1} \Sigma_{t+1|t+1}\big) + \sum_{t=1}^{T-1} \gamma g_{t,lin} \tag{5.44}$$

Proof. Note that the optimization problem (5.44) is equivalent to the following deterministic dynamic programming:

$$G_{T-1}(\theta_{T-1}) = \min_{f_{T-1}} \gamma g_{T-1,lin} \tag{5.45}$$
$$G_t(\theta_t) = \min_{f_t} Tr(P_{t+1} \Sigma_{t+1|t+1}) + \gamma g_{t,lin} + G_{t+1}(\theta_t, f_t) \tag{5.46}$$

where $\theta_t = (\theta_{t-1}, f_{t-1})$. Both $g_{t,lin}$ and $\Sigma_{t+1|t+1}$ are a function of (θ_t, f_t). For the optimal measurement policy proof, we will prove the equivalent characterization (5.45) and (5.46).

We will prove the result by induction. Let $W_t = [W_t^X, W_t^Z]$. By (5.36) and (5.37) in Proposition 19, we have

$$\begin{aligned} J_{T-1}(i_{T-1}) &= \mathbb{E}[S_{T-1}'(Q_{T-1} + A_{T-1}' Q_T A_{T-1}) S_{T-1} | i_{T-1}] \\ &+ \mathbb{E}[W_{T-1}' Q_T W_{T-1}] + \min_{f_{T-1}} \gamma g_{T-1,lin} \\ &+ \min_{u_{T-1}} \{u_{T-1}'(R_{T-1} + B_{T-1}' Q_T B_{T-1}) u_{T-1} \\ &+ 2\mathbb{E}[S_{T-1} | i_{T-1}] A_{T-1}' Q_T B_{T-1} u_{T-1}\} \end{aligned} \tag{5.47}$$

Hence, the optimal measurement control f_{T-1}^* is the solution of (5.45). By taking the derivative of the last two lines in (5.47) and setting to zero, we get $u_{T-1}^* = L_{T-1}\mathbb{E}[S_{T-1}|i_{T-1}]$.

Now, assume that the cost-to-go function at time $t+1$ takes the form

$$\begin{aligned} J_{t+1}(i_{t+1}) &= \mathbb{E}[S_{t+1}' K_{t+1} S_{t+1}|i_{t+1}] \\ &+ \mathbb{E}[(S_{t+1} - \mathbb{E}[S_{t+1}|i_{t+1}])' P_{t+1}(S_{t+1} - \mathbb{E}[S_{t+1}|i_{t+1}])|i_{t+1}] \\ &+ \sum_{\tau=t+1}^{T-1} \mathbb{E}[W_\tau' K_{\tau+1} W_\tau] + G_{t+1}(\theta_{t+1}) \end{aligned} \tag{5.48}$$

By Proposition 19, the cost-to-go at time t is

$$\begin{aligned} J_t(i_t) &= \mathbb{E}[S_t' Q_t S_t|i_t] + \mathbb{E}[S_t' A_t' K_t A_t S_t|i_t] \\ &+ \min_{u_t}\{u_t'(R_t + B_t' K_{t+1} B_t)u_t + \mathbb{E}[u_t' B_t' K_{t+1} A_t S_t|i_t, u_t]\} \\ &+ \min_{f_t}\{\gamma g_{t,lin} + G_{t+1}(\theta_t, f_t)\} \\ &+ \min_{u_t, f_t} \mathbb{E}[(S_{t+1} - \mathbb{E}[S_{t+1}|I_{t+1}])' P_{t+1} \cdot (S_{t+1} - \mathbb{E}[S_{t+1}|I_{t+1}])|i_t, u_t, f_t] \\ &+ \sum_{\tau=t}^{T-1} \mathbb{E}[W_\tau' K_{\tau+1} W_\tau] \end{aligned} \tag{5.49}$$

Since $\mathbb{E}[(S_{t+1} - \mathbb{E}[S_{t+1}|I_{t+1}])' P_{t+1}(S_{t+1} - \mathbb{E}[S_{t+1}|I_{t+1}])]$ is weighted error covariance produced by the Kalman filter, which does not depend the measurement received and plant control actions but rather on the plant and measurement parameters, it follows that

$$\begin{aligned} &\min_{u_t, f_t} \mathbb{E}[(S_{t+1} - \mathbb{E}[S_{t+1}|I_{t+1}])' P_{t+1} \cdot \\ &\qquad\qquad (S_{t+1} - \mathbb{E}[S_{t+1}|I_{t+1}])|i_t, u_t, f_t] \\ &= \min_{f_t} Tr(P_{t+1}\Sigma_{t|t}) \end{aligned} \tag{5.50}$$

Therefore, f_t^* is given by the solution of the following optimization problem

$$\min_{f_t} Tr(P_{t+1}\Sigma_{t|t}) + \gamma g_{t,lin} + G_{t+1}(\theta_t, f_t) \tag{5.51}$$

By setting the derivative of the second line in (5.49) to zero, we get the optimal plant control

$$u_t^* = -L_t \mathbb{E}[S_t|i_t] \tag{5.52}$$

Plug f_t^* and u_t^* back to $J_t(i_t)$, we have

$$\begin{aligned} J_t(i_t) &= \mathbb{E}[S_t' K_t S_t|i_t] + \mathbb{E}[(S_t - \mathbb{E}[S_t|i_t])' P_t (S_t - \mathbb{E}[S_t|i_t])|i_t] \\ &+ \sum_{i=t}^{T-1} \mathbb{E}[W_i' K_{i+1} W_i] + G_t(\theta_t) \end{aligned} \tag{5.53}$$

Herein, we have proved the correctness of the proposed form of the cost-to-go at each time step and obtained the expression of optimal plant and measurement control. □

Table 5.1: Summary of notations.

Parameter	Meaning	Value & Units
R	Average thermal resistance	$2°C/kW$
C	Average thermal capacitance	$10kWh/°C$
P	Average energy transfer rate	$14kW$
C_p	Average thermal load per person	$0.01°C/h$
X_a	Ambient temperature	$32°C$
η	Load efficiency	2.5
σ_{model}	Noise std of temperature model	$10^{-5}°Cs^{-0.5}$
σ_{meas}	Noise std of temperature measurement	$0.5°C$
X_d	Desired temperature	$25°C$

5.5 Case study: Occupancy-based Thermostat Control

In this section, we design a privacy-preserving control law for a thermostat that utilizes occupancy information to achieve energy savings and comfort improvement.

Thermostat model

The thermostat model presented herein follows [22] closely. With reference to the notations in Table 5.1, the temperature dynamics are modeled by a linear equation

$$X_{t+1} = aX_t + (1-a)(X_a - U_t RP) + C_p h Z_t + W_t^X \tag{5.54}$$

where $a = \exp(-h/CR)$ governs the thermal characteristics of the thermal mass and h is the time elapsed per discrete time step. $U_t \in \{0, 1\}$ indicates the ON/OFF actions. W_t^X is a zero-mean Gaussian noise process with variance $h\sigma_{model}^2$, accounting for all heat gain and loss not modeled explicitly. The term $C_p h Z_t$ captures the heat contributed by human presence, where Z_t indicates the number of people. For simplicity, we consider Z_t to be binary, which evolves as a Markov chain with transition probability

$$P(Z_{t+1}|Z_t) = \begin{cases} 1-q & \text{if } Z_{t+1} \neq Z_t \\ q & \text{if } Z_{t+1} = Z_t \end{cases} \tag{5.55}$$

The temperature is measured via a noisy sensor given by

$$Y_t = X_t + V_t^X \tag{5.56}$$

where $V_t^X \sim \mathcal{N}(0, \sigma_{meas}^2)$. To preserve privacy, the occupancy data Z_t will be obfuscated randomly in situ before being sent to the controller. To be specific, the occupancy sensor

measurement O_t is modeled by

$$P(O_t|Z_t) = \begin{cases} 1 - F_{t-1} & \text{if } O_t \neq Z_t \\ F_{t-1} & \text{if } O_t = Z_t \end{cases} \tag{5.57}$$

where $F_t \in \{0.5, 1\}$ is the control action on the occupancy sensor. U_t and F_t are designed by minimizing the expectation of the following objective

$$\sum_{t=0}^{T-1} \Big[\underbrace{\frac{1}{\eta} PhU_t}_{\text{energy cost}} + \gamma_p \underbrace{g_t(I_t, U_t, F_t)}_{\text{privacy loss}} + \gamma_c \underbrace{Z_t(X_t - X_d)^2}_{\text{comfort loss}} \Big] \tag{5.58}$$

where γ_p and γ_c stand for the amount of energy people are willing to pay in exchange of one bit of private information leakage, and one unit of uncomfortableness measured in the squared deviation of current temperature from the desired temperature X_d, respectively. Z_t is included in the comfort loss term to accommodate the fact that people only sustain comfort loss when they are present.

The optimal policy of the thermostat system presented above is intractable. To obtain a simple suboptimal controller, we resort to open-loop feedback control [13] and a heuristic that calibrates the estimate of future control cost by taking into account the effect of occupancy noise on the estimation of comfort loss.

Results

Since the thermostat model is not linear Gaussian, we use the sampling-based method presented in Section 5.3 to approximate the instantaneous privacy loss at each time step, where sample size is a free parameter that manifests the tension between computational efficiency and approximation accuracy. Fig. 5.2 provides a reference for the selection of the sample size. For each sample size and horizon length, we conduct 100 Monte Carlo simulations of privacy loss for each one of the 100 randomly generated action traces. The vertical axis of Fig. 5.2 is obtained by first calculating the standard deviation of privacy loss over all Monte Carlo simulations for every single action trace, and then computing the average of different action traces' privacy loss variation. It can be seen that a larger sample size can reduce the variation of the sampling-based privacy loss approximation. Moreover, for a given variation tolerance, more samples are needed for a longer horizon. In the following experiments, we choose the sample size to be 5000.

We then demonstrate the flexibility of jointly optimized control over plant and occupancy sensing fidelity, by comparing it with two other policies: (1) a minimum-privacy policy that reports the ground truth occupancy at all times (i.e., $F_t = 1, \forall t$) and optimizes the plant control action at each time step; (2) a maximum-privacy policy that always flips the true occupancy with probability 0.5 (i.e., $F_t = 0.5, \forall t$) and optimizes the plant control. The costs incurred by the three policies are simulated for users with different privacy preferences. We consider three different types of users, namely, the privacy fundamentalist who is very

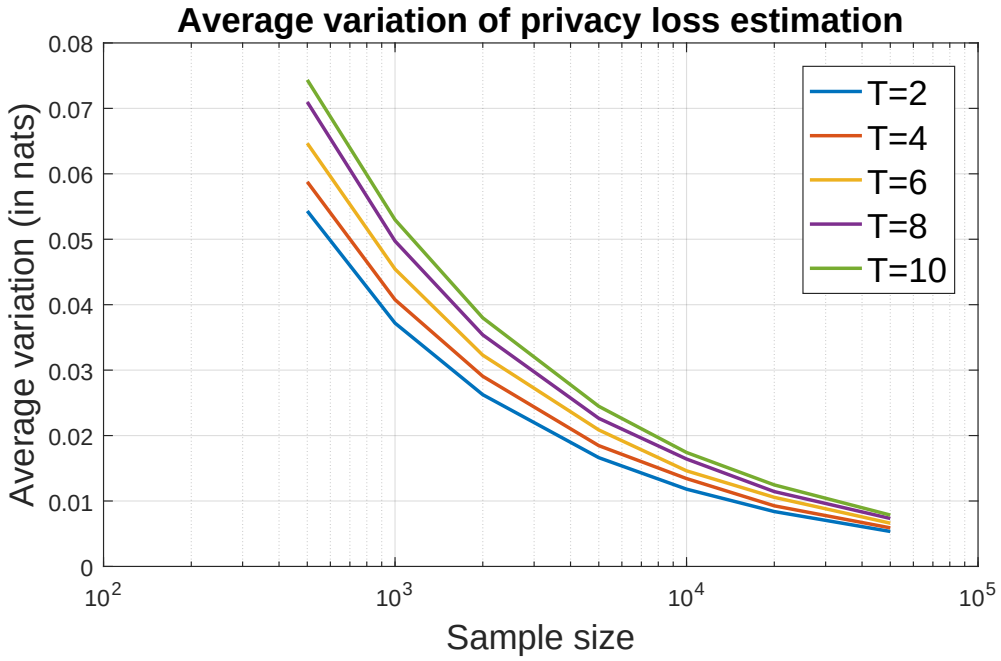
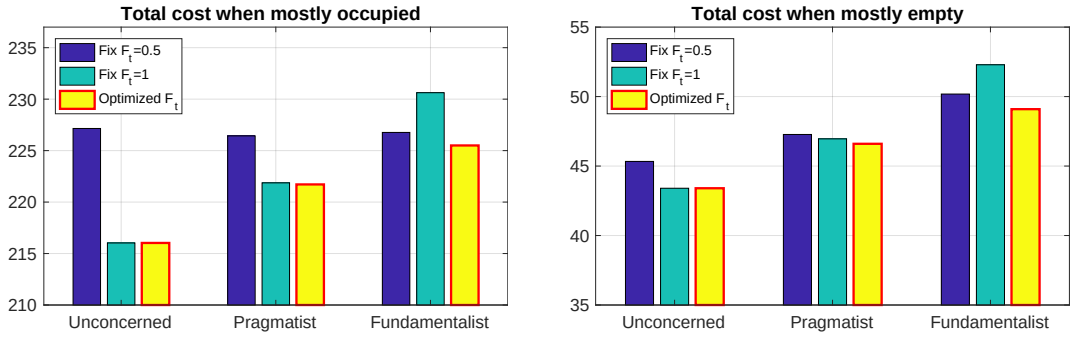


Figure 5.2: The variation of privacy loss approximation for different sample sizes and horizon lengths.



(a) (b)

Figure 5.3: The total cost incurred by three different policies: (1) flipping the occupancy with probability 0.5 and optimizing the plant controls (maximum privacy), (2) always reporting the true occupancy and optimizing the plant controls (minimum privacy), and (3) jointly optimizing the occupancy flipping probability and plant controls for users with different privacy preferences (optimized privacy), under different space occupation states.

concerned about privacy, the pragmatist that values the privacy to a moderate degree, and the unconcerned who does not care about privacy loss. We fix the comfort weight $\gamma_c = 1$ and varies the privacy weight $\gamma_p = 5, 1, 0$ for the three type of users. We cluster simulations

into two scenarios, namely mostly occupied or mostly empty, according to the ground truth occupancy of the space, and examine how the operation costs and privacy loss of different policies change for different types of users.

Figure 5.3 (a) and (b) illustrate the superiority of "optimally" flipped occupancy reports in terms of the total cost. Reporting the true occupancy tends to be the best strategy for users who are unconcerned about privacy loss while reporting randomly flipped occupancy is better if the user is a privacy fundamentalist. Figure 5.4 (a) and (b) show an interesting occupancy bias introduced by random occupancy flipping. It can be seen that when the space is mostly occupied the controller with randomly flipped occupancy ($F_t = 0.5$) incurs lower energy cost compared with the one that receives true occupancy reports; to the contrary, when the space is mostly empty the energy cost with randomly flipping occupancy is higher. This is because random flipping introduces artifacts of low (or high) occupancy when the space is actually occupied (or empty). Similar occupancy bias also manifests itself in Fig. 5.4 (c) and (d) by driving the comfort loss up when the space is occupied and bringing it down when the space is empty. The optimized flipping policy gradually reduces the privacy loss as the user puts more weight on the privacy loss, as shown in Fig. 5.4 (e) and (f) while the other two policies lack the agility to tune the privacy loss in response to users' preference.

5.6 Chapter Summary

This chapter studies the joint sensor-controller design problem in private-input-driven dynamical systems, which are motivated by occupancy-based thermostat control applications. We propose an instantaneous privacy loss measure that characterizes the privacy leakage of sequential data streams under standard inference attack, and provide a sampling-based method to approximate privacy loss of general systems with a possibly nonlinear form and arbitrary disturbance. We also characterize the optimal joint design policy and prove the separation of sensor design, plant control and system state estimation in LGQ systems. The numeric example on thermostat control demonstrates the tradeoff between plant operation cost and privacy leakage. The privacy-aware thermostat controller presented in the chapter is shown to realize better privacy protection while maintaining energy efficiency and thermal comfort.

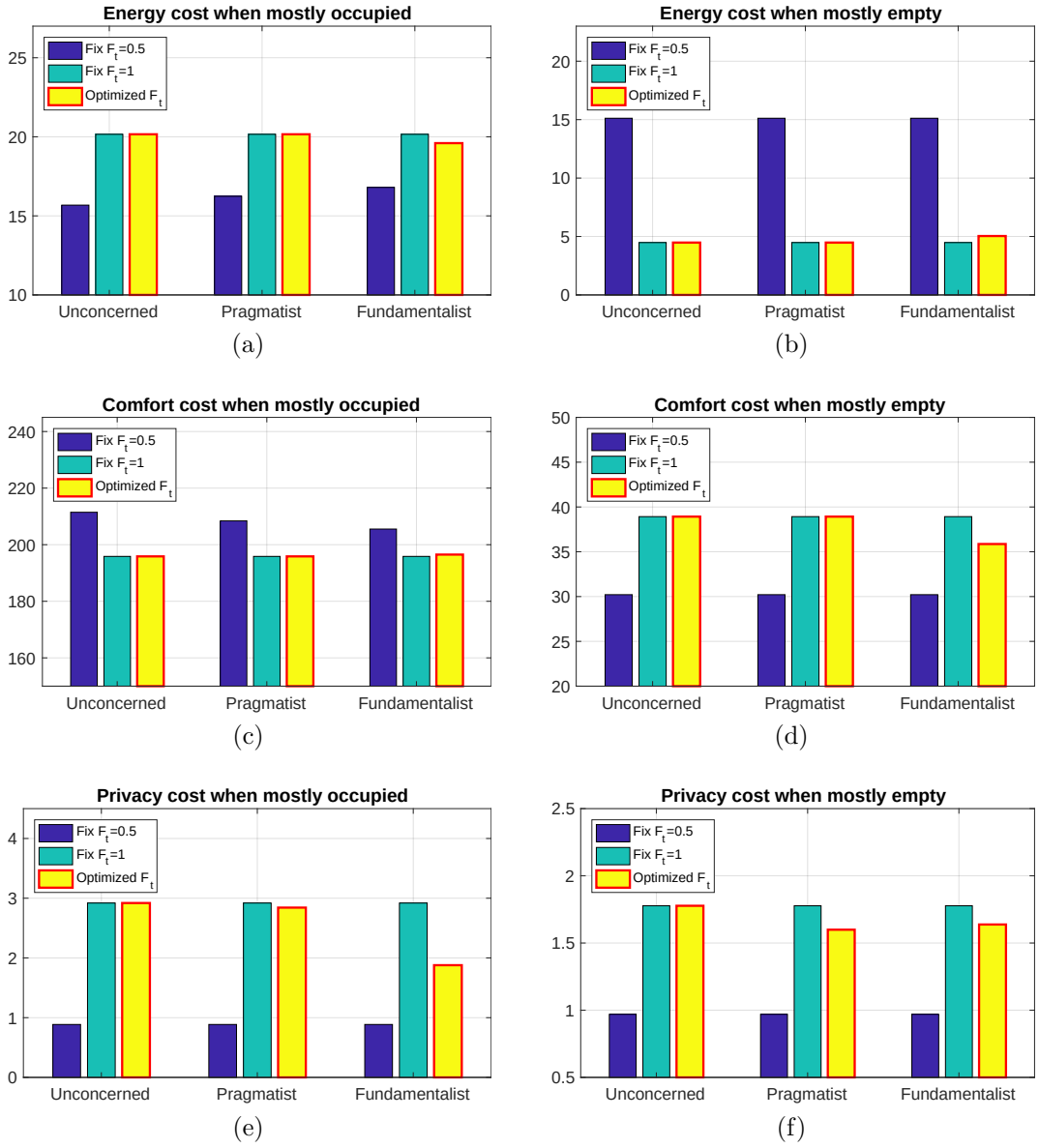


Figure 5.4: The energy cost, comfort cost, and privacy cost incurred by three different policies: (1) flipping the occupancy with probability 0.5 and optimizing the plant controls (maximum privacy), (2) always reporting the true occupancy and optimizing the plant controls (minimum privacy), and (3) jointly optimizing the occupancy flipping probability and plant controls for users with different privacy preferences (optimized privacy), under different space occupation states.

Chapter 6

Privacy-Aware Sensing for Model Predictive Control

6.1 Motivating Application: Occupancy-based HVAC Control

Large-scale sensing and actuation infrastructures have endowed buildings with the intelligence to perceive the status of their environment, energy usage, and occupancy, and to provide fine-grained and responsive controls over heating, cooling, illumination, and other facilities. However, the information that is collected and harnessed to enable such levels of intelligence may potentially be used for undesirable purposes, thereby raising the question of privacy. To spotlight the value of building sensory data and its potential for exploitation in the inference of private information, we consider as a motivating example the occupancy data, i.e., the number of occupants in a given space over time.

Occupancy data is a key component to perform energy-efficient and user-friendly building management. Particularly, it offers considerable potential for improving energy efficiency of the heating, ventilation, and air conditioning (HVAC) system, a significant source of energy consumption which contributes to more than 50% of the energy consumed in buildings [50]. Recent chapters [8, 94, 52] have demonstrated substantial energy savings of up to 40% by enabling intelligent HVAC control in response to occupancy variations. The value of occupancy data in building management has also inspired extensive research on occupancy sensing [45, 89, 88, 93, 170] as well as a number of commercial products which can provide high accuracy occupancy data.

While people have enjoyed the benefits brought by occupancy data, the privacy risks potentially posed by the data are largely overlooked (Figure 6.1). In effect, location traces of individual occupants can be inferred from the occupancy data with some auxiliary information [161]. Throughout this chapter, we refer to the individual location trace as the private information to be protected. The contextual information attached to location traces tells much about the individuals' habits, interests, activities, and relationships [104]. It can

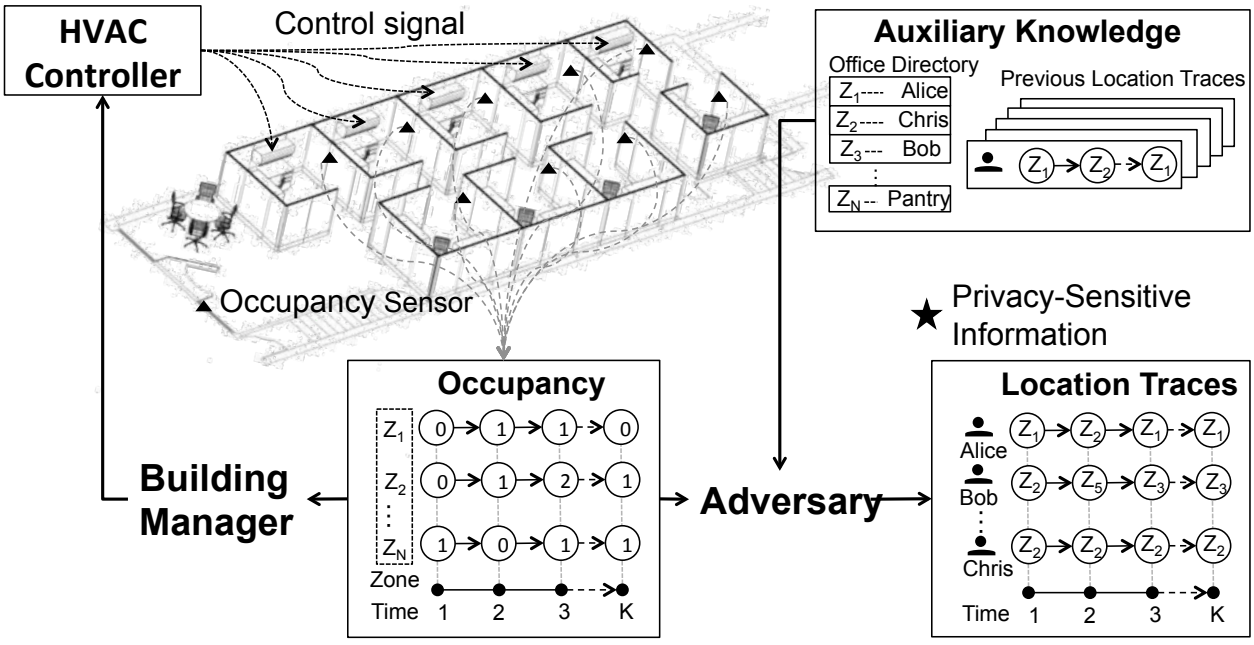


Figure 6.1: An overview of the problem of individual occupant location recovery. The building manager collects occupancy data to enable intelligent HVAC controls adapted to occupancy variations. However, an adversary with malicious intent may exploit occupancy data in combination with the auxiliary information to infer privacy details about indoor locations of building users.

also reveal their personal or corporate secrets, expose them to unwanted advertisement and location-based spams/scams, cause social reputation or economic damage, make them victims of blackmail or even physical violence [144].

At a first glance, it is surprising that occupancy data may incur risks of privacy breach, since it only reports the number of occupants in a given space over time without revealing the identities of the occupants. To illustrate why it is possible to infer location traces from seemingly "anonymized" occupancy data, consider the following scenario. We start by observing two users in one room and then one of them leaves the room and enters another room. We cannot tell which one of the two made this transition by observing the occupancy change. However, if the one who left entered an private office, the user can be identified with high probability based on the ownership of the office. Although a change in occupancy data may correspond to location shifts of many possible potential users, the knowledge of where the individuals mostly spend their time rules out many possibilities and renders the individual who made the transition identifiable. It has been shown in [161] that by simply combining some ancillary information, such as an office directory and user mobility patterns, individual location traces can be inferred from the occupancy data with the accuracy of more than 90%. It is, therefore, the objective of this chapter to enable an occupancy-based HVAC control system that provides privacy features for each user on a par with thermal comfort

and energy efficiency.

A simple yet effective way to preserve privacy is to obfuscate occupancy data by injecting noise to make the data itself less informative. This approach has been widely used in privacy disclosure control of various databases, ranging from healthcare [36], geolocation [4], web-browsing behavior data [55], etc. While reducing the risk of privacy breach, this approach would also deteriorate the utility of the data. There have been attempts to balance learning the statistics of interest reliably with safeguarding the private information [147]. Cryptography [40] and access control [159] are also effective means to ease privacy concerns, but they do not provide protection against all privacy breaches. There may be insiders who can access the private, decrypted data, or the building manager may not want to have access to (and responsibility for) the private data.

The objective of this chapter cannot be attained by simply extending the techniques developed previously. Our task is more challenging. Firstly, as opposed to learning some fixed statistics from static data in most database applications, the data is used for controlling a highly complex and dynamic system in our case, and the control performance relies on the data fidelity. With highly accurate occupancy data, the infrastructure can correctly sense the environment and enable proper response to occupancy variations; nevertheless, the location privacy is sacrificed. On the other hand, the usage of severely distorted occupancy data reduces the risks of privacy leakage, but may lead to even higher levels of energy consumption and discomfort. Essentially, we need to address the trade-off between the performance of a controller on a dynamical system, and, similarly, privacy of a time-varying signal, i.e. the location traces of individual occupants. Secondly, from the perspective of the building manager, the building performance is paramount: adding the privacy feature into the HVAC control system should not impair the performance of HVAC controller in terms of energy efficiency and thermal comfort. To achieve this, the injected noise should be calculated to minimally affect performance of the controller, while maximizing the amount privacy gained from the distortion.

In this chapter we develop a method which minimizes the privacy risks incurred by collection of occupancy data while guaranteeing the HVAC system operating in a "nearly" optimal condition. Our solution relies on an occupancy distortion mechanism, which can be implemented at the sensor level and "sanitizes" the occupancy data before any form of transport or storage of the data. We draw the inspiration from the information-theoretic approach in [133, 128, 157] for characterizing the privacy-utility trade-off, and choose the mutual information (MI) between reported occupancy measurements and individual location traces as our privacy metric. The design problem of finding the optimal occupancy distortion mechanism is cast as an optimization problem where the privacy risk is minimized for a set of constraints on the controller performance. This allows us to find points on the Pareto frontier in the utility-privacy trade-off, and to further analyze the economic side of privacy concerns [134]. The formulation can be easily generalized to resolve the tension between privacy and data utility in other cases where a control system utilizes some privacy-sensitive information as one of the control inputs, although in this chapter we limit our focus to addressing the privacy concern of occupancy-based HVAC controller. In addition, our work

here is complementary to the work being done in the cryptography communities: we can use our distortion mechanism to process sensor measurements, and then transmit the processed measurements across secure channels. Our work also serves as a complement for the privacy-preserving access control protocol in [159], as it provides distortion mechanisms against adversaries who might be able to subvert the protocol while still retaining the benefits for the occupancy data.

The main contributions of this chapter are as follow:

- We present a systematic methodology to characterize the privacy loss and control performance loss.
- We develop a holistic and tractable framework to balance the privacy pursuit and control performance.
- We evaluate the trade-off between privacy and HVAC control performance using the real-world occupancy data and simulated building dynamics.

6.2 Attack Model

Suppose the building of interest consists of N zones represented by $\mathcal{Z} = \{z_0, z_1, \cdots, z_N\}$, where a special zone z_0 is added to refer to the outside of the building. Let $\mathcal{O} = \{o_1, \cdots, o_M\}$ denote the set of occupants. The location of occupant o_m at time k is a random variable denoted by $X_k^{(m)}$ which takes values in the set $\mathcal{Z}$, for $m = 1, \cdots, M$. The true occupancy of zone z_n at time k is denoted by Y_k^n, $n = 0, 1, \cdots, N$. Y_k^n takes values from $\{0, 1, \cdots, M\}$, where M is the total number of occupants in the building. Note that the true occupancy and individual location traces are connected by $Y_k^n = \sum_{m=1}^{M} \mathbb{1}[X_k^{(m)} = z_n]$, where $\mathbb{1}[\cdot]$ is the indicator function.

We assume that the attacker observes a distorted version of the true occupancy, denoted by V_k^n which takes values from $\{0, 1, \ldots, M\}$. $\mathbb{P}(V_k^n | Y_k^n)$ represents the distortion mechanism we wish to design. If no distortion on the occupancy data is applied, then $V_k^n = Y_k^n$. We further define some shorthands: $X_k^{(1:M)} := \{X_k^{(1)}, \cdots, X_k^{(M)}\}$, $V_k^{1:N} := \{V_k^1, \cdots, V_k^N\}$. Then, the attacker is assumed to perform inference on the Factorial Hidden Markov model (FHMM), illustrated in Figure 6.2, in order to recover the location traces. The FHMM consists of several independent Markov chains evolving in parallel, representing the location trace of each occupant. Since the attacker only observes the aggregate occupancy information, the location traces are considered to be hidden states.

The FHMM model entails the following assumptions:

1. The location traces for different occupants are mutually independent: $\mathbb{P}(X_k^{(1:M)}) = \prod_{m=1}^{M} \mathbb{P}(X_k^{(m)})$.

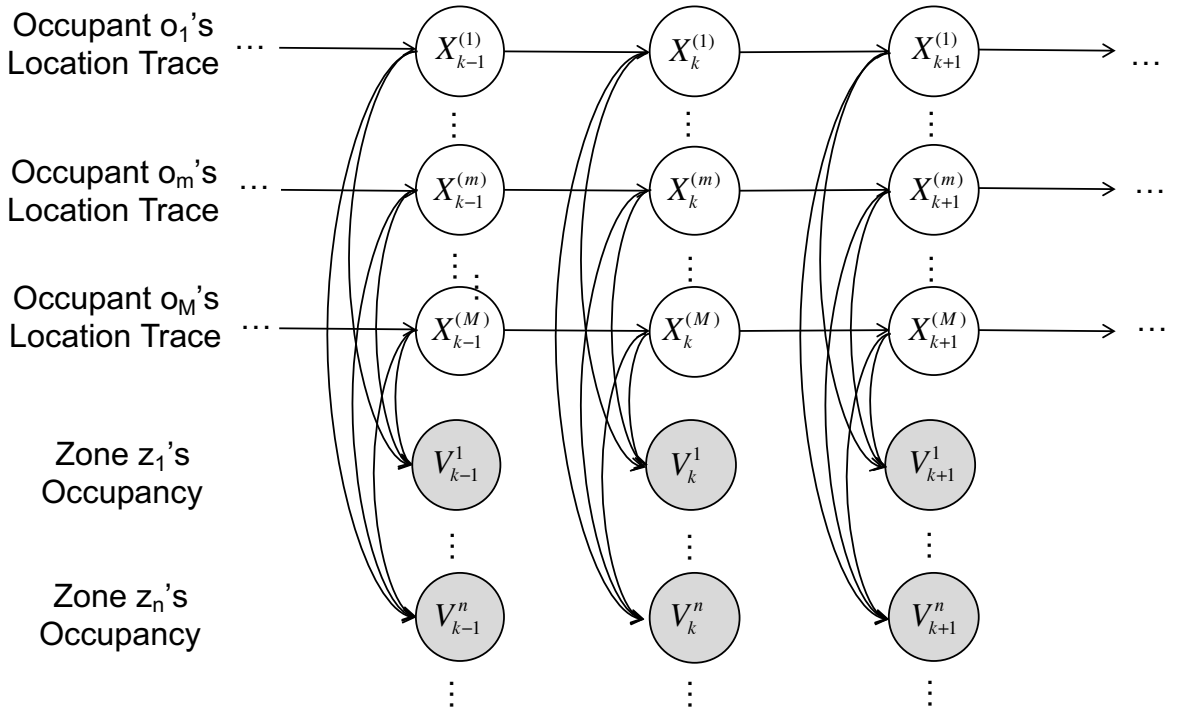


Figure 6.2: The graphical model representation of the FHMM model.

2. The location trace for any given occupant o_m, $m \in \{1, \ldots, M\}$, has the first-order Markov property:

$$\mathbb{P}(X_k^{(m)}|X_{k-1}^{(m)}, X_{k-2}^{(m)}, \ldots, X_1^{(m)}) = \mathbb{P}(X_k^{(m)}|X_{k-1}^{(m)}) \tag{6.1}$$

3. The distorted occupancy V_k^n depends only on Y_k^n. As a result, $\mathbb{P}(V_k^n|X_k^{(1:M)}) = \mathbb{P}(V_k^n|Y_k^n)$.

The FHMM model can be specified by the transition probabilities and emission probabilities. The transition probabilities describe the mobility pattern of an occupant, which is denoted as a $(N+1) \times (N+1)$ transition matrix. We define the transition matrix for occupant o_m as $A^{(m)} = [a_{ij}^{(m)}]$, $i, j = 0, 1, \cdots, N$, where $a_{ij}^{(m)} = \mathbb{P}(X_{k+1}^{(m)} = z_j|X_k^{(m)} = z_i)$ for $k = 0, 1, \cdots, K-1$. The transition parameters can be learned from the occupancy data based on maximum likelihood estimation. If the prior knowledge about the past location traces is also available, it can be encoded as the prior distribution of transition parameters from a Bayesian point of view, and then the transition parameters can be learned via *maximum a posteriori* (MAP) estimation. We refer the readers to [161] for the details of parameter learning. The emission probabilities characterize the conditional distribution of distorted

occupancy given the location of each occupant, defined by

$$\mathbb{P}(V_k^{1:N}|X_k^{(1:M)}) = \prod_{n=1}^{N} \mathbb{P}(V_k^n|X_k^{(1:M)}) = \prod_{n=1}^{N} \mathbb{P}(V_k^n|Y_k^n) \tag{6.2}$$

The above equalities follows from a conditional independence relation encoded by the FHMM—the distorted occupancy depends on individual location traces only via the true occupancy.

6.3 HVAC System Model

Suppose the thermal comfort of the building space of interest is regulated by the HVAC system shown in Figure 6.3, which provides a system-wide Air Handling Unit (AHU) and Variable Air Volume (VAV) boxes distributed at the zones. In this type of HVAC system, the outside air is conditioned at the AHU to a setpoint temperature T_a by the cooling coil inside. The conditioned air, which is usually cold, is then supplied to all zones via the VAV box at each zone. The VAV box controls the supply air flow rate to the thermal zone, and heats up the air using the reheat coils at the box, if required. The control inputs are temperature and flow rate of the air supplied to the zone by its VAV box. The AHU outlet air temperature setpoint T_a is assumed to be constant in this chapter. The HVAC system models described in the subsequent paragraphs will follow [92, 11, 63] closely.

State model. With reference to the notations in Table 6.1, the continuous time dynamics for the temperature T^n of zone z_n can be expressed as

$$C^n \frac{d}{dt} T^n = \mathbf{R}^n \cdot \mathbf{T} + Q^n + \dot{m}_s^n c_p (T_s^n - T^n) \tag{6.3}$$

where the superscript n indicates that the associated quantities are attached to zone z_n. $\mathbf{T} \coloneqq [T^1, \cdots, T^N]$ is a vector of temperature of all N zones. $\mathbf{R}^n$ indicates the heat transfer among different zones and outside. Q^n is the thermal load, which can be obtained by applying a thermal coefficient c_o to the number of occupants V^n, i.e., $Q^n = c_o V^n$. The control inputs $U^n \coloneqq [\dot{m}_s^n, T_s^n]$ are the supply air mass flow rate and temperature. Assuming $\dot{m}_s^n$, T_s^n and Q^n are zero-order held at sample rate Δt, we can discretize (6.3) using the trapezoidal method and obtain a discrete-time model, which can be expressed as

$$C^n \frac{T_{k+1}^n - T_k^n}{\Delta t} = R^n \cdot T_k + c_o V_k^n + \dot{m}_{s,k}^n c_p \left(T_{s,k}^n - \frac{T_{k+1}^n + T_k^n}{2} \right) \tag{6.4}$$

where k is the discrete time index and $T_k^n = T_t^n|_{t=k\Delta t}$. Q_k^n, $\dot{m}_{s,k}^n$ and $T_{s,k}^n$ are similarly defined.

Cost function. The control objective is to condition the room while minimizing the energy cost. The power consumption at time k consists of reheating power $P_{h,k}^n = \frac{c_p}{\eta_h} \dot{m}_{s,k}^n (T_{s,k}^n -$

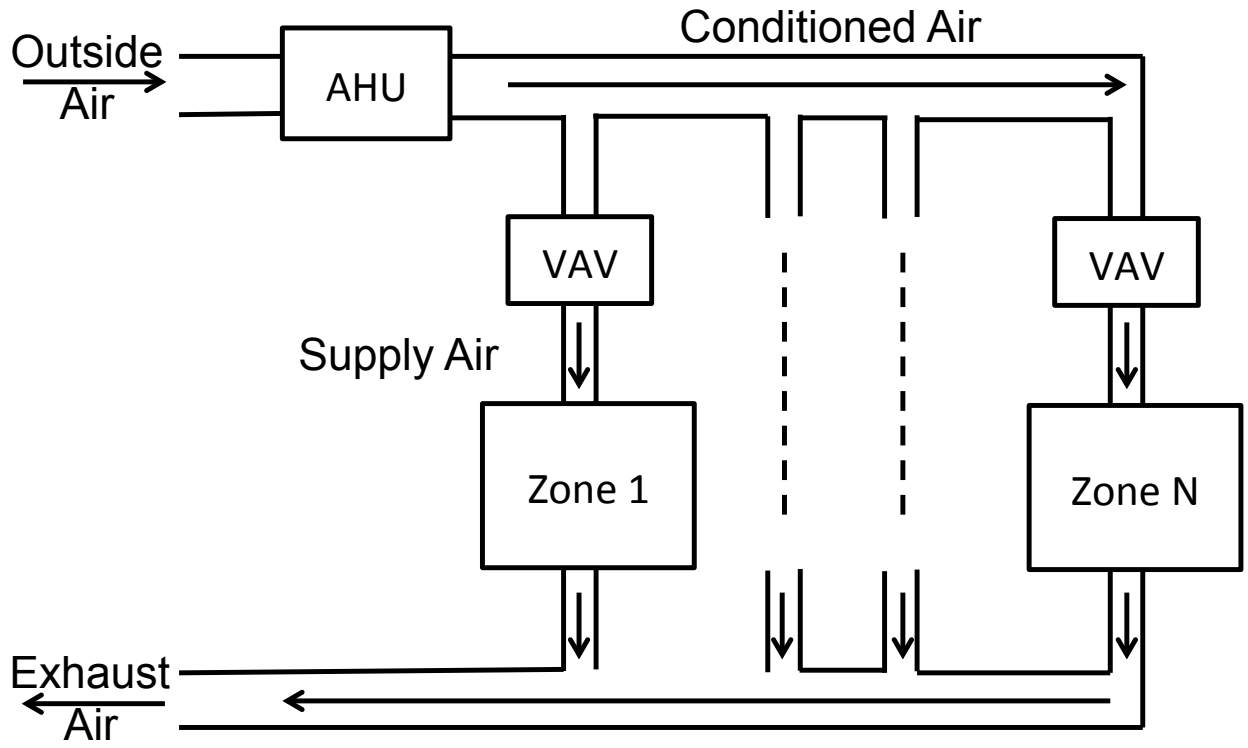


Figure 6.3: A schematic of a typical multi-zone commercial building with a VAV-based HVAC system.

T_a), cooling power $P^n_{c,k} = \frac{c_p}{\eta_c}\dot{m}^n_{s,k}(T_o - T_a)$ and fan power $P^n_{f,k} = \beta \dot{m}^n_{s,k}$, where η_h and η_c capture the efficiencies for heating and cooling side, respectively. β stands for a system dependent constant. We introduce several parameters to reflect utility pricing, r_e for electricity and r_h for heating fuel. These parameters may vary over time. Therefore, the total utility cost of zone z_n from time $k = 1, \cdots, K$ is $J^n = \sum_{k=1}^{K} \Big((r_{e,k} P^n_{f,k} + r_{h,k} P^n_{h,k} + r_{e,k} P^n_{c,k}) \Delta t \Big)$.

Constraints. The system states and control inputs are subject to the following constraints:

- C1: $\underline{T} \leq T^n_k \leq \overline{T}$, comfort range;
- C2: $\underline{\dot{m}}_s \leq \dot{m}^n_{s,k} \leq \overline{\dot{m}}_s$, minimum ventilation requirement and maximum VAV box capacity;
- C3: $T^n_{s,k} \geq T_a$, heating coils can only increase temperature;
- C4: $T^n_{s,k} \leq \overline{T}_h$, heating coil capacity.

These constraints hold at all times k and all zones $\{z_n\}_{n=1}^N$.

Table 6.1: Parameters used in the HVAC controller.

Param.	Meaning	Value & Units
Δt	Discretization step	$60s$
c_p	Thermal capacity of air	$1kJ/(kg \cdot K)$
C^n	Thermal capacity of the env.	$1000kJ/K$
c_o	Thermal load per person	$0.1kW$
R	Heat transfer vector	$0kW/K$
η_h	Heating efficiency	0.9
η_c	Cooling efficiency	4
β	System parameter	$0.5kW \cdot s/kg$
r_e	Electricity price	$1.5 \cdot 10^{-4}\$/kJ$
r_h	Heating fuel price	$5 \cdot 10^{-6}\$/kJ$
$\underline{T}$	Upper bound of comfort zone	$24°C$
$\overline{T}$	Lower bound of comfort zone	$26°C$
T_a	AHU outlet air temperature	$12.8°C$
$\underline{m_s}$	Minimum air flow rate	$0.0084kg/s$
$\overline{m_s}$	Maximum air flow rate	$1.5kg/s$
$\overline{T}_h$	Heating coil capacity	$40°C$

MPC controller. Knitting together the models described above, we present an MPC-based control strategy for the HVAC system to efficiently accommodate for occupancy variations. In this control algorithm, we assume that the predicted occupancy during the optimization horizon to be the same as the instanteneous occupancy observed at the beginning of control horizon. It was shown in [63] that the control algorithm with this assumption can achieve comparable performance with the MPC that constructs explicit occupancy model to predict occupancy for future time steps.

Let $U_{1:K}^{1:N}$ be the shorthand for $\{U_k^n | k = 1, \cdots, K, \quad n = 1, \cdots, N\}$. The optimal control inputs for the next K time steps are obtained by solving $\min_{U_{1:K}^{1:N}} \sum_{n=1}^{N} J^n$, subject to the inequality constraints C1-C4 and the equality constraint (6.4) and $T_1^n = T_{init}^n$, $\forall n = 1, \cdots, N$, where T_{init}^n is the initial temperature of zone z_n at each MPC iteration. We can see that the optimal control input is a function of the distorted occupancy that the controller sees and the initial temperature. We express this relationship explicitly by denoting the optimal control action at zone z_n as $U_{MPC}^n(V^n, T_{init}^n)$. In addition, the energy cost incurred by applying the optimal control action is denoted by $J_{MPC}^n(U_{MPC}^n(V^n, T_{init}^n), Y^n)$, where the second argument stresses that the actual control cost is dependent on the real occupancy.

6.4 Optimal Design for Privacy-Aware Sensing

With the HVAC model established, we can now develop the mathematical framework to discuss a privacy-enhanced architecture. We will first introduce MI as the metric we use

throughout the chapter to quantify privacy, and then present a method to optimally design the distortion mechanism which minimizes the privacy loss within a pre-specified constraint on control performance.

Privacy Metric

Definition 5 ([33]). *For random variables X and V, the* mutual information *is given by:*

$$I(X;V) = H(X) - H(X|V) \tag{6.5}$$

where $H(X)$ and $H(X|V)$ represent entropy *and* conditional entropy*, respectively. Let $\mathbb{P}_X(x) = \mathbb{P}(X = x)$, $H(X)$ and $H(X|V)$ are defined as*

$$H(X) = -\sum_x \mathbb{P}_X(x) \log(\mathbb{P}_X(x)) \tag{6.6}$$

$$H(X|V) = -\sum_v \mathbb{P}_V(v) \Big(\sum_x \mathbb{P}_{X|V}(x|v) \log \big(\mathbb{P}_{X|V}(x|v)\big) \Big) \tag{6.7}$$

Entropy measures uncertainty about X, and conditional entropy can be interpreted as the uncertainty about X after observing V. By the definition above, MI is a measure of the reduction in uncertainty about X given knowledge of V. We can see that it is a natural measure of privacy since it characterizes how much information one variable tells about another. It is also worth noting that inference technologies evolve and MI as a privacy metric does not depend on any particular adversarial inference algorithm [133] as it models the statistical relationship between two variables.

In this chapter, we will be using the MI between location traces and occupancy observations, i.e., $I(X_k^{(1:M)}; V_k^{1:N})$, as a metric of privacy loss. This metric reflects the reduction in uncertainty about location traces $X_k^{(1:M)}$ due to observations of $V_k^{1:N}$. As a proof of concept, we will verify that this metric serves as an accurate proxy for an adversary's ability to infer individual location traces in the experiments. We further introduce some assumptions which allow us to simplify the expression of the privacy loss and obtain a form of MI that has direct relationship with the distortion mechanism $P(V_k^n|Y_k^n)$ we wish to design.

Based on results in ergodic theory [91], we know that the probability distribution of individual location traces will converge to a unique stationary distribution under very mild assumptions[1]. For more details on stationary distributions, we refer the reader to [91]. This observation justifies the assumption that the Markov chains $X_k^{(m)}$ have a unique stationary distribution for all occupants o_m and are distributed according to those stationary distributions for all time steps k.

Combining this assumption and the occupancy-location model we presented in the preceding section, we present a proposition that allows us to greatly simplify the form of the privacy loss:

[1]Since there are only finitely many zones, a sufficient condition is the existence of a path from z_i to z_j with positive probability for any two zones z_i and z_j.

Proposition 21. *By Assumption 3, we have that*

$$I(X_k^{(1:M)}; V_k^{1:N}) = I(Y_k^{1:N}; V_k^{1:N}) \tag{6.8}$$

By the stationary distribution assumption, we have that $I(Y_k^{1:N}; V_k^{1:N})$ is a constant for all k, so we will drop the subscript: $I(Y^{1:N}; V^{1:N})$.

Finally, by the various conditional independences introduced in Assumption 3, it follows that

$$I(Y^{1:N}; V^{1:N}) = \sum_{n=1}^{N} I(Y^n; V^n) \tag{6.9}$$

The result that $I(Y_k^{1:N}; V_k^{1:N})$ is a constant value for all k allows us to design a single distortion mechanism $P(V^n|Y^n)$ for all time steps (note that we drop the subscript k to indicate the time-homogeneity of the distortion mechanism). By Proposition 21, minimization of privacy loss $I(X_k^{(1:M)}; V_k^{1:N})$ can be conducted by minimizing a simpler expression $\sum_{n=1}^{N} I(Y^n; V^n)$.

Optimal Distortion Design

We wish to find a distortion mechanism $P(Y^n|V^n)$ that can produce some perturbed occupancy data with minimum information leakage, while the performance of the controller using the perturbed occupancy data is on a par with that using true occupancy. To be specific, we will bound the difference of energy costs incurred by the controllers seeing distorted and real occupancy data.

Let T_{init1} and T_{init2} be initial temperature of the controller using distorted and real occupancy, respectively. Recall that $U_{MPC}^n(V^n, T_{init}^n)$ and $J_{MPC}^n(U_{MPC}^n(V^n, T_{init}^n), Y^n)$ stand for the optimal control actions and the associated cost based on the distorted occupancy; correspondingly, if the controller sees the real occupancy data, the optimal control action and the associated cost will be $U_{MPC}^n(Y^n, T_{init}^n)$ and $J_{MPC}^n(U_{MPC}^n(Y^n, T_{init}^n), Y^n)$, respectively. We denote the resulting temperature after applying optimal control actions as $T_{MPC}^n(U_{MPC}^n(V^n, T_{init}^n), Y^n)$, where the second argument emphasizes that the temperature evoluation depends on the true occupancy. We introduce the following constraints: $\forall |T_{init1} - T_{init2}| \leq \Delta_T'$, $y = 0, \cdots, M$, $n = 1, \cdots, N$,

C5: Cost difference constraint

$$E_{\mathbb{P}(V^n|Y^n=y)}\Big[J_{MPC}^n\big(U_{MPC}^n(T_{init1}, V^n), y\big) - J_{MPC}^n\big(U_{MPC}^n(T_{init2}, y), y\big)\Big] \leq \Delta \tag{6.10}$$

C6: Resulting temperature constraint

$$E_{\mathbb{P}(V^n|Y^n=y)}\Big[\big|T^n_{MPC}\big(U^n_{MPC}(T_{init1},V^n),y\big)- T^n_{MPC}\big(U^n_{MPC}(T_{init2},y),y\big)\big|\Big]\leq\Delta_T \tag{6.11}$$

C5 states that the cost difference between using the distorted occupancy measurements V^n and using the ground truth occupancy measurements Y^n is bounded by Δ in expectation, for any possible value of Y^n. The cost difference can be regarded as the control performance loss due to the usage of distorted data, and Δ stands for the tolerance on the control performance loss. C5 alone is a one-step performance guarantee, that is, it only bounds the cost difference associated with a single MPC iteration. In practice, MPC is repeatedly solved from the new initial temperature, yielding new control actions and temperature trajectories. In order to offer a guarantee for future cost difference, we introduce another constraint C6 on the resulting temperature difference of one MPC iteration. The idea is that the resulting temperature will become the new initial temperature of the next MPC iteration. If the resulting temperature difference between using distorted occupancy data and using true occupancy data is bounded within a small interval Δ_T, in the next MPC iteration C5 will provide a bound on cost difference for new initial temperatures that do not differ too much, since the cost difference constraint C5 is imposed to hold for all $|T_{init1}-T_{init2}|\leq\Delta'_T$. Typically, Δ'_T is set to be similar to Δ_T, but a small value of Δ'_T is preferred in order to assure the feasibility of the optimization problem (since the number of constraints increases with Δ'_T).

Now, we are ready to present the main optimization for privacy-enhanced HVAC controller by combining the privacy metric and performance constraint just presented. Suppose the assumptions of Proposition 21 hold. Given the control performance loss tolerance Δ, the *optimal distortion mechanism* is given by solving:

$$\min_{\substack{\mathbb{P}(V^n|Y^n)\\ n=1,\cdots,N}}\ \sum_{n=1}^{N} I(Y^n;V^n) \tag{6.12}$$

subject to the constraint C5-C6. Δ serves as a knob to adjust the balance between privacy and the controller performance loss. Increasing Δ leads to larger feasible set for the optimization problem, and thus a smaller value of MI (or privacy loss) is expected. Using the methodology presented in Section 6.2, we are able to calculate the terms inside the expectation in (6.10) and (6.11) for all $|T_{init1}-T_{init2}|\leq\Delta'_T$ and $y=0,\cdots,M$. Treating these as constants, calculating the optimal privacy-aware sensing mechanism is a convex optimization program, and can be efficiently solved. Additionally, since the constraints are enforced for each zone, the optimization (6.12) can actually be decomposed to N sub-problems and thus we can solve the optimal distortion scheme separately for each zone.

Remark on noisy occupancy data. In the preceding privacy-enhanced framework, we consider the occupancy can be accurately detected. In practice, the occupancy data

may be noisy itself, and thereby the distortion mechanism will be designed based on noisy occupancy W_k^n instead of true occupancy Y_k^n. In effect, the distortion designed using noisy occupancy provides an upper bound on the privacy loss. That is, in practice we could use noisy occupancy to design the distortion mechanism and the realized privacy loss can only be lower than the minimum privacy loss obtained from the optimization. Note that we have the Markov relationship: $Y_k^n \to W_k^n \to V_k^n$ when the distortion is applied to noisy data. Then the proof follows from the data processing inequality [33].

6.5 Evaluation

Experiment Setup

Occupancy dataset. The occupancy data used in this chapter is from the Augsburg Indoor Location Tracking Benchmark [127], which includes location traces for 4 users in a office building with 15 zones. The location data in the benchmark dataset was recorded every second over a period of 4 to 9 weeks. Since the dataset contains some missing observations due to technical issues or the vacation interruption, we finally use the dataset from November 5th to 24th in our experiment, during which the location traces of all the 4 users are complete, and subsample the dataset with 1-minute resolution. The ground truth occupancy data was synthesized by aggregating the locations trace of each user. Table 6.2 shows two statistics of the benchmark dataset. Notably, of all transitions per day, 66.7% to 84.6% either start from or end at one's own office, and office location can divulge one's identity. This sheds light on why location traces of individual users can be actually inferred from the "anonymized" occupancy data.

Table 6.2: The average number of transitions each user made in each workday, and the average percentage of transitions from or to one's office.

User	avg # of transitions per day	avg % of transitions from/to office per day
1	9.3	84.6%
2	20.2	75.4%
3	9.9	66.7%
4	7.6	75.5%

Adversary inference. We consider the adversary to be an *insider* with authorized building automation system access. One can think of it as the worst case of privacy breach, because insiders not only learn the ancillary information that is public-available, but are familiar with building operation policies. To be specific, the following auxiliary information is assumed to be available to the adversary: (1) Building directory and occupant mobility

patterns, encoded by the transition matrix of each occupant[2]; (2) Occupancy distortion mechanism designed by building manager.

The adversary attempts to reconstruct the most probable location trace given the occupancy data and the auxiliary information. That is, the attack is to find the MAP of location traces given the other information. The approach to finding MAP is well known as Viterbi algorithm in HMM. However, Viterbi is infeasible in the FHMM case as the location traces to be solved reside in a exponentially large state space ($N^M \times K$). We propose a fast inference method based on Mixed Integer Programming, and thus more efficiently evaluate the adversary's inference attack. The interested readers are referred to the code implementation of this chapter for the details of the fast inference algorithm.

Controller parameters. Without loss of generality, we consider the zones have the same thermal properties. The comfort range of temperature in the zones is defined to be within $24 - 26°C$ as in [120]. The minimum flow rate is set to be $0.084kg/s$ to fulfill the minimum ventilation requirement for $25m^2$-sized zone as per ASHRAE ventilation standard 62.1-2013 [5]. The optimization horizon of the MPC is 120 min, and the control commands are solved for and updated every 15 min [63]. Other design parameters are shown in Table 6.1, which bascially follows the choices in [92].

Platform. The algorithms are implemented in MATLAB; The interior-point algorithm is used to solve the bilinear optimization problem in MPC. To encourage the research on the privacy-preserving controller, the codes involved in this chapter will be open-sourced in `http://ruoxijia.github.io/4_code`.

Results

MI as proxy for privacy. We solve the MI optimization for different tolerance levels of control performance deterioration due to the usage of the distorted data, i.e., Δ, and obtain a set of optimal distortion designs and corresponding optimal values of MI. We then randomly perturb the true occupancy data using the different distortion designs, and infer location traces from the perturbed occupancy data. Monte Carlo (MC) simulations are carried out to assess results under the random distortion design. The inference accuracy is defined to be the ratio between the counts of correct location predictions over the total time steps. Figure 6.4 demonstrates the monotonically increasing relationship between adversarial location inference accuracy and MI, which justifies the usage of MI as a measure of privacy loss. When the adversary has perfect occupancy data, individual location traces can be inferred with accuracy of 96.81%. On the contrary, when the MI approaches zero, the adversary tends to estimate the location of each user to be constantly outside of the building, which is the best estimate the adversary can generate based on the uninformative occupancy data since people spend

[2] In the experiment, we use 4 days' occupancy data and 2 days' location traces to learn these parameters and the rest for evaluating our framework.

most of their time in a day outside. In this case, the inference accuracy is 77% but the adversary actually has no knowledge about users' movement. This serves as a baseline of the adversarial location inference performance.

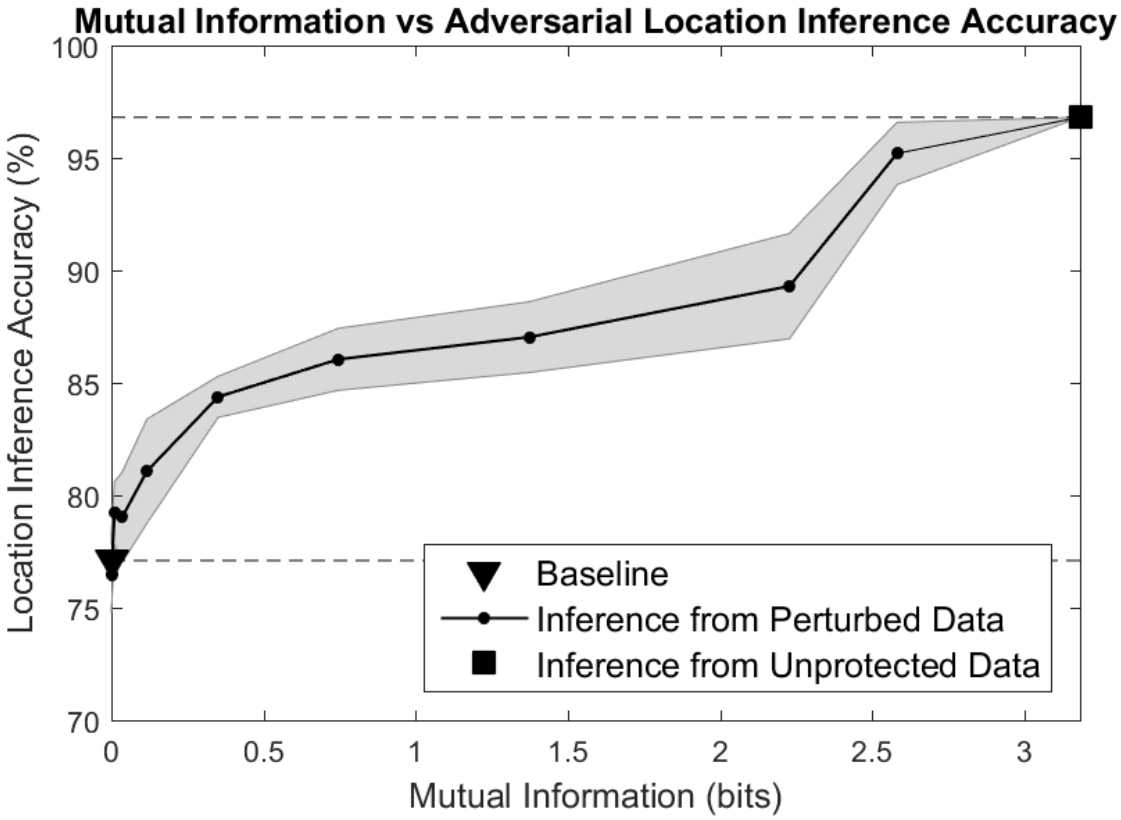


Figure 6.4: The adversary location inference accuracy increases as MI increases. The black line and the band around it show the mean and standard deviation of inference accuracy across ten MC simulations, respectively. The black square shows the location inference accuracy if the adversary sees true occupancy data. The black triangle gives the accuracy when the adversary outputs a constant location estimate.

Utility-Privacy Trade-off. Figure 6.5 shows the variation of privacy loss and controller performance loss with respect to different choices of Δ, which is the theoretical guarantee on controller performance loss. It is evident that privacy loss and control performance loss exhibit opposite trends as Δ changes. The privacy loss, measured by MI, monotonically decreases as Δ gets larger. This is the manifestation of the intrinsic utility-privacy trade-off embedded in the main optimization problem (6.12). As the performance constraint Δ is more relaxed, a smaller value of MI can be attained and thus privacy can be better preserved. The actual performance loss, measured by the HVAC control cost difference (between using distorted

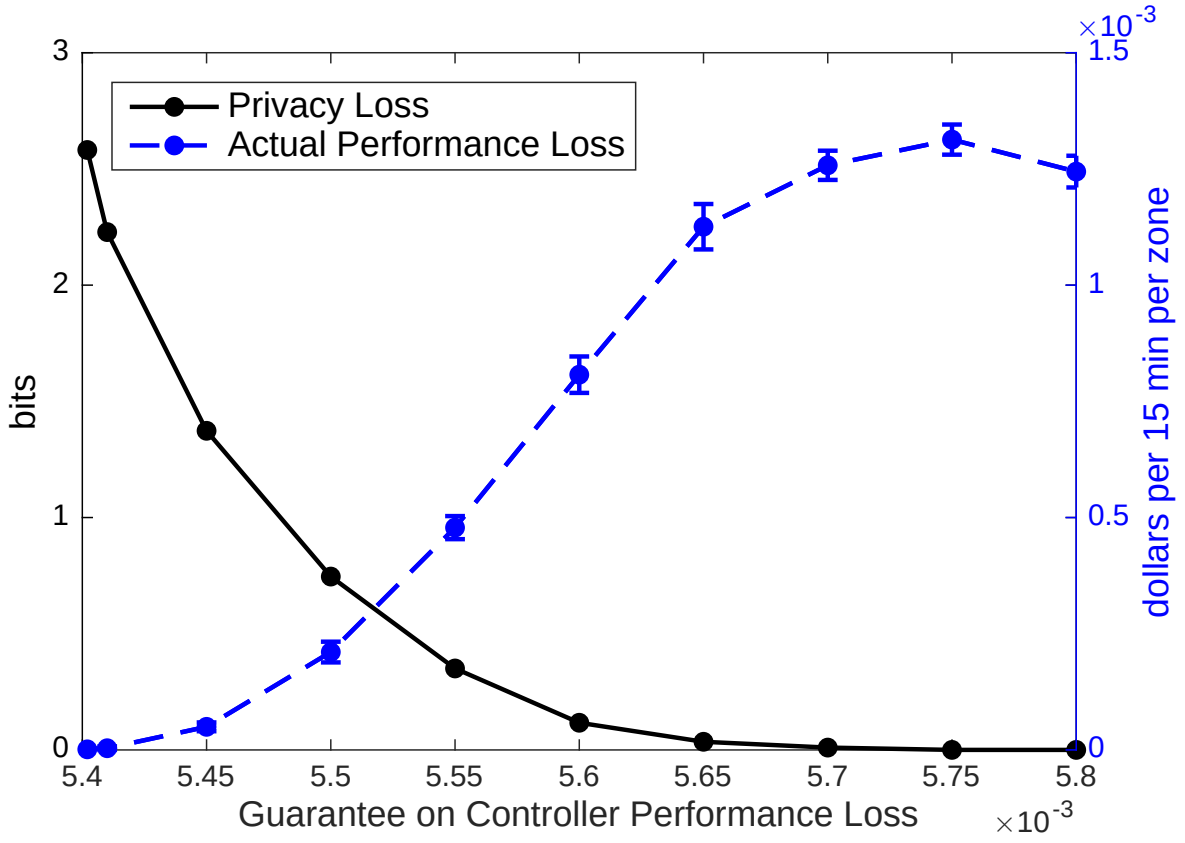


Figure 6.5: The changes of MI and actual control cost difference between using true and perturbed occupancy as the theoretical control cost difference changes. The blue dot line and errorbar demonstrate the mean and standard deviation of actual control cost difference across ten MC simulations, respectively.

and true data) averaged across different MPC iterations and difference zones, generally increases with Δ and is upper bounded by Δ. This indicates that the theoretical constraint on controller performance loss in our framework is effective and can actually provide a guarantee on the actual controller performance. We can see that the bound is far from tight, since the framework enforces the constraints on the controller performance for every possible true occupancy value to ensure the robustness while in practice the occupancy distribution is very spiked about the mean occupancy.

Figure 6.6 visualizes the distortion mechanism obtained by solving the MI under different choices of the tolerance on the control performance loss Δ. It can be clearly seen that the mechanism creates a higher level of distortion as Δ increases. When Δ is small, the resulting distortion matrix assigns most probability mass on the diagonal, i.e., the occupancy is very likely to keep unperturbed. As Δ gets larger, the distortion mechanism tends to have the same rows, in which case the distribution of distorted occupancy data is invariant under the

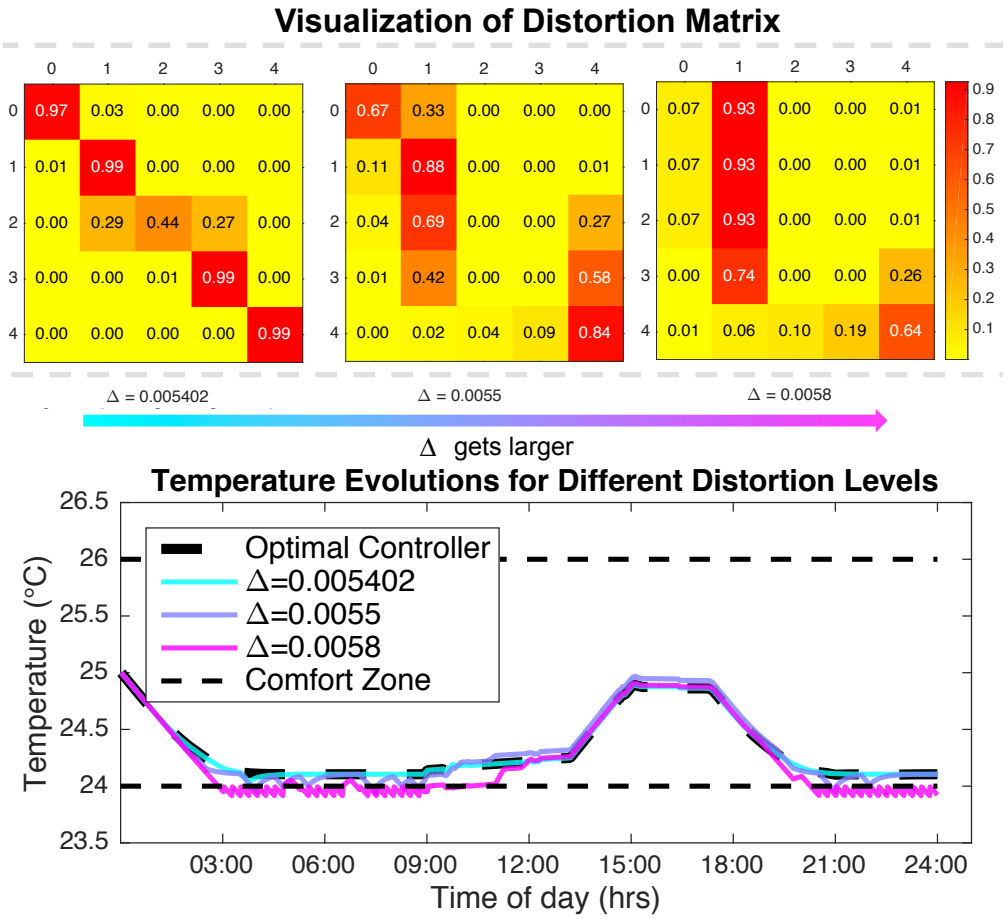


Figure 6.6: Illustration of distortion matrix $P(V|Y)$ under different controller performance guarantees. The row index corresponds to the value of Y, while column index corresponds to V. The zone temperature traces resulted from the controllers using occupancy data that is randomly distorted by different distortion matrices are also shown.

change of true occupancy and MI between true occupancy and perturbed occupancy, i.e., the privacy loss, tends to be zero. We also plot the temperature evolution under different distortion levels. Since we enforce a hard constraint on temperature, we can see that the zone temperature stays within the comfort zone for all Δ's. However, larger Δ would lead to a larger deviation from the temperature controlled using the true occupancy.

Comparison with Other Methods. We compare the performance of the HVAC controller using our optimally perturbed data against using unperturbed occupancy data, fixed occupancy schedule as well as randomly perturbed data by other distortion methods. In Figure 6.7 (a), we plot the privacy loss and control cost for controllers that use the various forms of occupancy data. Fixed occupancy schedule (assuming maximum occupancy during

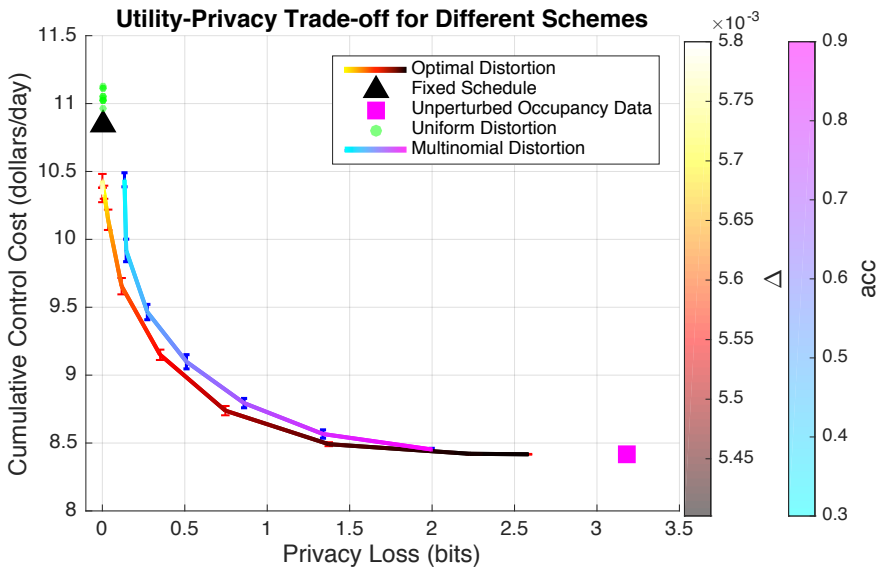
working hours and zero otherwise) exposes zero information about individual location traces, but cannot adapt to occupancy variations and thus incurs considerable control cost. The controller based on clean occupancy data is most cost-effective but discloses maximum private information. One of the random distortion method to be compared is uniform distortion scheme in which the true occupancy is perturbed to some value between zero to maximum occupancy with equal probability. We carry out 10 MC simulations to obtain the control cost incurred under this random perturbation scheme. It can be seen that the uniform distortion scheme protects the private information with compromised controller performance.

A natural question arising is if the current occupancy sensing systems provide intrinsic privacy-preserving features as there always exists occupancy estimation errors. Can we use a cheaper and inaccurate occupancy sensor to acquire privacy? As is suggested by the occupancy sensing results in [89], the estimation noise of a real occupancy sensing system can be modeled by a multinomial distribution which has most probability mass at zero. Inspired by this, we use the following multinomial distortion schemes to imitate a real occupancy sensing system with disparate accuracies acc,

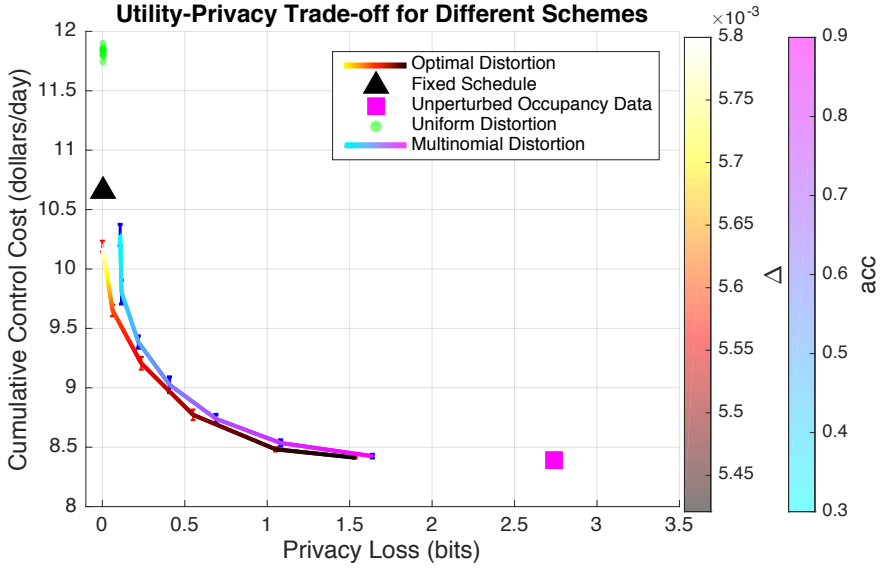
$$P(V^n|Y^n = y) = \begin{cases} acc, & V = y \\ \frac{1-acc}{2}, & V = y-1 \text{ or } y+1 \text{ if } y \neq 0 \\ \frac{1-acc}{2}, & V = 1 \text{ or } 2, \text{ if } y = 0 \end{cases} \tag{6.13}$$

Again, MC simulations are performed to evaluate the control performance under this random perturbation, and the results are shown in Figure 6.7 (a). It can be seen that when the privacy loss is relatively large (or data is slightly distorted), the control cost of our optimal noising scheme and the multinomial noising scheme do not differ too much. This is because at this level of privacy loss the two distortion schemes behave similarly, as shown in Figure 6.6, where the occupancy keeps untainted with high probability. But as the privacy loss decreases, our optimal noising scheme's intelligent noise placement begins to significantly improve control performance. In addition, our optimal distortion Pareto dominates the other schemes.

To investigate the scalability of our proposed scheme, we create synthetic data that simulates location traces for 15 occupants based on the Augsburg dataset. We extract the occupants' movement profile, i.e., transition parameters, from the original dataset and randomly assign the profiles to synthesized occupants. An occupant randomly chooses the next location according to the movement profile. The privacy-utility curve evaluated on this larger synthesized dataset is illustrated in Figure 6.7 (b), which demonstrates that the optimality of our distortion scheme is preserved when the experiment is scaled up. We can see that the privacy loss of the controller using the unperturbed occupancy gets lower when incorporating more occupants. Although privacy risks are lower as we scale up the experiment since with more people sharing the space it will be more difficult to identify each individuals, adding distortion to occupancy measurements can preserve the privacy even further as shown in Figure 6.7 (b).



(a)



(b)

Figure 6.7: Comparison of the privacy-utility trade-off of controllers using different forms of occupancy data, evaluated based on (a) real-world occupancy data and (b) synthesized data.

6.6 Chapter Summary

In this chapter, we present a tractable framework to model the trade-off between privacy and controller performance in a holistic manner. We take occupancy-based HVAC controller as an example where the objective is to utilize occupancy data to enable smart controls over

the HVAC system while protect individual location information from being inferred from the occupancy data. We use MI as the measure of privacy loss, and formulate the privacy-utility trade-off by a convex optimization problem that minimizes the privacy loss subject to a pre-specified controller performance constraint. By solving the optimization problem, we can obtain a mechanism that injects optimal amount of noise to occupancy data to enhance privacy with control performance guarantee. We verify our framework using real-world occupancy data and simulated building dynamics. It is shown that our theoretical framework is able to provide guidelines for practical privacy-enhanced occupancy-based HVAC system design, and achieve a better balance of privacy and control performance compared with other occupancy-based controllers.

One limitation of the current framework is the requirement of a detailed model of the HVAC system. In practice, the model can be learned from the data generated by the system, including control actions, state changes, etc. We plan to explore the impact of modeling errors on the privacy mechanism.

Chapter 7

Data Minimization and Free-Lunch Privacy

7.1 Background

One of the hallmarks of CPS, ranging from smart homes, smart transportation systems, smart energy systems, to smart cities, is that data collected from individuals or entities serve an indispensable part of decision making and control underlying the systems' operation. For instance, a household's occupancy is being sensed to inform the control over lighting and heating for greater energy efficiency; taxi drivers continually share their GPS data with a central dispatch solver in order to receive automated and optimized route suggestions. The individuals and entities that engage with these CPS infrastructures would naturally wish to preserve privacy of their data.

Different CPS operations require data with varying degrees of granularity, which can be generally categorized into *aggregate-level* and *individual-level*. In the former case, the CPS operation is contingent on the statistics extracted from aggregate data of a group of individuals. A typical example is smart grid load balancing, which utilizes the agglomeration of smart meter data to forecast future demand. In these types of operations, it can be assumed that there is a database that contains data record of relevant population and the operator queries the database to acquire the information of interest to system operation. There has been fruitful research of privacy preservation in this setup. Popular privacy metrics include differential privacy [49, 32] and k-anonymity [152], whose commonality is to hide an individual's private data in an aggregate of population, also known as "hiding in the crowd".

In contrast, there are also CPS applications, such as the aforementioned smart home control and taxi dispatch examples, where service is delivered based on individual data rather than aggregate information. The access to personal records is necessitated in this type of CPS due to its nature of personalized service provision. Privacy-preserving technologies developed under the "aggregate" setup cannot be directly applied here, as there is essentially "no crowd in which to hide". Several privacy metrics have been studied to protect data at the

individual level as opposed to at the database level. One of the oldest ones is called local differential privacy [163, 47], which provides plausible deniability by randomizing individual records. Another widely studied approach is to use information-theoretic measures to model privacy loss [129, 139], and protect privacy by adding optimally designed noise such that the privacy loss is minimized under some utility constraints.

In this chapter, we focus on privacy protection at an individual level in CPS applications. Our work is originally inspired by the observations in smart home heating system control. It is found that many of the occupancy measurements are redundant in the sense that their values do not change the optimal control of the heating system. We formalize our observations from the smart home application, and propose the concept of *free-lunch privacy*. The idea is to identify the critical data which must be reported truthfully in order to enjoy the optimal service, versus the unimportant data which can be concealed or disturbed without sacrificing control performance. Free-lunch privacy exists at a data point if it can be replaced by a falsified value without harming the utility. Free-lunch privacy is a pragmatic privacy metric since its objective is to always guarantee the optimality of service provided by CPS while reducing private data release by exploiting the possible insensitivity of CPS operation to data measurements.

The contributions of this chapter are as follows: Firstly, we develop a systematic approach to characterize the utility of data for control or a decision making process that can be characterized by an optimization problem. This approach can also be applied to information-theoretic privacy framework to rigorously study the tradeoff between privacy and control performance. We formalize the free-lunch privacy and study its existence and methods to compute it. Secondly, we propose a free-lunch privacy mechanism that processes original data into the one that contains minimum private information for realizing optimal control. It is a selective data disclosure procedure that materializes the Fair Information Principles [31], and can be also treated as an adaptive local differential privacy mechanism such that the reported data conforms with local differential privacy guarantee when it is not crucial for control. We also study the implication of free-lunch privacy mechanism for adversarial inference on the random process that generates an individual's data. Lastly, we present a case study to show the use of free-lunch privacy mechanism in heating, ventilation, and air conditioning (HVAC) control in smart buildings.

7.2 Problem Formulation

General setup

We consider a service provider that requires its end user to send personal information for delivering customized service. Let the user's data be $\theta \in \Theta$. The service is considered the result of a data-dependent decision making process, characterized by the following

optimization problem

$$\begin{aligned} x^*(\theta) &= \arg\min_x J(x,\theta) \\ s.t. \quad & g(x,\theta) \leq 0 \end{aligned} \tag{7.1}$$

where $x \in \mathcal{X}$ is the decision variable, and the data θ can be treated as the *parameter* of the optimization problem. *User data* and *parameter* will be used interchangeably to indicate θ.

1) Example 1 (Occupancy controlled thermostat in smart home): The smart thermostat continually monitors home's occupancy via an occupancy sensor, and uses it as an input to an Model Predictive Controller (MPC) that is designed to minimize the energy cost while maintaining users' comfort [80]. Here, $\theta \in \{0, 1\}$, indicating home owner's presence/absence. x stands for the control actions, such as air flow rate, that regulate indoor temperature. $J(x, \theta)$ is the objective function of MPC. $g(x, \theta)$ includes physical and comfort constraints in the MPC.

2) Example 2 (Taxi dispatch with real-time GPS data): Drivers' real-time GPS location data can be used for improving taxi dispatch efficiency. An optimal taxi dispatch problem is described in [114], where the objective $J(x, \theta)$ is to minimize the supply-demand mismatch and idle driving distance of all taxis, x includes dispatch order matrix and idling driving distance, θ is drivers' location information and considered to be privacy-sensitive, and $g(x, \theta)$ represents a set of operational constraints.

3) Example 3 (Integrated heat and electricity dispatch): Coordination between heat and electricity dispatch leads to a number of synergistic benefits including lower operational cost and more reliable energy supply. Often, in a large city there are many small district heating systems (DHSs) owned by different companies and only one electricity control center (ECC) managing the entire electricity system. To achieve combined heat and power, DHSs need to share with ECC the detailed information regarding the heat loads, which raises the concern of privacy [123]. In this example, θ includes information about the heat loads, x stands for the heat and electricity dispatch decision, and $J(x, \theta)$ and $g(x, \theta)$ represent the total operational cost and constraints, respectively.

In this chapter, we focus on the scenario where the required data for service provision is privacy-sensitive, as exemplified by the examples above. Our objective is to study the sensitivity of a service $x^*(\theta)$ to the variation of a data value θ. In particular, we are seeking an "equivalence set" that contains data points resulting in the same service as the true data. If the set exists, the true data can be replaced by an arbitrary point in its equivalence set without changing the service and thereby "free-lunch" privacy is achieved.

We will narrow down our scope to modeling the interaction between a single user and the service provider. In future work, we hope to extend the framework to multi-user scenario, where many interesting issues arise. For example, how does a system designer allocate privacy between users? There could be situations where only some of users can hide their parameters, and there needs to be a mechanism to decide which users get to experience "free-lunch privacy". Additionally, there is the potential for information leakage about other users through the sharing of equivalence sets.

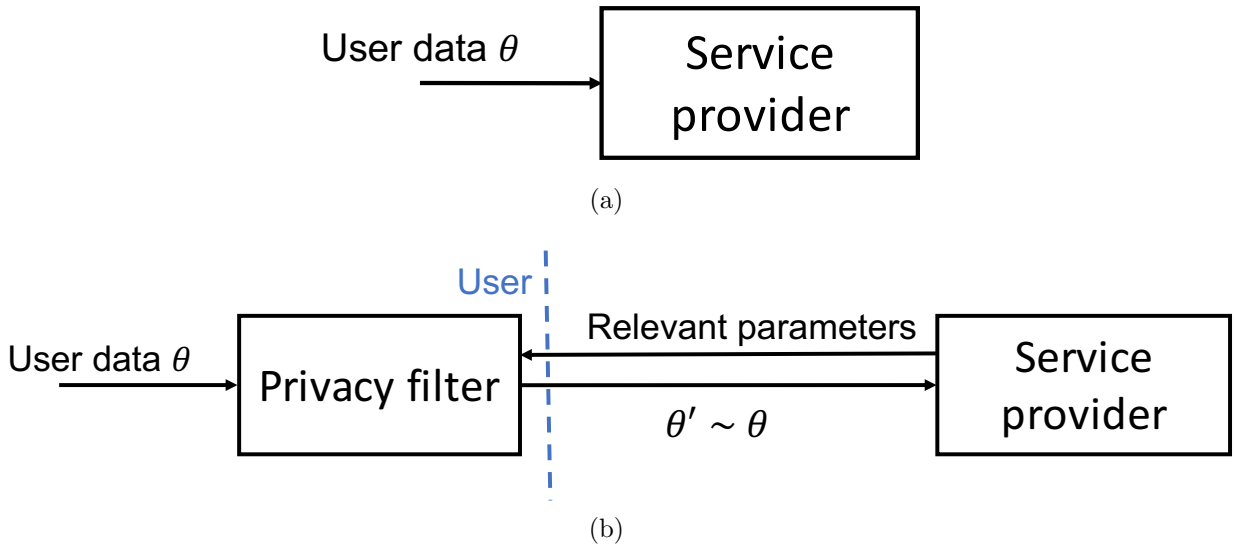


Figure 7.1: Diagram of (a) Traditional data sharing (b) Free-lunch privacy mechanism.

We now formally state the definition of "equivalence set" and "free-lunch privacy".

Definition 6. *For a given optimization problem in the form of (7.1), θ is said to be equivalent to θ', denoted by $\theta \sim \theta'$, if and only if $x^*(\theta) = x^*(\theta')$.*

Definition 7 (Equivalence class)**.** *The equivalence class of $\theta \in \Theta$ is the set $[\theta] = \{\theta' \in \Theta : \theta' \sim \theta\}$*

Proposition 22. *The relation defined in Definition 6 is in fact an equivalence relation, i.e. it is reflexive, symmetric, and transitive. Furthermore, the collection of equivalent classes of Θ is a partition $\mathcal{P}$ of Θ, i.e., every element of Θ belongs to one and only one equivalence class.*

Definition 8 (Free-lunch privacy)**.** *The optimal decision making problem given by (7.1) is said to enjoy free-lunch privacy at θ, if $[\theta]$ is not singleton.*

Proposition 22 allows us to devise a data sharing mechanism that exploits free-lunch privacy in an optimal decision making process, illustrated by Fig. 7.1 (b). Instead of directly requesting data from the user, the *free-lunch privacy mechanism* features a two-way communication as follows:

1. The service provider sends the equivalence partition of data space Θ, or the relevant information to compute the equivalence set of any given data to the user

2. The user randomly selects a data point in the equivalence set of the true data and report it to the service provider

If the equivalence set of the true data is a singleton, then the user will end up reporting the true data; otherwise, the proposed mechanism is able to hide private data without impacting the optimal decision.

We can see that the degree of privacy protection offered by the free-lunch privacy mechanism hinges on the existence of non-singleton equivalence sets. Next, we will focus on quadratic programming which has been extensively used in MPC, resource allocation, and financial applications, and study the geometry of the equivalence sets associated with the quadratic optimization problem.

Quadratic programming setup

Notations. If G is a matrix, then G_i denotes the ith row of G. In addition, if A is a index set, then G_A denotes the submatrix of the rows of G corresponding to the index set A. The dimension of a polyhedron P is the dimension of its affine hull and denoted by $\dim(P)$. If $\dim(P) = n$, then the polyhedron is said to be full-dimensional. The closure and interior of a set S is denoted by $\mathrm{cl}(S)$ and $\mathrm{int}(S)$, respectively.

We consider the following optimization with strictly convex quadratic objective and linear constraints,

$$\begin{aligned} J^*(\theta) = \min_x J(x,\theta) &= \frac{1}{2}x^T H x \\ \text{s.t.} \quad Gx &\leq w + S\theta \end{aligned} \tag{7.2}$$

where $x \in \mathbb{R}^s$, $\theta \in \mathbb{R}^n$, $G \in \mathbb{R}^{m\times s}$, $w \in \mathbb{R}^m$, and $S \in \mathbb{R}^{m\times n}$. $H \in \mathbb{R}^{s\times s}$ and $H \succ 0$. Note that the more general problem with $J(x,\theta) = \frac{1}{2}x^T H x + \theta^T F x$, where the objective and constraints are both dependent on the user data θ, can always be reformulated into the form of (7.2) by using the variable substitution $\tilde{x} = x + H^{-1}F^T\theta$. We denote by Θ^* the region of parameters such that (7.2) is feasible:

$$\Theta^* = \{\theta \in \mathbb{R}^n : \exists x \text{ satisfying } Gx \leq w + S\theta\} \tag{7.3}$$

Let $I = \{1, \cdots, m\}$ be the indices of constraints. In what follows, we define the key concepts for stating the main results of the chapter.

Definition 9 (Optimal active set)**.** *Let x be a feasible point of (7.2) for a given θ. The active constraints are the constraints that satisfy $G_i x - w_i - S_i\theta = 0$. The indices of the constraints that are active at $x^*(\theta)$ is referred to as the optimal active set, denoted by $\mathcal{A}(\theta)$, i.e.,*

$$\mathcal{A}(\theta) = \{i \in I : G_i x^*(\theta) - w_i - S_i\theta = 0\} \tag{7.4}$$

We also define as weakly active constraint an active constraint with an associated zero Lagrange multiplier, and as strongly active constraint an active constraint with a positive Lagrange multiplier. The optimal inactive set is similarly defined as

$$\mathcal{N}(\theta) = \{i \in I : G_i x^*(\theta) - w_i - S_i\theta < 0\} = I \setminus \mathcal{A}(\theta) \tag{7.5}$$

Definition 10 (Critical region)**.** *Given an index set $A \subseteq I$, the critical region CR_A associated with A is the set of parameters for which the optimal active set is equal to A, i.e.,*

$$CR_A = \{\theta \in \Theta^* : \mathcal{A}(\theta) = A\} \tag{7.6}$$

Definition 11 (LICQ)**.** *We say that Linear Independence Constraint Qualification (LICQ) holds at θ if the set of constraints indexed by $\mathcal{A}(\theta)$ are linearly independent, i.e., $G_{\mathcal{A}(\theta)}$ has full row rank.*

Theorem 23 ([19])**.** *Consider the parameteric quadratic programming in (7.2). The optimizer function $x^*(\theta) : \Theta^* \to \mathbb{R}^s$ is continuous and piecewise affine in the sense that there exists a finite set of full-dimensional polyhedral critical regions $\mathcal{CR} = \{CR_1, \cdots, CR_K\}$ such that $\Theta^* = \cup_{k=1}^K CR_k$, $\text{int}(CR_i) \cap \text{int}(CR_j) = \emptyset$ for all $i \neq j$ and $x^*(\theta)$ is affine inside each critical region CR_k, i.e., $x^*(\theta) = F_k\theta + g_k$, for all $k \in \{1, \cdots, K\}$.*

Note that if the subscript of CR is an index, e.g., CR_i, then it will be used to denote the i-th critical region; if the subscript of CR is a set, e.g., CR_A, then it stands for the critical region whose optimal active set is A.

Theorem 23 indicates that the parameter space can be partitioned into a collection of full-dimensional critical regions, and the optimizer function on each critical region is an affine function whose parameters can be explicitly written out.

The expression of critical regions and the optimizer function associated with each critical region can be obtained from the KKT conditions. The detailed derivation of F_k, g_k, and the characterization of CR_k can be found in [19].

7.3 Equivalence Set

Computation method

We start by addressing the problem of calculating the equivalence partition of the parameter space, with which the user can hide their original data while enjoying the same service as when reporting the truth. To do that, an algorithm is needed to compute the equivalence set of any given parameter.

Instead of exhaustively searching over the entire parameter space and checking the equivalence between every parameter and the given parameter, Theorem 23 allows us to search the parameter space in a region-by-region manner and directly solve for the equivalence set constrained to each critical region.

Theorem 24. *Given any $\theta^* \in \Theta^*$, let CR_{k^*} be be critical region whose closure contains θ^*. Then,*

$$[\theta^*] = \cup_{k=1}^K (CR_k \cap \mathcal{P}_k) \tag{7.7}$$

where

$$\mathcal{P}_k = \{\theta : F_k\theta = F_{k^*}\theta^* + g_{k^*} - g_k\} \tag{7.8}$$

Theorem 24 leads to Algorithm 7, which inspects every critical region in the parameter space, and calculates the intersection of each critical region with a linear subspace that achieves the same optimizer as the one at the true parameter θ^*.

Algorithm 7 Equivalence set computation

Input: Parameter θ^*, critical region CR_k and associated optimizer function parametrized by F_k and g_k $(k = 1, \cdots, K)$

Output: Equivalence set $[\theta^*]$

1: Find k^* such that CR_{k^*} contains θ^* $\mathcal{S} \leftarrow \emptyset$
2: **for** $k = 1, \cdots, K$ **do**
3: $\quad \mathcal{P}_k \leftarrow \{\theta : F_k\theta = F_{k^*}\theta^* + g_{k^*} - g_k\}$
4: $\quad \mathcal{S} \leftarrow \mathcal{S} \cup (CR_k \cap \mathcal{P}_k)$
5: **end for**

Next, we present some sufficient conditions that can be used to exclude the critical regions that do not contain any equivalent parameters to a given parameter without the need to explicitly compute the intersection.

Theorem 25. *Assume that A and B are the optimal active sets associated with two full dimensional critical regions CR_A and CR_B and that LICQ holds on A and B, i.e., G_A and G_B both have full row rank. Moreover, assume that there are no constraints which are weakly active at the optimizer $x^*(\theta)$ for all θ in $\text{int}(CR_A)$ or $\text{int}(CR_B)$. Let $U = A \cup B$. If G_U has linear independent rows, then $x^*(\theta_1) \neq x^*(\theta_2)$ for all $\theta_1 \in \text{int}(CR_A)$ and $\theta_2 \in \text{int}(CR_B)$.*

Proof. The KKT conditions for (7.2) are:

$$Hx^* + G^T u^* = 0, u^* \in \mathbb{R}^m \tag{7.9}$$
$$u_i^*(G_i x^* - w_i - S_i\theta) = 0 \tag{7.10}$$
$$Gx^* - w - S\theta \leq 0 \tag{7.11}$$
$$u_i^* \geq 0, \forall i \in \{1, \cdots, m\} \tag{7.12}$$

It follows that $x^* = -H^{-1}G^T u^*$. Since for inactive constraints $u_i^* = 0$, we have

$$x^* = -H^{-1}G_A^T u_A^*, \text{ on } CR_A \tag{7.13}$$
$$x^* = -H^{-1}G_B^T u_B^*, \text{ on } CR_B \tag{7.14}$$

We now prove the result by contradiction. Assume that there exist $\theta_1 \in \text{int}(CR_A)$ and $\theta_2 \in \text{int}(CR_B)$ such that $x^*(\theta_1) = x^*(\theta_2)$. Since H^{-1} is invertible and therefore a bijective mapping, we have $G_A^T u_A^* = G_B^T u_B^*$, i.e.,

$$\sum_{j \in A \cap B} u_{A,j}^* G_j^T + \sum_{j \in A \setminus B} u_{A,j}^* G_j^T = \sum_{j \in A \cap B} u_{B,j}^* G_j^T + \sum_{j \in B \setminus A} u_{B,j}^* G_j^T \tag{7.15}$$

which implies $u^*_{A,j} = 0$ for $j \in A \backslash B$ and $u^*_{B,j} = 0$ for $j \in B \backslash A$, which contradicts the assumption that there are no weakly active constraints in $\text{int}(CR_A)$ and $\text{int}(CR_B)$. □

Corollary 26. *Consider the same assumptions as in Theorem 25, if G_B can be obtained by adding or deleting rows from G_A, i.e., $A \subset B$ or $B \subset A$, then $x^*(\theta_1) \neq x^*(\theta_2)$, $\forall \theta_1 \in \text{int}(CR_A), \theta_2 \in \text{int}(CR_B)$.*

Lemma 7 ([148, 155])**.** *Let A be a given optimal set for some $\theta \in \Theta^*$. Assume that (1) LICQ holds for A, (2) there are no coinciding inequalities for facet of $\text{cl}(CR_A)$, and (3) there are no weakly active constraints at $x^*(\theta)$ for all $\theta \in \text{cl}(CR_A)$. Then, the optimal active set of any given adjacent critical region of CR_A only differs A in one element.*

Corollary 27. *For any two adjacent critical regions satisfying the assumptions in Lemma 7, denoted by CR_1 and CR_2, we have $x^*(\theta_1) \neq x^*(\theta_2)$, $\forall \theta_1 \in \text{int}(CR_1), \theta_2 \in \text{int}(CR_2)$.*

Example 28. *Consider the problem [19] where*

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, G^T = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{bmatrix}$$

$$S^T = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -2 \\ 1 & -1 & -1 & 1 & 3 & -1 \end{bmatrix} \tag{7.16}$$

$$w^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{7.17}$$

and $-1 \leq \theta_1, \theta_2 \leq 1$. The critical regions and corresponding optimal active sets are depicted in Fig. 7.2. For instance, we are interested in computing the equivalence set of $\theta = [-0.4, 0]^T$, which resides in $CR_{\{6\}}$. Corollary 26 indicates that critical region $CR_\emptyset$, $CR_{\{3,6\}}$, $CR_{\{4,6\}}$ can be excluded from the search as the corresponding optimal active set either includes or is contained by $\{6\}$.

Existence of non-singleton equivalence set

We now consider the question: under what conditions is free-lunch privacy guaranteed to exist at a given parameter? According to the definition of free-lunch privacy, this is equivalent to examining when the equivalence set of a parameter is a non-singleton. We will first focus on the geometry of the equivalence set constrained to the critical set that contains the given parameter, because (1) this critical region is only one that is always guaranteed to include equivalent points to the given parameter among all critical regions, and (2) the existence of a non-singleton equivalence set on this critical region serve as a sufficient condition for the existence on the entire parameter space.

Definition 12 (Constrained equivalence set)**.** *For a given parameter θ^*, let the critical region containing θ^* be CR_{k^*}. The constrained equivalence set of θ^* is the set $[\theta^*]_c = \{\theta : \theta \in [\theta^*], \theta \in CR_{k^*}\}$.*

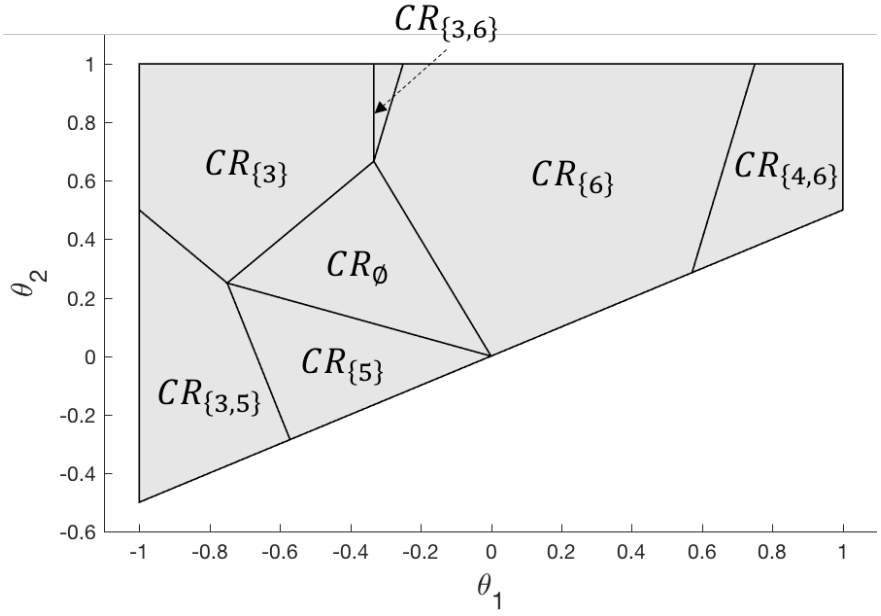


Figure 7.2: Critical regions for Example 28. [19]

Theorem 29. *Given a parameter θ^*, assume that $\theta^* \in \text{int}(CR_{k^*})$. Then,* $\dim([\theta^*]_c) = \dim(\ker(F_{k^*}))$.

Proof. By Theorem 23, $[\theta^*]_c$ is the intersection of CR_{k^*} with a linear subspace characterized by

$$\{\theta : F_{k^*}\theta = F_{k^*}\theta^*\} = \theta^* + \ker(F_{k^*}) \coloneqq L \tag{7.18}$$

where $\ker(\cdot)$ denotes the kernel of a matrix, i.e., $\ker(F_{k^*}) = \{x : F_{k^*}x = 0\}$. It is clear that $\dim([\theta^*]_c) \leq \dim(\ker(F_{k^*}))$. We next prove $\dim([\theta^*]_c) \geq \dim(\ker(F_{k^*}))$ to obtain the theorem. By the assumption that $\theta^* \in \text{int}(CR_{k^*})$, there exists a full-dimensional ball centered at θ^* with some radius r_1 that can be fitted in $\text{int}(CR_{k^*})$. Let the ball be denoted by $B_1 = \{\theta \in \mathbb{R}^n : \|\theta - \theta^*\|_2 \leq r_1\}$. Because θ^* also belongs to L, there exists another ball B_2 centered at θ^* with radius $r_2 < r_1$ on L, i.e., $B_2 = \{\theta \in L : \|\theta - \theta^*\|_2 \leq r_2\}$. Since $B_2 \subset B_1$, any θ in B_2 also belongs to $(B_1 \cap B_2) \subset [\theta^*]_c$. This implies that a ball with dimension same as L can be fitted in $[\theta]_{k^*}$. Therefore, we can conclude that $\dim([\theta^*]_c) = \dim(L) = \dim(\ker(F_{k^*}))$ □

The following corollary derived from Theorem 29 gives sufficient conditions for the existence of free-lunch privacy at a given parameter.

Corollary 30. *Given θ^* and assume $\theta^* \in \text{int}(CR_{k^*})$. Let A be the optimal active set at θ^*. It is ensured that $[\theta^*]$ is non-singleton if any one of the following conditions is met:*

- *Case (1): No constraints are active at θ^*, i.e., $A = \emptyset$;*

- *Case (2): The number of optimization variable is less than the number of parameters, i.e.,* $s < n$*;*
- *Case (3): The rank of* G_A *is less than the number of parameters, i.e.,* $\text{rank}(G_A) < n$*; particularly, if LICQ holds then this condition reduces to checking if the number of active constraints at* θ^* *is less than the number of parameters, i.e.,* $|A| < n$*.*

Proof. We consider the optimizer function in three different situations.

1): $A = \emptyset$. Since for inactive constraints $u^*_{I\setminus A} = 0$, we have $x^*(\theta) = -H^{-1}G^T u^* = -H^{-1}G^T u^*_{I\setminus A} = 0$. Therefore, free-lunch privacy holds on the entire critical region CR_A.

2): $A \neq \emptyset$ and LICQ holds. By [19],

$$F_{k^*} = H^{-1}G_A^T(G_A H^{-1} G_A^T)^{-1} S_A \tag{7.19}$$

which follows that $\text{rank}(F_{k^*}) \leq \min\{|A|, s, n\}$. In addition, by the rank-nullity theorem we have

$$\dim(\ker(F_{k^*})) = n - \text{rank}(F_{k^*}) \tag{7.20}$$

Therefore, if $s < n$ or $|A| < n$, then $\dim(\ker(F_{k^*})) \geq 1$ which implies free-lunch privacy at θ^* by Theorem 29.

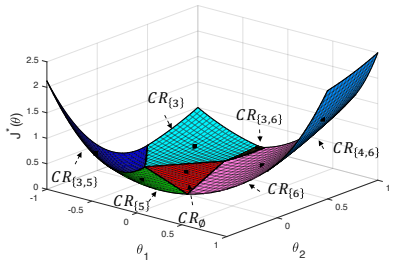
3): $A \neq \emptyset$ and LICQ does not hold. Let $l = \text{rank}(G_A)$. Again, using the results in [19] we have

$$F_{k^*} = H^{-1}G_{A,1}^T U_1^{-1} P \tag{7.21}$$

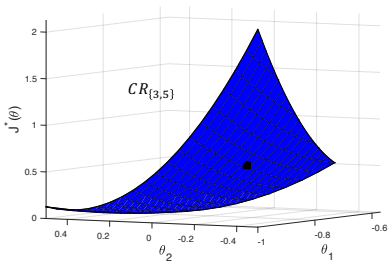
where $G_{A,1} \in \mathbb{R}^{s\times l}$, $U_1 \in \mathbb{R}^{l\times l}$, $P \in \mathbb{R}^{l\times n}$. Therefore, we have $\text{rank}(F_{k^*}) \leq \min\{s, l, n\}$. If $s < n$ or $l < n$, then $\dim(\ker(F_{k^*})) \geq 1$ is always ensured by the rank-nullity theorem.

Summarizing 1)-3), we prove the corollary. □

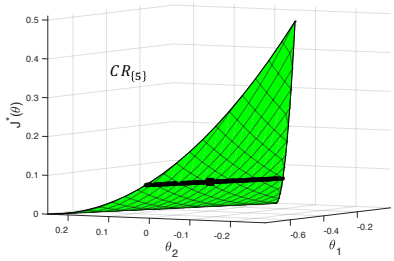
Example 1 (continued): We use the problem described in Example 28 to demonstrate the utility of Corollary 30. We consider the constrained equivalence set of the Chebyshev center of each critical region. Let the Chebyshev center of critical region CR_A be denoted by θ^*_A. First, notice that there is no constraint active at $\theta^*_\emptyset$. By Case (1) in Corollary 30, free-lunch privacy always exists at $\theta^*_\emptyset$. This is validated by Fig 7.3 (d), where every point in the critical region is equivalent to $\theta^*_\emptyset$. Secondly, since $\theta^*_{\{3\}}$, $\theta^*_{\{5\}}$, and $\theta^*_{\{6\}}$ all have one active constraint, and the free-lunch privacy at these points are guaranteed by Case (3) in Corollary 30, as illustrated by Fig. 7.3 (e), 7.3 (c) and 7.3 (f), respectively. For $\theta^*_{\{3,5\}}$, $\theta^*_{\{3,6\}}$, and $\theta^*_{\{4,6\}}$, the number of active constraints are equal to the number of parameters and free-lunch privacy at these parameters are not ensured. As shown in Fig. 7.3 (b), 7.3 (g), 7.3 (h), the equivalence set associated with each of these parameters collapses to a singleton.



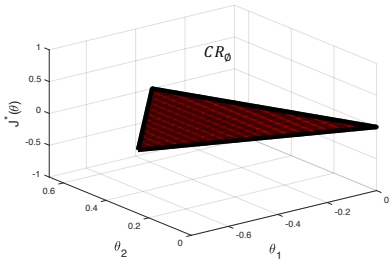
(a) Optimal value function



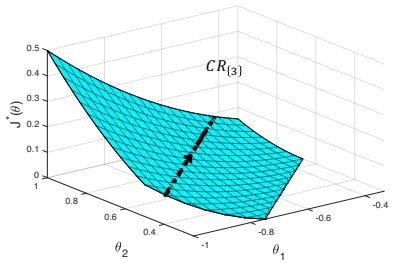
(b) $AC\# = 2$, $\dim(ES) = 0$



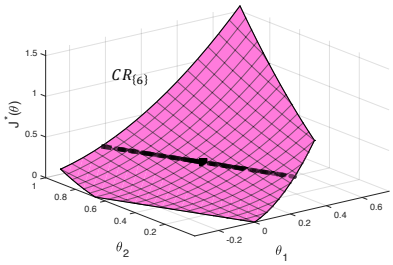
(c) $AC\# = 1$, $\dim(ES) = 1$



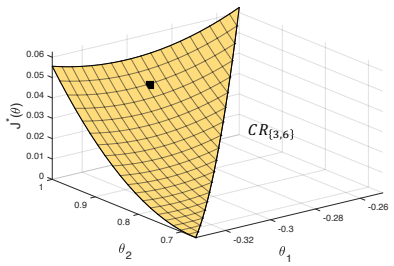
(d) $AC\# = 0$, $\dim(ES) = 2$



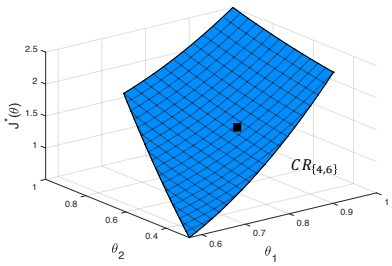
(e) $AC\# = 1$, $\dim(ES) = 1$



(f) $AC\# = 1$, $\dim(ES) = 1$



(g) $AC\# = 2$, $\dim(ES) = 0$



(h) $AC\# = 2$, $\dim(ES) = 0$

Figure 7.3: (a) illustrates the optimal value function on each critical region; (b)-(h) demonstrates the constrained equivalence set of the Chebyshev center of each critical region. The constrained equivalence set is shown in black line. $AC\#$ stands for the number of active constraints, and $dim(ES)$ represents the dimension of constrained equivalence set.

7.4 Implications to Statistical Estimation

The analysis in the preceding section implies that free-lunch privacy may not always exist. If the user's data is shared in stream, then there will be some data points which can be hidden without sacrificing the performance while sometimes the truthful data must be reported in order to maintain the optimality of a decision making process.

Assume that a user shares his/her data multiple times (continually or intermittently) to acquire the service. We consider two types of adversaries:

- A weak adversary, who can eavesdrop the communication link from the user to service provider and can intercept any data shared by the user;
- A strong adversary, who has access to not only the user data but the optimization problem that the service provider solves to offer the service.

Both adversaries are interested in learning the parameter of the underlying random process that generates the user's data. We further assume that users' data is exchanged via a simplified version of the free-lunch privacy mechanism described in 7.2. The simplified mechanism only calculates the constrained equivalence set of a given data point in order to improve computational efficiency. More specifically, in the simplified mechanism the user randomly selects a data point in the constrained equivalence set of the truthful data and report it to the service provider.

In this section, we will use a simple example to demonstrate that the free-lunch privacy mechanism is able to protect the statistics of a user's data against adversarial inference.

Assume the user's data is generated from an identically distributed and independent (i.i.d.) one-dimensional Gaussian source with mean μ and a known variance σ^2, i.e., $\theta_t \sim \mathcal{N}(\mu, \sigma^2)$, where θ_t represents t-th data sharing. The adversaries are interested in inferring μ from data observations under the simplified free-lunch privacy mechanism.

We first present a theorem that describes the geometry of constrained equivalence sets in the one-dimensional case, which helps us model the data reported under the privacy mechanism.

Lemma 8 ([123])**.** *The optimal value function $J^*(\theta) : \Theta^* \to \mathbb{R}$ is a continuous, convex, and piecewise quadratic function.*

Theorem 31. *For the optimization problem in 7.2 with $\theta \in \mathbb{R}$, the union of all non-singleton constrained equivalence sets form an connected interval, and the optimizer function is a constant on this connected interval.*

Proof. The proof simply follows from the convexity of $J^*(\theta)$. □

Corollary 32. *Consider the same assumptions as in Lemma 7. For all $\theta \in \Theta^*$ such that $[\theta]_c$ is not singleton, $[\theta]_c$ is the same for all such θ.*

Proof. Equivalently, we can prove that any non-singleton $[\theta]_c$ is equal to the union of all non-singleton constrained equivalence sets, denoted by U. Now, we use the contradiction technique for the proof. Assume U can be partitioned by multiple constrained equivalence sets. We consider two neighboring ones and denote them by CR_i and CR_j, respectively. By Corollary 27, we know that for any $\theta_i \in \text{int}(CR_i)$ and $\theta_j \in \text{int}(CR_j)$, $x^*(\theta_i) \neq x^*(\theta_j)$. Using the similar technique to the proof of Theorem 25, we can also prove that $x^*(\theta_i) \neq -x^*(\theta_j)$. Therefore, $J^*(\theta_i) \neq J^*(\theta_j)$, which contradicts the fact that the optimizer function is a constant on CR_i and CR_j. □

Corollary 32 indicates that the non-singleton constrained equivalence sets of all parameters (if exist) can be characterized by a unique closed interval. Therefore, we can model the data reported by the simplified free-lunch privacy mechanism as follows.

Let θ_t and $\tilde{\theta}_t$ denote the original and reported data, respectively. Let $[a, b]$ denote the unique constrained equivalence set. The generating process of $\tilde{\theta}_t$ can be described as follows:

- if $\theta_t \in [a, b]$, then

$$P(\tilde{\theta}_t = \theta) = \begin{cases} \frac{1}{a-b}, \text{ for } \theta_t \in [a, b] \\ 0, \text{ for } \theta_t \notin [a, b] \end{cases} \tag{7.22}$$

- if $\theta_t \notin [a, b]$, then $\tilde{\theta}_t = \theta_t$.

Note that when free-lunch privacy exists, i.e., $\theta_t \in [a, b]$, the reported data $\tilde{\theta}_t$ satisfy a strong local differential privacy [47] ($\epsilon = 0$) as $\frac{P(\tilde{\theta}_t|\theta_t=\theta')}{P(\tilde{\theta}_t|\theta_t=\theta)} = 1 \leq e^{\epsilon}$, $\forall \theta, \theta' \in [a, b]$.

We would like to compare the estimation of μ by some adversary with and without using the free-lunch privacy mechanism. We will assume that the weak adversary uses the sample mean to estimate μ, since, in the absence of more information, it is difficult for the adversary to design a better estimator. We will assume the strong adversary uses the maximum likelihood estimator (MLE) to estimate μ, since he has enough information to easily design this estimator.

Let $\hat{\mu}_{adv,data}$ denote the adversary's estimate, where $adv = \{w, s\}$ stands for the adversary type (weak/strong) and $data = \{o, r\}$ is the data observed by the adversary (original data/reported data via the free-lunch privacy mechanism).

Without using the free-lunch privacy mechanism, i.e., both strong and week adversaries have access to the original user data, the estimator used by both weak and strong adversary is

$$\tilde{\mu}_{adv,o} = \frac{\sum_{t=1}^{T} \theta_t}{T}, \qquad adv \in \{w, s\} \tag{7.23}$$

The bias and variance of the estimator are given by

$$\text{bias}[\tilde{\mu}_{adv,o}] = 0 \tag{7.24}$$

$$\text{var}[\tilde{\mu}_{adv,o}] = \frac{\sigma^2}{T} \tag{7.25}$$

Now we discuss the case where the simplified free-lunch privacy mechanism is adopted to sanitize the data. We first consider the weak adversary. Since the weak adversary only has the access to the data measurements, there is no information that can help the adversary to improve the estimation. Therefore, we suppose the estimator used by the weak adversary to be

$$\tilde{\mu}_{w,r} = \frac{\sum_{t=1}^{T} \tilde{\theta}_t}{T} \tag{7.26}$$

Proposition 33. *The systematic error of estimator, which we still call bias, is*

$$\begin{aligned} &\text{bias}[\tilde{\mu}_{w,r}] = \mathbb{E}[\tilde{\mu}_{w,r}] - \mu \\ &= (\frac{a+b}{2} - \mu)P[\theta_t \in [a,b]] + \frac{\sigma}{\sqrt{2\pi}}(e^{-\frac{(b-\mu)^2}{2\sigma^2}} + e^{-\frac{(a-\mu)^2}{2\sigma^2}}) \end{aligned} \tag{7.27}$$

By Proposition 33, using the sanitized data of the free-lunch privacy mechanism the weak adversary's estimate of the mean of the user's data is always subject to some nonzero bias.

A strong adversary has access to the parameters of the optimization problem solved by the service provider and thereby it knows exactly how the reported data is generated. It can derive an MLE estimator based on its knowledge about the value of a and b.

Proposition 34. *Let $A = \{i : \tilde{\theta}_i \notin [a,b]\}$. The MLE estimator for a strong adversary is*

$$\tilde{\mu}_{s,r} = \frac{\sum_{t\in A} \tilde{\theta}_t}{|A|} \tag{7.28}$$

and it associated bias is given by

$$\begin{aligned} \text{bias}[\tilde{\mu}_{s,r}] &= \mathbb{E}[\tilde{\mu}_{s,r}] - \mu \\ &= \frac{\sigma}{\sqrt{2\pi}} \frac{e^{-\frac{(b-\mu)^2}{2\sigma^2}} - e^{-\frac{(a-\mu)^2}{2\sigma^2}}}{P(\theta_t \in (-\infty, a] \cup [b, +\infty))} \end{aligned} \tag{7.29}$$

Proof. The likelihood of the data samples $\{\tilde{\theta}_t\}_{t=1}^{T}$ can be expressed as

$$L = \frac{1}{(b-a)^{(T-|A|)}} \frac{\prod_{t\in A} \exp\left(-\frac{(\tilde{\theta}_t - \mu)^2}{2\sigma^2}\right)}{\left(\sqrt{2\pi\sigma^2} P[\in (-\infty, a] \cup [b, +\infty)]\right)^{|A|}} \tag{7.30}$$

Taking the derivative of $\log(L)$ leads to the MLE estimator (7.28). The expectation of the estimator is given by

$$\mathbb{E}[\tilde{\mu}_{s,r}] = \mathbb{E}[\tilde{\theta}_t | \tilde{\theta}_t \in A] \tag{7.31}$$

and the probability distribution of $\tilde{\theta}_t \in A$ is given by

$$P(\tilde{\theta}_t|\tilde{\theta}_t \in A) = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(\tilde{\theta}_t-\mu)^2}{2\sigma^2})}{P[\in (-\infty, a] \cup [b, +\infty)]} \tag{7.32}$$

The bias in (7.29) can be obtained from (7.31) and (7.32). □

It is worth noting that the strong adversary simply drops all the data points that enjoy free-lunch privacy since they are "uninformative" due to the randomization in the privacy mechanism. But still, the MLE estimator above is subject to a systematic error. The bias is zero only if $a = b$ or $\mu = \frac{a+b}{2}$.

In summary, the free-lunch privacy mechanism is able to provide protection against adversaries that are interested in inferring the parameter of the random process that generates the user's data. The adversary, both weak and strong, cannot easily construct a unbiased estimator based on the data manifested through the free-lunch privacy mechanism.

7.5 Case Study: Occupancy-based HVAC Control

In this section, we demonstrate the use of free-lunch privacy mechanism to minimize the release of private data in HVAC control. In smart buildings, the control of HVAC systems is adapted to room occupancy for the purpose of energy saving and thermal comfort. However, the occupancy data, especially in offices or households, contains rich information about space owners' habits and behaviors, and is therefore considered privacy-sensitive. We present a single-room temperature control example with MPC, and study how much occupancy information can be concealed without affecting the performance of HVAC control.

Simulation Setup

Here, we briefly summarize the MPC model used for simulation with reference to the notations in Table 7.1. For the detailed derivation, we refer the readers to [11, 80].

State dynamics. The continuous temperature dynamics of a zone can be derived from the law of conservation of energy,

$$M\frac{d}{dt}T = \dot{Q} + c_p\dot{m}_z(T_s - T) \tag{7.33}$$

which includes the zone temperature T, the thermal capacity of the zone M, the mass air flow to the zone $\dot{m}_z$, the supply air temperature T_s, and the heat capacity c_p. In this example, we consider regulating the room temperature by controlling the temperature of the supplied air. Q represents the thermal load, which can be calculated by applying a thermal coefficient c_{occ} to the occupancy θ, i.e., $Q = \theta c_{occ}$. Here, we consider the occupancy is binary, i.e., $\theta \in \{0, 1\}$, indicating presence/absence.

Table 7.1: Parameters used in the HVAC controller.

Param.	Meaning	Value & Units
Δt	Discretization step	$15min$
c_p	Thermal capacity of air	$1kJ/(kg \cdot K)$
M	Thermal capacity of the env.	$1000kJ/K$
c_{occ}	Thermal coefficient	$0.1kW$
$\dot{m}_z$	Supply air flow rate	$0.0382kg/s$
$T_{s,lb}$	Lower bound of supply air temperature	$5°C$
$T_{s,ub}$	Upper bound of supply air temperature	$40°C$
$T_{o,lb}$	Lower bound of comfort zone (occupied)	$21.5°C$
$T_{o,ub}$	Upper bound of comfort zone (occupied)	$27.5°C$
$T_{u,lb}$	Lower bound of comfort zone (unoccupied)	$18.5°C$
$T_{u,ub}$	Upper bound of comfort zone (unoccupied)	$36.5°C$

Using the trapezoidal discretization, we obtain the following linear temperature dynamic model

$$(\frac{M}{\Delta t} + \frac{c_p\dot{m}_z}{2})T(k+1) = (\frac{M}{\Delta t} - \frac{c_p\dot{m}_z}{2})T(k) + c_p\dot{m}_z T_s(k) + \frac{c_{occ}(\theta(k) + \theta(k+1))}{2} \tag{7.34}$$

Constraints. The system states and control inputs are subject to the following constraints:

C1: $T_{s,lb} \leq T_s(k) \leq T_{s,ub}$, representing the heating coil capacity;

C2: $T_{lb} \leq T(k) \leq T_{ub}$, delineating the comfort range. T_{lb} and T_{ub} take different values for different occupied status.

$$T_{lb} = \begin{cases} T_{o,lb}, \text{ if } \theta(k) = 1 \\ T_{u,lb}, \text{ if } \theta(k) = 0 \end{cases} \tag{7.35}$$

$$T_{ub} = \begin{cases} T_{o,ub}, \text{ if } \theta(k) = 1 \\ T_{u,ub}, \text{ if } \theta(k) = 0 \end{cases} \tag{7.36}$$

where $T_{o,lb}, T_{o,ub}$ represent the lower and upper bound of room temperature when a room is occupied. $T_{u,lb}, T_{u,ub}$ correspond to the comfort range when a room is unoccupied. It is typically preferred that $[T_{o,lb}, T_{o,ub}] \subset [T_{u,lb}, T_{u,ub}]$ in order to save energy.

C3: $T(1) = T_{measurement}$, i.e., the measurement from temperature sensor is used for updating the initial state.

Cost function. Our objective is to minimize the energy consumption of the HVAC system, which is quantified as the $l2$-norm of the difference between the supply air temperature T_s and the temperature of the outside environment T_o. In addition, we would like to regulate

the room temperature T around a desired temperature T_{des}. Let the MPC horizon be N, the cost function is given by

$$\min_{T\in\mathbb{R}^N, T_s\in\mathbb{R}^{N-1}} \|T_s - T_o\|_2^2 + \lambda\|T - T_{des}\|_2^2 \tag{7.37}$$

Furthermore, the occupancy is assumed to be constant in the optimization horizon. Note that the cost function is a quadratic function of the decision variables, and the constraints are affine in decision variables. The occupancy data (or parameter) appears as a linear term in the optimization constraints. Therefore, we can use the machineries developed in previous sections to analyze the free-lunch privacy in the occupancy-based HVAC control.

Simulation results

We apply the free-lunch privacy mechanism, which reports random bits or not report anything when different occupancy status results in the same control action while reporting truthfully otherwise. The MPC horizon is fixed to be one hour. The value of other parameters are given in Table 7.1. We assume temperature sensor measurements are given by adding a Gaussian random noise with standard deviation 0.1°C to the temperature model (7.34).

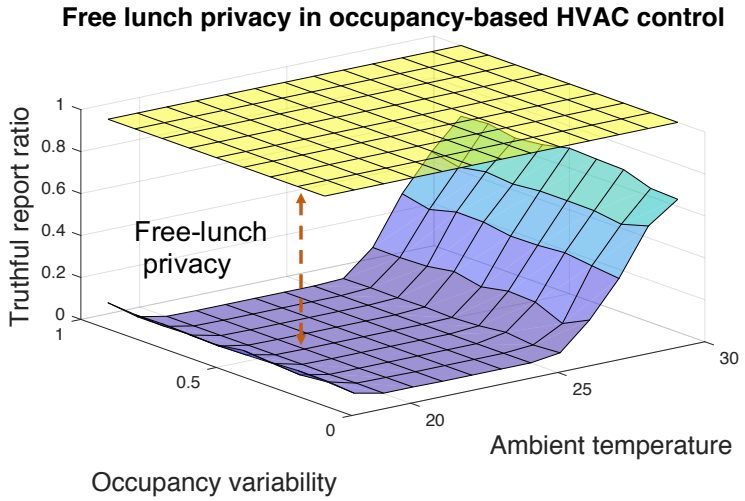


Figure 7.4: Simulation of the proportion of occupancy measurements that must be reported truthfully in a typical occupancy-based HVAC control application for different weather conditions and occupancy patterns under the free-lunch privacy mechanism (blue surface). As a comparison, the occupancy data reports without free-lunch privacy mechanism are also shown here (yellow plane).

We simulate the HVAC system behavior for two days for different outside weather conditions and occupancy patterns. The results are illustrated in Fig. 7.4. Different occupancy

patterns are simulated via the Markov chain with varying transition probabilities, i.e.,

$$P(\theta_{t+1}|\theta_t) = \begin{cases} q, \text{ if } \theta_{t+1} \neq \theta_t \\ 1-q, \text{ if } \theta_{t+1} = \theta_t \end{cases} \tag{7.38}$$

where q is referred to as *occupancy variability* in the figure. As shown in the figure, as outside ambient temperature becomes more extreme, the number of truthful occupancy report required increases. For instance, in the high-temperature scenario the change of occupancy status will dramatically change the control action. If there are no occupants, then the most energy-efficient way is to let the temperature drift and not react; nevertheless, if the occupant is present in the space, the HVAC system is obligated to drive the temperature to the comfort range specified by the building code. In contrast, if the outside temperature does not deviate much from the comfort range then the occupancy measurements are actually not necessary for executing the optimal control. The free-lunch privacy mechanism differentiates the critical measurement from the "trivial" measurement that does not contribute to the control, and conceal those measurements for better privacy. As a comparison, the yellow plane demonstrate the control without using free-lunch privacy mechanism. The gap between the yellow and blue surface signals the amount of measurements that can be concealed as a free lunch. It is also interesting to notice that there is quite a bit of free-lunch privacy that can be exploited in HVAC control applications due to slow dynamics of building environment.

Real-World Experiments

We further conduct experiments demonstrating the use of the free-lunch privacy mechanism in HVAC control with a real conference room. We utilize a simplified physical setup, characterizing the area into two zones (the conference room and the surrounding area). We construct temperature and occupancy models to implement the MPC controller and present two experiments validating the use of the free-lunch privacy mechanism described.

Temperature and Occupancy Dynamics. We build a temperature dynamics model for the conference room using 110 hours of data collection, monitoring the response of the room temperature to changes in setpoint S, current temperature, neighboring zone temperature, and the occupancy O (which presents a thermal load). This data is used to construct a linear model with the aforementioned predictors:

$$T^z_{k+1} = f(T^1_k, \ldots, T^Z_k, S^z_k, O^z_k) \tag{7.39}$$

Using the occupancy data from this data collection period we construct a K-Nearest Neighbors regressor to model the occupancy dynamics of the room.

MPC Controller. The objective function to be minimized for the MPC is the L2-norm of the difference between the supply air temperature T_s and the outside air temperature T_o,

serving as a proxy for the energy used by the HVAC for a given setpoint:

$$\min \|T_s - T_o\|^2 \tag{7.40}$$

where the supply air temperature is found to be a function of the current temperature and the setpoint for this particular building HVAC. In an occupied room, the comfort range is defined to be $23.9°C - 25.6°C$ and in an unoccupied room, $22.8°C - 26.7°C$. The minimum and maximum setpoints are $0°C$ and $37.8°C$, respectively. Paired with the temperature and occupancy dynamics models, the MPC minimizes the above objective function with aforementioned constraints on room temperature and control setpoint for a 15-minute prediction horizon. The optimal control action found for the first minute is applied and this optimization problem is solved every 10 minutes for the duration of the experiments.

Free-Lunch Mechanism. The occupancy is masked during the second day of experimentation by using an exhaustive search method to calculate the equivalence set of the true occupancy. This is possible due to the practical occupancy of the conference room being constrained. First, the control action is computed using the true occupancy. Then, the same is done for occupancies in the full range of the maximum observed occupancy of the conference room. If a non-singleton equivalence set (in which the control action mirrored that of the true occupancy) is found, then a false occupancy is randomly chosen from the set and reported to the MPC.

Results. The first day of experimentation uses the true occupancy for comparison, and the second day uses the free-lunch mechanism. The experiments are begun at 9am on consecutive weekdays and lasted for 6 hours.

The results are illustrated in Figure 7.5. As evidenced by the room temperature changes over the experimentation periods, the controller using the masked occupancy from the free-lunch privacy mechanism performs just as well as that using the true occupancy in maintaining the room temperature within the comfort bounds described earlier.

7.6 Chapter Summary

In this chapter, we present a framework to analyze free-lunch privacy in data-informed operation of CPS. Free-lunch privacy is the data ambiguity that can be allowed without affecting the optimal decision making. We present algorithms to efficiently compute free-lunch privacy given the optimization problem underlying the decision making process, as well as sufficient conditions for a quick check of existence of free-lunch privacy. We also show that the attackers are not able to recover the true statistics of a user's data under the free-lunch privacy mechanism. We demonstrate the use of the framework established in this chapter to analyze the amount of free-lunch privacy in a smart building control application. It is shown through simulations that when the outside temperature is in a moderate range much of the

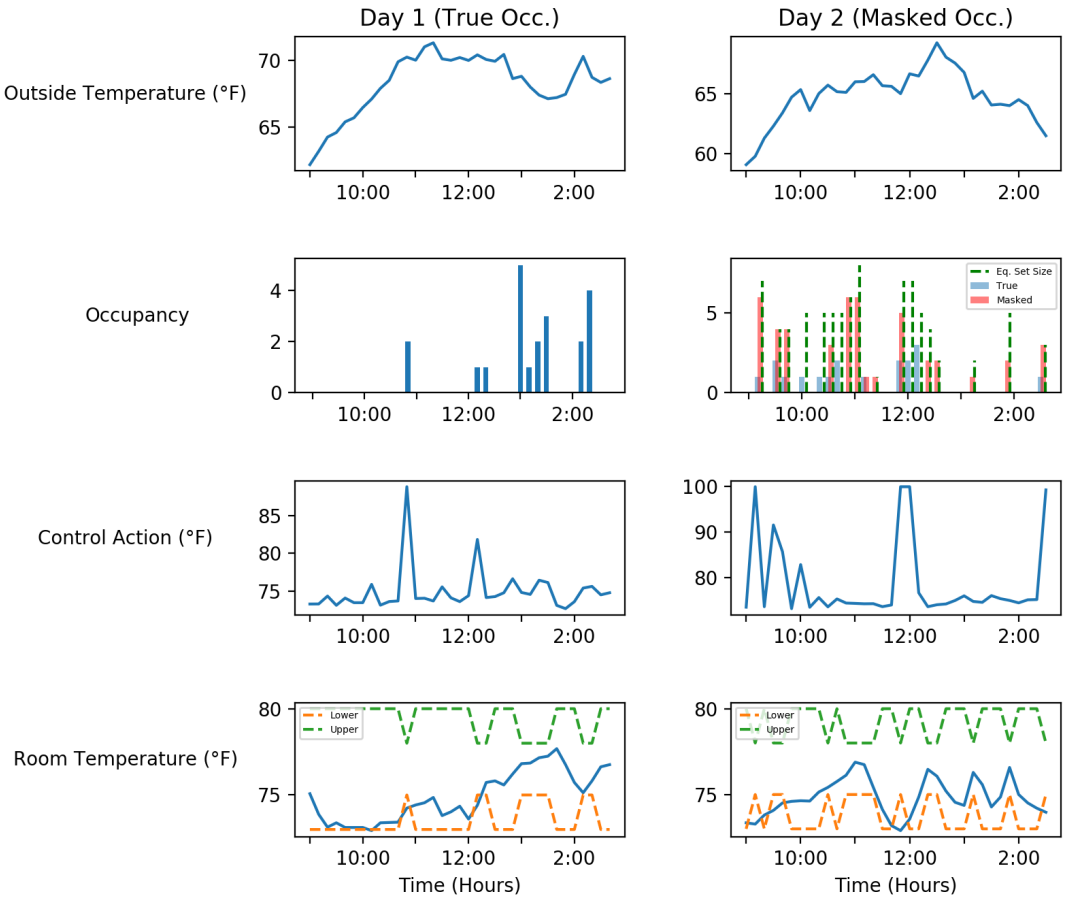


Figure 7.5: We conducted a two-day experiment of using MPC to control a real-world conference room. The MPC uses true occupancy in the first day and uses the masked occupancy generated by the free-lunch mechanism in the second day. The outside temperature, occupancy traces, control actions and room temperature during the two-day experiment are demonstrated.

occupancy measurements are redundant and therefore can be concealed to protect the user's privacy. We also demonstrate the usefulness of the free-lunch privacy mechanism through real-world experiments.

For future work, we plan to extend the analysis on quadratic programming presented in this chapter to other non-linear optimization problems. It is also interesting to apply the framework to some other CPS applications such as taxi dispatch and integrated energy planning.

The majority of work in privacy thus far has focused on how to allow consumers to "hide

in the crowd" for large databases. In this chapter, we consider ways in which a single user can modify their reported data without affecting their experienced quality of service *at all*. The free-lunch privacy framework ensures that the system will behave exactly as it would in the absence of any privacy-preserving mechanism, while still improving the privacy of users by reducing the amount of truthfully transmitted data. In this precise sense, our users can have their cake and eat it, too.

Chapter 8

Privacy-Preserving Data Publishing with Enhanced Utility

8.1 Background

A seamless integration of computation, networking and the physical world is being featured in a multitude of engineering systems such as civil infrastructure, energy grid, transportation and health care among others. In these systems, embedded computers and networks are used to monitor and control physical processes with feedback loops where these processes affect computation and vice versa. In light of the tight coupling between cyber and physical processes, these systems are commonly termed cyber-physical systems (CPSs). CPSs have enabled various applications where decisions are driven by the sensory information. For instance, the deployment of large-scale sensing and actuation networks in buildings has driven the evolution to "smart" buildings that can collect fine-grained information about indoor environments, energy usage and occupants. This information is further leveraged to control lighting, heating, ventilation, and air conditioning (HVAC), and other building equipment in an energy-efficient and occupant-responsive manner. Smart buildings, as a salient example of CPSs, will be considered throughout this chapter.

Due to the distributed nature and fast increase of system complexities, the operation of CPSs involves sensing, processing, and storage of massive amounts of data. Driven by benefits mutual to the stakeholders, there is a continually rising demand for publishing datasets collected in CPSs. In particular, publishing datasets collected in smart buildings is beneficial to occupants, building managers and research communities. Large-scale and high-quality datasets are often enablers of robust and sophisticated models. Promoting research on advanced data analytics will eventually give rise to building operations that provide more cost savings for building managers and better adapt to occupants' needs. Occupancy modeling and energy profiling are two good examples of building applications with a significant reliance on data-driven analytics. Occupancy modeling derives occupancy schedules from data and further enables on-demand control over lighting and HVAC systems [76, 138]. Energy profiling

refers to the characterization of occupants' energy use, which can help gain insights into buildings' operational conditions [67].

However, data published in the original form can come with the risk of privacy breach, especially when the CPSs involve humans in the loop. Pristine database may reveal detailed information about occupants' behaviors. Previous studies [41, 88] have shown that occupants' schedules and activities can be easily retrieved from occupancy and energy datasets. Tech-savvy criminals are already exploiting unintentional occupancy leaks to select victims for burglaries [18]. In addition, electricity data also indirectly reveal private information that is of interest to insurance companies, marketers, potential employers or the government for setting premium rates, directing ads, vetting an applicant's background or monitoring its citizens [111]. In light of the risks of privacy violation, European Commission has proposed a comprehensive reform of data protection rules in the European Union (EU) to protect personal data from misuse, and the regulation will apply from May 25, 2018 [131].

Current practice in publishing CPSs' datasets mainly relies on policy and agreements to regulate data use, sharing, and retention [172]. However, this prescriptive approach does not prevent privacy breaches from happening. Before publication, privacy-sensitive datasets are often anonymized by suppressing direct identifiers such as the identity of record owners. However, datasets resulting from applying simple suppression operations are vulnerable to adversaries with auxiliary knowledge. Given that an adversary possesses some prior knowledge of a person's data, the record of this person can be easily re-identified from the anonymized database by matching the records with the auxiliary information. This prior knowledge can often be easily obtained via external observations or interactions with the target.

K-anonymity [152] is a stronger notion for "being anonymous" than just suppressing direct identifiers. It can mitigate the risks of re-identification by allowing data owners to "hide in the crowd." To be specific, k-anonymity ensures that each record in a database is indistinguishable from at least $k-1$ other records in the database. Since k-anonymity is conceptually simple and can be easily implemented, it has been extensively used in publishing various datasets including location data collected from mobile devices [64]. Some states in the U.S., such as California, Colorado and Illinois, have enacted a privacy standard, often referred to as "15/15" rule, for utility companies in order to help ensure customer anonymity when energy data is released to third parties without customer consent [2]. The privacy standard is based on the k-anonymity concept, requiring that aggregated data include a minimum of 15 customers with no one customer's load exceeding 15 percent of the group's energy consumption.

The main challenge in applying k-anonymity to data publishing is the information loss introduced inevitably by the anonymization process, which is also remarked in the Article 29, "Opinion 05/2014 on Anonymization Techniques" [167], composed by representatives from all EU Data Protection Authorities, the European Data Protection Supervisor and the European Commission:

> *It is clear from case studies and research publications that the creation of a truly anonymous dataset from a rich set of personal data, whilst retaining as much of*

the underlying information as required for the task, is not a simple proposition.

The challenge becomes even acuter for publishing CPSs' data, as decision making and control in CPSs are highly sensitive to data quality. In the aforementioned occupancy modeling example, operating lighting and HVAC according to inaccurate occupancy schedules would affect the comfort and well-being of occupants. For the energy profiling example, without a truthful profiling grid operators can hardly preempt disturbances and ensure a stable and resilient energy supply.

In this chapter, we presented PAD [80] - an open-sourced system to publish data collected from CPSs with k-anonymity and enhanced utility. The underlying idea of PAD for improving data utility is to customize the data sanitization process to the subsequent usage of the data. To illustrate the idea, we can consider two researchers who are interested in performing different analysis on the same dataset. Suppose that one is interested in the occupancy patterns during lunch time while the other is interested in people's arrival time. It is evident that if we want to publish a dataset that is more useful for the first researcher, the occupancy records with similar patterns during lunch time should form a size-k group so that replacing the original record with any of the records in this group would not cause severe information loss for the lunchtime occupancy patterns. In contrast, to publish a sanitized dataset that is more valuable for the second researcher, the occupancy records with similar arrival time should be grouped in order to retain more information regarding arrival time.

Although customizing k-anonymization to the interest of data users is promising to increase data utility, due to the diversity of potential data uses it will be cumbersome to enumerate and hard-code every possible data use and design the corresponding anonymization process. In PAD, we proposed a unified protocol to comprehend users' diverse interests by learning from their interactions with the data publishing system. More specifically, PAD will first provide data users with some data that does not involve privacy risks such as publicly available datasets, and the data users will label the similarity of these data points according to the features of particular interest to them. PAD will then learn these features from the similarity labels provided by the data users and optimize the anonymization processes accordingly.

The contributions of the chapter are as follows.

- We design and implement an open-sourced system to publish building-related datasets with k-anonymity guarantees;
- We employ metric learning techniques to learn the intended data use from interactions with the data analyst and then use the learning result to reduce information loss incurred by data sanitization;
- We conduct extensive experiments using real-world building data on occupancy presence and plug-load energy consumption to demonstrate the usefulness of sanitized datasets.

8.2 K-Anonymity

In this section, we will discuss the privacy value of k-anonymity and attacker models, followed by a brief introduction of basic techniques for achieving k-anonymity. We will close the section by discussing the intrinsic tradeoff between privacy and data utility and some limitation of basic techniques to motivate the design of the proposed system.

Privacy Value

The concept of k-anonymity [152] was originally introduced in the context of relational data privacy. The idea behind k-anonymity can be described as "hiding in the crowd", as it requires that each individual cannot be identified within a set of k individuals in the released data. In this chapter, we deal with a slightly more general definition of k-anonymity, i.e., we consider a *row* in a database as *k-anonymous* if and only if it is indistinguishable from at least $k-1$ other rows. Depending on the contents of a row, this definition can incorporate the privacy guarantee at different levels. For instance, if each row is a daily energy or occupancy profile of a person, then this definition ensures that the profile of each day cannot be differentiated from $k-1$ other profiles. If we consider that each row in the database contains information of a person, then we recover user-level privacy which guarantees the indistinguishability of k persons and therefore offers a stronger privacy notion.

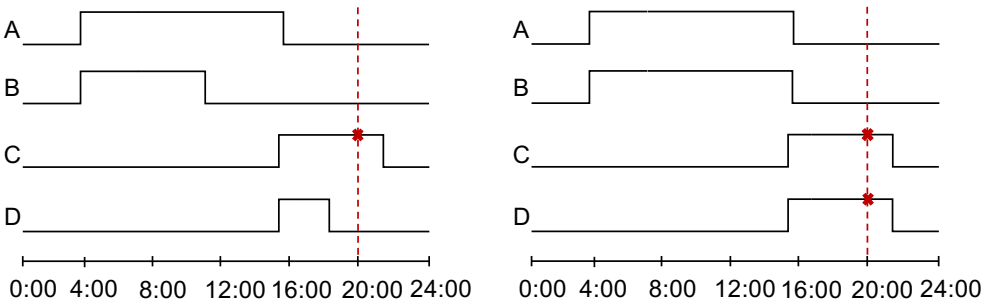
We illustrate the privacy value of the k-anonymity model by comparing it with the strategy that only masks the identifier of each row in a database. Assuming a data analyst requests data publishing and the database is sanitized solely by suppressing names of the data owners, we want to show that the information retained in this database can still create a threat against data privacy when combined with external observations or knowledge.

As an example, consider the scenario depicted in Figure 8.1 where the database contains four rows corresponding to the office occupancy status of four persons labeled as A, B, C, and D. If no k-anonymization is performed by the data curator, then the following linkage attack can be conducted: Suppose the adversary knows that C stays in this office at 20:00, then by linking this information with the data trajectories it has at hand it can find the complete occupancy status of C in the time horizon of the published data. However, such linkage attack is not effective if proper data perturbation is performed by the data curator to maintain k-anonymity. Consider the 2-anonymized version of the original dataset illustrated by Figure

In this chapter, we wish to achieve data protection against the adversaries with the following capabilities: (1) Having access to the published data; (2) Knowing short snippets of truthful private data by external observations.

Microaggregation

Microaggregation is a popular perturbation technique to achieve k-anonymity for databases with quantitative records. It processes the data in the following two steps prior to publication:



(a) Original data (b) 2-anonymous data

Figure 8.1: Linkage attack.

- **Step 1 (k-partition)**: All rows in the database are partitioned into small aggregates of k or more rows.
- **Step 2 (substitution)**: Each row is replaced with the centroid of the group it belongs to.

Following this procedure ensures that every record in the released database corresponds to at least k individual records; hence, k-anonymity is guaranteed.

Due to the data distortion introduced in the substitution step, the main problem in microaggregation is to retain as much information as possible while offering sufficient privacy protection. In order to minimize the information loss caused by microaggregation, groups should be formed by maximizing their within-group homogeneity. The more homogeneous the records in a group are, the lower information loss is incurred when replacing the true value of a record by the group average. The sum of squared distances (SSD) criterion is a common measure to estimate group heterogeneity and this is defined as

$$SSD = \sum_{i=1}^{g} \sum_{j=1}^{n_i} d(x_{ij}, \bar{x}_i) \tag{8.1}$$

where x_{ij} denotes the j-th row of i-th group, $\bar{x}_i$ represents the centroid of the group i, n_i is the number of elements in i-th group and g stands for the number of groups.

The distance metric $d(\cdot, \cdot)$ in equation (8.1) is often chosen to be an uninformed norm, such as Euclidean distance. Although Euclidean distance is simple and intuitive, it ignores the fact that the semantic meaning of "information loss" is inherently task- and data-dependent [166]. To illustrate this point, imagine two researchers who want to analyze the same occupancy dataset. The first one is interested in the occupancy patterns during electricity peak demand hours in order to estimate the demand response potential, whereas the second one is interested in the aggregate occupancy over the day for energy modeling purposes. Given the nature of their respective tasks, both should use very different distance metrics to measure the

information loss. If the purpose of the data is known at the time of publication, it can be taken into account during microaggregation to better retain information. But clearly, building a system to parse data users' interest is not the most robust and scalable approach due to the diversity of different data analysts' interest. It is, therefore, more desirable to have a standard protocol for different users to express their respective data purposes. Our approach implemented in PAD is to learn the distance metric explicitly for each specific application from data points' similarity labeled by the user.

8.3 Overview of PAD

We assume that the *data publisher* collects data records and releases the collected data to the *data recipient*, who will then conduct data mining on the published data. We will use "data recipient", "*data analyst*" and "*data user*" interchangeably in this chapter. Further, we assume that the data publisher is trustworthy yet the data recipients are not. This assumption is also referred to as the *trusted* model [59]. Since in our framework data analysts can interact with the data publication system to improve the usefulness of the published data, it is important to ensure that data analysts do not have access to the original database during any part of the data publication process.

Figure 8.2 illustrates the design of PAD. The objective of the system is to publish the dataset with k-anonymity guarantee as well as high quality in support of the required data analysis. The core idea of the system is to improve the data fidelity by learning how the data is intended to be used and then adjusting the data perturbation algorithm accordingly.

If the data is not used for specialized purposes, then PAD directly applies microaggregation and publishes the database. Otherwise, PAD processes the original database in the following four steps.

(1) Interaction preparation. The objective of this step is to provide a dataset for the data analyst to label data points' similarity, which will be later used to learn the purpose of data analysis. The dataset can either come from the original database or a dataset that is already public. Since this dataset should not cause privacy concerns, it must be pre-sanitized if the original database is used for interacting with the data analyst. In this step, the system has not received any inputs from the data analyst yet. Pre-sanitization is therefore performed via microaggregation with a generic distance metric, e.g., Euclidean distance.

(2) Subsampling. In the second step, PAD processes the rows in the prepared database into pairs and randomly selects some pairs to be returned to the data analyst, who will then assign a binary label to each pair of rows. The binary label indicates whether the two rows are similar or not in accordance with the particular data purpose. Consider, for example, the two pairs of occupancy records depicted in Figure 8.3. If the data analyst wants the published dataset to maximally retain the information regarding the occupancy patterns during lunch time, then he will assign "dissimilar" to the first pair and "similar" to the second one; however, if the data analyst is interested in the occupancy patterns during the entire day, then the first pair will be labeled as "similar" and the second one as "dissimilar". In

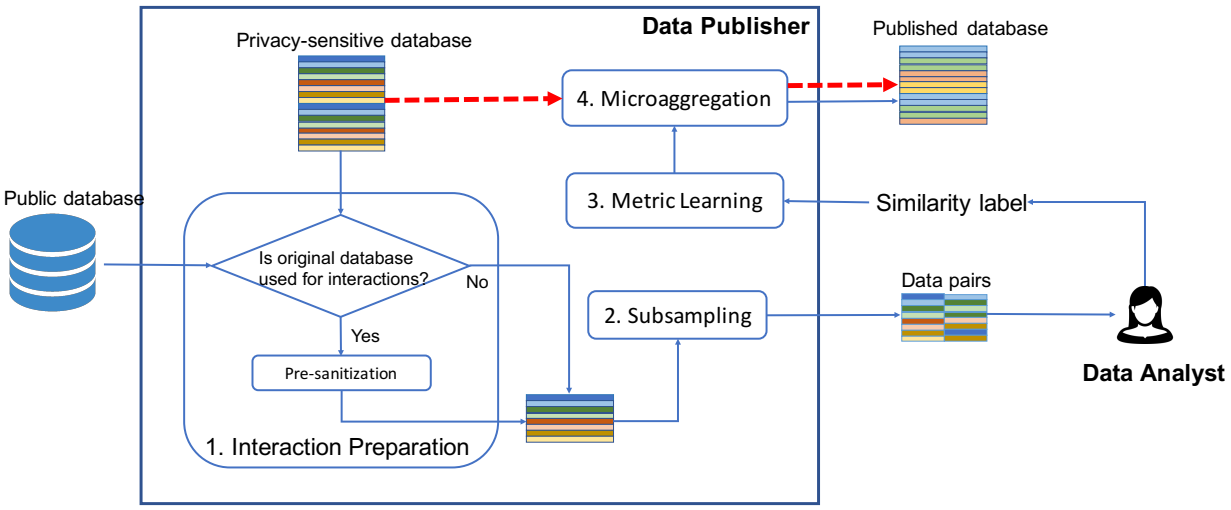


Figure 8.2: PAD diagram: If the purpose of the dataset to be published is not known prior to publication, then PAD directly applies microaggregation with an uninformed distance metric to sanitize the dataset (shown in red dashed arrow). Otherwise, PAD processes the data in the following steps: (1) Prepare the training data used for learning potential data uses. The training data can either come from original data base or a similar dataset that is already public. Pre-sanitize the data if the original database is used. (2) The data pairs are subsampled from the prepared training data and returned to the data analyst to solicit their labels on which data pairs are considered similar (The labels can be assigned manually or automatically using custom programs); (3) PAD learns a metric from the similarity labels; (4) The learned metric is used by microaggregation to generate the sanitized dataset for final publication.

the case where the desired metric for comparing similarity can be explicitly defined, labeling effort can be greatly alleviated by using computer programs to automatically label similarity of data points based on the desired metric.

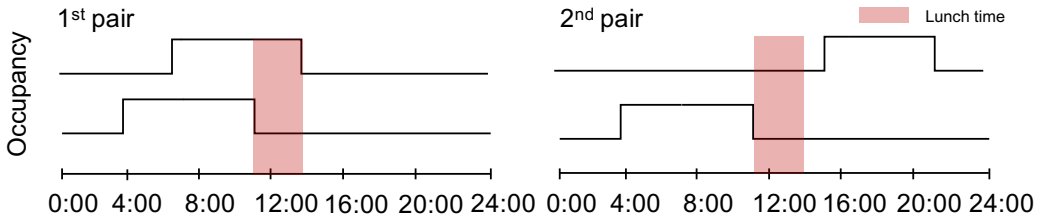


Figure 8.3: Illustration of determining similarity labels.

(3) Metric learning. In this step, a distance metric over the data record is automatically learned from data pairs and the corresponding similarity relationships specified by the data

analyst.

(4) Microaggregation. This step uses the distance metric learned from the previous step for microaggregation so that the database can be sanitized in a way that the information of interest to the data analyst is maximally retained.

The detailed algorithms for (3) metric learning and (4) microaggregation will be presented in Section 8.4 and Section 8.5, respectively. Before closing the section, we want to point out that the existence, amount and quality of similarity labels provided by the data analyst affect the usefulness of the published data; however, the privacy level remains the same regardless because the dataset is always microaggregated before publication.

8.4 Distance Metric Learning

We will firstly present the linear distance metric learning method. Then, we will introduce a more flexible metric learning method based on deep neural networks, which can learn both linear and nonlinear distances.

Linear Metric learning

Let the original, pre-sanitized, and finally published dataset be denoted by X, $\hat{X}$, and $\tilde{X}$, respectively. In the metric learning step, the data analyst is provided with some data pairs $(\hat{x}_k, \hat{x}_j)$ $(k, j = 1, \cdots, |\hat{X}|)$ from the pre-sanitized (or public) database, and assigns a similarity label to each of the data pairs. Our objective is to learn a distance metric $d(x, y)$ between points x and y so that "similar" points end up close to each other.

The idea underlying our metric learning is to parameterize the distance metric and find the parameters that best explain the similarity relationships labeled by the data analyst. To be specific, we consider the distance function of the following form

$$d(x, y) = d_A(x, y) = \sqrt{(x - y)^T A(x - y)} \tag{8.2}$$

where A is a semi-definite matrix to ensure $d(x, y)$ to be a well-defined metric that satisfies non-negativity and the triangle inequality. This distance metric, also termed Mahalanobis distance, is a generalization of Euclidean distance by admitting linear scalings and rotations of the original data space. A is often termed as inverse covariance (IC) matrix. Setting A to be the identity matrix I gives the Euclidean distance; Restricting A to be diagonal corresponds to learning a metric where the different axes are weighted differently. Note that $d_A(x, y) = \sqrt{(x - y)^T A(x - y)} = \|A^{\frac{1}{2}}x - A^{\frac{1}{2}}y\|_2$, and therefore learning a full matrix A is equivalent to finding a scaling and rotation of data that replaces each point x with $A^{\frac{1}{2}}x$ and applying the Euclidean distance to the tranformed data.

Suppose each row record has length m, i.e., $x \in R^m$, and the number of parameters to be estimated in total is m^2. Building-related datasets are often in the form of time series, so m is large. However, we would like to require as low labeling efforts as possible to facilitate the use of PAD. Consequently, the main technical challenge is to learn a distance metric in the

"high-dimensional" regime where the number of parameters to be determined is larger than the number of labeled samples.

Various distance metric learning techniques [169, 166] have been proposed in the literature, the core idea behind which is to form an optimization objective that minimizes the distance between the data pairs labeled as "similar" and pushes the "dissimilar" pairs far away. As for metric learning in the high-dimensional regime, a typical technique used is to pose some restrictions or prior knowledge on the distance metric model to regularize the model complexity. Consequently, only a smaller number of examples are required to learn a well-posed metric [132, 38].

Our approach adopts a similar idea and restricts the complexity of distance metric by imposing l_1 penalty. We propose the following l_1-regularized optimization to find the Mahalanobis distance from the data pairs with similarity labels:

$$\underset{A}{\text{minimize}} \quad \sum_{(\hat{x}_k,\hat{x}_j)\in\mathcal{S}} d_A^2(\hat{x}_k,\hat{x}_j) + \lambda\|A\|_1 \tag{8.3}$$

$$\text{subject to} \quad \sum_{(\hat{x}_k,\hat{x}_j)\in\mathcal{D}} d_A(\hat{x}_k,\hat{x}_j) \geq c \tag{8.4}$$

$$A \succeq 0 \tag{8.5}$$

where $\mathcal{S}$ and $\mathcal{D}$ are the sets of data pairs that are labeled as "similar" and "dissimilar" respectively. The above optimization demands similar points to have small squared distances between them while dissimilar points be separated by a margin c. The choice of the constant c is arbitrary but not important, and changing it to any other positive constant b results only in A being replaced by $(b/c)^2A$. Herein, we set $c = 1$ for simplicity. The l_1 norm penalty ensures the solution to be sparse and capable of being generalized to unseen data pairs.

Deep Metric Learning

One challenge with the Mahalanobis distance metric is that it performs well only when the feature is linear in the original data record, because learning the Mahalanobis distance metric from labeled data pairs is equivalent to seeking a linear transformation $A^{\frac{1}{2}}$. This implies that our previous approach cannot adequately capture the nonlinearities presented in a number of scenarios such as arrival and departure time analysis of an occupancy dataset.

To overcome this limitation, several approaches have been proposed to learn a nonlinear distance function, including the use of kernels [156, 171] and deep neural networks [74, 69]. Kernel methods map each data instance to a high-dimensional feature space and then learn a distance metric in the high-dimensional space. The challenge with kernel methods is that they require a user-specified kernel function. Conversely, deep metric learning can learn a nonlinear representation of data and enjoys more flexibility. To the best of our knowledge, no previous approach have applied DNNs for improving data utility with regard to k-anonymization.

DNNs pass the dataset through several layers of nonlinear transformations achieved by compositing linear transformations and nonlinear activation functions, as illustrated in

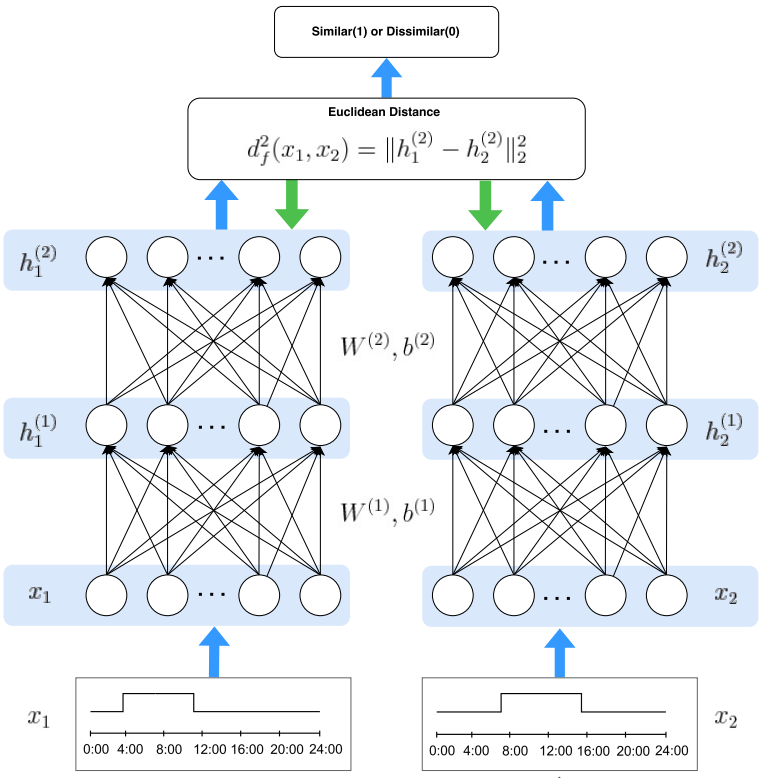


Figure 8.4: Deep Metric Learning with a two-layer neural network: A pair of data samples x_1 and x_2 are transformed to $h_1^{(2)}$ and $h_2^{(2)}$ through the same hierarchical non-linear transformation specified by the neural network. The Euclidean distance between $h_1^{(2)}$ and $h_2^{(2)}$ are computed to determine if x_1 and x_2 are similar.

Figure 8.4. The neural network we used comprises two identical branches, representing the nonlinear feature function applied to both points in a data pair. Let us first consider a single branch. Suppose that there are N layers in a deep network and that for each layer such as the nth layer, there are $k^{(n)}$ activation units, where $n = 1, 2, 3, \ldots, N$. The first layer takes one of the points in a data pair $x \in \mathbb{R}^d$ as input and outputs $h^{(1)} = g(W^{(1)}x + b^{(1)}) \in \mathbb{R}^{p^{(1)}}$. $W^{(1)} \in \mathbb{R}^{p^{(1)} \times d}$ is a projection matrix, $b^{(1)} \in \mathbb{R}^{p^{(1)}}$ is a bias vector and $g : \mathbb{R} \mapsto \mathbb{R}$ is the non-linear activation function. Examples of commonly used activation functions include `sigmoid`, `tanh` and `rectified` functions. The output $h^{(1)}$ from the first layer becomes the input for the second layer, and the output of the second layer is given by $h^{(2)} = g(W^{(2)}h^{(1)} + b^{(2)}) \in \mathbb{R}^{p^{(2)}}$, where $W^{(2)} \in \mathbb{R}^{p^{(2)} \times p^{(1)}}$, $b^{(2)} \in \mathbb{R}^{p^{(2)}}$. We can compute the outputs of other layers in a similar fashion and the output of the topmost layer, i.e., the Nth layer, is given as follows:

$$f(x) = h^{(N)} = g(W^{(N)}h^{(N-1)} + b^{(N)}) \in \mathbb{R}^{p^{(N)}} \tag{8.6}$$

where the $f : \mathbb{R}^d \mapsto \mathbb{R}^{p^{(N)}}$ is non-linear function determined by $W^{(n)}$ and $b^{(n)}$, $n =$

$1, 2, 3, \ldots, N$, as well as the nonlinear active function. Hence, we compute the distance between any pair of data samples x_i and x_j by firstly performing the transformation $f(x_i) = h_i^{(N)}$ and $f(x_j) = h_j^{(N)}$ and then calculating the Euclidean distance between $f(x_i)$ and $f(x_j)$:

$$d_f^2(x_i, x_j) = \|f(x_i) - f(x_j)\|_2^2 \tag{8.7}$$

The objective of our deep network is to find a non-linear mapping f such that for similar pairs $\mathcal{S}$ with the label $Y = 0$, $d_f^2(x_i, x_j)$ is smaller than that for dissimilar pairs $\mathcal{D}$ with the label $Y = 1$. Given (8.7), [69] proposed a contrastive loss function that learns the parameters of f such that data pairs in $\mathcal{S}$ are pulled closer and those in $\mathcal{D}$ are pushed apart. This contrastive loss function is defined as:

$$L(f, Y, x_i, x_j) = (1 - Y)\frac{1}{2}\left(d_f^2(x_i, x_j)\right) + (Y)\frac{1}{2}\left(max\{0, m - d_f^2(x_i, x_j)\}\right) \tag{8.8}$$

where $m > 0$ is a margin that separates $\mathcal{S}$ and $\mathcal{D}$. $\mathcal{D}$ only contributes to the loss function if their distance is within the margin [69].

It is worth noting that this network can also be adapted to learning linear distance metrics by simply replacing the activation functions in each hidden layer with identity functions.

8.5 Efficient Algorithm for Microaggregation

As discussed previously, microaggregation includes two steps, namely, k-partition that clusters the data into group sizes of at least k records and a substitution step that perturbs the data by replacing the true values by the group centroid. It is possible that the data type of group centroid is not consistent with the original data. For instance, the centroid of multiple occupancy time series is not necessary to be in an integer form. In such cases, proper post-processing, like rounding, should be conducted to make the published database meaningful.

The information loss in the published dataset is mainly determined by the k-partition step. An optimal k-partition is defined to be the one that minimizes the heterogeneity of group members characterized by equation (8.1). Note that k-partition is different from the classical clustering problem where the goal is to split the dataset into a fixed number of groups irrespective of the group size. In the case of k-partition, the constraints are on the group size instead of the number of groups. Nevertheless, we can modify the classical agglomerative clustering to make it serve for the k-partition purposes by terminating the agglomeration process at the proper level where the size of each group formed satisfies the constraints desired by the optimal k-partition.

The following proposition states the properties of the sizes of groups formed by optimal k-partition.

Proposition. *An optimal solution to the k-partition problem of a set of data exists such that its groups have size greater than or equal to k and less than 2k.*

The proof can be found in [44]. Proposition 1 indicates that the search space of the optimal k-partition can be reduced to the partition where all groups have size between k and $2k$. Therefore, we modify a widely used agglomerative clustering algorithm, Ward's method [42], to provide a heuristic and efficient solution that fulfills the group size requirements. The detailed algorithm is presented in Algorithm 8.

Algorithm 8 k-ward algorithm

Input: Database $X_i, i = 1, \cdots, n$

Group initialization
Define the extreme data points as the two which are most distant
For each of the extreme data points, take $k-1$ data closest to it and form the first two groups
The rest of data points in the dataset constitute single-element groups
Agglomerative clustering via Ward's method
while there exists some group of the size less than k **do**
    Find the nearest pair of distinct groups, at least one of which must have size less than k
    Merge the two groups and decrement the number of groups by one
end while
if there exists some group containing $2k$ or more data **then**
    Apply k-ward algorithm recursively on those groups
end if

8.6 Evaluation

We evaluate the performance of PAD using various datasets collected in real-world buildings. The questions we would like to answer from the experiments are:

- How useful are the sanitized datasets for typical data mining tasks?
- If the use purpose of a dataset is predetermined, can a dataset sanitized with the learned metric retain more relevant information than the one sanitized with an uninformed metric?

To answer these questions, we differentiate between three (3) evaluation cases, namely:

1. The utility of PAD with a generic distance metric
2. The utility of PAD with a customized distance metric when the feature of interest to the data analyst is linear in the original data record

3. The utility of PAD with a customized distance metric when the feature of interest the data analyst is nonlinear in the original data record

Experimental Setup

Datasets. Our datasets include occupancy and plug load power consumption, which represent typical building data types that may arouse occupants' privacy concerns. Two different occupancy datasets are employed in this study. One occupancy dataset, lasting about half a year, was collected at a resolution of 1 minute in four classrooms of the OU44 building at the University of Southern Denmark. In the following, this dataset will be referred to as *OU44 occupancy dataset*. Another occupancy dataset, which we call *smart home occupancy dataset*, was collected from thermostat motion sensors in 49 users' houses. Each user's data has a resolution of 5 minuates and lasts one day. Both datasets contain binary occupancy time series, indicating whether or not the room is occupied. Occupancy data can potentially reveal privacy-sensitive information such as daily routines and detailed schedules of the inhabitants. The *plug load dataset* consists of 15-minute-resolution power consumption data over three months. This dataset was collected at the individual desks of five occupants in Cory Hall on UC Berkeley campus. Plug load data also raises privacy concerns. As shown in the previous studies [83, 118], occupants' presence or even more detailed activities can be easily identified from power data. Since OU 44 occupancy dataset and plug load dataset contain a relatively small population of individuals, we will consider anonymity protection at the daily profile level instead of the user level. That is, k-anonymity ensures that k day profiles, rather than k users, are indistinguishable. In this regard, we process these two datasets into the form where each row corresponds to a person's daily occupancy or energy profile. We would like to stress that the framework can also protect the anonymity at the user level by feeding a dataset where each row corresponds to the data of a different user, such as the smart home occupancy dataset.

Implementation. The deep metric learning algorithm was implemented using *Keras* [28] and *Tensorflow* [110]. We used `rectified` as the activation function for the hidden layers. The `Adam` algorithm was adopted for learning the weights of the network. The implementation code of the framework is open-sourced at `https://github.com/PAD-Protecting-Anonymity/PAD`.

Evaluation Procedure. We evaluate PAD in two training scenarios. One is where public datasets are available to train a distance metric. In that case, we divide the data into two parts. The first part is assumed to be the privacy-sensitive database that will be sanitized by PAD. The second part plays the role of a publicly available dataset and is used for training distance metric functions. Another training case considered in our experiments is where publicly available datasets are difficult to find and thus the pre-sanitized version of the original database is used for training the distance metric. We demonstrate the results of both

cases. We further split training data into two portions: one for fitting the distance function and another for testing the fitted function. We do not implement hyper-parameter tuning due to the lack of training data. In addition, to examine the performance variation of the learned metrics caused by changes in the training dataset, we conduct five Monte Carlo (MC) simulations and in each MC simulation 80% of the training samples are randomly drawn to learn the distance metrics.

Utility of PAD with Generic Distance Metric

We first focus on a general scenario where the system does not have access to similarity labels. This can happen either when the purpose of the data is not known before publication, or when the data analyst does not want to interact with PAD. In that case, a generic metric, i.e., Euclidean distance, is used for performing micro-aggregation. We validate the usefulness of the k-anonymized dataset through several typical data mining tasks, including occupancy prediction and occupancy statistics extraction.

Prediction

K-nearest neighbor (KNN) based occupancy prediction models are built using the original and sanitized database respectively with varying anonymity levels. To make prediction at time t, we compute the distance between the testing profile and all profiles in the training set during the interval $[t - \Delta t, t - 1]$ where Δt is the length of the window used for prediction, and then pick the most common occupancy value at t among the K nearest training profiles. Cross-validation is performed to compute the average prediction accuracy across all time steps in the day. The results are shown in Fig. 8.5 (a), where the prediction accuracies with original and sanitized dataset are both above 90%. There is a tradeoff between anonymity protection level and data utility. We can see that the prediction accuracy drops as the anonymity level of the published dataset is increased.

It is important to note that moderate degree of anonymization is helpful for improving model's robustness and better fitting unseen data. Particularly, KNN model constructed with 2-anonymized dataset achieves higher prediction accuracy than that built with original dataset. We also implement an occupancy prediction model based on Support Vector Machine (SVM) and the corresponding results are shown in Fig. 8.5 (b) where we can observe the similar patterns. This is because the training data points usually contain both the useful information that can be used to predict unseen cases, as well as the useless noise that can degrade the model. Essentially, k-anonymization reduces the "harmful" noise by aggregating similar data points and avoids overfitting. This suggests that for a data publication with moderate anonymity requirement the sanitized dataset is more advantageous than the original dataset since the sanitized one can achieve privacy protection as well as an improved model quality.

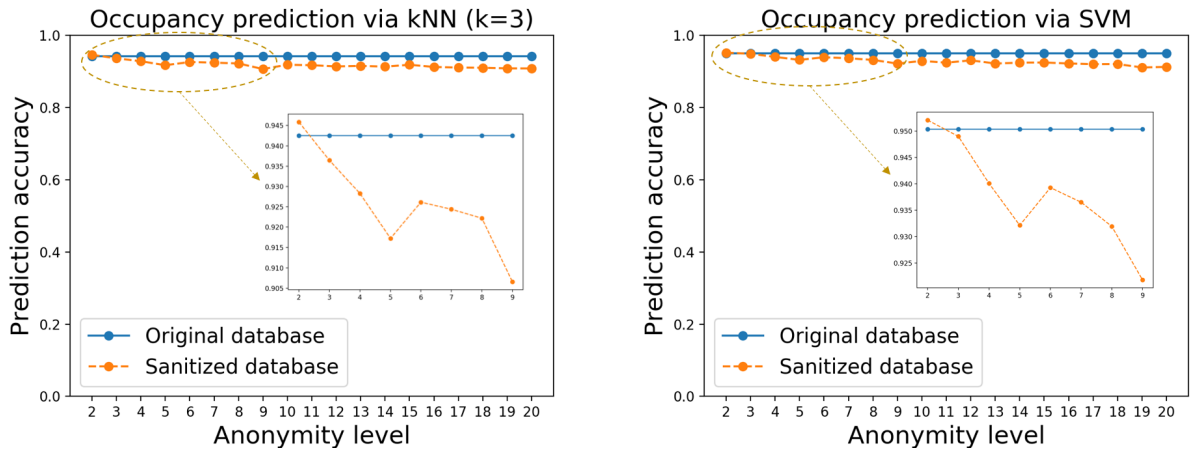
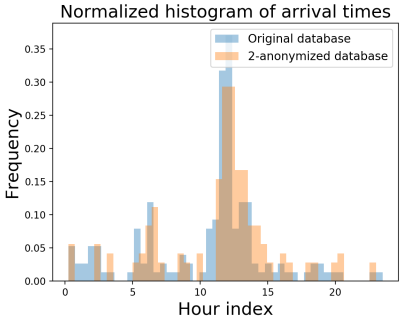


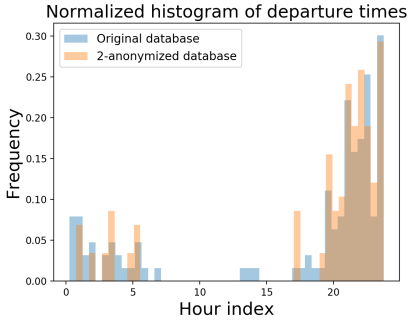
Figure 8.5: Comparison of prediction performance of occupancy models constructed by using the original vs. sanitized database.

Statistics

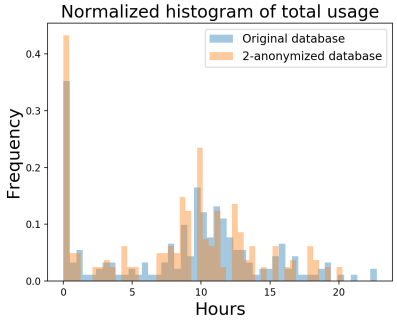
The raw time series collected in buildings are often processed into some key information that is directly useful for informing various control applications. For instance, occupancy statistics, such as arrival time, are particularly useful for designing occupant-responsive HVAC control algorithms. In light of this, we want to test if the sanitized database can retain these useful statistics. We compare the histograms of the useful occupancy statistics including arrival time, departure time, and total occupation time extracted from the original and sanitized database, respectively. Fig. 8.6 and Fig. 8.7 illustrate the results on the OU44 occupancy dataset and the smart home occupancy dataset, respectively. We can see that the anonymized datasets can preserve the distribution of these statistics, especially the mean and modes of the distribution. Take the OU44 occupancy dataset for example: the relative errors of using the 2-anonymized datasets to estimate the mean of arrival time, departure time and total occupation time are 8.13%, 8.37% and 6.21%, respectively; for 7-anonymized datasets, the relative errors are 6.80%, 5.34% and 0.47%, respectively. In other words, we can still retrieve accurate information about typical behaviors of occupants from the sanitized database. However, it is worth noting that data sanitization reduces the variability of the dataset, which is getting more pronounced when the anonymity level is increased to 7 as shown in Figure 8.6 (d), 8.6 (e) and 8.6 (f). For instance, the departures at noon cannot be detected with the anonymized dataset. This is a direct consequence of "hide in the crowd" philosophy of k-anonymity. Therefore, it will be easier to mine population properties than atypical patterns from the sanitized data.



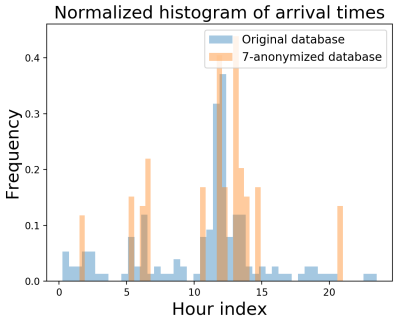
(a) Original vs. 2-anonymized



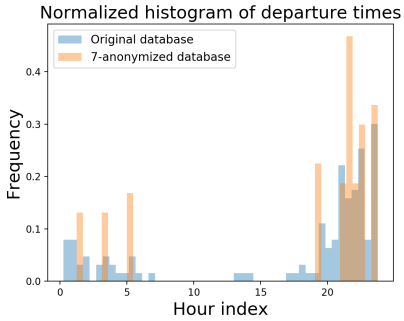
(b) Original vs. 2-anonymized



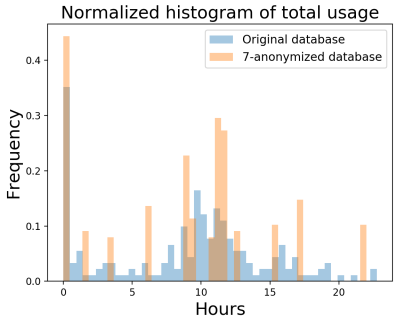
(c) Original vs. 2-anonymized



(d) Original vs. 7-anonymized



(e) Original vs. 7-anonymized



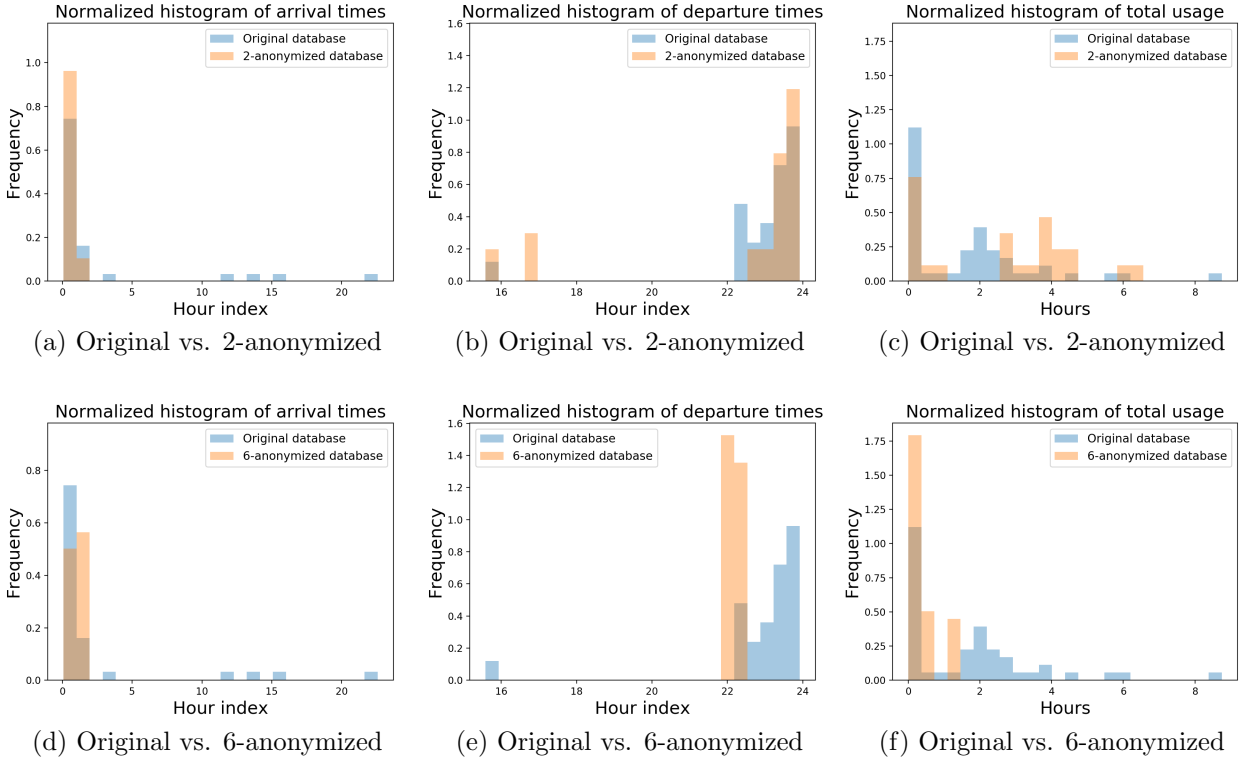
(f) Original vs. 7-anonymized

Figure 8.6: Comparison of occupancy statistics extracted from the OU44 occupancy dataset and the corresponding sanitized dataset.

Utility of PAD With Customized Distance Metric for Linear Features

In this part, we investigate scenarios where the purpose of the data is known at the time of publication and there exists a "best" distance metric for microaggregation, which retains the maximum amount of information pertaining to the data analyst's interest. For instance, if the data is used for studying occupancy patterns of a building during lunch time, then the best metric will be the Euclidean distance over the lunch period. The data records with similar lunch patterns will be grouped by the "best" metric; as a result, the information loss with respect to lunchtime occupancy patterns incurred by the substitution step will be minimized.

First, we consider that the feature that interests the data analyst is a linear function of the original data record. The aforementioned lunchtime occupancy pattern is an example the linear feature because the lunchtime occupancy is equivalent to multiplying the whole-day occupancy data by a diagonal matrix that has non-zero entries only at the coordinates corresponding to the lunchtime. In the sequel, we will use two examples, namely, occupancy data segments and peak-hour energy usage, to demonstrate the utility of PAD for linear



(a) Original vs. 2-anonymized (b) Original vs. 2-anonymized (c) Original vs. 2-anonymized

(d) Original vs. 6-anonymized (e) Original vs. 6-anonymized (f) Original vs. 6-anonymized

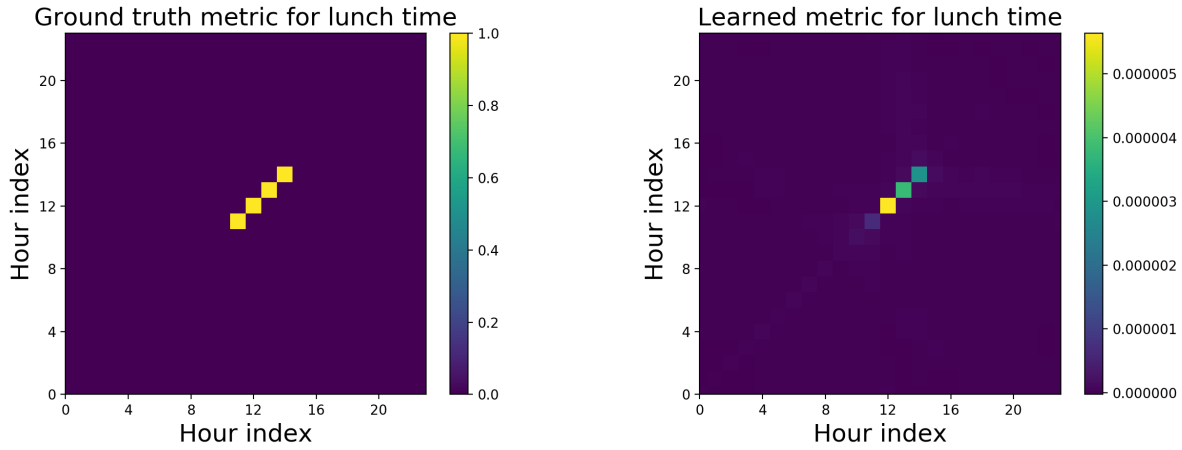
Figure 8.7: Comparison of occupancy statistics extracted from the smart home occupancy dataset and the corresponding sanitized dataset.

features.

Note that although there has been fruitful previous research on data publishing, different approaches may not be directly comparable because they may have different viewpoints on what is considered "private." Existing work on k-anonymization always relies on a generic metric in the microaggregation step. Therefore, PAD with the generic distance metric is used as the baseline approach for comparison here.

Segment

We use the following example to demonstrate the role of distance metric learning in the workflow of PAD. Consider that the data analyst wants to study the occupancy patterns during lunch time, i.e., $11:00-14:00$. The IC matrix A associated with the best metric that minimizes the information loss during lunch period is illustrated in Figure 8.8 (a), which is equivalent to cutting off the lunch period and applying the Euclidean distance. We call this best metric as *ground truth metric*. The distance metric learned by PAD is shown in Figure 8.8 (b), which visually exhibits the same pattern as the ground truth metric. The values on the diagonal pertaining to the lunch period dominate.



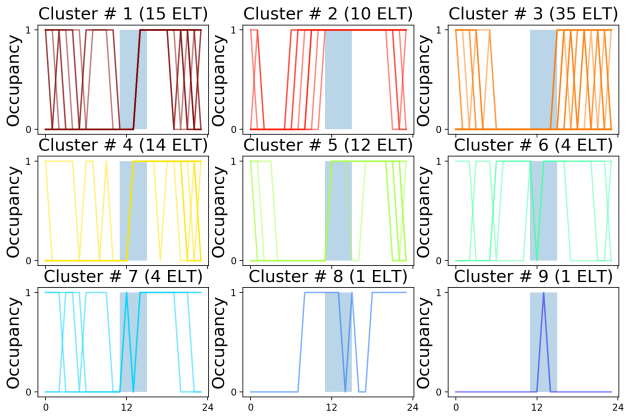
(a) (b)

Figure 8.8: Comparison of ground truth metric and the learned metric for specialized data publication for lunch times.

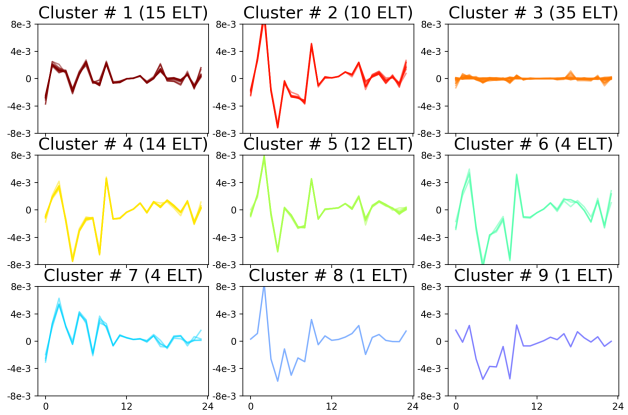
Close scrutinization on the learned metric shows that it contains some small nonzero off-diagonal entries which intuitively correspond to the rotation and rescaling of the original occupancy times series. In order to understand the effect of these small nonzero entries, we cluster all the daily profiles in the database according to the lunch time patterns and apply the linear transformation implied by the learned metric to the occupancy series in each cluster. The results are shown in Figure 8.9 (a) and 8.9 (b), respectively. We can see that the distance learning procedure finds a linear transformation under which the data points that are "similar" in the data analyst's view are close to each other in terms of Euclidean distance.

Figure 8.10 compares sanitization procedures that use a generic metric, the learned metric, and the ground truth metric, respectively, in terms of the tradeoff between anonymity level and information loss. We want to emphasize that the information loss for special-purpose publication measures the difference between the interested information in the original data record and that in the sanitized record. Here, the information loss refers to the Euclidean distance of the lunch periods between the record in the original database and its sanitized version in the published database. We can see that the information loss can be significantly reduced by learning a proper metric for microaggregation.

Figure 8.11 demonstrates that with more labeled data pairs PAD can achieve better data quality. The pre-sanitized database contains 16 different entries, and the maximum number of data pairs for labeling is $\binom{16}{2} = 120$. Although it requires extra labeling effort to reduce information loss, we want to point out that the data analyst can use computer program to achieve automatic labeling in the case where the desired metric is explicitly defined. For example, in this experiment we write a script to label the similarity of data points by first clustering the data points and assigning the similarity label to a data pair according to whether the pair of points reside in the same cluster. We can also observe from Figure 8.11



(a)



(b)

Figure 8.9: (a) Clustering of occupancy time series according to lunch patterns; (b) Applying the linear transformation implied by the learned metric to the data in each cluster. The number before "ELT" in the parenthesis gives the number of elements in each cluster.

that more labeled data pairs can reduce the variance of published data quality as well.

Peak hour energy usage

We consider an energy data use case that mines occupants' peak-hour energy use patterns. More specifically, the data analyst is interested in acquiring accurate information on total energy consumption during the peak hours, i.e., $17:00-20:00$. The ground truth metric associated with this example can be defined as $d_p(x, x') = \|f(x) - f(x')\|_2$ where f calculates the sum of the coordinates during peak hours for x and x'. Figure 8.12 shows the information loss of peak time usage in the published datasets using the generic metric, the learned metric

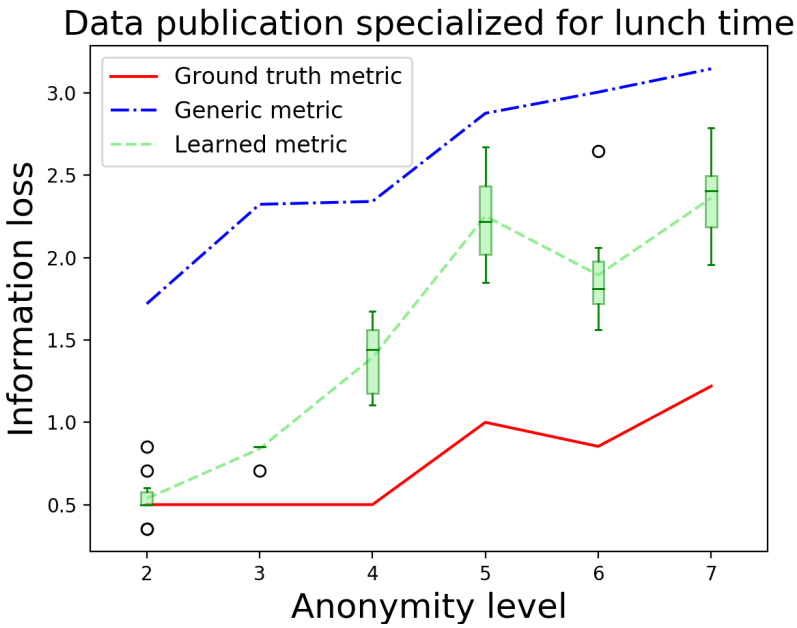


Figure 8.10: The tradeoff between anonymity level and information loss for the specialized publication for lunch time.

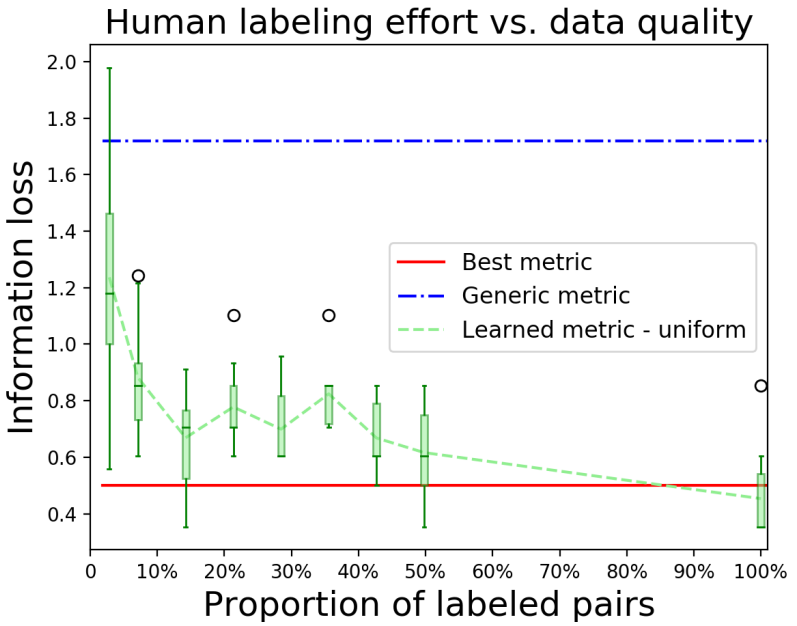


Figure 8.11: The tradeoff between labeling effort and information loss.

and the ground truth metric, respectively, under different anonymity guarantees. Again, the information loss is measured by the difference between peak-hour total usage of the original record and that of the sanitized version in the published database. We can observe a similar tradeoff between privacy and data quality to what we have seen in the use case of lunch-time segment. The information loss can be reduced by replacing a generic metric with the learned metric.

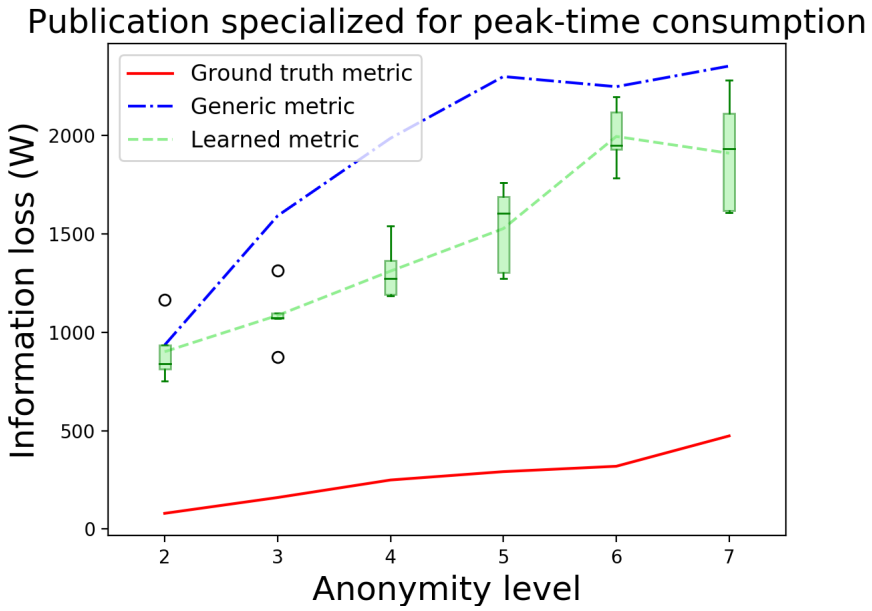


Figure 8.12: The tradeoff between anonymity and information loss for data publication specialized for peak hour energy usage.

Figure 8.13 shows the distribution of peak-time energy usage in the original database and the databases sanitized via the generic and learned metric. We can see that the learned metric better retains the modes of the original distribution. For instance, the peak-hour energy usage below 5000 W is completely neglected by the sanitized database with generic metric while the learned metric successfully grasps this probability mass and better captures the variation embedded in the original dataset.

Utility of PAD with Customized Distance Metric for Nonlinear Features

In this part, we will switch our focus to nonlinear features, which are quite common in mining smart building datasets. For instance, the data analyst is interested in modeling the arrival and departure time of a building from the occupancy datasets. Assume that each row in the database contains occupancy measurements throughout the day, denoted by a vector x. Let

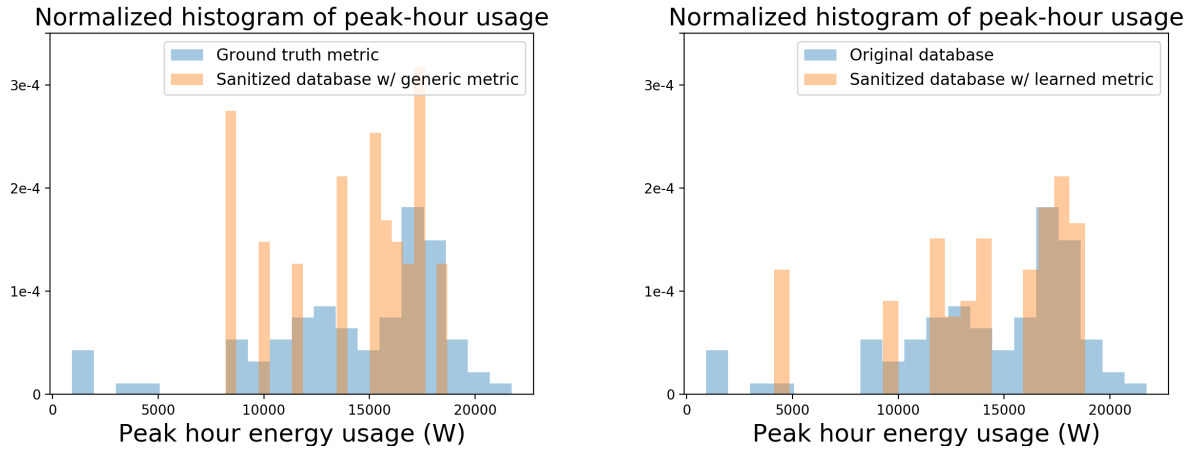
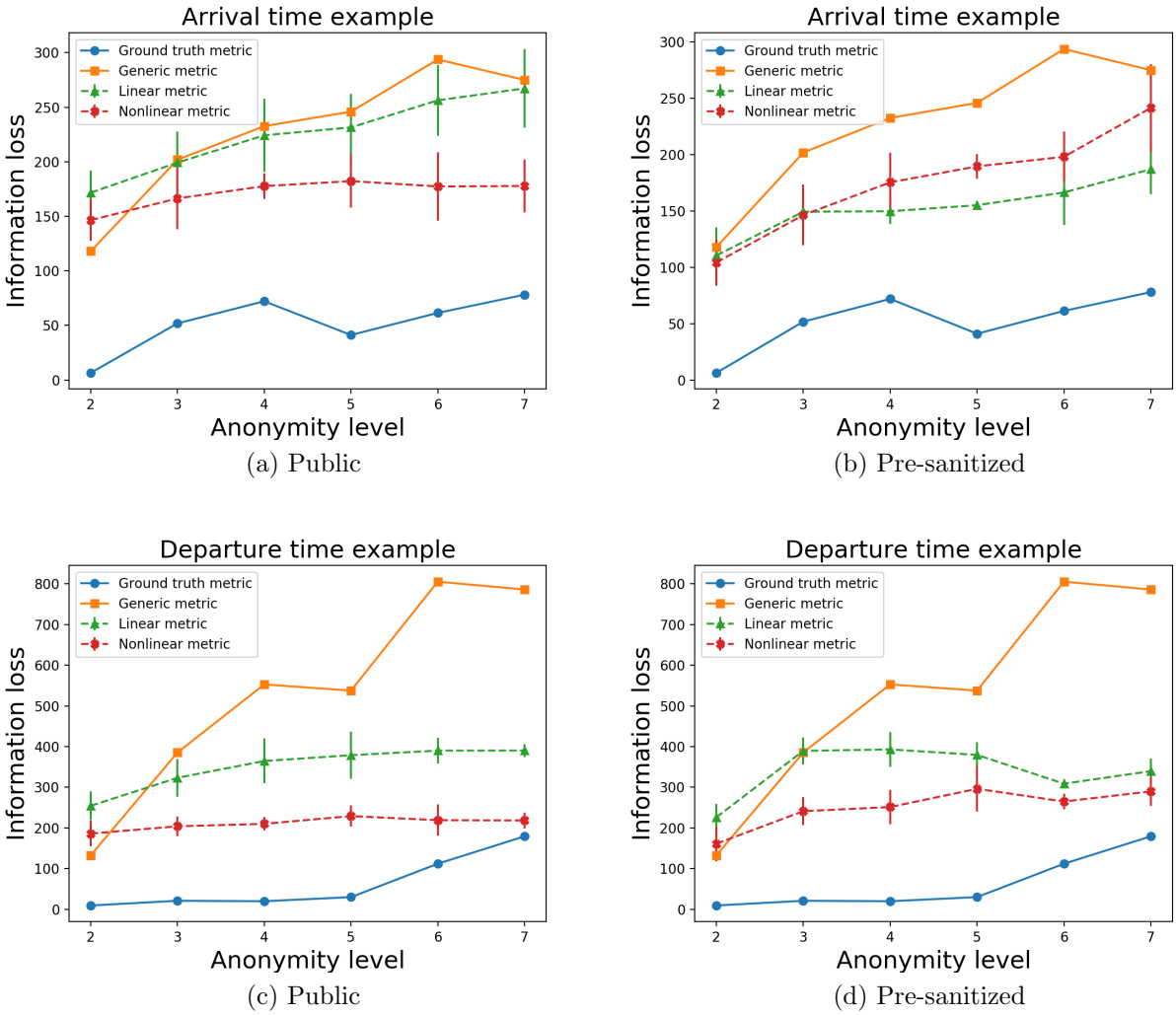


Figure 8.13: Comparison of 5-anonymized datasets with the Euclidean distance metric and the learned metric on peak-hour energy usage information recovery.

f be the function that calculate the arrival time of x. Then, we have $f(x)$ is equal to the first non-zero element in x, which is apparently a nonlinear function of x.

Fig. 8.14 (a) and Fig. 8.14 (b) compare the ability of different metrics to retain arrival time information. We can see that learning a proper non-linear metric for microaggregation is beneficial to the preservation of nonlinear features. Linear metrics require fewer examples to train because they have fewer parameters. Since the number of unique training samples decreases as anonymity level increases, we observed in Fig. 8.14 (b) that the linear metric is more performant than the nonlinear one when a high level of anonymity is desired. The results corresponding to departure time are illustrated in Fig. 8.14 (c) and Fig. 8.14 (d), where the advantage of non-linear metrics can be observed for both training scenarios.

In order to better understand the reason for the performance discrepancy between different distance metrics, we calculate the correlation between the learned distances and the ground truth distances for the four aforementioned examples, namely, lunch time occupancy pattern, peak hour energy usage, arrival time, and departure time. The correlation is measured in terms of the Pearson correlation coefficient, which has a value between $+1$ and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation. The results are listed in Table 8.1. The correlation between the generic distances (i.e., Euclidean distance) and the ground truth distances is also listed as a baseline. We can see that when the ground truth metric is nonlinear, the nonlinear metrics can produce distances that have highest correlation with the ground truth distances. On the other hand, if the ground truth metric is linear, linear metrics has the highest correlation with the ground truth distances. In addition, we can observe that both linear and nonlinear metrics are more indicative of the ground truth, compared to the Euclidean distance. In practice, the data



(a) Public (b) Pre-sanitized

(c) Public (d) Pre-sanitized

Figure 8.14: The tradeoff between anonymity and information loss for data publication specialized for arrival and departure times.

publishing system does not have access to data analysts' interests *a priori*; instead, only a set of data pairs with similarity labels are provided for the system. Therefore, the data publishing system can implement a nonlinear metric (e.g., via neural networks) since it can work sufficiently well for both linear and nonlinear features.

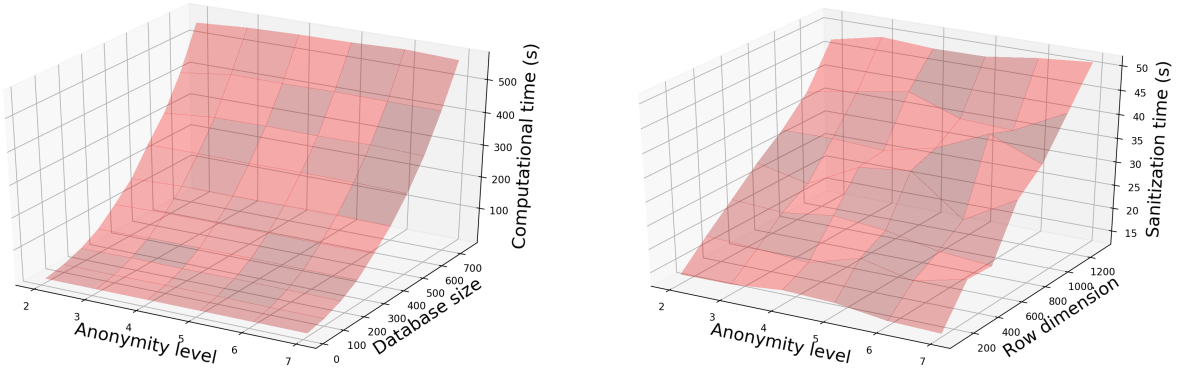
Computational Overhead

We study the computational overhead associated with PAD. We first look into the complexity of the microaggregation part. Let the size of the database be n, the dimension of the row be m, and the anonymity level be k. The microaggregation complexity mainly comprises

Table 8.1: Correlation between the learned distances and the ground truth distances for different use cases. Correlation is measured in terms of Pearson correlation coefficients. The correlation between the generic distance (i.e., Euclidean distance) and the ground truth distance is also listed as a baseline. The Pearson correlation coefficients are calculated at anonymity level 4 and averaged over 5 MC simulations.

Distance metrics	**Use cases**			
	Lunchtime occupancy	Peak hour energy usage	Arrival time	Departure time
Euclidean	0.32	0.64	0.41	0.39
Linear	**0.95**	**0.93**	0.34	0.36
Nonlinear	0.43	0.72	**0.68**	**0.96**

$O(n^2m)$ computations of distance values and the complexity of the clustering process which is shown to be $n(1-1/k)$ in the best case and $(n/k-1)(n/2+k-2)$ in the worst case [43]. Figure 8.15 demonstrates the computation time of microaggregation as a function of n, m and k. We can see that the overhead is approximately quadratic in the database size and linear in the dimension of the row. In addition, changing the anonymity level requirement does not affect the computational time significantly.



(a) (b)

Figure 8.15: Computational complexity of microaggregation.

The complexity of the deep metric learning step depends on the actual algorithm used for optimization and the convergence criterion. Fig. 8.16 illustrates the relationship between computational time of metric learning and database dimension. `Adam` is used for solving the optimization involved in the deep metric learning. Given a fixed number of epochs (the number of times that the learning algorithm goes through the training data pairs), learning rate and batch size, computational time associated with the metric learning part increases with the number of labeled data pairs and the dimension of the data records. Moreover, the number of labeled pairs dominates the computational overhead of the metric learning step.

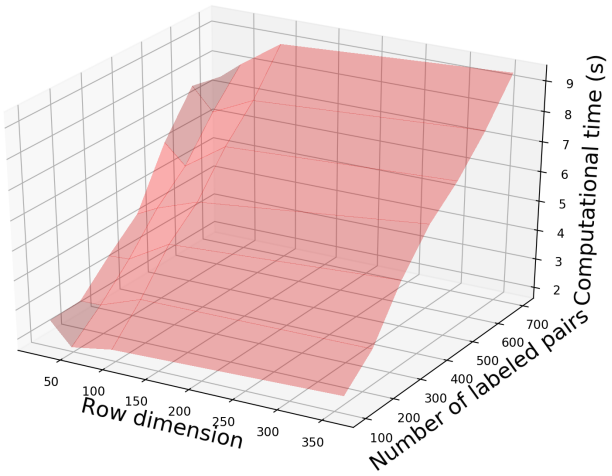


Figure 8.16: Computational overhead of the deep metric learning step.

8.7 Chapter Summary

In this chapter, we present an open-sourced data publication system, PAD, for protecting k-anonymity of time series data collected in buildings. Particularly, PAD can achieve better data utility than traditional anonymization techniques. This feat is achieved by customizing the data sanitization process to the potential data use. In order to tackle the scalability issues with hard-coding different data uses and their corresponding optimized anonymization procedures, we propose a simple protocol for data users to convey their diverse interests—the system provides a batch of data pairs and the analyst labels the similarity of each data pair according to his desired data use. PAD can then learn a more context-aware distance metric from the labeled data. We show through extensive experiments on real-world datasets that PAD can better preserve the usefulness of the published data while providing privacy protection. By proposing PAD we hope to revolutionize the way that CPSs' datasets are published.

In the current implementation of PAD, the published dataset is optimized for a particular data purpose. If the analyst has multiple purposes, it remains a question if there is a distance metric that better serves for multiple specific purposes than a simple generic metric. In addition, it needs further study on the privacy implication in the case where several data analysts who receive different sanitized datasets collude with each other.

Chapter 9

Final Words

CPSs are permeating our everyday lives. The data collected in CPSs brings unprecedented opportunities to promote the efficiency, resilience, and sustainability of the systems. However, the data is often heterogenous and may contain privacy-sensitive information that triggers public concern. To fulfill the potentials of new CPS technologies, we need to develop *integrative*, *accountable*, and *privacy-preserving* data analysis tools that can reliably transform large volumes of data into understandable and actionable insights that facilitate system operation and improve people's well-being.

This dissertation has been a key step toward the goal of designing a framework for accountable and privacy-preserving data analysis in CPSs. In this chapter, we would like to conclude this dissertation by reflecting on our approaches and discussing some of the limitations and future directions.

Our work on *accountable data fusion* focuses on three questions, namely, how to combine prior knowledge with real-time sensor measurements for modeling and inference, how to understand black-box predictions, how to identify low-quality and adversarial data, as discussed in Chapter 2, Chapter 3, and Chapter 4. We have addressed the first question by adopting a Bayesian modeling approach, in which the interdependence between measurements, unobserved states, and prior knowledge are represented by conditional probability distributions. Under the Bayesian framework, we can naturally obtain prediction confidence from the posterior distributions. However, in practice, the relationship between different random variables in the Bayesian models varies over time; as a result, we are required to modify the model on the fly. We have introduced a notion of data value to understand model predictions through the lens of the training data. The data value notion can also be used in future data marketplace for pricing data instances. However, the proposed data value can only be computed efficiently for a handful of machine learning models. In future, we plan to develop practical data valuation algorithms for a broader variety of models.

In the work of *privacy protection in CPS*, we first investigated the techniques to protect privacy when privacy-utility tradeoffs are allowed. We further differentiate between the case where one can modify both controllers and sensors to ensure privacy (Chapter 5) and the case in which the controller is fixed and only the sensor can be adjusted (Chapter 6). Then,

we considered a more pragmatic case where optimal utility is always desired (Chapter 7) and studied the minimal amount of data needed to achieve optimal utility. However, for system operations characterized by complex and large-scale optimization problems, analyzing the minimal data requirement is often computationally expensive or even intractable. We are required to design lightweight privacy mechanisms which can be run with limited computational and communication resources, say, on a wireless sensor node. Lastly, we addressed the privacy issues in data publication. However, the privacy notion used in our work is susceptible to attacks when different data users collude with each other. Hence, we need to design data publishing systems that can provide stronger privacy guarantees.

9.1 Future Directions

Accountable Reinforcement Learning

Reinforcement Learning (RL) is aimed at learning high-performance control actions through interactions with the environment. Significant progresses have been made recently by combining advances in deep learning for feature representation [99] with reinforcement learning. Notable examples include playing Go [145] and Atari games [116], acquiring advanced robot manipulation skills using raw sensory inputs [102]. However, one limitation of applying RL to real-world CPSs is the lack of interpretabilty for the RL control strategies. For instance, consider the application of RL-based control in buildings. A set of questions need to be answered before the building manager can trust and effectively manage an RL-based controller: How does the controller work? Why does the controller produce a certain control action for a given state? Are there any safety and performance guarantees associated with the controller? When should the controller hand over to a human expert?

Usability for Algorithmic Privacy

Beyond the algorithmic advances, if we are to achieve societal impacts around the issues of privacy, progress needs to be made in communicating to the public the protections offered by parameterized privacy-preserving algorithms. Firstly, we need to provide justification for the choice of parameter values, empowering users to exercise an individual choice on the parameter values. Secondly, the analysis of privacy-utility tradeoff needs to be make straightforward for people with no special training in privacy, working under tight time and resource constraints.

Practical Data Valuation for Data Marketplaces

As the amount of data collected by different entities increases and data becomes increasingly recognized as an asset, data marketplaces have proliferated in recent years. In data marketplaces, people can find and provision data for profit. Due to the pervasion of CPSs,

more and more people are empowered to collect their data and sell it; as a result, promoting data marketplaces might be able to mitigate the income inequality issues that we are facing today. The cornerstone of a data marketplace is a fair data valuation mechanism. In this thesis, we have made attempts to remunerate training data instances. However, it will be meaningful to extend the current work to value the entities who contribute computation, infrastructure, and intellectual property for data analytics (Figure 9.1). Hereinbefore, we only addresses the problem of attributing the given utility of services enabled by a model to each training data instance. It is crucial to develop techniques to appraise the financial value of the services. The value of a service can be definite or not, depending on the context. For the targeted advertisement services, the service value will be the profits obtained from deployed ads. However, in other cases, we will need to design mechanisms to elicit the service value from users. For instance, Facebook posts activities based on users' location and interest; the value of such services is not clear without an explicit pricing mechanism. Another important aspect for practical data valuation systems is the ability to track the provenance of data, without which one could easily inflate his profits by replicating other people's data.

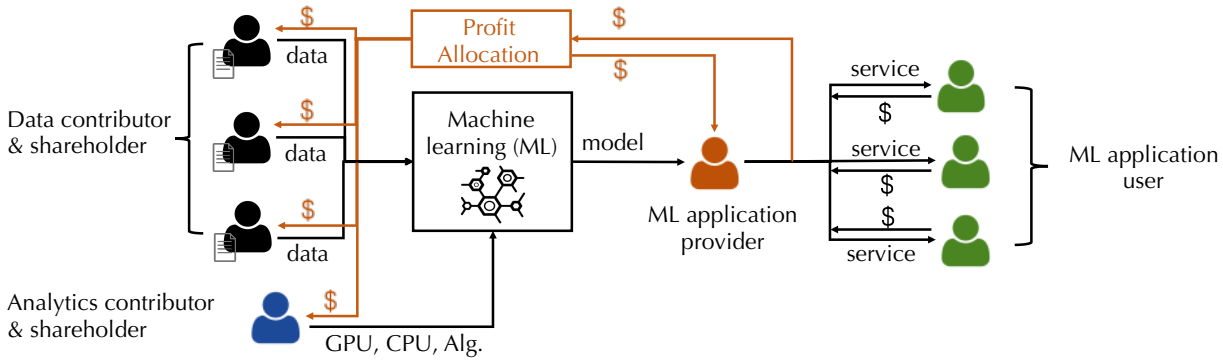


Figure 9.1: Illustration of a data marketplace.

9.2 Closing Thoughts

Our work presented in this dissertation is only a step that attempts to bring ideas from control, optimization, learning, and game theory to address the accountability and privacy issues of data analysis in CPSs. As discussed in this chapter, there are still many existing challenges that need to be addressed. We envision a strong collaboration between various disciplines for fulfilling the potential of CPSs.

Bibliography

[1] https://nest.com/support/article/Learn-more-about-Home-Away-Assist.

[2] *Aggregated Data Access: The 15/15 Rule in Illinois and Beyond*. http://www.elevateenergy.org/wp/wp-content/uploads/1515-Rule-Factsheet-FINAL.pdf. [Online; accessed 15-Jun-2017]. 2013.

[3] Mohamed H Albadi and Ehab F El-Saadany. "Demand response in electricity markets: An overview". In: *Power Engineering Society General Meeting, 2007. IEEE*. IEEE. 2007, pp. 1–5.

[4] Miguel E Andrés et al. "Geo-indistinguishability: Differential privacy for location-based systems". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM. 2013, pp. 901–914.

[5] *ANSI/ASHRAE Standard 62.1-2013: Ventilation for Acceptable Indoor Air Quality*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2013.

[6] M Sanjeev Arulampalam et al. "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". In: *Signal Processing, IEEE Transactions on* 50.2 (2002), pp. 174–188.

[7] M Sanjeev Arulampalam et al. "A variable structure multiple model particle filter for GMTI tracking". In: *Information Fusion, 2002. Proceedings of the Fifth International Conference on*. Vol. 2. IEEE. 2002, pp. 927–934.

[8] Bharathan Balaji et al. "Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings". In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2013, p. 17.

[9] Stephane Beauregard and Harald Haas. "Pedestrian dead reckoning: A basis for personal positioning". In: *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*. 2006, pp. 27–35.

[10] Abdelmoula Bekkali, Horacio Sanson, and Mitsuji Matsumoto. "RFID indoor positioning based on probabilistic RFID map and Kalman filtering". In: *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*. IEEE. 2007, pp. 21–21.

[11] Alex Beltran and Alberto E Cerpa. "Optimal HVAC building control with occupancy prediction". In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM. 2014, pp. 168–171.

[12] George Bennett. "Probability inequalities for the sum of independent random variables". In: *Journal of the American Statistical Association* 57.297 (1962), pp. 33–45.

[13] Dimitri P Bertsekas et al. *Dynamic programming and optimal control*. Vol. 1. 2. Athena Scientific Belmont, MA, 1995.

[14] Battista Biggio, Blaine Nelson, and Pavel Laskov. "Poisoning attacks against support vector machines". In: *arXiv preprint arXiv:1206.6389* (2012).

[15] Battista Biggio et al. "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks". In: *International workshop on multiple classifier systems*. Springer. 2011, pp. 350–359.

[16] Battista Biggio et al. "Poisoning attacks to compromise face templates". In: *Biometrics (ICB), 2013 International Conference on*. IEEE. 2013, pp. 1–7.

[17] Christopher M Bishop. "Pattern recognition". In: *Machine Learning* 128 (2006), pp. 1–58.

[18] Andy Bloxham. *Most burglars using Facebook and Twitter to target victims, survey suggests*. `http://www.telegraph.co.uk/technology/news/8789538/Most-burglars-using-Facebook-and-Twitter-to-target-victims-survey-suggests.html`. [Online; accessed 26-Sep-2011]. 2011.

[19] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[20] Atreyi Bose and Chuan Heng Foh. "A practical path loss model for indoor WiFi positioning enhancement". In: *Information, Communications & Signal Processing, 2007 6th International Conference on*. IEEE. 2007, pp. 1–5.

[21] Olivier Bousquet and André Elisseeff. "Stability and generalization". In: *Journal of machine learning research* 2.Mar (2002), pp. 499–526.

[22] Duncan S Callaway. "Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy". In: *Energy Conversion and Management* 50.5 (2009), pp. 1389–1400.

[23] Nicholas Carlini and David Wagner. "Towards evaluating the robustness of neural networks". In: *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE. 2017, pp. 39–57.

[24] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. "Differentially private empirical risk minimization". In: *Journal of Machine Learning Research* 12.Mar (2011), pp. 1069–1109.

[25] Xinyun Chen et al. "Targeted backdoor attacks on deep learning systems using data poisoning". In: *arXiv preprint arXiv:1712.05526* (2017).

[26] Yudong Chen, Constantine Caramanis, and Shie Mannor. "Robust high dimensional sparse regression and matching pursuit". In: *arXiv preprint arXiv:1301.2725* (2013).

[27] Zhenghua Chen et al. "Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization". In: *Sensors* 15.1 (2015), pp. 715–732.

[28] François Chollet et al. *Keras*. `https://github.com/keras-team/keras`. 2015.

[29] Gilad Cohen, Raja Giryes, and Guillermo Sapiro. "DNN or k-NN: That is the Generalize vs. Memorize Question". In: *arXiv preprint arXiv:1805.06822* (2018).

[30] Shay Cohen, Eytan Ruppin, and Gideon Dror. "Feature selection based on the Shapley value". In: *In other words* 1 (2005), 98Eqr.

[31] Federal Trade Commission et al. "Fair information practice principles". In: *last modified June* 25 (2007).

[32] Jorge Cortés et al. "Differential privacy in control and network systems". In: *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE. 2016, pp. 4252–4272.

[33] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[34] Lorrie Faith Cranor. "Necessary but not sufficient: Standardized mechanisms for privacy notice and choice". In: *J. on Telecomm. & High Tech. L.* 10 (2012), p. 273.

[35] George E Dahl et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition". In: *IEEE Transactions on audio, speech, and language processing* 20.1 (2012), pp. 30–42.

[36] Fida Kamal Dankar and Khaled El Emam. "Practicing Differential Privacy in Health Care: A Review." In: *Transactions on Data Privacy* 6.1 (2013), pp. 35–67.

[37] Anirban Dasgupta et al. "Sampling algorithms and coresets for \ell_p regression". In: *SIAM Journal on Computing* 38.5 (2009), pp. 2060–2078.

[38] Jason V Davis et al. "Information-theoretic metric learning". In: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, pp. 209–216.

[39] Xiaotie Deng and Christos H Papadimitriou. "On the complexity of cooperative solution concepts". In: *Mathematics of Operations Research* 19.2 (1994), pp. 257–266.

[40] Whitfield Diffie and Martin E Hellman. "Privacy and authentication: An introduction to cryptography". In: *Proceedings of the IEEE* 67.3 (1979), pp. 397–427.

[41] Simona D'Oca and Tianzhen Hong. "Occupancy schedules learning process through a data mining framework". In: *Energy and Buildings* 88 (2015), pp. 395–408.

[42] J. Domingo-Ferrer and J. M. Mateo-Sanz. "Practical data-oriented microaggregation for statistical disclosure control". In: *IEEE Transactions on Knowledge and Data Engineering* 14.1 (Jan. 2002), pp. 189–201. ISSN: 1041-4347. DOI: `10.1109/69.979982`.

[43] Josep Domingo-Ferrer. "Microaggregation for database and location privacy". In: *International Workshop on Next Generation Information Technologies and Systems*. Springer. 2006, pp. 106–116.

[44] Josep Domingo-Ferrer and Josep Maria Mateo-Sanz. "Practical data-oriented microaggregation for statistical disclosure control". In: *IEEE Transactions on Knowledge and data Engineering* 14.1 (2002), pp. 189–201.

[45] Bing Dong et al. "An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network". In: *Energy and Buildings* 42.7 (2010), pp. 1038–1046.

[46] Dingzhu Du, Frank K Hwang, and Frank Hwang. *Combinatorial group testing and its applications*. Vol. 12. World Scientific, 2000.

[47] John C Duchi, Michael I Jordan, and Martin J Wainwright. "Local privacy and statistical minimax rates". In: *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE. 2013, pp. 429–438.

[48] Cynthia Dwork. "Differential privacy". In: *Proc. of the Int. Colloq. on Automata, Languages and Programming*. Springer, 2006, pp. 1–12.

[49] Cynthia Dwork. "Differential privacy: A survey of results". In: *International Conference on Theory and Applications of Models of Computation*. Springer. 2008, pp. 1–19.

[50] US EIA. "Annual energy review". In: *Energy Information Administration, US Department of Energy: Washington, DC www. eia. doe. gov/emeu/aer* (2011).

[51] Murat A Erdogdu and Nadia Fawaz. "Privacy-utility trade-off under continual observation". In: *Information Theory (ISIT), 2015 IEEE International Symposium on*. IEEE. 2015, pp. 1801–1805.

[52] Varick L Erickson and Alberto E Cerpa. "Occupancy based demand response HVAC control strategy". In: *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*. ACM. 2010, pp. 7–12.

[53] Frédéric Evennou and François Marx. "Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning". In: *Eurasip journal on applied signal processing* 2006 (2006), pp. 164–164.

[54] Frédéric Evennou, François Marx, and Emil Novakov. "Map-aided indoor mobile positioning system using particle filter". In: *Wireless Communications and Networking Conference, 2005 IEEE*. Vol. 4. IEEE. 2005, pp. 2490–2494.

[55] Liyue Fan et al. "Monitoring web browsing behavior with differential privacy". In: *Proceedings of the 23rd international conference on World wide web*. ACM. 2014, pp. 177–188.

[56] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. "A linear approximation method for the Shapley value". In: *Artificial Intelligence* 172.14 (2008), pp. 1673–1699.

[57] Jiashi Feng et al. "Robust logistic regression and classification". In: *Advances in neural information processing systems*. 2014, pp. 253–261.

[58] Brian Ferris, Dirk Haehnel, and Dieter Fox. "Gaussian processes for signal strength-based location estimation". In: *In proc. of robotics science and systems*. Citeseer. 2006.

[59] Benjamin Fung et al. "Privacy-preserving data publishing: A survey of recent developments". In: *ACM Computing Surveys (Csur)* 42.4 (2010), p. 14.

[60] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[61] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[62] Neil J Gordon, David J Salmond, and Adrian FM Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE Proceedings F (Radar and Signal Processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

[63] Siddharth Goyal, Herbert A Ingley, and Prabir Barooah. "Occupancy-based zone-climate control for energy-efficient buildings: Complexity vs. performance". In: *Applied Energy* 106 (2013), pp. 209–221.

[64] Marco Gruteser and Dirk Grunwald. "Anonymous usage of location-based services through spatial and temporal cloaking". In: *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM. 2003, pp. 31–42.

[65] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. "Badnets: Identifying vulnerabilities in the machine learning model supply chain". In: *arXiv preprint arXiv:1708.06733* (2017).

[66] Faruk Gul. "Bargaining foundations of Shapley value". In: *Econometrica: Journal of the Econometric Society* (1989), pp. 81–95.

[67] Mehreen S Gul and Sandhya Patidar. "Understanding the energy consumption and occupancy of a multi-purpose academic building". In: *Energy and Buildings* 87 (2015), pp. 155–165.

[68] Dominik Gusenbauer, Carsten Isert, and Jens Krösche. "Self-contained indoor positioning on off-the-shelf mobile devices". In: *Indoor positioning and indoor navigation (IPIN), 2010 international conference on*. IEEE. 2010, pp. 1–9.

[69] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping". In: *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. Vol. 2. IEEE. 2006, pp. 1735–1742.

[70] Mike Hazas and Andy Hopper. "Broadband ultrasonic location systems for improved indoor positioning". In: *Mobile Computing, IEEE Transactions on* 5.5 (2006), pp. 536–547.

[71] Chunrong He, Songtao Guo, and Yuanyuan Yang. "Voronoi diagram based indoor localization in wireless sensor networks". In: *Communications (ICC), 2015 IEEE International Conference on*. IEEE. 2015, pp. 3269–3274.

[72] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[73] Sebastian Hilsenbeck et al. "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2014, pp. 147–158.

[74] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. "Discriminative deep metric learning for face verification in the wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1875–1882.

[75] Peter J Huber. "Robust statistics". In: *International Encyclopedia of Statistical Science*. Springer, 2011, pp. 1248–1251.

[76] Ruoxi Jia and Costas Spanos. "Occupancy modelling in shared spaces of buildings: a queueing approach". In: *Journal of Building Performance Simulation* 10.4 (2017), pp. 406–421.

[77] Ruoxi Jia et al. "Optimal Sensor-Controller Codesign for Privacy in Dynamical Systems". In: *56th IEEE Conference on Decision and Control*. 2017.

[78] Ruoxi Jia et al. "Poisoning Attacks on Data-Driven Utility Learning in Games". In: *The 2018 American Control Conference*. 2018.

[79] Ruoxi Jia et al. "Privacy-Enhanced Architecture for Occupancy-based HVAC Control". In: *arXiv preprint arXiv:1607.03140* (2016).

[80] Ruoxi Jia et al. "Privacy-enhanced architecture for occupancy-based HVAC control". In: *Proceedings of the 8th International Conference on Cyber-Physical Systems*. ACM. 2017, pp. 177–186.

[81] Jiantao Jiao et al. "Justification of logarithmic loss via the benefit of side information". In: *IEEE Transactions on Information Theory* 61.10 (2015), pp. 5357–5365.

[82] Ming Jin, Ruoxi Jia, and Costas Spanos. "APEC: Auto Planner for Efficient Configuration of Indoor Positioning Systems". In: *The 9th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'15)*. 2015, pp. 100–107.

[83] Ming Jin, Ruoxi Jia, and Costas Spanos. "Virtual occupancy sensing: Using smart meters to indicate your presence". In: *IEEE Transactions on Mobile Computing* (2017).

[84] Ming Jin, Javad Lavaei, and Karl Johansson. "A Semidefinite Programming Relaxation under False Data Injection Attacks against Power Grid AC State Estimation". In: *55th Annual Allerton Conference on Communication, Control, and Computing*. 2017.

[85] Ming Jin, Lin Zhang, and Costas Spanos. "Power Prediction through Energy Consumption Pattern Recognition for Smart Buildings". In: *IEEE International Conference on Automation Science and Engineering (IEEE CASE 2015)*. 2015, pp. 419–424.

[86] Ming Jin et al. "Environmental Sensing by Wearable Device for Indoor Activity and Location Estimation". In: *40th Annual Conference of the IEEE Industrial Electronics Society (IECON 2014)*. 2014, pp. 5369–5375.

[87] Ming Jin et al. "Inverse Reinforcement Learning via Deep Gaussian Process". In: *The Conference on Uncertainty in Artificial Intelligence (UAI)*. 2017.

[88] Ming Jin et al. "Presencesense: Zero-training algorithm for individual presence detection based on power monitoring". In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM. 2014, pp. 1–10.

[89] Ming Jin et al. "Sensing by Proxy: Occupancy Detection Based on Indoor CO2 Concentration". In: *UBICOMM 2015* (2015), p. 14.

[90] Susanna Kaiser, Mohammed Khider, and Patrick Robertson. "A human motion model based on maps for navigation systems". In: *EURASIP Journal on Wireless Communications and Networking* 2011.1 (2011), pp. 1–14.

[91] Olav Kallenberg. *Foundations of Modern Probability*. Springer, 2002.

[92] Anthony Kelman and Francesco Borrelli. "Bilinear model predictive control of a HVAC system using sequential quadratic programming". In: *Ifac world congress*. Vol. 18. 2011, pp. 9869–9874.

[93] Md Abdullah Al Hafiz Khan, HM Hossain, and Nirmalya Roy. "Infrastructure-less occupancy detection and semantic localization in smart environments". In: *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2015, pp. 51–60.

[94] Wilhelm Kleiminger, Silvia Santini, and Friedemann Mattern. "Smart heating control with occupancy prediction: how much can one save?" In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM. 2014, pp. 947–954.

[95] Pang Wei Koh and Percy Liang. "Understanding Black-box Predictions via Influence Functions". In: *International Conference on Machine Learning*. 2017, pp. 1885–1894.

[96] Ioannis C Konstantakopoulos et al. "A Robust Utility Learning Framework via Inverse Optimization". In: *IEEE Transactions on Control Systems Technology* 99 (2017), pp. 1–17.

[97] Ricky Laishram and Vir Virander Phoha. "Curie: A method for protecting SVM Classifier from Poisoning Attack". In: *arXiv preprint arXiv:1606.01584* (2016).

[98] Jerome Le Ny and George J Pappas. "Differentially private filtering". In: *IEEE Transactions on Automatic Control* 59.2 (2014), pp. 341–354.

[99] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436.

[100] Bo Li et al. "Data poisoning attacks on factorization-based collaborative filtering". In: *Advances in neural information processing systems*. 2016, pp. 1885–1893.

[101] Lin Liao et al. "Voronoi tracking: Location estimation using sparse and noisy sensor data". In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 1. IEEE. 2003, pp. 723–728.

[102] Timothy P Lillicrap et al. "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971* (2015).

[103] RHA Lindelauf, HJM Hamers, and BGM Husslage. "Cooperative game theoretic centrality analysis of terrorist networks: The cases of jemaah islamiyah and al qaeda". In: *European Journal of Operational Research* 229.1 (2013), pp. 230–238.

[104] M.A. Lisovich, D.K. Mulligan, and S.B. Wicker. "Inferring Personal Information from Demand-Response Systems". In: *IEEE Security & Privacy* 8 (2010), pp. 11–20. ISSN: 1540-7993. DOI: 10.1109/MSP.2010.40.

[105] Chang Liu et al. "Robust linear regression against training data poisoning". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM. 2017, pp. 91–102.

[106] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4768–4777.

[107] Xuan Luo et al. "Electric load shape benchmarking for small-and medium-sized commercial buildings". In: *Applied Energy* 204 (2017), pp. 715–725.

[108] Sasan Maleki. "Addressing the computational issues of the Shapley value with applications in the smart grid". PhD thesis. University of Southampton, 2015.

[109] Sasan Maleki et al. "Bounding the estimation error of sampling-based Shapley value approximation". In: *arXiv preprint arXiv:1306.4265* (2013).

[110] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[111] Eoghan McKenna, Ian Richardson, and Murray Thomson. "Smart meter data: Balancing consumer privacy concerns with legitimate applications". In: *Energy Policy* 41 (2012), pp. 807–814.

[112] Carlos Medina, José Carlos Segura, and Angel De la Torre. "Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy". In: *Sensors* 13.3 (2013), pp. 3501–3526.

[113] Lewis Meier, John Peschon, and R Dressler. "Optimal control of measurement subsystems". In: *IEEE Transactions on Automatic Control* 12.5 (1967), pp. 528–536.

[114] Fei Miao et al. "Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach". In: *IEEE Transactions on Automation Science and Engineering* 13.2 (2016), pp. 463–478.

[115] Tomasz Michalak et al. "Computational analysis of connectivity games with applications to the investigation of terrorist networks". In: (2013).

[116] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), p. 529.

[117] Fatiha Mokdad et al. "Determination of an optimal feature selection method based on maximum Shapley value". In: *Intelligent Systems Design and Applications (ISDA), 2015 15th International Conference on*. IEEE. 2015, pp. 116–121.

[118] Andrés Molina-Markham et al. "Private memoirs of a smart meter". In: *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*. ACM. 2010, pp. 61–66.

[119] Luis Muñoz-González et al. "Towards poisoning of deep learning algorithms with back-gradient optimization". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM. 2017, pp. 27–38.

[120] Srinarayana Nagarathinam et al. "Centralized Management of HVAC Energy in Large Multi-AHU Zones". In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM. 2015, pp. 157–166.

[121] Blaine Nelson et al. "Exploiting Machine Learning to Subvert Your Spam Filter." In: *LEET* 8 (2008), pp. 1–9.

[122] Kohei Ogawa, Yoshiki Suzuki, and Ichiro Takeuchi. "Safe screening of non-support vectors in pathwise SVM computation". In: *International Conference on Machine Learning*. 2013, pp. 1382–1390.

[123] Zhaoguang Pan, Qinglai Guo, and Hongbin Sun. "Feasible region method based integrated heat and electricity dispatch considering building thermal inertia". In: *Applied Energy* 192 (2017), pp. 395–407.

[124] Jun-geun Park. "Indoor localization using place and motion signatures". PhD thesis. Citeseer, 2013.

[125] Andrea Paudice et al. "Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection". In: *arXiv preprint arXiv:1802.03041* (2018).

[126] Leon Petrosjan and Georges Zaccour. "Time-consistent Shapley value allocation of pollution cost reduction". In: *Journal of economic dynamics and control* 27.3 (2003), pp. 381–398.

[127] Jan Petzold. "Augsburg Indoor Location Tracking Benchmarks". In: (2004).

[128] F. du Pin Calmon and N. Fawaz. "Privacy against statistical inference". In: *2012 50th Annu. Allerton Conf. on Commun., Control, and Computing (Allerton)*. Oct. 2012, pp. 1401–1408. DOI: 10.1109/Allerton.2012.6483382.

[129] Flávio du Pin Calmon and Nadia Fawaz. "Privacy against statistical inference". In: *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE. 2012, pp. 1401–1408.

[130] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. "The cricket location-support system". In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM. 2000, pp. 32–43.

[131] *Protection of personal data*. `http://ec.europa.eu/justice/data-protection/`. [Online; accessed 13-Jan-2017]. 2012.

[132] Guo-Jun Qi et al. "An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 841–848.

[133] S. R. Rajagopalan et al. "Smart meter privacy: A utility-privacy framework". In: *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. Oct. 2011, pp. 190–195. DOI: `10.1109/SmartGridComm.2011.6102315`.

[134] Lillian J. Ratliff et al. "Effects of Risk on Privacy Contracts for Demand-Side Management". In: *arXiv:1409.7926v3* (2015).

[135] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[136] Samer S Saab and Zahi S Nakad. "A standalone RFID indoor positioning system using passive tags". In: *Industrial Electronics, IEEE Transactions on* 58.5 (2011), pp. 1961–1970.

[137] Salman Salamatian et al. "Managing your private and public data: Bringing down inference attacks against your privacy". In: *IEEE Journal of Selected Topics in Signal Processing* 9.7 (2015), pp. 1240–1255.

[138] Fisayo Caleb Sangogboye et al. "Performance comparison of occupancy count estimation and prediction with common versus dedicated sensors for building model predictive control". In: *Building Simulation* 10.6 (Dec. 2017), pp. 829–843. ISSN: 1996-8744. DOI: `10.1007/s12273-017-0397-5`. URL: `https://doi.org/10.1007/s12273-017-0397-5`.

[139] Lalitha Sankar, S Raj Rajagopalan, and Soheil Mohajer. "Smart meter privacy: A theoretical framework". In: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 837–846.

[140] S Sasikala, S Appavu alias Balamurugan, and S Geetha. "A novel feature selection technique for improved survivability diagnosis of breast cancer". In: *Procedia Computer Science* 50 (2015), pp. 16–23.

[141] Ali Shafahi et al. "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks". In: *arXiv preprint arXiv:1804.00792* (2018).

[142] Lloyd S Shapley. "A value for n-person games". In: *Contributions to the Theory of Games* 2.28 (1953), pp. 307–317.

[143] Boris Sharchilev et al. "Finding Influential Training Samples for Gradient Boosted Decision Trees". In: *arXiv preprint arXiv:1802.06640* (2018).

[144] Reza Shokri et al. "Quantifying location privacy". In: *Security and privacy (sp), 2011 ieee symposium on*. IEEE. 2011, pp. 247–262.

[145] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), p. 484.

[146] Steven S Skiena. *The algorithm design manual: Text*. Vol. 1. Springer Science & Business Media.

[147] Jordi Soria-Comas et al. "Enhancing data utility in differential privacy via microaggregation-based k-anonymity". In: *The VLDB Journal* 23.5 (2014), pp. 771–794.

[148] JøRgen SpjøTvold et al. "On the facet-to-facet property of solutions to convex parametric quadratic programs". In: *Automatica* 42.12 (2006), pp. 2209–2214.

[149] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. "Certified defenses for data poisoning attacks". In: *Advances in Neural Information Processing Systems*. 2017, pp. 3517–3529.

[150] Xin Sun et al. "Using cooperative game theory to optimize the feature selection problem". In: *Neurocomputing* 97 (2012), pp. 86–93.

[151] Nattapong Swangmuang and Prashant Krishnamurthy. "Location fingerprint analyses toward efficient indoor positioning". In: *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*. IEEE. 2008, pp. 100–109.

[152] Latanya Sweeney. "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[153] Takashi Tanaka and Henrik Sandberg. "SDP-based joint sensor and controller design for information-regularized optimal LQG control". In: *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE. 2015, pp. 4486–4491.

[154] Sekhar Tatikonda, Anant Sahai, and Sanjoy Mitter. "Stochastic linear control over a communication channel". In: *IEEE transactions on Automatic Control* 49.9 (2004), pp. 1549–1561.

[155] Petter TøNdel, Tor Arne Johansen, and Alberto Bemporad. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions". In: *Automatica* 39.3 (2003), pp. 489–497.

[156] Ivor W Tsang et al. "Distance metric learning with kernels". In: *Proceedings of the International Conference on Artificial Neural Networks*. 2003, pp. 126–129.

[157] Parv Venkitasubramaniam, Jiyun Yao, and Parth Pradhan. "Information-theoretic security in stochastic control systems". In: *Proceedings of the IEEE* 103.10 (2015), pp. 1914–1931.

[158] He Wang et al. "Unsupervised indoor localization". In: *MobiSys. ACM* (2012).

[159] Hua Wang, Lili Sun, and Elisa Bertino. "Building access control policy model for privacy preserving and testing policy conflicting problems". In: *Journal of Computer and System Sciences* 80.8 (2014). Special Issue on Theory and Applications in Parallel and Distributed Computing Systems, pp. 1493–1503. ISSN: 0022-0000. DOI: http://dx.doi.org/10.1016/j.jcss.2014.04.017. URL: http://www.sciencedirect.com/science/article/pii/S0022000014000610.

[160] Hui Wang et al. "Enhancing the map usage for indoor location-aware systems". In: *Human-Computer Interaction. Interaction Platforms and Techniques*. Springer, 2007, pp. 151–160.

[161] Xiao Wang and Patrick Tague. "Non-Invasive User Tracking via Passive Sensing: Privacy Risks of Time-Series Occupancy Measurement". In: *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*. ACM. 2014, pp. 113–124.

[162] Roy Want et al. "The active badge location system". In: *ACM Transactions on Information Systems (TOIS)* 10.1 (1992), pp. 91–102.

[163] Stanley L Warner. "Randomized response: A survey technique for eliminating evasive answer bias". In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69.

[164] Kevin Weekly et al. "Building-in-Briefcase: A Rapidly-Deployable Environmental Sensor Suite for the Smart Building". In: *Sensors (Basel, Switzerland)* 18.5 (2018).

[165] Kevin Weekly et al. "Indoor occupant positioning system using active RFID deployment and particle filters". In: *Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on*. IEEE. 2014, pp. 35–42.

[166] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. "Distance metric learning for large margin nearest neighbor classification". In: *Advances in neural information processing systems*. 2006, pp. 1473–1480.

[167] WP216. "Opinion 05/2014 on Anonymisation Techniques". In: (2014).

[168] Wei Wu and Ari Arapostathis. "Optimal sensor querying: General markovian and lqg models with controlled observations". In: *IEEE Transactions on Automatic Control* 53.6 (2008), pp. 1392–1405.

[169] Eric P Xing et al. "Distance metric learning with application to clustering with side-information". In: *Advances in neural information processing systems*. 2003, pp. 521–528.

[170] Zheng Yang and Burcin Becerik-Gerber. "Cross-space building occupancy modeling by contextual information based learning". In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM. 2015, pp. 177–186.

[171] Dit-Yan Yeung and Hong Chang. "A kernel approach for semisupervised metric learning". In: *IEEE Transactions on Neural Networks* 18.1 (2007), pp. 141–149.

[172] *Zodiac dataset publication agreement*. `http://www.synergylabs.org/bharath/datasets.html`. [Online; accessed 15-Jun-2017]. 2015.

[173] Han Zou et al. "WinIPS: WiFi-based non-intrusive IPS for online radio map construction". In: *IEEE Conference on Computer Communications Workshops*. 2016, pp. 1081–1082.