Towards optimal strategies for the management of online information and activity: privacy and utility tradeoffs

by

Chandra Sharma

B.E., Tribhuvan University, 2016

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

The unprecedented growth of big-data applications suggests that there is a growing competition in the technological world to collect and harness tremendous amounts of user information. Tech companies and other online service providers are always seeking to enhance the quality of their products and services by collecting massive amounts of information from their user base. The collected data is typically used by the service providers to enhance the utility of the services. For instance, e-commerce services use the information about a user's purchases to recommend new products that may be of interest to the user. Similarly, streaming services use a user's ratings of various movies to recommend new and potentially interesting movies to the user. Unfortunately, the pursuit of utility often entails the loss of user privacy as the collected information often reveals sensitive information about the users, often through correlations not immediately apparent at the surface. This is aggravated by the fact that service providers often share, and even sell, their customers' information with third parties, which makes protecting the users' private information ever so critical.

This dissertation seeks to address two important privacy problems. First, ensuring user privacy is not a trivial problem. At one end, service providers need customers' information to offer customized contents and personalized recommendations. The utility provided to a user is therefore positively correlated with the amount and the accuracy of the information that the user discloses to the service provider. On the other end, the collected information can be subject to inference attacks that reveal various private attributes of the user such as their income, race, political affiliation, and sexual orientation. The privacy of the user is therefore negatively correlated with the amount of disclosed information. The problem, as such, naturally manifests as a privacy-utility tradeoff problem. In this dissertation, we develop models to capture the precise notions of privacy and utility and design privacy

mechanisms that maximize the utility of the disclosed information while limiting the privacy leakage.

The second problem that this dissertation seeks to address is of extreme relevance: given users' tendency to continuously disclose their personal information, as in the case of social media, modeling the privacy leakage over time is paramount to devising privacy mechanisms that limit the accumulated leakage. Further, there is a natural concern regarding the effect of our current online activities on our future privacy. Modeling the problem is extremely intricate as capturing future privacy is not trivial given the inherent uncertainties surrounding the future. In this dissertation, we capture the dynamics of privacy leakage over time using a probabilistic framework. Via experimental evaluations, we demonstrate that there exist multiple promising strategies that a user can utilize to limit their future privacy leakage while maximizing their perceived utility over time.

Towards optimal strategies for the management of online information and activity: privacy and utility tradeoffs

by

Chandra Sharma

B.E., Tribhuvan University, 2016

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
George Amariucai

Copyright

Abstract

The unprecedented growth of big-data applications suggests that there is a growing competition in the technological world to collect and harness tremendous amounts of user information. Tech companies and other online service providers are always seeking to enhance the quality of their products and services by collecting massive amounts of information from their user base. The collected data is typically used by the service providers to enhance the utility of the services. For instance, e-commerce services use the information about a user's purchases to recommend new products that may be of interest to the user. Similarly, streaming services use a user's ratings of various movies to recommend new and potentially interesting movies to the user. Unfortunately, the pursuit of utility often entails the loss of user privacy as the collected information often reveals sensitive information about the users, often through correlations not immediately apparent at the surface. This is aggravated by the fact that service providers often share, and even sell, their customers' information with third parties, which makes protecting the users' private information ever so critical.

This dissertation seeks to address two important privacy problems. First, ensuring user privacy is not a trivial problem. At one end, service providers need customers' information to offer customized contents and personalized recommendations. The utility provided to a user is therefore positively correlated with the amount and the accuracy of the information that the user discloses to the service provider. On the other end, the collected information can be subject to inference attacks that reveal various private attributes of the user such as their income, race, political affiliation, and sexual orientation. The privacy of the user is therefore negatively correlated with the amount of disclosed information. The problem, as such, naturally manifests as a privacy-utility tradeoff problem. In this dissertation, we develop models to capture the precise notions of privacy and utility and design privacy

mechanisms that maximize the utility of the disclosed information while limiting the privacy leakage.

The second problem that this dissertation seeks to address is of extreme relevance: given users' tendency to continuously disclose their personal information, as in the case of social media, modeling the privacy leakage over time is paramount to devising privacy mechanisms that limit the accumulated leakage. Further, there is a natural concern regarding the effect of our current online activities on our future privacy. Modeling the problem is extremely intricate as capturing future privacy is not trivial given the inherent uncertainties surrounding the future. In this dissertation, we capture the dynamics of privacy leakage over time using a probabilistic framework. Via experimental evaluations, we demonstrate that there exist multiple promising strategies that a user can utilize to limit their future privacy leakage while maximizing their perceived utility over time.

Table of Contents

List of Figures x

List of Tables xii

Acknowledgements xiii

Dedication xiv

1 Introduction 1

1.1 Related Works on Privacy-Utility Tradeoff Optimization 4

1.2 Overview of the Subsequent Chapters 6

2 Privacy-Utility Tradeoff in a Static Setting 8

2.1 Introduction 8

2.1.1 Acknowledgements 9

2.2 Problem Setup 9

2.2.1 Problem Setting 9

2.2.2 Problem Formulation 10

2.2.3 Relationship between Mutual Information and Fisher Information . . 17

2.3 Greedy Algorithm 17

2.3.1 Algorithmic Complexity 19

2.4 Experimental Analysis 20

2.4.1 Performance on Synthetic Datasets 20

2.4.2 Evaluation on a real-world dataset 25

3 Privacy-Utility Tradeoff in a Dynamic Setting . . . 31
3.1 Introduction . . . 31
3.2 Problem Description . . . 32
3.2.1 Problem Setup . . . 32
3.2.2 Privacy and Utility Requirements . . . 36
3.3 Formulation as a Markov Decision Process Problem . . . 37
3.4 Sub-optimal Algorithms . . . 42
3.4.1 Value Iteration with Discretization . . . 43
3.4.2 Pessimistic algorithm . . . 43
3.4.3 Optimistic algorithm . . . 45
3.5 Privacy-Utility Tradeoff Under Estimated Privacy Leakage . . . 45
3.6 Simulations . . . 48

4 Privacy-Utility Tradeoff in a Generalized Setting . . . 54
4.1 Introduction . . . 54
4.2 Problem Description . . . 55
4.2.1 Problem Setting . . . 55
4.2.2 Min-Entropy Leakage . . . 56
4.2.3 System Model . . . 57
4.2.4 Problem Formulation . . . 60
4.2.5 From Bayesian Estimation to Kalman Filter . . . 62
4.3 Experimental Evaluations . . . 65

5 Conclusion . . . 71
5.1 Review of the Contributions . . . 72

Bibliography . . . 74

List of Figures

2.1 Visual representation of the privacy and the utility aspects 10

2.2 Comparison of the performances of different algorithms on the two sample datasets. 22

2.3 Performance of the greedy algorithm across different values of δ and γ on the two sample datasets. 23

2.4 Sensitivity of the greedy algorithm to ϵ_0. 24

2.5 Histogram for some of the attributes in the dataset. 26

2.6 Comparison of the actual Mutual information between pairs of features, $(X_p, X_i)\ \forall i = 1 \dots 14$, with the corresponding Gaussian-approximated Mutual Information. 27

2.7 Comparison of the actual Mutual information between pairs of features, $(X_u, X_i)\ \forall i = 1 \dots 14$, with the corresponding Gaussian-approximated Mutual Information. 28

2.8 DFF NN Architecture. 29

3.1 The dynamics of the finite-horizon privacy-utility tradeoff problem. 35

3.2 Comparison of the performances of different algorithms in terms of the probability of privacy outage and the average utility loss for system model 1 ($\delta = 0.3$, $n = 5$ and $M = 1$). 50

3.3 Privacy-utility tradeoff achieved by different strategies for system model 1 ($\delta = 0.3$, $n = 5$ and $M = 1$). 51

3.4 Privacy-utility tradeoff achieved by different strategies for system model 2 ($\delta = 0.3$, $n = 5$ and $M = 1$). 52

3.5 Privacy-utility tradeoff achieved by different strategies for system model 3 ($\delta = 0.3$, $n = 5$ and $M = 1$). 52

4.1 The dynamics of the privacy-utility tradeoff problem over a finite period of time. 59
4.2 Performance of the privacy mechanism across different values of δ for System Model 1 ($n = 5$). 67
4.3 Privacy-utility tradeoff for different covariance matrices of the process noise for System Model 1 ($n = 5$). 68
4.4 Privacy-utility tradeoff due to randomization compared against the privacy-utility tradeoff due to compression for System Model 4 with $Q = 0.1I$ ($n = 5, M = 1$). 69
4.5 Privacy-utility tradeoff due to randomization compared against the privacy-utility tradeoff due to compression for System Model 4 with $Q = 0.001I$ ($n = 5, M = 1$). 69
4.6 Comparison of the performance of different techniques in optimizing the privacy-utility tradeoff across different choices of Q for System Model 4 ($n = 5, M = 1$). 70

List of Tables

2.1 Prediction of the private feature (Income). 30

2.2 Prediction of the utility feature (Employed). 30

3.1 Description of some commonly used symbols and notations 34

Acknowledgments

As I reach the completion of my PhD, I am swamped with mixed feelings. At one end, I am excited about embarking on a new career path while at the other end, I have already started missing my years as a PhD student at Kansas State University. My PhD ride has been extremely joyous which for the most part, has been made possible by my advisor, George Amariucai, who has always been supportive, encouraging and most importantly, the guiding force for making my PhD less stressful. He is, without a doubt, one of the most inspiring people I have ever met.

I am thankful to my PhD committee members, Eugene Vasserman, Arslan Munir, Doina Caragea and Bala Natarajan, for shepherding me in this journey and guiding me toward promising research directions. Their frequent constructive feedback undoubtedly has helped me become a better researcher.

A lot of research that materialized in this dissertation was co-advised by Shuangqing Wei from Louisiana State University whose brilliant insights and sheer knowledge of the field helped break numerous barriers that I stumbled upon during my research. I am truly honored to experience the opportunity to work under his supervision.

Lots of credit goes to Bishwas Mandal who helped me in setting up many of the experiments covered in this dissertation. His support and insights have made this dissertation possible.

I am grateful to my friends and family for their support, especially to my brother, Hemraj Sharma, for believing in me. I do not think I can ever find proper adjectives to articulate the support I have received from my family. I owe all my accomplishments to them.

Lots of love to my wife. This is your achievement as much as it is mine.

Dedication

Dedicated to my father and my mother who would have been extremely proud of this accomplishment.

Chapter 1

Introduction

With the growth of social media and other platforms where the users share their information, the privacy of the published information has been a subject of much interest. Users often disclose their personal information in exchange for some utility. The exact nature of the expected utility differs with the platform on which a user discloses their information–in social media, for instance, the utility received by the user can be associated with the gratification received as a result of approval from other users (via likes and shares, for instance) whereas in online market places, the utility received by the user can be associated with the recommendation of potentially interesting products to the user by the service provider. Users often disclose their personal information either by directly engaging with the service providers, such as in the case of social media and online shopping, or indirectly, and often inadvertently, by simply possessing IoT and other smart devices, such as location trackers. Over time, massive amounts of individual data are acquired by the service providers and can be subject to malicious use, often via seemingly innocuous release to other parties.

The biggest concern with the shared data is the privacy of the data. Users are often oblivious to the actual scale and nature of the information being disclosed to the service providers. The shared information often contains sensitive information about the users such as their current location, income, religious beliefs, political affiliation and sexual orientation. Further, service providers often share, and even sell their customers' information to third

parties, which makes protecting the users' private information ever so critical. In light of this, there have been increasing efforts to devise privacy-preserving mechanisms that make it difficult for external entities to infer a user's private information from the disclosed data. Such mechanisms protect the user's information often via means of randomization [1, 2, 3, 4, 5, 6, 7, 8] and/or compression [9, 10, 11, 12, 13] before disclosure. Unfortunately, the resulting distortion entails a loss of some useful information from the disclosed data which can otherwise be utilized by the service provider to provide a customized service to the user. The challenge, therefore, is to find an optimal tradeoff between protecting the user's privacy and enhancing the utility of the disclosed data.

An important first step toward solving this problem is to sufficiently capture the intuitive understanding of privacy and utility in the problem formulation. A common information disclosure pattern typically involves a randomization or a compression mechanism which takes some input data and produces a perturbed output, which makes it difficult to extract sensitive information from the output data, while maintaining the perceived utility. In essence, any privacy metric should nontrivially relate between the disclosed data and the sensitive information. In the literature, such relation is often captured using privacy metrics such as *Differential Privacy* [14, 15, 16], *Correlated Differential Privacy* [17, 18], *Mutual Information* [2, 19, 20], *Changes in min-entropy* [14, 21, 22], *Fisher Information* [23, 24, 25], *Maximal Leakage* [26, 27] and *Maximal Correlation* [19, 28, 29]. The choice of a particular privacy metric depends on the use case; if it is desirable to hide the identity of users in a database, metrics such as *Differential Privacy* are adequate whereas if the goal is to hide specific private attributes of a user, information-theoretic metrics such as *Mutual Information* or *Maximal Leakage* are more useful.

As with the case of privacy, it is also critical to mathematically capture the intuitive understanding of utility. Loosely speaking, utility can be thought of as a meaningful use of the shared information. The subject of the utility could either be the person who shares the information or the party that uses the shared information, or both. In any case, the better the information is used, the higher the utility, which makes utility highly reliant on the quality of the information. Any randomization or compression mechanism, therefore,

essentially decreases the utility of the information. Utility is commonly quantified using a distortion function (see, for instance, [2], [19], [20] and [30]) which captures an overall loss of information between the data before and after randomization. Such a formulation of utility makes an important distinction from privacy: privacy is considered an *individual* concept whereas utility is considered an *aggregate* concept [31]. In such models, utility is associated with the entire data set of a user and consequently, utility loss due to randomization is often overestimated. However, from a more practical viewpoint, the overall utility can often be associated with only a small subset of user attributes. A utility model where the overall utility of a user is associated with a specific and small subset of their attributes not only facilitates a more straightforward privacy-utility tradeoff formulation, but also extends the notion of subjective utility. Furthermore, in existing models, the only constraint on the utility is the maximum acceptable utility loss; such models can sometimes sacrifice significant utility (up to the specified limit imposed by the constraint) while only attaining a minimal privacy gain, which is often undesirable. It is hence necessary to incorporate an additional constraint into the problem formulation, that imposes a limit on the loss in utility per unit gain in privacy due to randomization – in effect, a constraint on the utility's gradient with respect to privacy.

Just as it is important to capture the precise notions of privacy and utility, it is equally important to consider the privacy leakage over time resulting from a continuous and correlated disclosure of information. While existing works try to capture different notions of privacy and derive theoretical bounds on the privacy leakage, they do so in a static setting which either assumes that a user discloses their information only once, or it treats each disclosure of the user's information independently of the previous disclosures. This model of privacy falls short in many practical settings in which users continuously disclose their personal information over time (as in social media) and each disclosure is temporally correlated with the previous disclosures. Therefore, static privacy models are not only incomplete but also inaccurate to be applied in dynamic settings.

The goal of this dissertation is threefold: first, we seek to model the privacy-utility tradeoff problem in a static setting capturing the precise notions of privacy and utility. Second, we aim to develop a dynamic model of privacy which considers the impact of the past, the

present and the future disclosures on future privacy. And lastly, we aim to develop a general privacy model which considers the accumulated privacy leakage at all finite future time steps resulting from all preceding disclosures. Using a mix of compression and randomization mechanisms, we aim to design robust privacy mechanisms that offer long-term privacy guarantees while optimizing the cumulative utility of the disclosed information.

1.1 Related Works on Privacy-Utility Tradeoff Optimization

The problem of optimizing the privacy-utility tradeoff in a static setting has been widely studied under different notions of privacy but similar interpretations of utility. Existing works mostly focus on precisely defining privacy based on the context and/or deriving bounds on the privacy leakage. Li et al. [31] formulate privacy loss as the information gained about the sensitive values of individuals and utility loss as the information lost about the sensitive values of the whole population. *JS-divergence* and *KL-divergence* measures are used to quantify the privacy loss and the utility loss respectively. Similarly, Makhdoumi et al. [19] define privacy loss as the information leaked about some private data from a randomized and disclosed non-private data and quantify it using two metrics: mutual information and maximal leakage. They define utility loss as the average distortion between the perturbed and the original data and quantify it using a general distortion measure. Similar formulations for utility can be found in [2, 14, 20] and [32].

A more general formulation for privacy can be found in [33] where privacy is captured with a generalized cost function with the cost gain measuring the amount of information obtained about the private data after observing the disclosed non-private data. Two privacy metrics, *average information leakage* and *maximum information leakage* are studied under the self-information cost function. Similar to other works, utility is quantified as an average distortion in the disclosed data. The same framework lays the foundation for [34] where the log-loss function (self-information) is used for both privacy and utility metrics. Here,

the privacy leakage is measured as the mutual information between the private data and the disclosed data and the average distortion (utility) as the mutual information between the non-private data and the disclosed data. The privacy-utility trade-off problem is then formulated as an optimization problem that minimizes the mutual information between the private and the disclosed data over all feasible randomization mechanisms that guarantee the desired distortion level. The problem is referred to as the *Privacy Funnel* and is shown to be non-convex.

From the viewpoint of dynamic privacy, there are but a few works that model the continuous disclosure of a user's information and capture the temporal correlation between subsequent disclosures. In [30], the authors investigate the privacy leakage resulting from a continuous release of a time-series data that is correlated, both spatially and temporally, with a user's sensitive data (also considered to be time-series but non-disclosable). The privacy mechanism seeks to distort the time series data before each disclosure to impede inference attacks on the sensitive data while preserving the utility of the disclosed data. This model seeks to limit the privacy leakage at the present time using the information from the past disclosures and by carefully regulating the current disclosure (other similar models can be found in [35, 36]). In contrast, our dynamic privacy models seek to limit the privacy leakage in the future using the information from the past disclosures and by carefully regulating the present and all future disclosures. Our models are more general and easily simplify to a model similar to that in [30] under a particular instantiation (namely, $n = k$ where n represents the finite time horizon and k represents the current time step).

Recently, researchers have started to explore the privacy issues in a dynamic setting with regard to future privacy leakage. In [37], the authors investigate the privacy issues related with the continuous disclosure of sensor measurements containing some private and some public information. They formulate the problem as a filtering problem in which they seek to find the optimal compression that maximizes the variance of the estimation error associated with the estimation of the private data while minimizing the variance of the estimation error associated with the estimation of the public data. Under their model, they investigate the privacy-utility tradeoff at the current time step and one time step into the future. The

same work is further extended in [13] where the authors investigate the tradeoff multiple time steps into the future. In their formulation, they make predictions about the system's future state using the observations available up until the current time step. This resembles a setting in which a user's past and present actions are considered to estimate their future privacy leakage. However, the impact of the user's future actions are discounted in making the prediction; therefore, while this model can be useful, it is not quite complete as it does not accurately reflect the actual future privacy leakage. In contrast, our dynamic privacy models explicitly account for a user's past, present as well as future actions in evaluating their future privacy leakage.

1.2 Overview of the Subsequent Chapters

Chapter 2 considers the privacy-utility tradeoff in a static setting. In this chapter, we consider the privacy implications of a single online disclosure of personal information. We consider a user who desires to share her personal information on an online platform in hope of deriving some utility–the problem of interest is minimizing the potential privacy leakage resulting from the disclosure. In this regard, we design a privacy mechanism that randomizes the user's data before disclosure and produces a perturbed output–the goal of the privacy mechanism is to make it difficult to extract sensitive information from the disclosed data while maintaining the perceived utility of the data. To facilitate the development of precise privacy mechanisms, we introduce a novel model of utility where the utility of the user is associated with a small subset of the user attributes, referred to as utility attributes. We also consider a more restrictive but pragmatic constraint that captures the acceptable loss of utility per unit gain in privacy due to the privacy mechanism. Finally, we present a heuristic greedy algorithm with polynomial time and space complexity to solve the tradeoff problem and demonstrate its efficacy with synthetic and real-world performance tests.

In Chapter 3, we consider a dynamic setting in which users continuously disclose their personal information over time resulting in an accumulated leakage of their sensitive information. In this setting, we model a privacy-aware user who seeks to cautiously disclose her

personal information to a data analyst over a finite time horizon. The objective of the user is to maximize her instantaneous utilities, which the data analyst provides by extracting useful information from the disclosed information at each time step, while limiting the amount of leakage about her sensitive information at the end of the finite time horizon. We consider a simple privacy mechanism that involves compressing the user's data before each disclosure to minimize the privacy leakage at a future time. We then formulate a novel privacy-utility tradeoff problem capturing the dynamics of privacy leakage over a finite time period and investigate different strategies that allow the user to maximize her net utility subject to the specified privacy requirements. We discuss challenges associated with finding optimal strategies for real world problems and motivate sub-optimal algorithms to solve the tradeoff problem. Further, we formulate a simpler dynamic privacy problem that is computationally less intensive to solve but conserves the essence of the original problem. Finally, we evaluate the performance of the sub-optimal algorithms on synthetic datasets and demonstrate that despite being sub-optimal, the proposed algorithms perform extremely well in achieving a good privacy-utility tradeoff.

Chapter 4 extends the dynamic setting introduced in Chapter 3 to consider privacy leakage not just at the end of a finite time horizon but at all finite future time steps. As such, we consider a user who seeks to cautiously disclose her personal information with the goal of maximizing her perceived cumulative utility while limiting the accumulated privacy leakage over a finite time period in the future. To capture a real world setting, we assume that the user is interested in limiting her privacy leakage only for a finite period of time and that after this time period, the user deems her privacy less important. In this chapter, we develop a privacy mechanism using a mix of compression and randomization techniques. The privacy and utility models introduced in this chapter are fundamentally a generalization of the privacy and the utility models introduced in Chapters 2 and 3. Via experimental evaluations, we show that our privacy mechanism is extremely effective in optimizing the privacy-utility tradeoff in various dynamic settings.

Finally, in Chapter 5, we summarize the motivation, scope and contributions of this dissertation.

Chapter 2

Privacy-Utility Tradeoff in a Static Setting

2.1 Introduction

This chapter considers the problem of minimizing the privacy leakage resulting from a single online disclosure of personal information. Due to the recent popularity of online social networks, coupled with people's propensity to disclose personal information in an effort to achieve certain gratifications, the problem of navigating the tradeoff between privacy and utility has attracted a lot of recent interest. A critical prerequisite to solving the problem is to appropriately capture the privacy and the utility aspects in the problem formulation. Most of the existing works focus on the notion of privacy, while utility loss is often treated as the undesirable but necessary distortion of the true data, introduced by the privacy mechanism. By contrast, we are interested in modeling utility differently, by associating it with specific attributes of a user, just like privacy is associated with specific private attributes in the literature. Such model of utility facilitates a better and more precise privacy mechanism. This chapter introduces a new formulation of the privacy-utility tradeoff problem centred on a more practical notion of utility. Our problem formulation also incorporates a practical constraint on acceptable loss in utility per unit gain in privacy, which allows users

to customize the privacy mechanisms in order to account for the relative values that each user associates with their own privacy and utility.

2.1.1 Acknowledgements

This work was supported in part by the U.S. National Science Foundation under Grants No. CNS-1527579, CNS-1619201. A part of this work is published in [38].

2.2 Problem Setup

2.2.1 Problem Setting

We consider a setting where a user shares some personal information, for instance, on social media, in hope of deriving some utility. In particular, we are interested in the privacy leakage and the utility received resulting from a single disclosure. In this setting, we first characterize each user by a set of features. We assume that each user has some *private features* represented by the random vector X_p and some *utility features* represented by the random vector X_u. Examples of private features include political affiliation, sexual orientation etc., while examples of utility features include how the user would rate a certain movie, number of likes/dislikes on the user's social media post etc. For generality, we do not require that private and utility features be distinct. The utility received by a user is directly related to her utility features. For instance, based on how the user rates previously watched movies on a streaming platform, the streaming service can feed her new potentially interesting movies. Next, we denote by X all the other features that are non-private and non-utility, and assume that X is correlated with X_p and X_u with the goal to release a perturbed version of X, say Y, that helps gain reasonably large information about X_u but only minimal information about X_p. Notice that we are considering a setting where a user does not disclose X_u, but rather aims to convey the information in X_u by disclosing Y. There are at least two reasons for this: first, the exact value of X_u may not be known to the user (think recommender systems) and second, it may be in the best interest of the user to not disclose X_u. The latter case, for

instance, is common in social media, where disclosing certain information in hope of gaining gratification may qualify an individual as a narcissist [39, 40, 41]. Instead, a savvy user may hope to subtly suggest that information to their friends. With this setup, we relate privacy inversely to the information gained about X_p from Y and utility directly to the information gained about X_u from Y. Note that $(X_p, X_u) \rightarrow X \rightarrow Y$ form a Markov chain. The privacy and the utility aspects of the problem are captured visually in Figure 2.1.

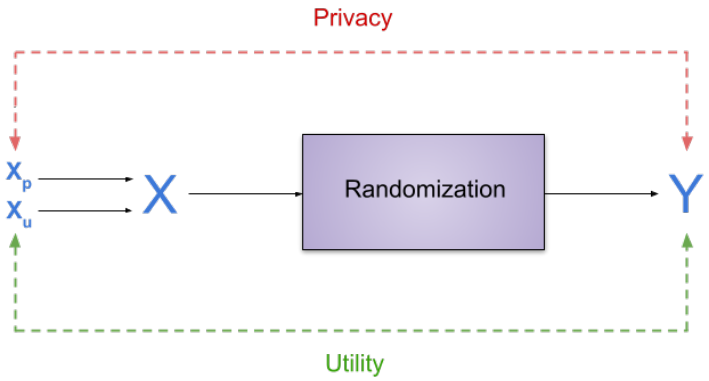


Figure 2.1: Visual representation of the privacy and the utility aspects

2.2.2 Problem Formulation

For our problem formulation, we consider a single user setup. Let $\{x_1, x_2, \cdots x_n\}$ be n random variables representing the actual features of the user. Note that each x_i, where $1 \leq i \leq n$, represents a unique feature. In particular, we denote the n features by a single random vector X. We also assume that the random variables are correlated with each other.

In addition to the n features, let the user have two, potentially non-disjoint, sets of special features: a set containing n_p private features and a set containing n_u utility features denoted by the random vectors X_p and X_u respectively. Note that $X_p \cap X = \phi$ and $X_u \cap X = \phi$. However, we assume that X_p and X_u are correlated with X. Also, let $X \cup X_p = [x_1, x_2, \cdots, x_n, x_1^p, x_2^p, \cdots, x_{n_p}^p]^T = [X^T, X_p^T]^T$ and $X \cup X_u = [x_1, x_2, \cdots, x_n, x_1^u, x_2^u, \cdots, x_{n_u}^u]^T = [X^T, X_u^T]^T$.

Let, for any random vector V, Σ_V represent its covariance matrix. For this problem, we assume that X, X_u, X_p are jointly Gaussian. The Gaussian assumption facilitates the

development of much simpler yet practical models and is frequently used in the literature [20, 42, 43, 44]. Now, let Y represent a perturbed version of X. As a randomization mechanism, we assume the addition of Gaussian white noise, i.e. $Y = X + N$, where N represents a random vector with independent components, following a multivariate Gaussian distribution with zero mean and diagonal covariance matrix. Also, let $Y \cup X_p$ and $Y \cup X_u$ denote the perturbed version of $X \cup X_p$ and $X \cup X_u$ respectively. Similarly, let 0^m denote a vector of m zeros. Observe that if $\hat{N} = [N^T, 0^{n_p}]^T$ and $\bar{N} = [N^T, 0^{n_u}]^T$, then $Y \cup X_p = X \cup X_p + \hat{N}$ and $Y \cup X_u = X \cup X_u + \bar{N}$. Observe that $\Sigma_Y = \Sigma_X + \Sigma_N$, $\Sigma_{Y \cup X_p} = \Sigma_{X \cup X_p} + \Sigma_{\hat{N}}$ and $\Sigma_{Y \cup X_u} = \Sigma_{X \cup X_u} + \Sigma_{\bar{N}}$.

The natural next step in formulating the problem is to quantify the privacy and utility aspects of the problem. While there are several metrics that can be used and the practicality of one over the other can be argued upon, we quantify both privacy and utility using the mutual information metric. Mutual information as a privacy measure has been used in [2, 19, 20] and as a utility measure has been used in [45, 46]. The mutual information between X_p and Y constitutes the privacy aspect and the mutual information between X_u and Y constitutes the utility aspect of the problem. Accordingly, the privacy-utility tradeoff problem can be formulated as the following optimization problem:

$$\underset{\Sigma_N}{\text{minimize}} \;\; I(X_p; Y)$$

subject to:

$$I(X_u; X) - I(X_u; Y) \leq \widehat{\delta}, \tag{2.1}$$

$$\frac{\Delta I(X_u; Y)}{\Delta I(X_p; Y)} \leq \gamma, \tag{2.2}$$

$$\Sigma_N \geq 0, \tag{2.3}$$

where $I(U; V)$ represents the mutual information between random vectors U and V. The mutual information between two random vectors is a scalar quantity which measures the total average (over all vector components) reduction in uncertainty about a random vector due to the observation of the other random vector.

Put $\delta = I(X_u; X) - \widehat{\delta}$. The first constraint can then be re-written as $I(X_u; Y) \geq \delta$. The first constraint enforces a requirement that the *utility loss* due to randomization must be no more than the preset baseline, $\widehat{\delta}$, or equivalently, the end utility (that can be extracted from Y) must be no less than δ. The parameter δ determines the tradeoff between the privacy leakage and utility loss due to the privacy mechanism. The second constraint enforces a requirement that the ratio of the end utility loss to the end privacy gain must be no more than the preset threshold, γ. The third constraint requires that the diagonal matrix, Σ_N, must be positive semi-definite–put simply, the variance of all added noises must be non-negative. Notice that δ and γ are user-customizable parameters. For brevity, the first constraint will be referred to as the δ-constraint, while the second as the γ-constraint.

Now,

$$\begin{aligned} I(X_p; Y) &= H(X_p) + H(Y) - H(X_p, Y) \\ &= \frac{1}{2} log\big((2\pi e)^{n_p} |\Sigma_{X_p}|\big) + \frac{1}{2} log\big((2\pi e)^{n} |\Sigma_{Y}|\big) - \frac{1}{2} log\big((2\pi e)^{n+n_p} |\Sigma_{Y \cup X_p}|\big) \\ &= \frac{1}{2} (log|\Sigma_{X_p}| + log|\Sigma_Y| - log|\Sigma_{Y \cup X_p}|). \end{aligned}$$

Similarly,

$$\begin{aligned} I(X_u; Y) &= \frac{1}{2} (log|\Sigma_{X_u}| + log|\Sigma_Y| - log|\Sigma_{Y \cup X_u}|), \\ I(X_u; X) &= \frac{1}{2} (log|\Sigma_{X_u}| + log|\Sigma_X| - log|\Sigma_{X \cup X_u}|). \end{aligned}$$

Before we discuss the intricacies of solving the problem, we will look into another interesting metric that can be used to quantify utility: the *Fisher Information*, which measures how much information about a parameter can be obtained by observing a random variable where the probability of the random variable depends on the parameter. The lower bound on the variance of any unbiased estimator of the parameter (formally, referred to as Cramer-Rao bound) is given by the inverse of the Fisher information. Notice that our parameter of

interest is the vector of utility attributes and we are interested in measuring the variance of any estimator. In this setting, Fisher information fits perfectly which allows us to formulate the privacy-utility tradeoff problem as follows:

$$\underset{\Sigma_N}{\text{minimize}} \;\; I(X_p; Y)$$

subject to:

$$\frac{1}{Tr(\mathcal{I}^{-1}(X_u))} \geq \delta, \tag{2.4}$$

$$\Delta I(X_p; Y) \cdot \Delta Tr(\mathcal{I}^{-1}(X_u)) \geq \gamma, \tag{2.5}$$

$$\Sigma_N \geq 0. \tag{2.6}$$

where $Tr(W)$ represents the trace of the matrix W while $\mathcal{I}(X_u)$ represents the Fisher Information about X_u and is defined as

$$\mathcal{I}(X_u) = -\mathrm{E}[\ell''(Y|X_u)], \tag{2.7}$$

where $\ell''(Y|X_u)$ represents the second partial derivative of the log-likelihood function, $\ell(Y|X_u)$, with respect to X_u.

The constraint in (2.4) reveals interesting facets of the utility formulation using Fisher information. First, note that for a vector X_u, $\mathcal{I}(X_u)$ is a matrix, called the *Fisher Information Matrix*. Recall that the inverse of the scalar Fisher information gives the lower bound on the variance of any unbiased estimator of a parameter. Similarly, the inverse of the Fisher information matrix gives the variances and covariances of the n_u estimators of the n_u utility parameters. Notice that the trace is chosen as a scalar function on $\mathcal{I}^{-1}(X_u)$. Other common choices include the determinant and the largest eigenvalue [47]. Again, we do not insist on using one particular scalar function, rather, we focus on capturing more important characteristics of the problem.

We make the same assumptions about the distribution of X, X_p and X_u we made earlier

and follow the same notations for all relevant covariance matrices. In addition, we denote by M_u and $M_{Y \cup X_u}$ the mean vectors of X_u and $Y \cup X_u$, respectively. Note that all differentiations, henceforth, are with respect to X_u unless otherwise stated.

Theorem 1. *For $i, j \in [1 \,..\, n_u]$, let $a_{i,j}$ denote the elements in the i^{th} row and j^{th} column of the inverse matrix, $\Sigma_{X_u}^{-1}$, and $b_{n+i,n+j}$ denote the elements in the $(n+i)^{th}$ row and $(n+j)^{th}$ column of the inverse matrix, $\Sigma_{Y \cup X_u}^{-1}$. Then,*

$$\mathcal{I}(X_u) = -\mathcal{H}, \tag{2.8}$$

where $\mathcal{H}$ is a Hessian matrix of dimension $n_u \times n_u$ with elements given by

$$\mathcal{H}_{i,j} = \frac{1}{2} \cdot \begin{cases} 2(a_{i,i} - b_{n+i,n+i}) & i = j \\ a_{i,j} + a_{j,i} - b_{n+i,n+j} - b_{n+j,n+i} & i \neq j \end{cases}$$

Proof. By definition, we have

$$\mathcal{I}(X_u) = -\mathrm{E}[\ell''(Y|X_u)], \tag{2.9}$$

Here,

$$\begin{aligned} \ell(Y|X_u) &= \log f(Y|X_u) \\ &= \log f(Y, X_u) - \log f(X_u). \end{aligned}$$

Differentiating both sides, we get

$$\ell'(Y|X_u) = \frac{f'(Y, X_u)}{f(Y, X_u)} - \frac{f'(X_u)}{f(X_u)}. \tag{2.10}$$

We have

$$f(X_u) = \frac{1}{(2\pi)^{n_u/2}|\Sigma_{X_u}|^{1/2}} \cdot \exp\Big(-\frac{1}{2}(X_u - M_u)^T \Sigma_{X_u}^{-1}(X_u - M_u)\Big).$$

Differentiating both sides,

$$\begin{aligned} f'(X_u) &= \frac{1}{(2\pi)^{n_u/2}|\Sigma_{X_u}|^{1/2}} \cdot \exp\Big(-\frac{1}{2}(X_u - M_u)^T \Sigma_{X_u}^{-1}(X_u - M_u)\Big) \\ &\quad \cdot \frac{\partial}{\partial X_u}\Big(-\frac{1}{2}(X_u - M_u)^T \Sigma_{X_u}^{-1}(X_u - M_u)\Big) \\ &= f(X_u) \cdot \frac{\partial}{\partial X_u}\Big(-\frac{1}{2}(X_u - M_u)^T \Sigma_{X_u}^{-1}(X_u - M_u)\Big). \end{aligned}$$

Similarly,

$$\begin{aligned} f(Y, X_u) = \ & \frac{1}{(2\pi)^{(n+n_u)/2}|\Sigma_{Y\cup X_u}|^{1/2}} \\ & \cdot \exp\Big(-\frac{1}{2}(Y \cup X_u - M_{Y\cup X_u})^T \cdot \Sigma_{Y\cup X_u}^{-1}(Y \cup X_u - M_{Y\cup X_u})\Big). \end{aligned}$$

Differentiating both sides,

$$f'(Y, X_u) = f(Y, X_u) \cdot \frac{\partial}{\partial X_u}\Big(-\frac{1}{2}(Y \cup X_u - M_{Y\cup X_u})^T \cdot \Sigma_{Y\cup X_u}^{-1}(Y \cup X_u - M_{Y\cup X_u})\Big).$$

Now, (2.10) can be written as

$$\begin{aligned} \ell'(Y|X_u) = & \frac{\partial}{\partial X_u}\Big(-\frac{1}{2}(Y \cup X_u - M_{Y\cup X_u})^T \cdot \Sigma_{Y\cup X_u}^{-1}(Y \cup X_u - M_{Y\cup X_u})\Big) \\ & - \frac{\partial}{\partial X_u}\Big(-\frac{1}{2}(X_u - M_u)^T \Sigma_{X_u}^{-1}(X_u - M_u)\Big). \end{aligned}$$

Differentiating both sides,

$$\ell''(Y|X_u) = \frac{\partial^2}{\partial X_u^2}\Big(-\frac{1}{2}(Y \cup X_u - M_{Y\cup X_u})^T \cdot \Sigma_{Y\cup X_u}^{-1}(Y \cup X_u - M_{Y\cup X_u})\Big) \\ - \frac{\partial^2}{\partial X_u^2}\Big(-\frac{1}{2}(X_u - M_u)^T\Sigma_{X_u}^{-1}(X_u - M_u)\Big). \tag{2.11}$$

For $i, j \in [1 \,..\, n_u]$, if $a_{i,j}$ denote the elements in the i^{th} row and j^{th} column of the inverse matrix, $\Sigma_{X_u}^{-1}$, and $b_{n+i,n+j}$ denote the elements in the $(n+i)^{th}$ row and $(n+j)^{th}$ column of the inverse matrix, $\Sigma_{Y\cup X_u}^{-1}$, then (2.11) simplifies to a Hessian matrix, $\mathcal{H}$, with dimension $n_u \times n_u$ where

$$\mathcal{H}_{i,j} = \frac{1}{2} \cdot \begin{cases} 2(a_{i,i} - b_{n+i,n+i}) & i = j \\ a_{i,j} + a_{j,i} - b_{n+i,n+j} - b_{n+j,n+i} & i \neq j \end{cases}$$

Therefore, (2.9) can be written as

$$\mathcal{I}(X_u) = -\mathrm{E}(\mathcal{H}) = -\mathcal{H}, \tag{2.12}$$

where the negative expectation of the Hessian is commonly referred to as the *Fisher Information Matrix*. □

A special case of (2.8) is when there is a single utility feature ($n_u = 1$). In this case, the Hessian matrix is a scalar with the first and only element given by

$$\mathcal{H}_{1,1} = \frac{1}{\sigma_{x_u}^2} - b_{n+1,n+1}.$$

Then, from (2.8),

$$\mathcal{I}(X_u) = b_{n+1,n+1} - \frac{1}{\sigma_{x_u}^2}, \tag{2.13}$$

where $\sigma_{x_u}^2$ represents the variance of the utility feature.

2.2.3 Relationship between Mutual Information and Fisher Information

Expanding on the expression given in (2.13), we now establish a relationship between Mutual information and Fisher information for $n_u = 1$. First, observe that

$$b_{n+1,n+1} = \frac{|\Sigma_Y|}{|\Sigma_{Y \cup X_u}|}.$$

Multiplying both sides by $\sigma^2_{x_u}$ and computing $\frac{1}{2}\log$ on both sides, we get

$$\begin{aligned} \frac{1}{2}\log\left(b_{n+1,n+1} \cdot \sigma^2_{x_u}\right) &= \frac{1}{2}\log\left(\frac{\sigma^2_{x_u} \cdot |\Sigma_Y|}{|\Sigma_{Y \cup X_u}|}\right) \\ &= I(X_u; Y). \end{aligned} \tag{2.14}$$

Then, from (2.13) and (2.14),

$$I(X_u; Y) = \frac{1}{2} log(\sigma^2_{x_u} \cdot \mathcal{I}(X_u) + 1). \tag{2.15}$$

Observe that the expression in (2.15) is analogous to the expression for *channel capacity* in communication systems, where $I(X_u; Y)$ corresponds to the channel capacity, $\sigma^2_{x_u}$ corresponds to the signal power and $\frac{1}{\mathcal{I}(X_u)}$ corresponds to the noise power. The relationship in (2.15) highlights that for the special case where X_u is a scalar, any problem formulation using Fisher information as the utility metric is analogous to the problem formulation using Mutual information as the utility metric.

2.3 Greedy Algorithm

The optimization problem formulated in Section 2.2.2 cannot be readily solved using existing methods due to the additional $\gamma-$constraint. Furthermore, the convexity of the objective function is not known, which makes the existing convex optimization techniques inapplicable.

Analytical methods based on KKT conditions that restrictively work on certain non-convex problems also fall short for two reasons: first, it is unknown whether strong duality holds for the problem and second, these methods often do not scale well for higher dimensional problems. These restrictions motivate us to develop a custom heuristic algorithm to solve the problem.

We take a greedy iterative approach to solving the problem: at each step, we add a small amount, say $\Delta\theta$, to the variance of the noise to be applied to one of the variables $x_1, x_2, \cdots, x_n$. The selection of the variable to add noise to is determined by the *gain factor* which is defined as the ratio of the *privacy gain* to the *utility loss* due to the increased noise variance. Essentially, in each iteration, we select the variable with the highest gain factor, add $\Delta\theta$ to the noise variance to be applied to that variable, and test for the δ and the γ constraints. If both constraints are slack, we commit to the noise variance addition, else we reduce $\Delta\theta$ by a factor of 2 and proceed to the next iteration without committing. We stop when $\Delta\theta$ becomes less than or equal to a preset value ϵ.

An interesting situation arises when a variable yields the highest gain factor among all variables but only achieves negligible privacy gain (as a result of a small utility loss). Clearly, it is not worthwhile to add any more noise to the variable. This is called a *saturation* phase and the variable is said to be *saturated*. If a variable is saturated, we ignore the variable for the current iteration and continue with the other variables. If all variables are saturated in the same iteration, this is referred to as *total saturation*, in which case, we stop.

In what follows, we summarize our approach by presenting the heuristic greedy algorithm. For simplicity, let $Y_{i+\Delta\theta}$ represent the vector that has the same elements as Y but with $\Delta\theta$ added to the variance of the noise applied to the i^{th} component.

Greedy algorithm for the privacy-utility tradeoff problem:

1. **Initialization**. Initialize $\Delta\theta$ to a small positive value, set Y = X.

2. **Evaluation**. For each variable, i $(1 \leq i \leq n)$, compute
   - privacy_gain$(i) = I(X_p; Y) - I(X_p; Y_{i+\Delta\Theta})$
   - utility_loss$(i) = I(X_u; Y) - I(X_u; Y_{i+\Delta\Theta})$
   - gain_factor(i) = privacy_gain(i) / utility_loss(i)
   - If privacy_gain$(i) < \epsilon_0$, set gain_factor$(i) = -1$

   (This corresponds to the saturation phase)

3. **Selection**. Select the variable with the highest gain_factor. Let j be the index of this variable.

4. **Stopping criteria**. If gain_factor(j) <= 0, stop.

   (If the highest gain factor <= 0, all other gain factors are also <= 0 which implies total saturation)

5. **Update**. If $I(X_u; Y_{j+\Delta\theta}) \geq \delta$ and gain_factor$(j) \geq \gamma$, set $Y = Y_{j+\Delta\theta}$. Else, set $\Delta\theta = \Delta\theta/2$.

6. **Repeat**. If $\Delta\theta < \epsilon$, stop. Else, go to 2.

Although in the above algorithm we have quantified both privacy gain and utility loss using Mutual information, it is straightforward to modify the algorithm and quantify utility loss using Fisher information. However, note that the values of δ and γ need to be adjusted for the new metric. For scalar X_u, the new values of δ and γ can be determined using the relationship in (2.15).

2.3.1 Algorithmic Complexity

There are two relatively computationally intensive parts of the algorithm:

1. Computing the Mutual information

2. Determining the variable with the highest gain factor

Computing the Mutual information requires computing the determinants of the covariance matrices. The dimension of the largest matrix is $(n+1) \times (n+1)$, so computing its determinant requires roughly $O(n^3)$ time. Similarly, determining the variable with the highest gain factor requires sorting which takes, on average, $O(n \log(n))$ time. Overall, the time-complexity of the algorithm is $O(n^3)$.

In regard to the space complexity, there are two relatively space intensive parts of the algorithm:

1. Storing the covariance matrices

2. Storing the gain factors of n variables

The space requirement for the covariance matrices is in the order of $O(n^2)$. Similarly, the space requirements for storing the gain factors of n variables is in $O(n)$. The overall space-complexity of the algorithm is, therefore, $O(n^2)$.

2.4 Experimental Analysis

2.4.1 Performance on Synthetic Datasets

In this section, we analyze our model by running the greedy algorithm on synthetic datasets. All datasets are sampled from a multivariate normal distribution and reflect various features of a user. Sample dataset 1 consists of 4 features, including a private and a utility feature, whereas sample dataset 2 consists of 8 features. The covariance matrices for the sample datasets and the corresponding privacy-utility trade-off graphs are presented below. For each covariance matrix, the penultimate row (and column) corresponds to the private feature and the last row (and column) corresponds to the utility feature.

Sample dataset 1:

$$\Sigma = \begin{bmatrix} 138.27 & 165.66 & 26.36 & 11.28 \\ 165.66 & 240.07 & 43.86 & 6.84 \\ 26.36 & 43.86 & 8.76 & 0.01 \\ 11.28 & 6.84 & 0.01 & 2.26 \end{bmatrix}$$

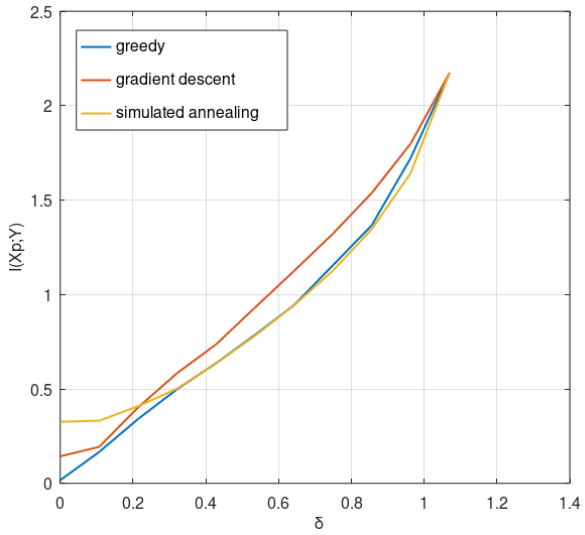
Sample dataset 2:

$$\Sigma = \begin{bmatrix} 66.4 & 57.4 & 83.9 & 80.0 & 0.1 & 121.4 & 9.3 & 11.2 \\ 57.4 & 229.2 & 146.9 & 232.6 & 0.0 & 69.3 & 45.1 & 2.4 \\ 83.9 & 146.9 & 142.9 & 169.8 & 0.1 & 140.2 & 27.2 & 11.3 \\ 80.0 & 232.6 & 169.8 & 247.4 & 0.1 & 114.4 & 45.2 & 6.9 \\ 0.1 & 0.0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.0 & 0.0 \\ 121.4 & 69.3 & 140.2 & 114.4 & 0.1 & 233.3 & 9.4 & 22.4 \\ 9.3 & 45.1 & 27.2 & 45.2 & 0.0 & 9.4 & 9.0 & 0.0 \\ 11.2 & 2.4 & 11.3 & 6.9 & 0.0 & 22.4 & 0.0 & 2.2 \end{bmatrix}$$

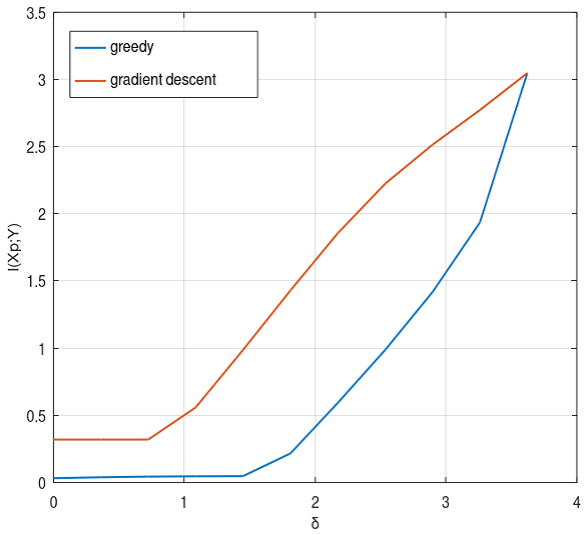
Figure 2.2a compares the performance (in terms of minimizing the objective function) of the greedy algorithm against the gradient descent and simulated annealing algorithms for sample dataset 1.[1] Similarly, Figure 2.2b compares the performance of the greedy algorithm against the gradient descent algorithm for sample dataset 2.[2] As the gradient descent algorithm is not compatible with the γ-constraint, for the sake of comparison, we set $\gamma = 0$ for our simulations (which is equivalent to omitting the γ-constraint). Observe in the two figures

[1] A simple gradient descent algorithm that numerically approximates the gradient was used for the simulation. Simulations of the gradient descent and the simulated annealing algorithms involved the addition of quadratic loss functions to the objective function to transform the constrained optimization problem into an equivalent unconstrained optimization problem.

[2] The complexity of designing a robust neighbor function for the higher dimensional problem hindered us from running the simulated annealing algorithm on sample dataset 2.



(a) Performance of different algorithms on sample dataset 1; the graph shows the plot of minimum $I(X_p;Y)$ for different values of δ with $\gamma = 0$ ($\epsilon_0 = 10^{-6}$).



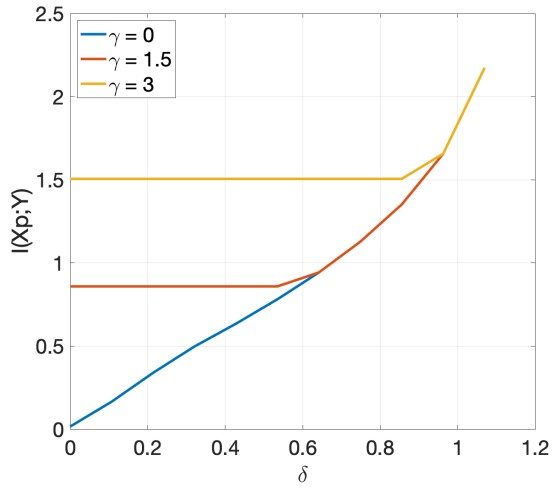
(b) Performance of different algorithms on sample dataset 2; the graph shows the plot of minimum $I(X_p;Y)$ for different values of δ with $\gamma = 0$ ($\epsilon_0 = 10^{-6}$).

Figure 2.2: Comparison of the performances of different algorithms on the two sample datasets.

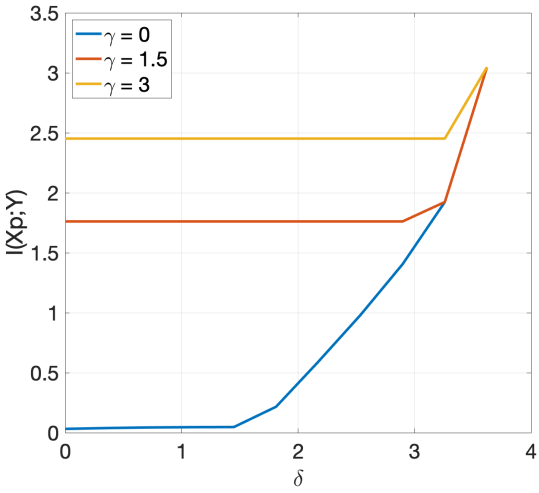
that our greedy algorithm consistently performs better than the gradient descent algorithm across different values of delta. Where applicable, the results are comparable to that of the simulated annealing algorithm. We note that the figures are not meant to highlight the superiority of our algorithm[3] but to show that our algorithm converges to a reasonably good solution. We, stress that the added $\gamma-$constraint in the original problem often mandates the use of our algorithm (or a modified version of it).

Figure 2.3a and Figure 2.3b show the minimum values of the objective function, $I(X_p;Y)$, across different values of δ and γ. As can be seen in the figures, both δ and γ parameters determine the end mutual information between X_p and Y, and the corresponding privacy gain. For $\gamma = 0$, the privacy gain is higher for lower values of δ (to the point of saturation) as a result of lower mutual information between X_p and Y as compared to the higher values of δ. However, when $\gamma \neq 0$, the end privacy gain may be the same over a range of δ values. Note that these inferences are consistent with our intuitive understanding of the functionalities of

[3]Gradient descent algorithms are typically faster and can be optimized for better accuracy.



(a) Plot of minimum $I(X_p;Y)$ across different values of δ and γ ($\epsilon_0 = 10^{-6}$) for sample dataset 1.

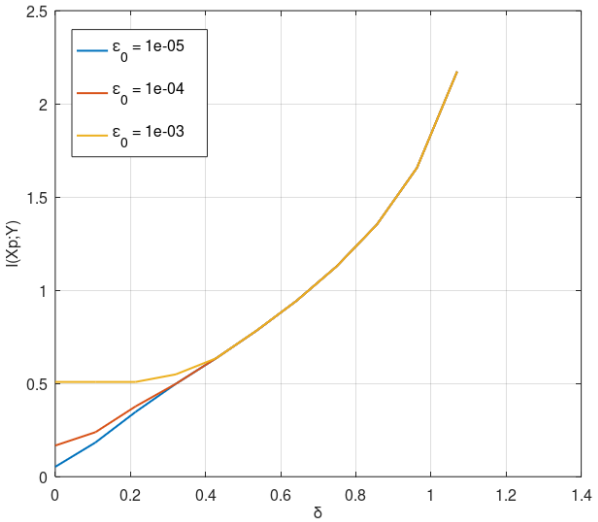


(b) Plot of minimum $I(X_p;Y)$ across different values of δ and γ ($\epsilon_0 = 10^{-6}$) for sample dataset 2.

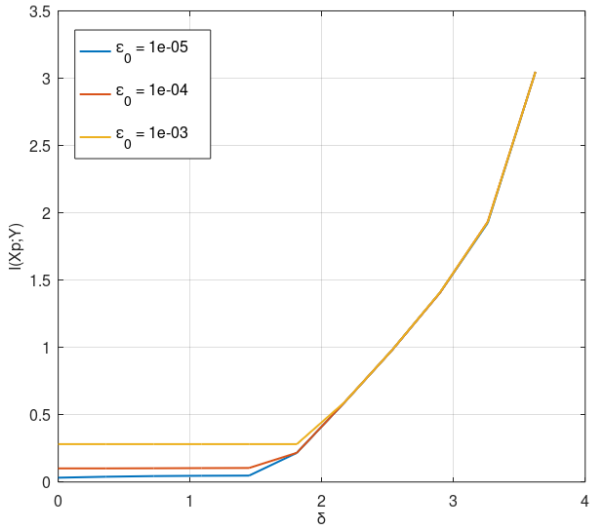
Figure 2.3: Performance of the greedy algorithm across different values of δ and γ on the two sample datasets.

the δ and γ parameters: $\gamma = 0$ implies that the user does not care about the gain in privacy per unit loss in utility and therefore, we expect the end privacy gain to be solely dependent on the desired level of utility loss, δ. However, $\gamma \neq 0$ implies that the gain in privacy per unit loss in utility must also be prioritized when maximizing the privacy gain (by minimizing the objective function). Under such constraints, we expect the smaller values of δ to be less relevant in determining the minimum $I(X_p;Y)$ as the user demands higher gain factors.

To further understand the effect of the δ and the γ parameters on the minimum value of the objective function, the graphs shown in Figure 2.3a and Figure 2.3b can be virtually divided into two regions, a δ-dominated region and a γ-dominated region. For a given γ, if the minimum $I(X_p;Y)$ is constant across a range of δ values (not accounting for the saturation), we say that the corresponding region is γ-dominated. Similarly, for a given γ, if the minimum $I(X_p;Y)$ varies with δ, we say that the corresponding region is δ-dominated. Note that while it may appear that larger values of γ result in larger γ-dominated regions and smaller values of γ result in smaller γ-dominated regions, the extent of the region significantly depends on the characteristics of the user attributes as well. Consider, for instance, two attributes of a



(a) Plot of minimum $I(X_p;Y)$ across different values of δ and ϵ_0 ($\gamma = 0$) for sample dataset 1.



(b) Plot of minimum $I(X_p;Y)$ across different values of δ and ϵ_0 ($\gamma = 0$) for sample dataset 2.

Figure 2.4: Sensitivity of the greedy algorithm to ϵ_0.

user, X_1 and X_2, where both X_1 and X_2 are highly correlated with the private attribute, X_p, and less correlated with the utility attribute, X_u. The optimal privacy mechanism intuitively involves adding a lot of noise to both X_1 and X_2 without losing much utility. Essentially, the gain factor is expected to be very high throughout the addition of incremental noises. For this setup, higher values of γ (up to a threshold) do not necessarily imply larger γ-dominated regions. Put simply, the extent of the γ-dominated and the δ-dominated regions depends on the covariances of the attributes as much as the parameters themselves.

Figure 2.4a and Figure 2.4b highlight the sensitivity of our algorithm to the ϵ_0 parameter. The ϵ_0 parameter defines the threshold for saturation and consequently, influences the resulting solution. For smaller values of δ, observe that the objective function is more sensitive to the ϵ_0 parameter. Also observe that the smaller values of ϵ_0 consistently produce smaller minimum values for $I(X_p;Y)$ across all values of δ and are therefore, desirable. However, we note that for smaller values of ϵ_0, the algorithm converges more slowly.

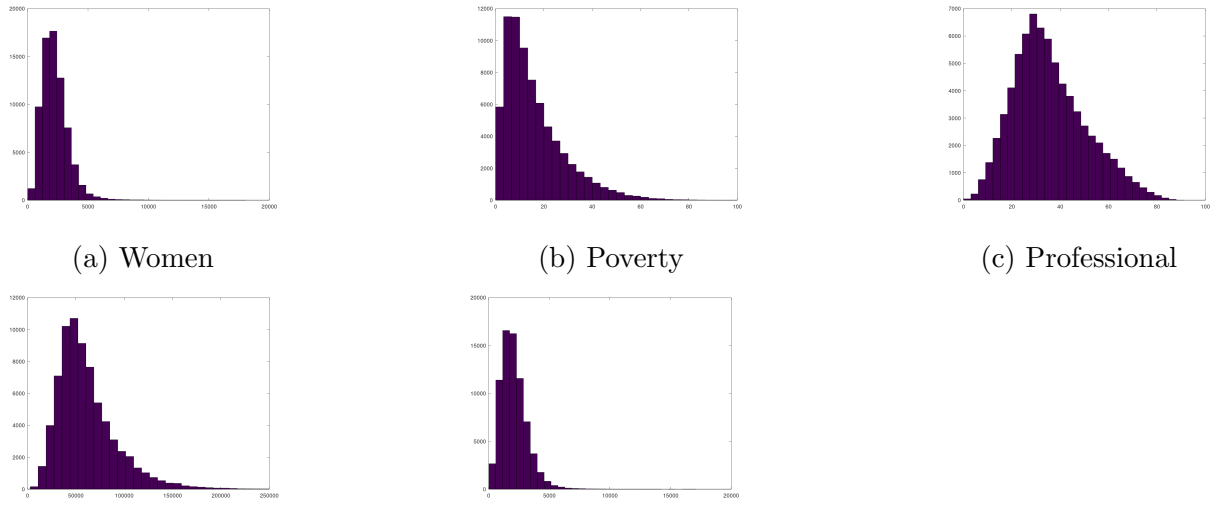
Running the greedy algorithm on the two datasets, in the case in which the utility is expressed as the Fisher information and δ adjusted accordingly as in (2.15), yields exactly the same privacy vs utility curves as above. This identity was consistently observed through all

our simulations. Nevertheless, because the problem is potentially non-convex, we conjecture that under certain instantiations the two utility metrics will provide different minimum privacy values.

2.4.2 Evaluation on a real-world dataset

The American Community Survey data for 2017 consisting of 74,001 records [48] is used as a real-world dataset to evaluate the performance of the greedy algorithm. The dataset contains demographic data for each census tract in the United States. Sixteen out of the 37 attributes in the dataset are selected as interesting features. The selected features include the number of men, the number of women, the number of citizens eligible to vote, per capita income, the percentage of the population under the poverty level, the percentage of the population employed in management, business, science, and arts sector, the percentage of the population employed in service jobs, the percentage of the population unemployed, the percentage of the population that are Hispanic or Latino, the percentage of the population that are White, the percentage of the population that are Black, the percentage of the population that are Asian, the percentage of the population that are Native American or Native Alaskan, the percentage of the population that are Native Hawaiian or Pacific Islander, the number of employed population, and the median household income in US dollars. For our experimental evaluations, the median household income is selected as the only private feature and the number of employed population is selected as the only utility feature. We emphasize here that the selection of the private and the utility features is completely random, for the purpose of testing our algorithm, and that no utility or privacy values are associated with these features in reality. The reader should not, therefore, try to understand why for example anyone would be interested in keeping private a county's median household income.

Although the selected attributes are not normally distributed (in Figure 2.5, we show the histograms of the private and utility features, as well as some of the other features, most correlated with them), the greedy algorithm still performs well in achieving a good privacy-utility tradeoff.



(a) Women (b) Poverty (c) Professional

(d) Income (e) Employed

Figure 2.5: Histogram for some of the attributes in the dataset.

The performance of the greedy algorithm in optimizing the privacy-utility tradeoff by selectively adding more noise to features with higher gain factors is compared against a naïve algorithm where equal incremental noise is added to all features–both algorithms ensuring the same level of utility, δ. The performances are evaluated using three regression models: a combinatorial model and two neural networks. These models serve as exemplary tools that can be used by an adversary and a data analyst to predict the private and the utility features of the subjects, respectively. In particular, we are interested in how well the adversary is able to predict the private features of the subjects, and the data analyst the utility features, given the subjects disclose the perturbed version of their data. The error in prediction is quantified using two common error metrics: Mean Absolute Percentage Error (MAPE) and Root Mean Square Percentage Error (RMSPE).

A Deep Feed Forward Neural Network (DFF NN) model comprising of one input, two hidden and one output layers (see Figure 2.8) is used as one of the regression models. Tensorflow, an open-source machine learning library, is used to implement this neural network. Two similar DFF NNs are created to predict the private and the utility features. The DFF NN for predicting the private feature has an input layer that consists of 14 inputs which output to 128 neurons in the first hidden layer and to 128 neurons in the second hidden

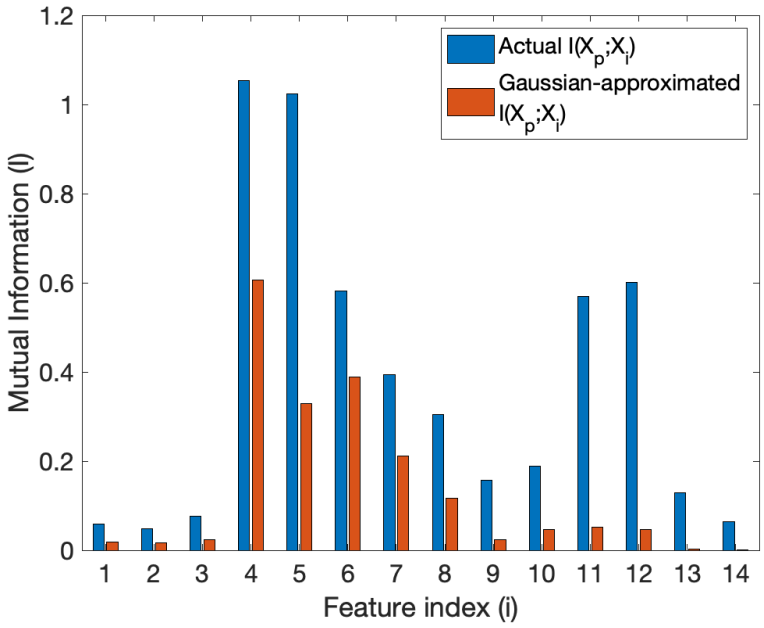


Figure 2.6: Comparison of the actual Mutual information between pairs of features, (X_p, X_i) $\forall i = 1 \ldots 14$, with the corresponding Gaussian-approximated Mutual Information.

layer. All these layers have RELU activation function. However, for the output layer, a linear activation function is used with one neuron for the output. This model is trained for 300 epochs. Likewise, the DFF NN for predicting the utility feature has two hidden layers comprising of 64 neurons. It is quite similar to the other DFF NN in that it has the same activation functions in the input, hidden and output layers (i.e. RELU in the input and the hidden layers and a linear activation function in the output layer). This model is trained for 350 epochs. In both DFF NNs, all neurons in one layer are connected to every other neurons in the preceding layer.

The other two regression models, the Combinatorial model and the GMDH Neural Network, are integrated as a part of the *GMDH Shell for Data Science* software. The appropriate parameters for the models are automatically selected by the software. The software automatically optimizes and selects the best models for regression based on the training and validation datasets.

Each model is trained with a training set consisting of 50,000 data points. Another set, consisting of 1000 data points, is used to generate two noisy test sets, one using the naïve

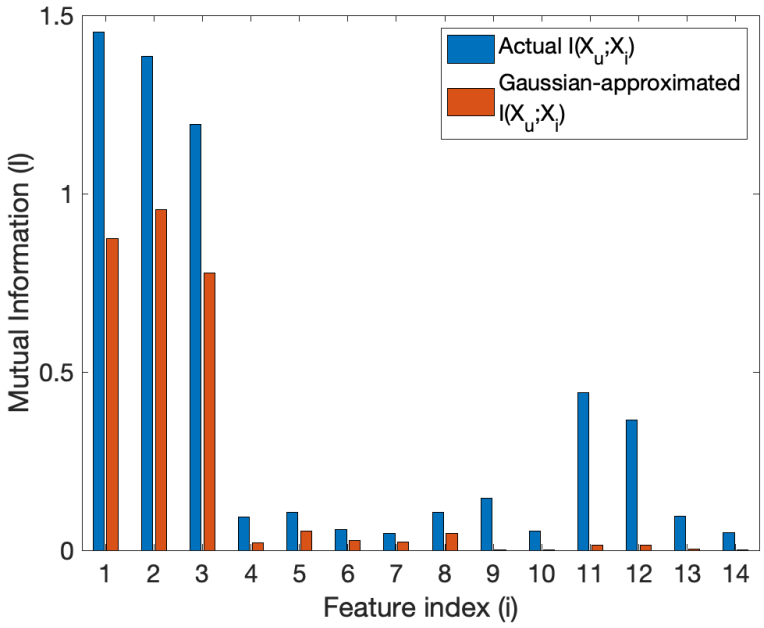


Figure 2.7: Comparison of the actual Mutual information between pairs of features, (X_u, X_i) $\forall i = 1 \ldots 14$, with the corresponding Gaussian-approximated Mutual Information.

algorithm and the other using the greedy algorithm. Each noisy test set consists of a million data points as a result of the addition of *Gaussian White Noise* samples. The results of running the regression models on the two noisy test sets are shown in Table 2.1 and Table 2.2. For reference, the results of running the regression models on the noise-free version of the test set is also included.

Three different values of δ that cover a wide range of resulting noise variances are selected to evaluate the performance of the greedy algorithm ($\delta = 1.19 \implies tr(\Sigma_N) = 2.999$, $\delta = 1.18 \implies tr(\Sigma_N) = 655.25$, $\delta = 1.17 \implies tr(\Sigma_N) = 6803.3$). Note that δ is inversely related to the prediction error for the utility feature, and therefore, we expect smaller values of δ to induce higher errors in the prediction of the utility feature. However, a given value of δ does not directly map to particular values of percentage errors in the prediction of the utility feature. This can be attributed to three main reasons: first, the dataset used for computing the noise variances is not perfectly Gaussian, and therefore, the computed Mutual Information (assuming an underlying multivariate Gaussian distribution) differs from the actual Mutual Information (see, for instance, Figure 2.6 and Figure 2.7 which compare

the actual Mutual Information between pairs of features with the corresponding Gaussian-approximated Mutual Information); second, the models used for regression are inherently empirical and do not necessarily parallel theoretical results; and third, the smaller values of δ result in higher noise variances which, when added to the data points as a part of the perturbation process, may result in negative values which must be capped to 0, thereby skewing the results slightly.

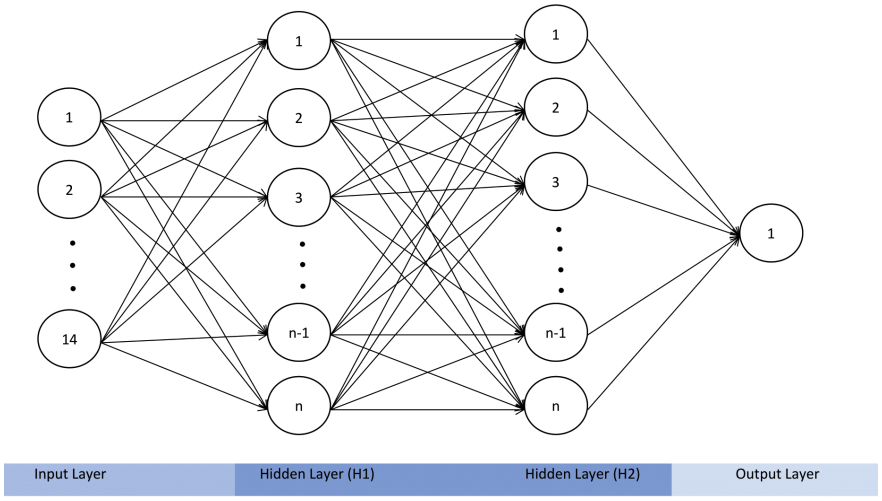


Figure 2.8: DFF NN Architecture.

As can be seen in Table 2.1, in the prediction of the private feature, the greedy algorithm consistently results in higher prediction errors under different regression models and across different values of δ which, compared to the naïve algorithm, translates to better privacy gain. In the prediction of the utility feature (Table 2.2), the performance of the greedy algorithm is comparable to that of the naïve algorithm for $\delta = 1.19$ and $\delta = 1.18$ whereas for $\delta = 1.17$, the greedy algorithm performs slightly worse than the naïve algorithm. However, notice that for $\delta = 1.17$, the greedy algorithm performs significantly better in terms of minimizing the privacy leakage (Table 2.1) suggesting a much higher ratio of privacy gain to utilty loss than the naïve algorithm. Overall, the greedy approach achieves considerably better privacy-utility tradeoffs than the naïve approach.

Table 2.1: Prediction of the private feature (Income).

δ	Regression Model	MAPE (No noise)	MAPE (Naïve)	MAPE (Greedy)	RMSPE (No noise)	RMSPE (Naïve)	RMSPE (Greedy)
1.19	Combinatorial	17.03	17.128	18.446	24.45	24.556	27.069
	GMDH NN	15.644	15.812	16.744	22.620	22.689	24.470
	DFF NN (Tensorflow)	13.298	13.325	13.612	18.820	18.851	19.221
1.18	Combinatorial	17.03	19.48	85.35	24.45	28.74	146.69
	GMDH NN	15.644	17.684	55.960	22.620	25.804	82.925
	DFF NN (Tensorflow)	13.298	14.302	44.291	18.820	20.077	71.184
1.17	Combinatorial	17.03	20.989	84.939	24.45	30.646	131.641
	GMDH NN	15.644	19.779	69.864	22.620	29.835	110.602
	DFF NN (Tensorflow)	13.298	16.020	60.622	18.820	22.771	107.58

Table 2.2: Prediction of the utility feature (Employed).

δ	Regression Model	MAPE (No noise)	MAPE (Naïve)	MAPE (Greedy)	RMSPE (No noise)	RMSPE (Naïve)	RMSPE (Greedy)
1.19	Combinatorial	20.11	20.307	20.116	34.82	35.067	34.816
	GMDH NN	11.245	11.471	11.246	14.655	15.111	14.655
	DFF NN (Tensorflow)	10.185	10.201	10.204	13.310	13.325	13.334
1.18	Combinatorial	20.11	21.49	20.34	34.82	36.57	35.14
	GMDH NN	11.245	13.165	11.245	14.655	17.486	14.655
	DFF NN (Tensorflow)	10.185	10.610	11.733	13.310	13.872	15.716
1.17	Combinatorial	20.11	20.536	22.332	34.82	34.938	37.656
	GMDH NN	11.245	12.609	18.235	14.655	16.349	23.644
	DFF NN (Tensorflow)	10.185	11.362	16.814	13.310	14.696	23.261

Chapter 3

Privacy-Utility Tradeoff in a Dynamic Setting

3.1 Introduction

In this chapter, we consider a dynamic model of privacy and capture privacy leakage over time, specifically focusing on the leakage at the end of a finite time horizon. We consider a simple privacy mechanism that involves compressing the user's data before each disclosure to minimize the privacy leakage at a future time. Subject to constraints on future privacy, we investigate different strategies that yield different net utilities for the user. Notice that the dynamic privacy-utility tradeoff problem under consideration is directly analogous to the investment problem from economics where a user seeks to maximize her rate of return over a finite time horizon by carefully choosing to invest a certain amount of money and spend the rest from her periodic income. Some examples of practical settings where this privacy model is useful are: 1. A person aims to run for an office n years from the present. The person seeks to cautiously regulate their social media usage in the meantime so that they can limit the amount of private information that can be inferred when they run for the office. 2. A chemical plant is designed to undergo a complete overhaul after a certain number of years. In the meantime, the plant operator hires a third-party consultant, with expertise in

data-based control strategies, with whom they decide to share some of the sensor data. At the same time, the operator aims to hide information about proprietary chemical processes at the time of the overhaul and therefore, looks to cautiously control the amount of shared information when the plant is in operation.

3.2 Problem Description

3.2.1 Problem Setup

We consider a setting where a privacy-aware user seeks to cautiously disclose her personal information to a data analyst over a finite time horizon. The objective of the user is to maximize her instantaneous utilities, which the data analyst provides by extracting useful information from the disclosed information at each time step, while limiting the amount of leakage about her sensitive information at the end of the finite time horizon. In contrast to the static setting which models the information disclosure at a single time step, the dynamic setting under consideration models the incremental disclosure of information at every time step until the end of the finite time horizon. The solution to this dynamic privacy problem involves finding an optimal strategy that maximizes the sum of instantaneous utilities while ensuring that the privacy leakage at the end of the finite horizon remains below a pre-specified threshold with high probability.

In the dynamic privacy setting, we assume that each user has a set of features, represented by the random vector X, which evolves over time. We use the subscript k to denote the feature vector at time step k. At any given time step k, the feature vector consists of the user's private features represented by the random vector X_k^p and the user's public features represented by the random vector X_k^u such that $X_k = X_k^p \cup X_k^u$. We consider a general setting where X_k^p and X_k^u are correlated. In many real world settings, the user's observations of her own feature vectors are only available as noisy measurements (for instance, heart-rate readings from a smart watch). To model this, we assume that the true values of X_k (and consequently, X_k^p and X_k^u) may not be directly observable; instead, there is an

observable process Z_k that carries information about X_k such that $Z_k = f(X_k)$. The user's instantaneous privacy and utility is directly associated with X_k^p and X_k^u, respectively. Next, we assume that the user is willing to disclose Z_k in return for some utility. However, as Z_k contains information about both X_k^p and X_k^u, disclosing Z_k inevitably leaks some information about X_k^p, and this leakage carries over to the future time-steps which the user seeks to avoid. To address this, we consider a privacy mechanism which perturbs Z_k before disclosure. The privacy mechanism involves transforming the entire observation vector, Z_k, into a lower-dimensional noisy vector, $\tilde{Z}_k$. The transformation is essentially non-invertible and therefore, certain information about Z_k (and consequently, X_k) is lost due to the transformation. An ideal transformation function maximizes the information loss regarding X_k^p while minimizing the information loss regarding X_k^u. However, due to the correlation between X^p and X^u, this may not always be possible.

A Linear Dynamical System (LDS), which is a continuous state-space generalization of a Hidden Markov Model, can be used to model the evolution of the user's feature vectors over time as well as the observation process. LDS has been widely used to model the underlying system in the context of privacy-preserving information disclosure [13, 37, 49, 50, 51]. We consider a first-order LDS model in which X_k evolves according to the linear equation:

$$X_k = F_k X_{k-1} + W_k, \tag{3.1}$$

where F_k is the *state-transition matrix* and W_k is the zero-mean Gaussian process noise with covariance Q_k. Similarly, the observation process can be represented by the linear equation:

$$Z_k = H_k X_k + V_k, \tag{3.2}$$

where H_k denotes the *observation matrix* and V_k denotes the zero-mean Gaussian measurement noise with covariance R_k. We consider both F_k and H_k to be full-ranked square matrices and assume that all system parameters are publicly known. For quick reference, the description of all system parameters, system vectors and other symbols used throughout

this paper can be found in Table 3.1.

Table 3.1: Description of some commonly used symbols and notations

Symbol	**Description**
n	Finite time horizon
N_p	Number of private features
N_u	Number of utility features
N	Total number of features ($N = N_p + N_u$)
M	Compression size ($M < N$)
X_k	Random vector representing the user's features ($X_k \in \mathbb{R}^{N\times 1}$)
Z_k	Observation vector ($Z_k \in \mathbb{R}^{N\times 1}$)
$\tilde{Z}_k$	Compressed observation vector ($\tilde{Z}_k \in \mathbb{R}^{M\times 1}$)
F_k	State-transition matrix ($F_k \in \mathbb{R}^{N\times N}$)
H_k	Observation matrix ($H_k \in \mathbb{R}^{N\times N}$)
Q_k	The covariance of the process noise ($Q_k \in \mathbb{R}^{N\times N}$)
R_k	The covariance of the measurement noise ($R_k \in \mathbb{R}^{N\times N}$)
C_k	Compression matrix ($C_k \in \mathbb{R}^{N\times M}$)
X_k^p	Random vector representing the user's true private feature at time k ($X_k^p \in \mathbb{R}^{N_p\times 1}$)
X_k^u	Random vector representing the user's true public (utility) feature at time k ($X_k^u \in \mathbb{R}^{N_u\times 1}$)
$\widehat{X}_{k\|j}^{Z,p}$	Data owner's estimation of X_k^p using observations $Z_1, Z_2, \cdots, Z_j$ $(k \geq j)$
$\widehat{X}_{k\|j}^{Z,u}$	Data owner's estimation of X_k^u using observations $Z_1, Z_2, \cdots, Z_j$ $(k \geq j)$
$\widehat{X}_{k\|j}^{\tilde{Z},p}$	Adversary's estimation of X_k^p using observations $\tilde{Z}_1, \tilde{Z}_2, \cdots, \tilde{Z}_j$ $(k \geq j)$
$\widehat{X}_{k\|j}^{\tilde{Z},u}$	Data analyst's estimation of X_k^u using observations $\tilde{Z}_1, \tilde{Z}_2, \cdots, \tilde{Z}_j$ $(k \geq j)$
$P_{k\|k}^{Z}$	Error covariance associated with the data owner's estimate of X_k
$P_{k\|k}^{\tilde{Z}}$	Error covariance associated with the data analyst and adversary's estimate of X_k
$\mathsf{P}(X\|Y)$	The conditional probability of X given Y

The privacy mechanism involves mapping the observation vector Z_k to a lower-dimensional vector $\tilde{Z}_k$ using a compression matrix C_k such that $\tilde{Z}_k = C_k^T Z_k$. We assume that an adversary has complete knowledge about the system dynamics as well as the privacy mechanism. The goal of the data-owner (user) is to prudently select the compression matrices, $C_1, C_2, \cdots C_n$, that maximize the sum of instantaneous utilities while limiting the amount of information leaked about the private features at the end of the finite time horizon, n. Note that the sequence $C_1, C_2, \cdots, C_n$ constitutes the *strategy* for the data-owner. The data analyst is tasked with inferring X_k^u from the disclosed sequence $\tilde{Z}_1, \tilde{Z}_2, \cdots, \tilde{Z}_k$ at each time step k

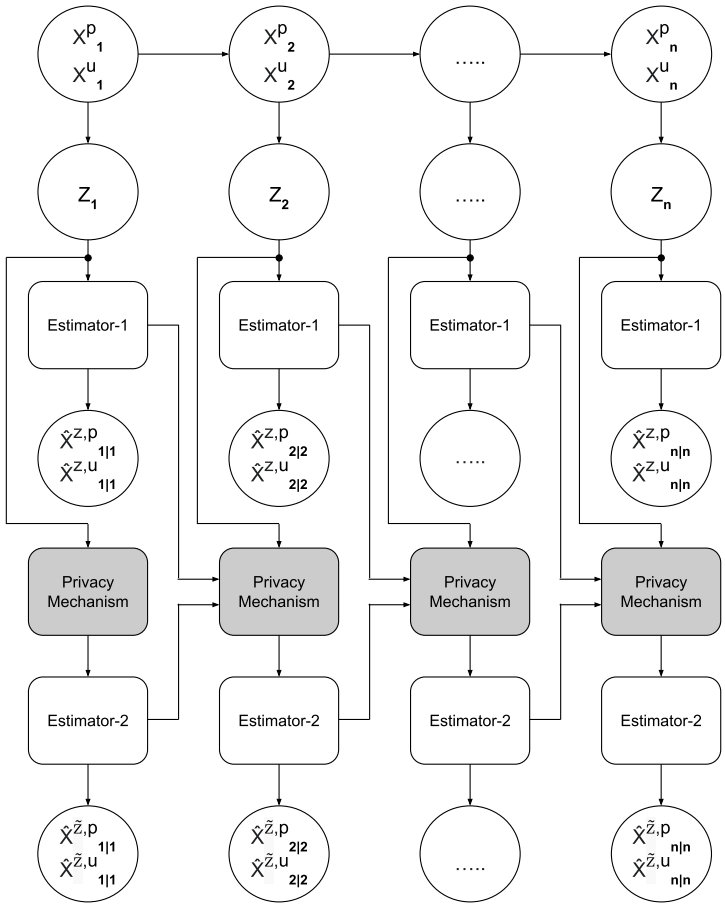


Figure 3.1: The dynamics of the finite-horizon privacy-utility tradeoff problem.

while the future adversary seeks to infer X_n^p from the disclosed sequence $\tilde{Z}_1, \tilde{Z}_2, \cdots, \tilde{Z}_n$. The problem, therefore, naturally relates to an estimation problem. The dynamics of the problem are depicted in Figure 3.1.

Before discussing the formal models of privacy and utility, we first focus on the problem of estimating the latent system states, X_k^p and X_k^u, given a series of observations. This estimation problem can be solved using the *Kalman filter* which is an optimal linear filter in terms of minimizing the *Mean Squared Error* of the estimates [52]. Estimation using the Kalman filter involves two steps: the prediction step in which the system states are predicted a priori and the update step in which the current measurements/observations are incorporated to update the state estimates. Formally, the Kalman filter for the LDS represented by (3.1) and (3.2) can be expressed as [52]:

Prediction step:

$$\widehat{x}_{k|k-1} = F_k \widehat{x}_{k-1|k-1}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T$$

Update step:

$$\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k(Z_k - H_k \widehat{x}_{k|k-1})$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1},$$

where $\widehat{x}_{k|k-1}$ is the a priori estimate of X_k given the observations up to time $k-1$, $\widehat{x}_{k|k}$ is the a posteriori estimate of X_k given the observations up to time k, $P_{k|k-1}$ is the a priori error covariance of the estimate $\widehat{x}_{k|k-1}$, and $P_{k|k}$ is the a posteriori error covariance of the estimate $\widehat{x}_{k|k}$. The Kalman gain, K_k, is given by $K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$.

3.2.2 Privacy and Utility Requirements

Let $\widehat{X}_{k|k}^{Z,u}$ and $\widehat{X}_{k|k}^{Z,p}$ represent the data owner's estimate of X_k^u and X_k^p, respectively, given the series of observations, $Z_1, Z_2, \cdots, Z_k$. Similarly, let $\widehat{X}_{k|k}^{\tilde{Z},u}$ and $\widehat{X}_{k|k}^{\tilde{Z},p}$ represent the data analyst's estimate of X_k^u and the adversary's estimate of X_k^p, respectively, given the series of observations, $\tilde{Z}_1, \tilde{Z}_2, \cdots, \tilde{Z}_k$. Also, let $d(X, Y)$ denote some distance function that measures the distance between random vectors X and Y. An example of the distance function is the L^2-norm. From the utility point of view, it is desirable that $d(\widehat{X}_{k|k}^{\tilde{Z},u}, \widehat{X}_{k|k}^{Z,u})$ is as small as possible for all k. A zero distance between the estimates, $\widehat{X}_{k|k}^{\tilde{Z},u}$ and $\widehat{X}_{k|k}^{Z,u}$, is achievable if the data owner discloses her true observations, $Z_1, Z_2, \cdots, Z_k$, with no privacy mechanisms. Similarly, from the future privacy point of view, it is desirable that $d(\widehat{X}_{n|n}^{Z,p}, \widehat{X}_{n|n}^{\tilde{Z},p})$ is as large as possible. However, due to the correlation between X_k^p and X_k^u, in general, it is not feasible to both minimize the instantaneous utility losses and maximize the perceived future privacy. The problem, therefore, naturally manifests as a privacy-utility trade-off

optimization problem. Intuitively, the optimization problem involves finding an optimal strategy that minimizes the sum of instantaneous utility losses, $\sum_{k=1}^{n} d(\widehat{X}_{k|k}^{\tilde{Z},u}, \widehat{X}_{k|k}^{Z,u})$, under the constraint that the privacy leakage at the end of the finite horizon, $\frac{1}{d(\widehat{X}_{n|n}^{\tilde{Z},p}, \widehat{X}_{n|n}^{Z,p})}$, must not exceed a pre-specified threshold $\frac{1}{\delta}$. Formulating the optimization problem, however, exposes several challenges. For one, $\sum_{k=1}^{n} d(\widehat{X}_{k|k}^{\tilde{Z},u}, \widehat{X}_{k|k}^{Z,u})$ and $d(\widehat{X}_{n|n}^{\tilde{Z},p}, \widehat{X}_{n|n}^{Z,p})$ are both random variables due to the uncertainties in the future observations, $Z_{k+1}, Z_{k+2}, \cdots, Z_n$. Further, without the knowledge of the future observations, it is difficult to devise an optimal strategy that satisfies the constraint on future privacy leakage. In fact, given the Gaussian assumption for both the process noise and the measurement noise, it may not even be possible to ensure a non-trivial constraint on the future privacy leakage using any feasible sequence of actions, $C_1, C_2, \cdots, C_n$; it is, therefore, more appropriate to characterize strategies in terms of the probability of privacy outage, $\mathsf{P}(d(\widehat{X}_{n|n}^{\tilde{Z},p}, \widehat{X}_{n|n}^{Z,p}) < \delta)$, in addition to the total utility loss. The probability of privacy outage reflects the probability that the privacy constraint is not satisfied in the future.

3.3 Formulation as a Markov Decision Process Problem

The finite horizon privacy-utility trade-off problem fits nicely with a Markov Decision Process (MDP). In a discrete time continuous state MDP model, at every time step k, an agent observes the current state of some Markov process S_k, takes an action a_k and receives a reward R_k. The state of the Markov process at time step k, in general, depends on the state and the action at time step $k-1$ and some stochastic process noise ω_k. The reward received by the agent at time step k depends on the current state of the Markov process, the current action taken by the agent and the next state of the Markov process.

Formally, a discrete time continuous state continuous action Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ where $\mathcal{S} \in \mathbb{R}^J$ represents the state space, $\mathcal{A} \in \mathbb{R}^L$ represents the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the state-transition function such that $P(s_k | a_k, s_{k+1})$

gives the probability of transitioning to the next state s_{k+1} from the current state s_k by taking an action a_k. Let the random vectors S_k, A_k and S_{k+1} represent the current state, the current action and the next state, respectively. As the state space is continuous, P is specified as a probability density function such that $\int_{\Psi} P\left(s_k|a_k, s_{k+1}\right) \mathrm{d}s_{k+1} = \mathsf{P}(S_{k+1} \in \Psi|S_k = s_k, A_k = a_k)$, where $\Psi \subseteq \mathcal{S}$, with $\mathcal{S}$ the space of S_k. Similarly, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ represents the reward function such that $R_k(s_k, a_k, s_{k+1})$ gives the reward received by the agent at time step k by taking an action a_k when the current and the next states of the Markov process are s_k and s_{k+1}, respectively. The discount factor $\gamma \in [0, 1]$ captures how the agent values her future rewards compared to her current reward – if $\gamma = 1$, the agent values all her future rewards equally to her current reward and if $\gamma = 0$, the agent only values her current reward and disregards all her future rewards.

The goal of the agent is to maximize the expected sum of her current and future discounted rewards, $\mathrm{E}[\sum_k \gamma^k R_k(S_k, A_k, S_{k+1})]$. The agent seeks to find the optimal sequence of actions that allows her to optimize the expected sum of discounted rewards. In this regard, it is useful to define a function, called the optimal state-value function, that provides a measure of the maximum achievable sum of expected rewards from a particular state. Let $V_k^*(s_k)$ denote the optimal state-value function at time step k given the current state s_k. Then, the optimal state-value function can be written as a recursive equation using Bellman's Principle of Optimality as:

$$V_k^*(s_k) = \max_{a_k} \int_{\mathcal{S}} P\left(s_k|a_k, s_{k+1}\right) \left(R_k(s_k, a_k, s_{k+1}) + \gamma V_{k+1}^*(s_{k+1})\right) \mathrm{d}s_{k+1}.$$

The Bellman equation formulation offers a dynamic programming approach to solve the resulting optimization problem.

The finite-horizon privacy-utility problem can be directly translated to a finite-horizon discrete time continuous state continuous action MDP problem. Recall that in the finite-horizon privacy setting, the user seeks to find an optimal sequence of actions that allows her to maximize the sum of the instantaneous utilities while ensuring that the privacy leakage at the end of the finite horizon remains below a pre-specified threshold with high probability. This

is inherently a decision problem that incorporates a meaningful notion of reward captured as the expected sum of instantaneous utilities and future privacy leakage.

Let $\widehat{X}^Z_{k|k}$ represent the data owner's estimate of X, given the series of observations, $Z_1, Z_2, \cdots, Z_k$ and $P^Z_{k|k}$ represent the error covariance associated with the estimate[1]. Similarly, let $\widehat{X}^{\tilde{Z}}_{k|k}$ represent the data analyst's (or adversary's) estimate of X, given the series of observations, $\tilde{Z}_1, \tilde{Z}_2, \cdots, \tilde{Z}_k$ and $P^{\tilde{Z}}_{k|k}$ represent the error covariance associated with the estimate. Now, define

$$
\begin{aligned}
&d^u(k,k) \triangleq d(\widehat{X}^{Z,u}_{k|k}, \widehat{X}^{\tilde{Z},u}_{k|k}),\\
&d^p(j,k) \triangleq d(\widehat{X}^{Z,p}_{j|k}, \widehat{X}^{\tilde{Z},p}_{j|k}) \quad (n \geq j \geq k),\\
&S_k \triangleq \{Z_k, \widehat{X}^Z_{k-1|k-1}, \widehat{X}^{\tilde{Z}}_{k-1|k-1}, P^Z_{k-1|k-1}, P^{\tilde{Z}}_{k-1|k-1}\},
\end{aligned}
$$

where S_k represents the state of the MDP (which is fundamentally different from the state of the LDS, X_k). The state variables in S_k can be recursively computed using the following sequence of Kalman filter equations:

$$
\begin{aligned}
\widehat{X}^Z_{k|k-1} &= F_k \widehat{X}^Z_{k-1|k-1},\\
\widehat{X}^{\tilde{Z}}_{k|k-1} &= F_k \widehat{X}^{\tilde{Z}}_{k-1|k-1},\\
P^Z_{k|k-1} &= F_k P^Z_{k-1|k-1} F_k^T + Q_k,\\
P^{\tilde{Z}}_{k|k-1} &= F_k P^{\tilde{Z}}_{k-1|k-1} F_k^T + Q_k,\\
K^Z_k &= P^Z_{k|k-1} H_k^T (H_k P^Z_{k|k-1} H_k^T + R_k)^{-1},\\
K^{\tilde{Z}}_k &= P^{\tilde{Z}}_{k|k-1} H_k^T C_k (C_k^T H_k P^{\tilde{Z}}_{k|k-1} H_k^T C_k + C_k^T R_k C_k)^{-1},
\end{aligned}
$$

[1] $P^Z_{k|k}$ is not a function of Z_k. The superscript Z is used as a convention to imply that the symbol being defined directly concerns the data owner, who has observations $\{Z_k\}$, rather than the adversary or data analyst, who have observations $\{\tilde{Z}_k\}$.

$$\widehat{X}^{Z}_{k|k} = \widehat{X}^{Z}_{k|k-1} + K^{Z}_{k}(Z_k - H_k\widehat{X}^{Z}_{k|k-1}),$$
$$\widehat{X}^{\tilde{Z}}_{k|k} = \widehat{X}^{\tilde{Z}}_{k|k-1} + K^{\tilde{Z}}_{k}(C^T_k Z_k - C^T_k H_k\widehat{X}^{\tilde{Z}}_{k|k-1}),$$
$$P^{Z}_{k|k} = P^{Z}_{k|k-1} - K^{Z}_{k} H_k P^{Z}_{k|k-1},$$
$$P^{\tilde{Z}}_{k|k} = P^{\tilde{Z}}_{k|k-1} - K^{\tilde{Z}}_{k} C^T_k H_k P^{\tilde{Z}}_{k|k-1}.$$

Initially, $\widehat{X}^{Z}_{0|0} = \widehat{X}^{\tilde{Z}}_{0|0} = \mathrm{E}[X_0]$ and $P^{Z}_{0|0} = P^{\tilde{Z}}_{0|0} = \mathrm{Cov}(X_0)$.

We now define the reward function, R_k, as

$$R_k(S_k, C_k, S_{k+1}) = \begin{cases} \alpha\big(d^p(n, k+1) - d^p(n,k)\big) - d^u(k,k) & \text{when } k < n, \\ \alpha d^p(n,k) - d^u(k,k) & \text{when } k = n, \end{cases} \tag{3.3}$$

where α is the privacy-utility tradeoff parameter.

At any given time k, the user's goal is to chose an action $C_k = f(S_k)$ that allows her to maximize the expected sum of the current and future rewards, $\mathrm{E}[\sum_{t=k}^{n} R_t(S_t, C_t, S_{t+1})]$.

Let $\mathbf{C}^* = \{C^*_1, C^*_2, \cdots, C^*_n\}$ be the set of optimal actions. Notice that at the beginning of the finite time horizon, the sum of the rewards can be expressed as

$$\sum_{k=0}^{n} R_k(S_k, C_k, S_{k+1}) = \alpha\big(2d^p(n,n) - d^p(n,0)\big) - \sum_{k=0}^{n} d^u(k,k).$$

Since $d^P(n,0)$ and $d^u(0,0)$ are both zero (which follows from the assumption that $\widehat{X}^{Z}_{0|0} = \widehat{X}^{\tilde{Z}}_{0|0}$ and $P^{Z}_{0|0} = P^{\tilde{Z}}_{0|0}$), substituting $2\alpha = \beta$, we get

$$\sum_{k=0}^{n} R_k(S_k, C_k, S_{k+1}) = \beta d^p(n,n) - \sum_{k=1}^{n} d^u(k,k). \tag{3.4}$$

As the reader may have noticed by now, the reward function is defined such that the sum of rewards captures both the privacy and the utility aspects of the problem in a single expression given in (3.4). The value of γ is taken to be 1 for the same reason. Note that the parameter β in (3.4) directly relates to the probability of privacy outage $\mathsf{P}(d(\widehat{X}^{\tilde{Z},p}_{n|n}, \widehat{X}^{Z,p}_{n|n}) < \delta)$

and the resulting privacy-utility tradeoff; larger values of beta are expected to result in higher utility losses with lower probabilities of privacy outage while smaller values of beta are expected to result in lower utility losses with higher probabilities of privacy outage. The privacy-utility tradeoff region corresponding to different values of β will be determined experimentally.

We now use the Bellman equation to formulate the finite horizon privacy-utility tradeoff optimization problem. Let V_k denote the state-value function at timestep k and $V_k^*(s_k)$ denote the optimal state-value function given the state s_k where

$$s_k = \{z_k, \widehat{x}^z_{k-1|k-1}, \widehat{x}^{\tilde{z}}_{k-1|k-1}, P^z_{k-1|k-1}, P^{\tilde{z}}_{k-1|k-1}\}.$$

Then, using the Bellman equation of optimality, the optimization problem can be formulated as:

$$\begin{aligned}
V_k^*(s_k) &= \max_{C_k} \int_{\mathcal{S}} \mathsf{P}(s_{k+1}|s_k, C_k) \cdot \Big(R_k(s_k, C_k, s_{k+1}) + V_{k+1}^*(s_{k+1})\Big) \mathrm{d}s_{k+1} \\
&= \max_{C_k} \int_{\mathcal{Z}} \mathsf{P}(z_{k+1}|s_k, C_k) \int_{\Lambda} \mathsf{P}(\widehat{x}^z_{k|k}|s_k, C_k) \cdot \int_{\Delta} \mathsf{P}(\widehat{x}^{\tilde{z}}_{k|k}|s_k, C_k) \int_{\Phi} \mathsf{P}(P^z_{k|k}|s_k, C_k) \cdot \\
&\qquad \int_{\Omega} \mathsf{P}(P^{\tilde{z}}_{k|k}|s_k, C_k) \Big(R_k(s_k, C_k, s_{k+1}) + V_{k+1}^*(s_{k+1})\Big) \cdot \mathrm{d}P^{\tilde{z}}_{k|k}\, \mathrm{d}P^z_{k|k} \mathrm{d}\widehat{x}^{\tilde{z}}_{k|k}\, \mathrm{d}\widehat{x}^z_{k|k}\, \mathrm{d}z_{k+1},
\end{aligned}$$

where $\mathcal{Z}, \Lambda, \Delta, \Phi$ and Ω are the feasible spaces of $z_{k+1}, \widehat{x}^z_{k|k}, \widehat{x}^{\tilde{z}}_{k|k}, P^z_{k|k}$ and $P^{\tilde{z}}_{k|k}$, respectively.

Given s_k and C_k, the state variables, $\widehat{x}^z_{k|k}$, $\widehat{x}^{\tilde{z}}_{k|k}$, $P^z_{k|k}$ and $P^{\tilde{z}}_{k|k}$, are all deterministic (which directly follows from the application of the Kalman filter equations). Therefore,

$$V_k^*(s_k) = \max_{C_k} \int_{\mathcal{Z}} \mathsf{P}(z_{k+1}|s_k, C_k) \cdot \Big(R_k(s_k, C_k, s_{k+1}) + V_{k+1}^*(s_{k+1})\Big) \mathrm{d}z_{k+1} \tag{3.5}$$

$$= \max_{C_k} \int_{\mathcal{Z}} \mathsf{P}(z_{k+1}|z_k) \cdot \Big(R_k(s_k, C_k, s_{k+1}) + V_{k+1}^*(s_{k+1})\Big) \mathrm{d}z_{k+1}. \tag{3.6}$$

If X_k is a Gaussian process and V_k is a Gaussian white noise process, then, $Z_{k+1}|Z_k \sim$

$\mathrm{N}(\bar{\mu}, \bar{\Sigma})$ where

$$\begin{aligned}
\bar{\mu} &= \mathrm{E}[Z_{k+1}|Z_k] \\
&= \mathrm{E}[HX_{k+1} + V_{k+1}|Z_k] \\
&= \mathrm{E}[HX_{k+1}|Z_k] + \mathrm{E}[V_{k+1}|Z_k] \\
&= H\mathrm{E}[X_{k+1}|Z_k] \\
&= H\widehat{x}^z_{k+1|k} \\
&= HF\widehat{x}^z_{k|k}
\end{aligned}$$

and

$$\begin{aligned}
\bar{\Sigma} &= \mathrm{Cov}(Z_{k+1}|Z_k) \\
&= \mathrm{Cov}(HX_{k+1} + V_{k+1}|Z_k) \\
&= \mathrm{Cov}[HX_{k+1}|Z_k] + \mathrm{Cov}[V_{k+1}|Z_k] \\
&= H\mathrm{Cov}(X_{k+1}|Z_k)H^T + R \\
&= HP^z_{k+1|k}H^T + R \\
&= H(FP^z_{k|k}F^T + Q)H^T + R.
\end{aligned}$$

The optimization problem in (3.6) reflects the user's objective of maximizing the expected sum of the current and future rewards starting at a particular state s_k. The optimizing argument $C^*_k(s_k) = \arg\max_{C_k} V^*_k(s_k)$ constitutes the best action taken toward the goal of maximizing the expected sum of rewards.

3.4 Sub-optimal Algorithms

The optimization problem formulated in (3.6) suffers from the curse of dimensionality as both the state space and the action space are continuous. Analytical methods to solve the

problem are infeasible for practical problems as they do not yield closed-form solutions for higher dimensional problems. Numerical algorithms, such as the *value iteration* algorithm and the *policy iteration* algorithm which advance by sweeping through all possible states at each time step, also fail as there are infinitely many states to sweep through. The problem is therefore not easily tractable without further assumptions about the state space or the action space, or both.

A popular approach to solving similar optimization problems involves discretizing the state space (see, for instance, [53, 54, 55, 56, 57]). This approach is typically suboptimal, however, it can still yield promising solutions. In what follows, we highlight different algorithms that are based on the discretization of the state space to solve the optimization problem formulated in (3.6).

3.4.1 Value Iteration with Discretization

The value iteration approach to solving a finite-horizon discrete state space MDP problem involves solving the Bellman equation to find the optimal values for every possible state at every time step, starting at the end of the finite time horizon and working backwards. At time step n, the optimal value of each state is computed using the terminal reward given by $R_n(S_n, C_n) = \beta d^p(n, n) - d^u(n, n)$. The algorithm then iteratively calculates the optimal values at previous time steps as given in Algorithm 1.

The major characteristic of the value iteration algorithm is that it calculates optimal values and the optimal actions associated with the states at all time steps before the actual observations are available. The calculated values are optimal for the discrete MDP, however, due to the additional discretization step (which is not intrinsic to the value iteration algorithm itself), they are typically sub-optimal for the original MDP.

3.4.2 Pessimistic algorithm

The pessimistic algorithm is a customized algorithm to solve the finite-horizon discrete state space MDP problem. The pessimistic algorithm captures an agent who always expects to

Algorithm 1 Value Iteration Algorithm with Discretization

Define the feasible state space, $\mathcal{S}$.
Select a discretization rule, $\mathcal{D}$, and discretize $\mathcal{S}$ according to $\mathcal{D}$.
procedure VALUE ITERATION
    Initialize $V_n^*(s_n)$ for all $s_n \in \mathcal{S}$ with terminal rewards.
    for $k = n-1$ to 1 **do**
        for each $s_k \in \mathcal{S}$ **do**
            $V_k^*(s_k) = \max_{C_k} \sum_{z_{k+1}} \mathsf{P}(z_{k+1}|z_k) \cdot \Big(R_k(s_k, C_k, s_{k+1}) + V_{k+1}^*(s_{k+1})\Big)$
            $C_k^*(s_k) = \arg\max_{C_k} V_k^*(s_k)$
        end for
    end for
end procedure

transition to the worst possible state at every time step. The pessimistic algorithm seeks to optimize the value and the action associated with a state while assuming that the next transition leads to the state with the least value. The advantage of using the algorithm is that it is computationally less intensive as the state transition in the underlying model is assumed to be deterministic. The pessimistic algorithm is highlighted in Algorithm 2.

Algorithm 2 Pessimistic Algorithm

Define the feasible state space, $\mathcal{S}$.
Select a discretization rule, $\mathcal{D}$, and discretize $\mathcal{S}$ according to $\mathcal{D}$.
Initialize $V_n^*(s_n)$ for all $s_n \in \mathcal{S}$ with terminal rewards.
$v_n^{\#} = \min\{V_n^*(s_n) \,:\, s_n \in \mathcal{S}\}$
$s_n^{\#} = \arg\min_{s_n}\{V_n^*(s_n) \,:\, s_n \in \mathcal{S}\}$
for $k = n-1$ to 1 **do**
    for each $s_k \in \mathcal{S}$ **do**
        $V_k^*(s_k) = \max_{C_k} \Big(R_k(s_k, C_k, s_{k+1}^{\#}) + v_{k+1}^{\#}\Big)$
        $C_k^*(s_k) = \arg\max_{C_k} V_k^*(s_k)$
    end for
    $v_k^{\#} = \min\{V_k^*(s_k) \,:\, s_k \in \mathcal{S}\}$
    $s_k^{\#} = \arg\min_{s_k}\{V_k^*(s_k) \,:\, s_k \in \mathcal{S}\}$
end for

A quick remark on the notations: $\min\{.\}$ represents the minimum of the set whereas $\arg\min_Y\{.\}$ represents the parameter Y that corresponds to the minimum value of the set.

3.4.3 Optimistic algorithm

In contrast to the pessimistic algorithm, the optimistic algorithm captures an agent who always expects to transition to the best possible state at every time step. The optimistic algorithm seeks to optimize the value and the action associated with a state while assuming that the next transition leads to the state with the highest value. The optimistic algorithm is highlighted in Algorithm 3.

Algorithm 3 Optimistic Algorithm

Define the feasible state space, $\mathcal{S}$.
Select a discretization rule, $\mathcal{D}$, and discretize $\mathcal{S}$ according to $\mathcal{D}$.
Initialize $V_n^*(s_n)$ for all $s_n \in \mathcal{S}$ with terminal rewards.
$v_n^* = \max\{V_n^*(s_n) \,:\, s_n \in \mathcal{S}\}$
$s_n^* = \arg\max_{s_n}\{V_n^*(s_n) \,:\, s_n \in \mathcal{S}\}$
for $k = n-1$ to 1 **do**
    for each $s_k \in \mathcal{S}$ **do**
        $V_k^*(s_k) = \max_{C_k}\left(R_k(s_k, C_k, s_{k+1}^*) + v_{k+1}^*\right)$
        $C_k^*(s_k) = \arg\max_{C_k} V_k^*(s_k)$
    end for
    $v_k^* = \max\{V_k^*(s_k) \,:\, s_k \in \mathcal{S}\}$
    $s_k^* = \arg\max_{s_k}\{V_k^*(s_k) \,:\, s_k \in \mathcal{S}\}$
end for

3.5 Privacy-Utility Tradeoff Under Estimated Privacy Leakage

The finite-horizon privacy-utility tradeoff problem can also be formulated with the constraint on estimated privacy leakage instead of the actual privacy leakage at the end of the finite time horizon. The resulting optimization problem is much simpler to solve, nevertheless, the dynamic privacy requirements are still captured into the problem formulation. To this end, we consider a user who seeks to maximize her instantaneous utility while limiting the estimated future leakage about her sensitive information. The resulting optimization problem is:

$$\min_{C_k} \; d(\widehat{X}^{Z,u}_{k|k}, \widehat{X}^{\tilde{Z},u}_{k|k}) \tag{3.7}$$

subject to:

$$d(\widehat{X}^{Z,p}_{n|k}, \widehat{X}^{\tilde{Z},p}_{n|k}) \geq \delta \tag{3.8}$$

Intuitively, the user seeks to disclose as much as possible (allowed by the privacy constraint) at the current time step so as to maximize her current utility with a complete disregard for her future utilities. This strategy is, therefore, referred to as a *maximum disclosure strategy*. In contrast, the optimization problem formulated in (3.6) captures a user who seeks to cautiously disclose her personal information *piecewise*.

In a dynamic setting, a user following the maximum disclosure strategy needs to solve the optimization problem at every time step as new observations are made. As the user approaches the end of the finite time horizon, the privacy constraint is more restrictive due to the accumulated leakage resulting from the past disclosures. In some cases, the actual leakage may already exceed the estimated leakage and therefore, no choice of C_k may satisfy the constraint, especially closer to the end of the finite time horizon. Therefore, it is more appropriate to formulate an unconstrained optimization problem that captures the semantics of the constrained problem. This leads us to the following optimization problem:

$$\begin{aligned} &\min_{C_k} \; d(\widehat{X}^{Z,u}_{k|k}, \widehat{X}^{\tilde{Z},u}_{k|k}) - \beta \left(d(\widehat{X}^{Z,p}_{n|k}, \widehat{X}^{\tilde{Z},p}_{n|k}) \right) \\ &= \max_{C_k} \beta \left(d(\widehat{X}^{Z,p}_{n|k}, \widehat{X}^{\tilde{Z},p}_{n|k}) \right) - d(\widehat{X}^{Z,u}_{k|k}, \widehat{X}^{\tilde{Z},u}_{k|k}) \end{aligned} \tag{3.9}$$

The parameter β in (3.9) directly relates to the constraint in (3.8) and influences the probability of privacy outage at the end of the finite time horizon, $\mathsf{P}(d(\widehat{X}^{\tilde{Z},p}_{n|n}, \widehat{X}^{Z,p}_{n|n}) < \delta)$.

Although the optimization problem formulated in (3.9) can easily be solved without further transformation, it is nevertheless possible to transform the optimization problem

into an equivalent MDP formulation. First, define

$$d^u(k,k) \triangleq d(\widehat{X}^{Z,u}_{k|k}, \widehat{X}^{\tilde{Z},u}_{k|k}),$$
$$d^p(n,k) \triangleq d(\widehat{X}^{Z,p}_{n|k}, \widehat{X}^{\tilde{Z},p}_{n|k}),$$
$$S_k \triangleq \{Z_k\},$$

where S_k represents the state of the MDP. We now define the reward function, R_k, as

$$R_k(S_k, C_k) = \beta d^p(n,k) - d^u(k,k)$$

Notice that the reward function is independent of the future observations and therefore, deterministic. At any given time k, the user's goal is to chose an action $C_k = f(S_k)$ that allows her to maximize the instantaneous reward, $R_k(S_k, C_k)$. Since the user is oblivious to future rewards, we set $\gamma = 0$. The MDP equivalent of the optimization problem in (3.9) can then be expressed as:

$$V^*_k(s_k) = \max_{C_k} R_k(s_k, C_k) \tag{3.10}$$

The optimization problem in (3.10) reflects the user's objective of maximizing her instantaneous reward at a particular state s_k. The argument of the optimization $C^*_k(s_k) = \arg\max_{C_k} V^*_k(s_k)$ constitutes the best action taken toward the goal of maximizing the instantaneous reward.

The main advantage of the optimization problem formulated in (3.10) (and equivalently, in 3.9) is that it does not require sweeping through all possible states at each time step (which would otherwise be required if the current reward depended on future states) and therefore, computationally much less intensive to solve. Further, the optimization problem is solved forwards as new observations become available. Algorithm 4 highlights the steps involved in solving the finite-horizon privacy-utility tradeoff optimization problem under estimated

privacy leakage using the maximum disclosure strategy.

Algorithm 4 Maximum Disclosure Algorithm

1: At each time step k, **do**
2: $V_k^*(s_k) = \max_{C_k} R_k(s_k, C_k)$
3: $C_k^*(s_k) = \arg\max_{C_k} V_k^*(s_k)$
4: **end**

3.6 Simulations

In this section, we evaluate the performance of the value iteration algorithm, the pessimistic algorithm, the optimistic algorithm and the maximum disclosure algorithm via synthetic simulations. For our simulations, we consider an LDS with $N_p = 1$ and $N_u = 2$. We assume that F_k and H_k are time invariant such that $F_1 = F_2 = \cdots = F_n = F$ and $H_1 = H_2 = \cdots = H_n = H$. Further, we assume that X_k is a zero mean Gaussian process and W_k and V_k are independent and identically distributed standard Gaussian random vectors.

The elements of F and H are sampled independently from a uniform distribution in the unit interval. F is further normalized such that its eigenvalues lie within a unit circle which ensures that the LDS is stable. As $P^Z_{k-1|k-1}$ is not a function of the observations or the actions, it is computed offline. Similarly, $P^{\tilde{Z}}_{k-1|k-1}$ is estimated with $P^{\tilde{Z}}_{k-1|0}$. For the value iteration, pessimistic and optimistic algorithms, a discretization function, $\mathcal{D}$, is used to approximate the components of Z_k, $\widehat{X}^Z_{k-1|k-1}$ and $\widehat{X}^{\tilde{Z}}_{k-1|k-1}$ as binary variables. As a simplest discretization strategy, we chose the function $\mathcal{D}$ such that:

$$\mathcal{D}(y) = \begin{cases} \mathrm{E}[y] - 0.1 & \text{when } y < \mathrm{E}[y], \\ \mathrm{E}[y] + 0.1 & \text{when } y \geq \mathrm{E}[y]. \end{cases} \tag{3.11}$$

The choice of 0.1 as the distance to the quantization points from the mean is arbitrary.

The performances of the four algorithms are evaluated in terms of the probability of privacy outage and the average utility loss. First, each algorithm, with the exception of the maximum disclosure algorithm, is run in turn to determine the optimal actions associated

with every discretized state of the Markov Decision Process. Next, $10,000$ Monte-Carlo simulations of the LDS are carried out. In each simulation of the LDS, a sequence of observations, $z_1, z_2, \cdots, z_n$ are generated. When an observation z_k is generated, the Kalman filter equations are used to compute the actual state, s_k. For the value iteration, the pessimistic and the optimistic algorithms, the Bellman equation (3.6) is solved to determine the optimal action, C_k^*, associated with the state. For the maximum disclosure algorithm, the non-recursive equation (3.10) is solved to determine the optimal action, C_k^*, associated with the state. This process is repeated until the end of the finite time horizon. At the end of the finite time horizon, any violation of the privacy constraint: $d(\widehat{X}_{n|n}^{\tilde{Z},p}, \widehat{X}_{n|n}^{Z,p}) < \delta$, is checked, which concludes one simulation. After all simulations have been completed, the probability of privacy outage and the average utility loss are calculated using

$$\mathsf{P}(\text{outage}) = \frac{\text{number of constraint violations}}{\text{total number of simulations}}$$

and

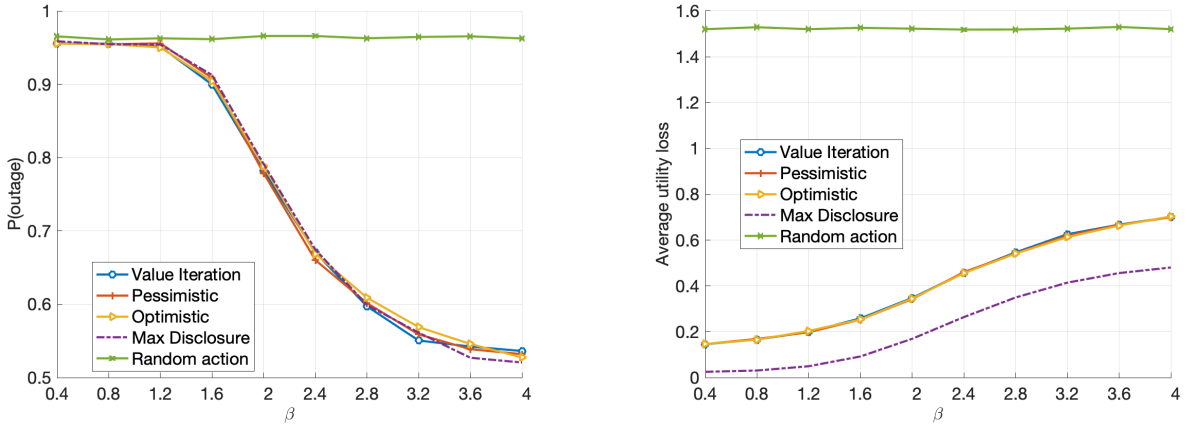
$$\text{Average utility loss} = \frac{\sum \left(\sum_{k=1}^{n} d(\widehat{X}_{k|k}^{Z,u}, \widehat{X}_{k|k}^{\tilde{Z},u}) \right)}{\text{total number of simulations}},$$

respectively. The experiment is repeated multiple times for different randomly generated samples of H and F.

From among multiple system models used in the simulations, three representative system models are selected that provide various insights on the performances of the four algorithms:

Model 1:

$$F = \begin{bmatrix} 0.06218 & 0.08373 & 0.12324 \\ 0.07386 & 0.04809 & 0.11332 \\ 0.13481 & 0.09099 & 0.06936 \end{bmatrix}, \quad H = \begin{bmatrix} 0.30780 & 0.77969 & 0.29994 \\ 0.37514 & 0.67681 & 0.45616 \\ 0.98334 & 0.94292 & 0.45824 \end{bmatrix}.$$



(a) Probability of privacy outage for different values of β

(b) Average utility loss for different values of β

Figure 3.2: Comparison of the performances of different algorithms in terms of the probability of privacy outage and the average utility loss for system model 1 ($\delta = 0.3$, $n = 5$ and $M = 1$).

Model 2:

$$F = \begin{bmatrix} 0.02712 & 0.01067 & 0.00073 \\ 0.00792 & 0.01444 & 0.01576 \\ 0.01029 & 0.00998 & 0.01596 \end{bmatrix}, \quad H = \begin{bmatrix} 0.02712 & 0.01067 & 0.00073 \\ 0.00792 & 0.01444 & 0.01576 \\ 0.01029 & 0.00998 & 0.01596 \end{bmatrix}.$$

Model 3:

$$F = \begin{bmatrix} 0.12246 & 0.51340 & 0.14024 \\ 0.45475 & 0.02484 & 0.53664 \\ 0.35442 & 0.70248 & 0.05728 \end{bmatrix}, \quad H = \begin{bmatrix} 0.75237 & 0.31551 & 0.85396 \\ 0.93524 & 0.03364 & 0.62274 \\ 0.01605 & 0.36138 & 0.05232 \end{bmatrix}.$$

Figure 3.2 shows the performances of the four algorithms in terms of the probability of privacy outage and the average utility loss across different values of β for system model 1. For reference, the performance of a naïve strategy in which the user randomly selects C_k from a uniform distribution in the unit interval at each time step k is also included. In Figure 3.2a and Figure 3.2b, we observe that all four algorithms consistently outperform the *random action* strategy across all values of β. Also, for all four algorithms, we observe

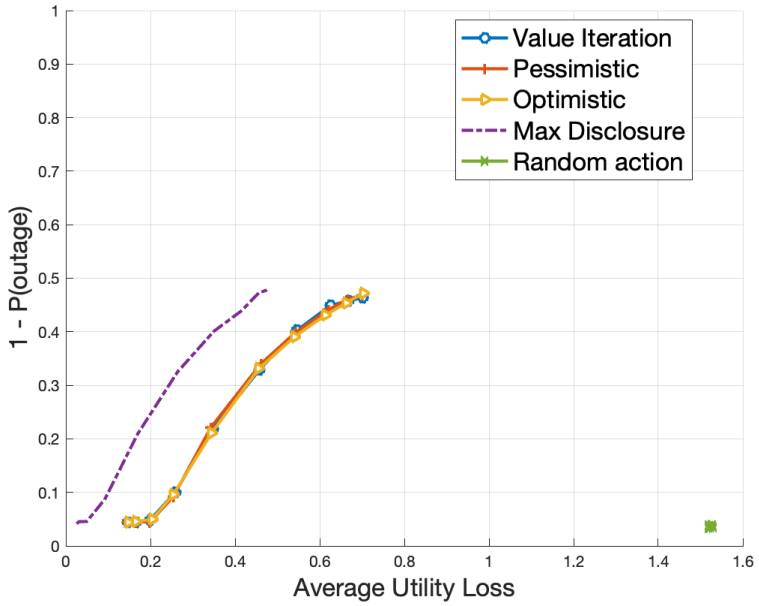


Figure 3.3: Privacy-utility tradeoff achieved by different strategies for system model 1 ($\delta = 0.3$, $n = 5$ and $M = 1$).

a decrease in the probability of privacy outage, and an increase in the average utility loss, as β increases. This observation is consistent with the intuition that larger values of β put more weight on the privacy requirement than the utility and therefore, result in a decrease in the probability of privacy outage and an increase in the utility loss. For the random action strategy, however, we observe that the probability of privacy outage (Figure 3.2a) and the average utility loss (Figure 3.2b) are both virtually constant across all values of β. This is expected as the random action strategy does not account for β in the selection of C_k.

In Figure 3.2, we also observe that the performances of the value iteration, the pessimistic and the optimistic algorithms are similar across different values of β. We consistently observed similar performances of the three algorithms for different random samples of H and F and for different values of n. In light of this, we conclude that the average performances of the three algorithms in terms of the probability of outage and the average utility loss are similar. Consequently, it may be desirable to use the pessimistic or the optimistic algorithm over the value iteration algorithm for speed benefits. However, it should be noted that the algorithms may perform differently for a more robust discretization strategy.

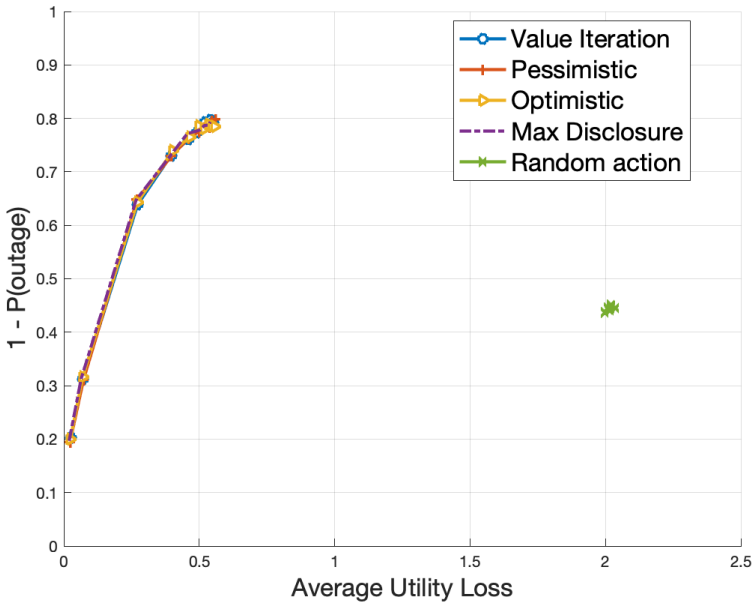


Figure 3.4: Privacy-utility tradeoff achieved by different strategies for system model 2 ($\delta = 0.3$, $n = 5$ and $M = 1$).

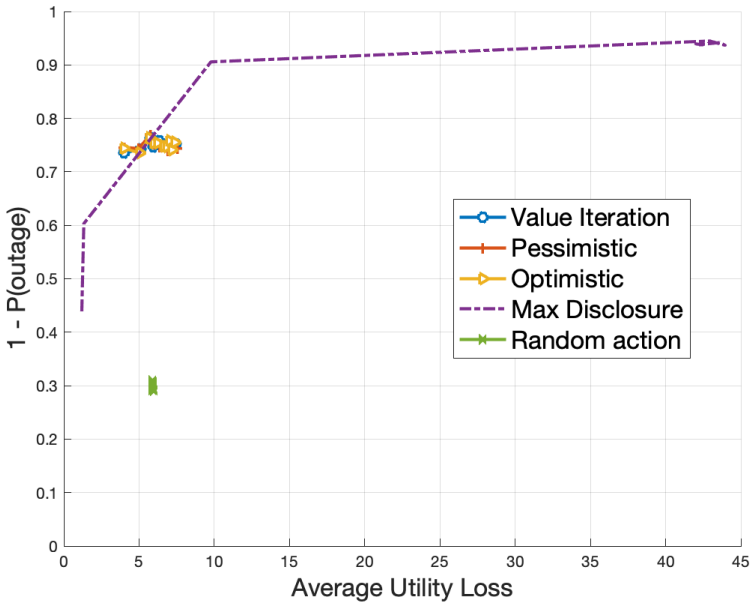


Figure 3.5: Privacy-utility tradeoff achieved by different strategies for system model 3 ($\delta = 0.3$, $n = 5$ and $M = 1$).

Figure 3.3, Figure 3.4 and Figure 3.5 highlight the privacy-utility tradeoff achieved by the maximum disclosure strategy/algorithm against the value iteration, pessimistic and optimistic algorithms for the three system models. For system model 1 (Figure 3.3), we observe that the maximum disclosure strategy significantly outperforms the other algorithms and achieves much better privacy-utility tradeoff. For system model 2 (Figure 3.4), we observe that the performance of the maximum disclosure strategy is similar to that of the value iteration, pessimistic and optimistic algorithms. Similarly, for system model 3 (Figure 3.5), the dynamic range for the privacy-utility tradeoff achieved by the maximum disclosure strategy is significantly higher than the other strategies– the higher dynamic range translates to more room for tuning the privacy-utility tradeoff.

The performance of the four algorithms, in general, depends on the system model. The relatively poor performances of the value iteration, pessimistic and optimistic algorithms against the maximum disclosure strategy across all representative system models can be attributed to the choice of the binary discretization strategy. For a more robust discretization strategy, we expect the three algorithms to perform better than the maximum disclosure strategy. However, increasing the quantization points in pursuit of a better discretization strategy significantly increases the computational requirements and may not be feasible for all systems. For high dimensional practical problems, maximum disclosure strategy is therefore the only computationally feasible option to solve the dynamic privacy problem.

Chapter 4

Privacy-Utility Tradeoff in a Generalized Setting

4.1 Introduction

This chapter considers the accumulated privacy leakage in a generalized dynamic setting. In contrast to the privacy model introduced in Chapter 3 which only focuses on the privacy leakage at a specific time in the future, the privacy model introduced in this chapter considers the accumulated privacy leakage over all finite future time steps. Essentially, the dynamic setting discussed in this chapter is an extension of the dynamic setting introduced in Chapter 3. This chapter also leverages a mix of compression and randomization techniques introduced in earlier chapters to develop a novel privacy mechanism that optimizes the perceived cumulative utility of a user while limiting the accumulated leakage over a finite period in the future. The main advantage of the enhanced privacy model is that it is more general, and therefore more practical, than the models introduced in the earlier chapters. Some practical applications of this model include privacy protection in smart grids, protection against gps location tracking, IoT device privacy etc. In all these applications, the service providers continuously collect the user's data; therefore, the privacy mechanism developed in this chapter is very fitting.

4.2 Problem Description

4.2.1 Problem Setting

We consider the problem of limiting the privacy leakage of a user who routinely shares her personal information with a service provider. We assume that the service provider uses the information shared by the user to offer personalized service to the user which constitutes the utility received by the user. In addition, the service provider may use the user's information for its own business benefits; however, we are only interested in the utility from the user's end.

The act of routinely sharing/disclosing one's personal information entails privacy loss over time. The spatial and temporal correlations between data disclosed at two different time instants may be exploited to reveal various sensitive information about a user such as their current location, web activity, religious beliefs etc. In addition, each disclosure contributes to an increasing privacy loss as a malicious party may be able to infer additional sensitive information from each subsequent disclosure. Therefore, from the privacy point of view, it is desirable to limit the cumulative leakage resulting from all disclosures.

To capture a real world setting, we assume that the user is interested in limiting her privacy leakage only for a finite period of time–for after this time period, the user deems her privacy less important. During this finite time period, the user seeks to cautiously disclose her personal information to a service provider so as not to leak too much information that she deems sensitive. At the same time, the user is interested in maximizing the cumulative utility of the disclosed information. As such, we are interested in developing a privacy mechanism that allows the user to maximize her perceived cumulative utility while limiting the accumulated leakage over time.

To quantify the information leaked about the sensitive information from the disclosed information, our metric of choice is *min-entropy leakage*. Min-entropy leakage captures the increase in probability of correctly guessing some secret in one try before and after observing some disclosed data that is correlated with the secret. This one-shot characterization of

privacy makes it a popular choice to capture privacy leakage [21, 58, 59, 60].

4.2.2 Min-Entropy Leakage

Consider an information channel (X, Z, P_{XZ}) where $X \in \mathbb{R}^N$ represents the input data (potentially, secret), $Z \in \mathbb{R}^M$ represents the output data and $P_{XZ}(x, z)$ represents the probability that the input is x and the output is z. Reńyi's min-entropy captures the uncertainty about a random variable in terms of the probability of correctly guessing the actual value of the random variable in one try. For discrete random variables (or vectors) X and Z, the initial entropy of X before observing Z is defined as [58]:

$$H_\infty(X) = -\log \max_{x \in \mathcal{X}} P_X(x) \tag{4.1}$$

where P_X represents the probability distribution over X such that $P_X(x)$ gives the probability that $X = x$.

Similarly, the posterior entropy of X conditioned on Z is defined as:

$$H_\infty(X|Z) = -\log \sum_{z \in \mathcal{Z}} P_Z(z) \max_{x \in \mathcal{X}} P_{X|Z}(x|z) \tag{4.2}$$

where $P_{X|Z}$ represents the probability distribution over X conditioned on Z such that $P_{X|Z}(x, z)$ gives the probability that $X = x$ given $Z = z$.

Now, min-entropy leakage from X to Z is defined as:

$$\begin{aligned} \mathcal{L}_{XZ} &= H_\infty(X) - H_\infty(X|Z) \\ &= \log \frac{\sum_{z \in \mathcal{Z}} P_Z(z) \max_{x \in \mathcal{X}} P_{X|Z}(x|z)}{\max_{x \in \mathcal{X}} P_X(x)} \end{aligned} \tag{4.3}$$

The notion of Rényi's entropy and min-entropy leakage has also been extended to the case of continuous random variables. For continuous random variables (or vectors) X and

Z, the initial entropy of X before observing Z is defined as [61]:

$$h_\infty(X) = -\log \max_{x \in \mathcal{X}} f_X(x) \tag{4.4}$$

where f_X represents the probability density function of X. Given an arbitrary guess $\widehat{x}$ for the actual value of X, $\int_{\widehat{x}-\epsilon}^{\widehat{x}+\epsilon} f_X(x)\mathrm{d}x$ gives the probability that the true value of x is within the ϵ interval of $\widehat{x}$. As $\epsilon \to 0$, this probability is maximized (up to a zero measure) at $\widehat{x} = \arg\max_{x \in \mathcal{X}} f_X(x)$ and therefore, $\arg\max_{x \in \mathcal{X}} f_X(x)$ constitutes the best guess for the value of X before observing Z.

Similarly, the posterior entropy of X conditioned on Z can be defined as:

$$h_\infty(X|Z) = -\log \int_{z \in \mathcal{Z}} f_Z(z) \max_{x \in \mathcal{X}} f_{X|Z}(x|z)\,\mathrm{dz} \tag{4.5}$$

where $f_{X|Z}$ represents the probability density function of X conditioned on Z.

Now, min-entropy leakage from X to Z can be defined as:

$$\begin{aligned} \mathcal{L}_{XZ} &= h_\infty(X) - h_\infty(X|Z) \\ &= \log \frac{\int_{z \in \mathcal{Z}} f_Z(z) \max_{x \in \mathcal{X}} f_{X|Z}(x|z)\,\mathrm{dz}}{\max_{x \in \mathcal{X}} f_X(x)} \end{aligned} \tag{4.6}$$

The expression for min-entropy leakage for continuous random vectors X and Z (4.6) is analogous to the expression for the min-entropy leakage for discrete random vectors X and Z (4.3).

4.2.3 System Model

In the domain of information privacy, it is common to model a user's data in terms of attributes/features. Some examples of features include the number of web pages visited by a user in a particular day, the user's current location, number of likes in the user's social media post etc. We assume that the user of interest has some sensitive features, represented by random vector X^p, that the user wishes to keep private. Some examples of sensitive features

include the user's current location, current web activity etc. Similarly, we assume that the user is willing to disclose the true value of some of her other features, represented by random vector X^o, in exchange for some utility. Some examples of such features may include rating given to a certain movie, number of movies watched in a particular period of time etc. In this setting, we assume that the user's utility is associated with a specific feature set, called the utility features, that is to be inferred by the service provider from the disclosed data. For instance, based on the user's ratings for previously watched movies, a service provider tries to infer the rating that the user would give to a new movie and potentially recommend it to the user. We denote the user's utility feature by X^u and assume that the actual value of X^u is to be inferred by the service provider and is potentially unknown to the user at the current time.

To model a dynamic setting, we assume that the true values of all features of the user evolve over time. Let the random vectors $X_j^p \in \mathbb{R}^{N_p \times 1}$ and $X_j^u \in \mathbb{R}^{N_u \times 1}$ represent the user's private and the utility features, respectively, at time-step j. Similarly, let $X_j^o \in \mathbb{R}^{N_o \times 1}$ represent all other features of the user that she is willing to disclose at time j. Let $X_j \in \mathbb{R}^{N \times 1}$ represent the entire feature vector of the user at time j such that $X_j = [X_j^{oT}, X_j^{pT}, X_j^{uT}]^T$.

We use a first-order Linear Dynamical System (LDS) equation to model the evolution of X_j over time. The state transition model in a LDS is given by:

$$X_j = F_j X_{j-1} + W_j \tag{4.7}$$

where $F_j \in \mathbb{R}^{N \times N}$ is the state-transition matrix and $W_j \in \mathbb{R}^{N \times 1}$ represents the process noise at time j. W_j is assumed to be a Gaussian process with noise components having zero mean and covariance specified by the matrix Q_j.

At every time step j, the privacy mechanism involves two distinct steps: first, the user's features $X_j^o \in \mathbb{R}^{N_o \times 1}$ are compressed to a lower-dimensional vector $\tilde{X}_j^o \in \mathbb{R}^{M \times 1} (M \leq N_o)$ using a linear transformation. Next, the compressed vector is randomized by means of the addition of Gaussian white noise samples drawn from a multivariate Gaussian distribution with zero mean and diagonal covariance matrix specified by R_j. As such, the privacy mech-

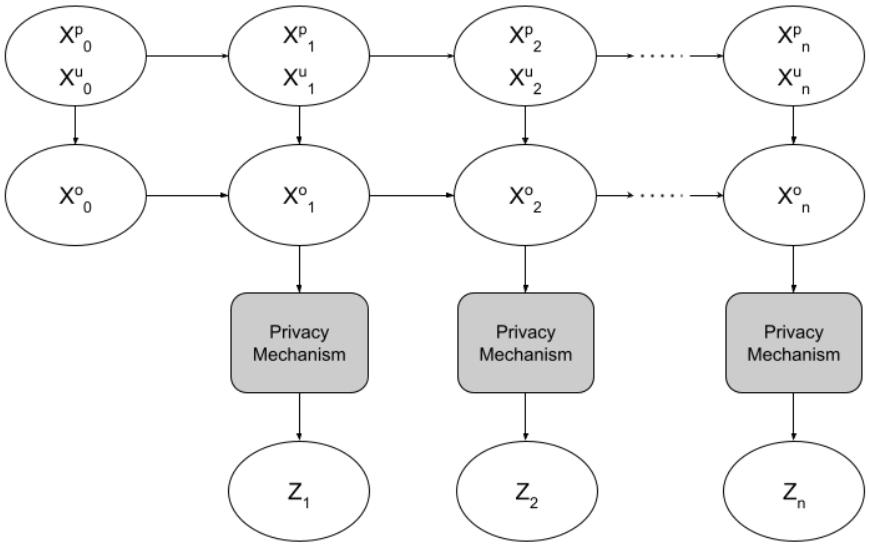


Figure 4.1: The dynamics of the privacy-utility tradeoff problem over a finite period of time.

anism at each time step j can be represented by the linear equation:

$$Z_j = H_j X_j + V_j \tag{4.8}$$

where $H_j \in \mathbb{R}^{M \times N}$ is the compression matrix and $V_j \in \mathbb{R}^{M \times 1}$ represents the Gaussian noise to be added before each disclosure. The elements in the i^{th} column of H_j are defined to be zeros when $i > N_o$. This structure of H removes any information about X_j^p (which the user wishes to keep private) and X_j^u (which the user may not yet know at time j) from the disclosed data Z_j. The dynamics of the problem are shown in Figure 4.1. The privacy mechanism involves finding the optimal H_j (for compression) and R_j (for randomization) that satisfy certain privacy and utility requirements. It is assumed that both an adversary and the utility provider have complete knowledge about the system dynamics as well as the privacy mechanism.

It should be noted that privacy protection using compression may not be practicable in certain settings. For instance, in the context of database, compression entails elimination of certain attributes from the database which may not be desirable. In such settings, we set $M = N_o$ and define $H_j \in \mathbb{R}^{N_o \times N}$ such that the element in the p^{th} row and the q^{th} column is

given by

$$H_j^{p,q} = \begin{cases} 1 & \text{when p = q} \\ 0 & \text{otherwise.} \end{cases}$$

The structure of the matrix H reflects the mapping from X_j to X_j^o at every time step j. Essentially, with this instantiation, the privacy mechanism in (4.8) simplifies to a randomization mechanism which involves randomizing X_j^o with no compression.

4.2.4 Problem Formulation

From the privacy viewpoint, at present time k, it is desirable to limit the information leaked about X_k^p resulting from all disclosures up until time k, i.e. $Z_1, Z_2, \cdots, Z_k$. At the same time, from the utility viewpoint, it is desirable to retain maximum amount of information about X_k^u in Z_k in addition to the information that can be inferred about X_k^u from $Z_1, Z_2, \cdots, Z_{k-1}$. In a single disclosure setting, a potential formulation of the optimization problem would be:

$$\arg\max_{R_k, H_k} \mathcal{L}_{X_k^u Z^k} \tag{4.9}$$

subject to:

$$\mathcal{L}_{X_k^p Z^k} \leq \delta \tag{4.10}$$

where $Z^k = \{Z_1, Z_2, \cdots, Z_k\}$, and δ specifies the maximum acceptable privacy leakage at time k quantified in terms of min-entropy leakage. However, we are more interested in optimizing the privacy and utility trade-off over multiple disclosures which involves regulating the present disclosure as well as planning for all future disclosures. This setting makes the privacy requirement more stringent than for the single disclosure setting.

In consideration of the multi-disclosure setting, we seek to find the optimal sequence of compression matrices $H_k, H_{k+1}, \cdots, H_n$ and of noise covariance matrices $R_k, R_{k+1}, \cdots, R_n$ at

present time k that allows a user to maximize the sum of their present and future perceived utilities while limiting the sum of their present and future privacy leakages up until the end of the specified finite time period n. The sum of the present and future perceived utilities constitutes the cumulative utility of the user whereas the sum of the present and future privacy leakages constitutes the accumulated privacy leakage over the finite time period. The resulting optimization problem can be mathematically expressed as:

$$\underset{\substack{R_k,R_{k+1},\cdots,R_n,\\ H_k,H_{k+1},\cdots,H_n}}{\arg\max} \sum_{j=k}^{n} \alpha_j \mathcal{L}_{X_j^u Z^j} \tag{4.11}$$

subject to:

$$\sum_{j=k}^{n} \beta_j \mathcal{L}_{X_j^p Z^j} \leq \delta. \tag{4.12}$$

The optimization problem formulated in (4.9) is a specific case ($n = k$ and $\alpha_k = \beta_k = 1$) of the optimization problem formulated in (4.11). The parameter $\alpha_j \in [0, 1]$ determines the relative value of the user's utility at time j. For instance, if $\alpha_k = 1$ and $\alpha_{k+1} = \alpha_{k+2} = \cdots = \alpha_n = 0$, then the user only values her present utility and disregards all her future utilities. On the other hand, if $\alpha_k = \alpha_{k+1} = \cdots = \alpha_n = 1$, then the user values all her future utilities equal to her present utility. Similarly, the parameter $\beta_j \in [0, 1]$ determines the relative value of the user's privacy at time j.

While $Z^j = \{Z_1, Z_2, \cdots, Z_j\}$ represents the sequence of random vectors representing the disclosed data up to time j, let $z^j = \{z_1, z_2, \cdots, z_j\}$ represent the sequence of realizations of $\{Z_1, Z_2, \cdots, Z_j\}$. Then,

$$\mathcal{L}_{X_j^u Z^j} = \log \frac{\int_{z^j \in \mathcal{Z}} f_{Z^j}(z^j) \max_{x_j^u \in \mathcal{X}^u} f_{X_j^u|Z^j}(x_j^u|z^j)\, \mathrm{dz}^j}{\max_{x_j^u \in \mathcal{X}^u} f_{X_j^u|Z^{j-1}}(x_j^u|z^{j-1})} \tag{4.13}$$

and

$$\mathcal{L}_{X_j^p Z^j} = \log \frac{\int_{z^j \in \mathcal{Z}} f_{Z^j}(z^j) \max_{x_j^p \in \mathcal{X}^p} f_{X_j^p|Z^j}(x_j^p|z^j)\, \mathrm{d}z^j}{\max_{x_j^p \in \mathcal{X}^p} f_{X_j^p|Z^{j-1}}(x_j^p|z^{j-1})} \tag{4.14}$$

where $\mathcal{Z}$ represents the space of Z^j, $\mathcal{X}^u$ represents the space of X_j^u and $\mathcal{X}^p$ represents the space of X_j^p.

The joint probability distribution over all states and observations/disclosures up to time k can be written as

$$\begin{aligned}
&\mathsf{P}(X_0, X_1, \cdots, X_j, Z_1, \cdots, Z_j) \quad (0 \le j \le n) \\
&= \mathsf{P}(X_0)\, \mathsf{P}(X_1|X_0)\, \mathsf{P}(Z_1|X_1)\, \mathsf{P}(X_2|X_1)\, \mathsf{P}(Z_2|X_2) \cdots \mathsf{P}(X_j|X_{j-1})\, \mathsf{P}(Z_j|X_j) \\
&= \mathsf{P}(X_0) \prod_{i=1}^{j} \mathsf{P}(X_i|X_{i-1})\, \mathsf{P}(Z_i|X_i)
\end{aligned}$$

Here,

$$\begin{aligned}
\mathrm{E}[X_i|X_{i-1}] &= \mathrm{E}[FX_{i-1} + W_i|X_{i-1}] = FX_{i-1}, \\
\mathrm{Cov}[X_i|X_{i-1}] &= \mathrm{Cov}[FX_{i-1} + W_i|X_{i-1}] = Q_i.
\end{aligned}$$

Therefore, if X_j is a Gaussian process, then $\mathsf{P}(X_j|X_{j-1}) \sim \mathcal{N}(FX_{j-1}, Q_j)$. Similarly, $\mathsf{P}(Z_j|X_j) \sim \mathcal{N}(HX_j, R_j)$.

4.2.5 From Bayesian Estimation to Kalman Filter

For the Gaussian process X_j, the problem of estimating the actual value of X_j (and consequently, X_j^p and X_j^u) at time j given the observations $z_1, z_2, \cdots, z_j$ nicely relates to the Kalman filter estimation [62]. It is known that $\mathsf{P}(X_j|z^j) \sim \mathcal{N}(\widehat{x}_{j|j}, P_{j|j})$ where $\widehat{x}_{j|j}$ represents the Kalman filter estimate of X_j given the observations $z_1, z_2, \cdots, z_j$ and $P_{j|j}$ represents the error covariance associated with the estimate.

Notice that

$$\mathsf{P}(X_j, X_{j-1}|z^{j-1}) = \mathsf{P}(X_{j-1}|z^{j-1})\,\mathsf{P}(X_j|X_{j-1}, z_{j-1}) = \mathsf{P}(X_{j-1}|z^{j-1})\,\mathsf{P}(X_j|X_{j-1}),$$
$$\mathsf{P}(X_j|z^{j-1}) = \int_{x_{j-1}\in\mathcal{X}} \mathsf{P}(x_{j-1}|z^{j-1})\,\mathsf{P}(X_j|x_{j-1})\mathrm{d}x_{j-1}.$$

By analogy with the probability distribution associated with the Kalman filter update step, $\mathsf{P}(X_j|z^{j-1}) \sim \mathcal{N}(\widehat{x}_{j|j-1}, P_{j|j-1})$ where $\widehat{x}_{j|j-1}$ represents the Kalman filter estimate of X_j given the observations $z_1, z_2, \cdots, z_{j-1}$ and $P_{j|j-1}$ represents the error covariance associated with the estimate.

Now, for a Gaussian random vector $X \in \mathbb{R}^N$ with mean μ_X and covariance matrix Σ_X,

$$\max_{x\in\mathcal{X}} f_X(x) = f_X(\mu_X) = \frac{1}{(2\pi)^{N/2}|\Sigma_X|^{1/2}}.$$

Since X_j^u and X_j^p are both Gaussian processes, we have

$$\max_{x_j^u\in\mathcal{X}^u} f_{X_j^u|Z^{j-1}}(x_j^u|z^{j-1}) = \frac{1}{(2\pi)^{N_u/2}|P_{j|j-1}^u|^{1/2}}$$

and

$$\max_{x_j^u\in\mathcal{X}^u} f_{X_j^u|Z^{j}}(x_j^u|z^{j}) = \frac{1}{(2\pi)^{N_u/2}|P_{j|j}^u|^{1/2}}$$

where $P_{j|j-1}^u$ and $P_{j|j}^u$ represent the error covariance matrices associated with the Kalman filter estimate of X_j^u given the observations z^{j-1} and z^j, respectively.

Similarly,

$$\max_{x_j^p \in \mathcal{X}^p} f_{X_j^p|Z^{j-1}}(x_j^p|z^{j-1}) = \frac{1}{(2\pi)^{N_p/2}|P_{j|j-1}^p|^{1/2}}$$

and

$$\max_{x_j^p \in \mathcal{X}^p} f_{X_j^p|Z^{j}}(x_j^p|z^{j}) = \frac{1}{(2\pi)^{N_p/2}|P_{j|j}^p|^{1/2}}$$

where $P_{j|j-1}^p$ and $P_{j|j}^p$ represent the error covariance matrices associated with the Kalman filter estimate of X_j^p given the observations z^{j-1} and z^j, respectively.

Now, (4.13) and (4.14) simplify to

$$\mathcal{L}_{X_j^u Z^j} = \log \frac{\int_{z^j \in \mathcal{Z}} f_{Z^j}(z^j) \frac{1}{(2\pi)^{Nu/2}|P_{j|j}^u|^{1/2}} \, \mathrm{d}z^j}{\frac{1}{(2\pi)^{Nu/2}|P_{j|j-1}^u|^{1/2}}}$$

and

$$\mathcal{L}_{X_j^p Z^j} = \log \frac{\int_{z^j \in \mathcal{Z}} f_{Z^j}(z^j) \frac{1}{(2\pi)^{N_p/2}|P_{j|j}^p|^{1/2}} \, \mathrm{d}z^j}{\frac{1}{(2\pi)^{N_p/2}|P_{j|j-1}^p|^{1/2}}}$$

Since $P_{j|j}^u$ and $P_{j|j}^p$ are not functions of z^j and $\int_{z^j \in \mathcal{Z}} f_{Z^j}(z^j) \, \mathrm{d}z^j = 1$,

$$\mathcal{L}_{X_j^u Z^j} = \log \frac{|P_{j|j-1}^u|^{1/2}}{|P_{j|j}^u|^{1/2}} = \frac{1}{2} \log \frac{|P_{j|j-1}^u|}{|P_{j|j}^u|}$$

and

$$\mathcal{L}_{X_j^p Z^j} = \log \frac{|P_{j|j-1}^p|^{1/2}}{|P_{j|j}^p|^{1/2}} = \frac{1}{2} \log \frac{|P_{j|j-1}^p|}{|P_{j|j}^p|}.$$

At this point, it should be apparent that for the Gaussian process X_j (and consequently, X_j^p and X_j^u), the min-entropy leakage at time j does not not depend on the actual observations $z_1, z_2, \cdots, z_j$ but rather depends on the system parameters, the second order statistics of the system variables, the compression matrices $H_1, H_2, \cdots, H_j$, and the noise covariance

matrices $R_1, R_2, \cdots, R_j$. This property facilitates the design of a privacy mechanism in which the optimal $H_k, R_k, H_{k+1}, R_{k+1}, \cdots, H_n, R_n$ for all future disclosures can be computed offline at present time k.

4.3 Experimental Evaluations

To evaluate the performance of our privacy mechansim in a dynamic setting, we simulate a LDS with $N_o = 3$, $N_p = 1$ and $N_u = 1$ and assume that F_j is time invariant with elements sampled independently from a uniform distribution in the unit interval. Further, we assume that X_j is a zero mean Gaussian process and W_j is a standard Gaussian white noise process. The elements of F are normalized such that its eigenvalues lie within a unit circle which ensures that the LDS is stable. The performance is evaluated in terms of the cumulative utility and accumulated privacy leakage over a finite period of time specified by n. To model a user who values her utility and privacy at all time steps equally, we set $\alpha_j = 1$ and $\beta_j = 1$ $\forall j$ in all of our experiments.

From among multiple system models used in the experiments, some of the models are presented below for reference. We follow the convention that I represents an identity matrix whose size can be inferred from the context.

System Model 1:

$$F = \begin{bmatrix} 0.07586 & 0.22054 & 0.47590 & 0.58715 & 0.26984 \\ 0.58150 & 0.29952 & 0.32197 & 0.35355 & 0.70091 \\ 0.42221 & 0.40270 & 0.23323 & 0.50196 & 0.38292 \\ 0.51853 & 0.82650 & 0.96737 & 0.64124 & 0.01898 \\ 0.33712 & 0.44125 & 0.84970 & 0.31183 & 0.35277 \end{bmatrix},$$

System Model 2:

$$F = \begin{bmatrix} 0.44846 & 0.76046 & 0.00718 & 0.22204 & 0.57138 \\ 0.58215 & 0.26733 & 0.38272 & 0.68266 & 0.84095 \\ 0.16258 & 0.09929 & 0.95259 & 0.94049 & 0.63785 \\ 0.56447 & 0.87370 & 0.21176 & 0.03405 & 0.81411 \\ 0.52462 & 0.61846 & 0.56026 & 0.16526 & 0.19117 \end{bmatrix},$$

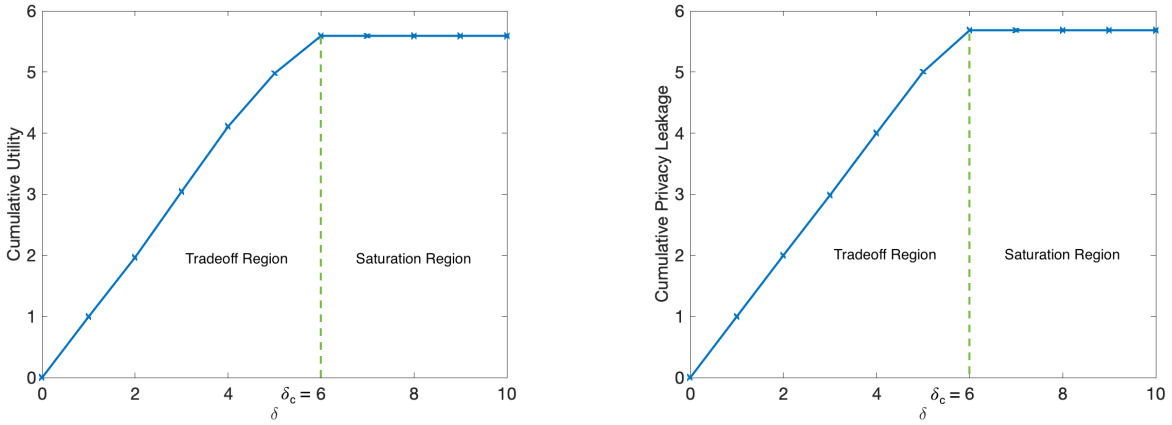
System Model 3:

$$F = \begin{bmatrix} 0.93844 & 0.63047 & 0.34832 & 0.76994 & 0.44301 \\ 0.42264 & 0.43039 & 0.87913 & 0.26645 & 0.27141 \\ 0.88704 & 0.31267 & 0.00057 & 0.57640 & 0.64625 \\ 0.88285 & 0.84604 & 0.54653 & 0.46600 & 0.89349 \\ 0.50608 & 0.91276 & 0.99671 & 0.69914 & 0.29007 \end{bmatrix},$$

System Model 4:

$$F = I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We first consider the randomization aspect of our privacy mechanism to evaluate its performance in a setting where compression is impractical, as in the case of privacy protection in a database consisting of user attributes. Figure 4.2 shows the performance of our privacy mechanism across different values of δ for System Model 1. In the figure, we observe that as the value of δ increases, we see an increase (up to a threshold) in both the cumulative utility (Figure 4.2a) and the cumulative privacy leakage (Figure 4.2b). This is justified



(a) Cumulative utility for different values of δ

(b) Cumulative privacy leakage for different values of δ

Figure 4.2: Performance of the privacy mechanism across different values of δ for System Model 1 ($n = 5$).

by the fact that larger values of δ imply that the user is willing to accept higher privacy leakage and therefore the privacy mechanism optimizes the privacy-utility tradeoff to yield higher cumulative utility for the user. However, this increase in the cumulative utility and the accumulated privacy leakage is only observed up to a particular value of δ–called the critical value–represented by δ_c. When δ is increased further, we see virtually no change in the cumulative utility and the privacy leakage afforded by the privacy mechanism suggesting that a local maximum is reached for the objective function when $\delta \geq \delta_c$. The region in which different privacy-utility tradeoffs can be achieved by varying the values of δ is referred to as the *tradeoff region*. Similarly, the region in which the cumulative utility and the cumulative privacy leakage remains constant across different values of δ is referred to as the *saturation region*.

An important intuition is that the process noise W_j also induces some randomness in the observations. Therefore, we expect the choice of Q (which determines the process noise) to have an impact on the size of the tradeoff region. In turn, this will directly influence the dynamic range achieved by the privacy mechanism. Indeed, in Figure 4.3, we observe that as the diagonal components of Q are increased (which implies higher variances of the process noise), the dynamic range for the privacy-utility tradeoff decreases. In fact, for $Q = 0.01I$,

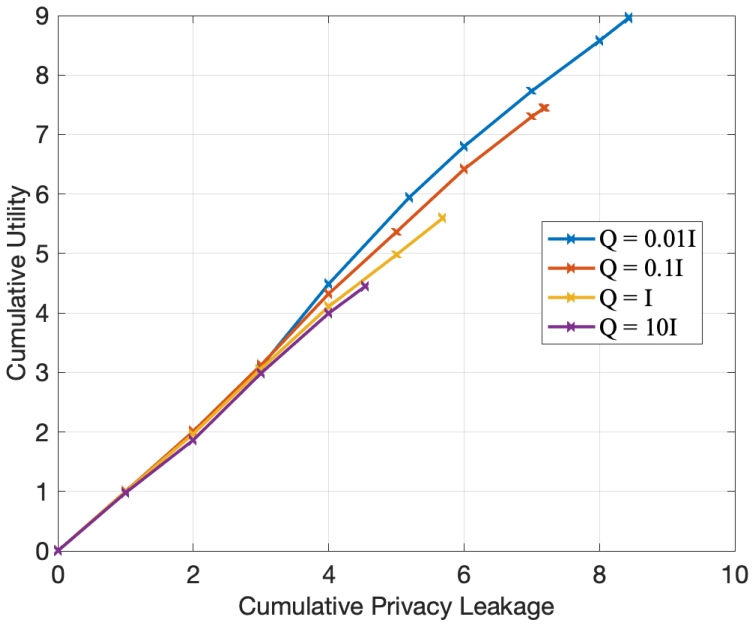


Figure 4.3: Privacy-utility tradeoff for different covariance matrices of the process noise for System Model 1 ($n = 5$).

the privacy mechanism offers much better privacy-utility tradeoff. This can be justified by the fact that, when the variances of the process noise components are small, the randomness in the observations is mostly due to the randomization noise V_j which depends on, and is controlled by, R_j.

We made similar observations regarding the performance of the randomization mechanism across different LDS models and with the number of features increased to $N_o = 10$, $N_p = 2$ and $N_u = 3$. In conclusion, the randomization aspect of our privacy mechanism facilitates in achieving promising privacy-utility tradeoffs in various dynamic settings.

Next, we evaluate how the compression aspect of our privacy mechanism compares against the randomization aspect. Figure 4.4 shows the privacy-utility tradeoff due to compression versus the privacy-utility due to randomization for a simple system represented by System Model 4. For reference, we have also included the tradeoff when no privacy mechanism is in place. As can be seen in the figure, for $Q = 0.1I$, the tradeoff due to compression is significantly better than the tradeoff due to randomization. We observed similar results for larger variances of the process noise components and across different samples of F. However,

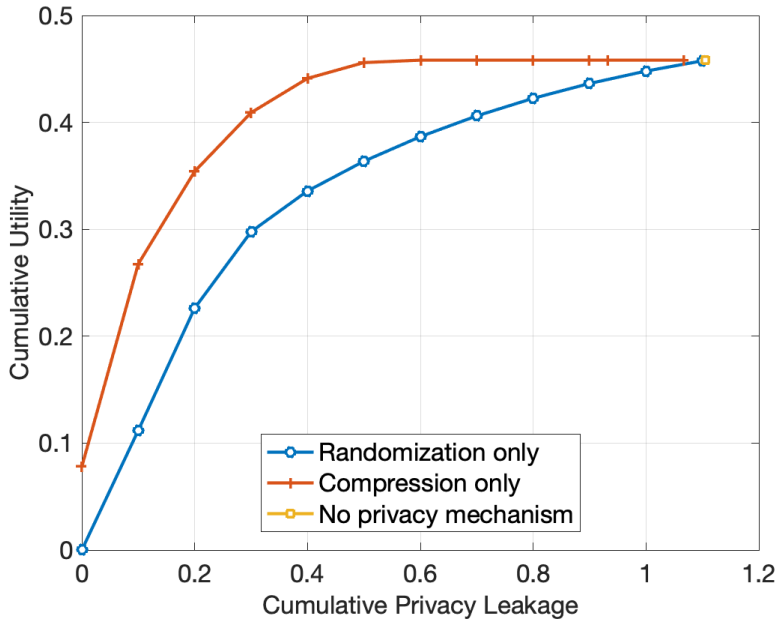


Figure 4.4: Privacy-utility tradeoff due to randomization compared against the privacy-utility tradeoff due to compression for System Model 4 with $Q = 0.1I$ $(n = 5, M = 1)$.

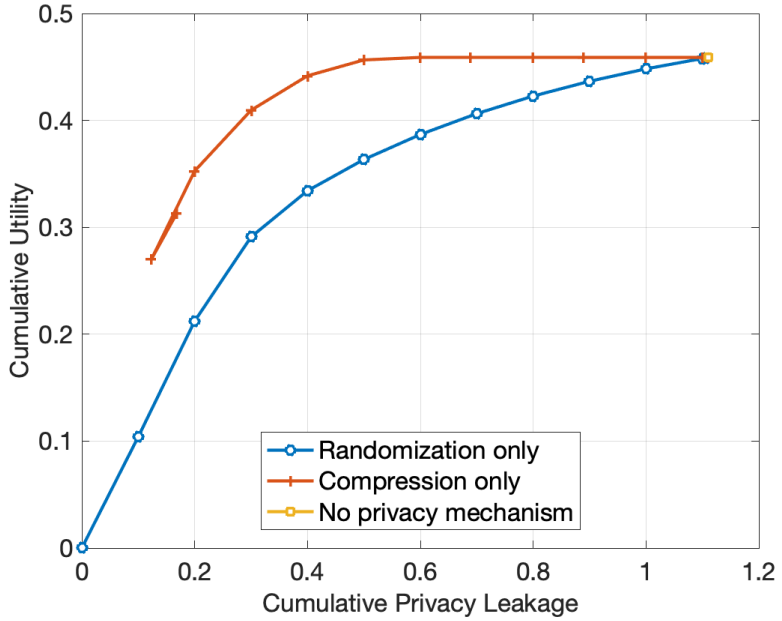
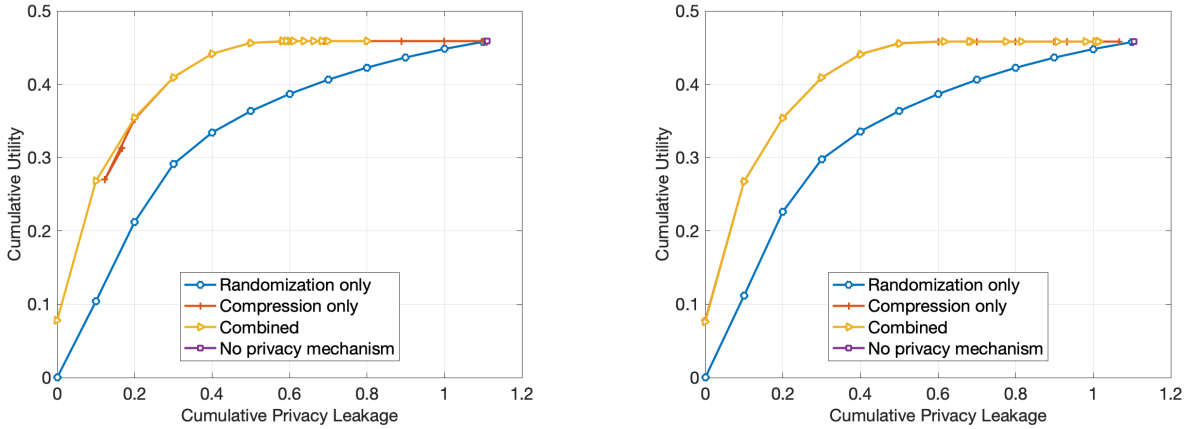


Figure 4.5: Privacy-utility tradeoff due to randomization compared against the privacy-utility tradeoff due to compression for System Model 4 with $Q = 0.001I$ $(n = 5, M = 1)$.



(a) $Q = 0.001I$ (b) $Q = 0.1I$

Figure 4.6: Comparison of the performance of different techniques in optimizing the privacy-utility tradeoff across different choices of Q for System Model 4 ($n = 5, M = 1$).

when the variances of the process noise components are decreased (Figure 4.5), we observe a missing tradeoff region in between the cumulative leakage values of 0 and 0.1, inclusive. This is because when $\delta = 0$ and $\delta = 0.1$ (which capture strong privacy requirements), no choice of the compression matrix is found to satisfy the privacy constraint which results in the violation of the constraint at these values. However, when the variances of the process noise components are large (Figure 4.4), the compression mechanism leverages the randomness due to the process noise to meet the privacy constraints in this region. Intuitively, therefore, by combining the compression and randomization techniques together, an optimal privacy mechanism can be designed that achieves the best privacy-utility tradeoff for any choice of the system parameters. This is supported by the observations made in Figure 4.6a and Figure 4.6b which shows that the combined technique is at least as good as the randomization and the compression techniques across two notably different choices of Q. In our experiments, this observation was consistent across various samples of F and across various choices of Q. In conclusion, the privacy mechanism given in (4.8) is not only general but also optimal and offers the best privacy-utility tradeoff in various dynamic settings.

Chapter 5

Conclusion

User privacy in social media, streaming services, and other online platforms have recently started to become a hot topic as the general population has started realizing the sheer scale of potential privacy breach surrounding the use and misuse of their information. The main challenge in ensuring user privacy on such platforms is the expectation of a certain utility from the platform – the very reason the users share their information in the first place. Seeking absolute privacy and maximum utility when actively engaging in online platforms are contradictory goals as the fundamental nature of the problem makes it infeasible to maximize the privacy without sacrificing some utility and vice-versa.

The scope and the meaning of privacy and utility can substantially differ based on the context and the setting. The main motivation of this dissertation is to improve on the existing static models capturing the precise notions of privacy and utility and more importantly, to consider the intrinsic dynamic nature of privacy. This motivation drove us to introducing several formulations of the privacy-utility tradeoff problem carefully tailored based on the context, setting and scope. We further designed several privacy mechanisms with diverse privacy goals and experimentally validated their efficacy. We also developed several algorithms to solve the tradeoff problem in various settings and discussed different strategies for both the short-term and the long-term privacy protection.

5.1 Review of the Contributions

Chapter 2 considered the privacy-utility tradeoff in a static setting. In this setting, we presented a novel utility model in which utility is associated with a small subset of user attributes, referred to as utility attributes. We imposed an additional constraint on the utility model that captured each user's requirement on maximum acceptable loss in utility per unit gain in privacy. Under the new utility model, we formulated a novel privacy-utility tradeoff problem and presented a heuristic greedy algorithm with polynomial time and space complexity to solve the problem. We showed that the performance of the greedy algorithm is comparable to other popular algorithms wherever applicable. We presented experimental results on the performance of the greedy algorithm on both synthetic and real-world datasets. Using a naïve algorithm as a reference, we demonstrated that the greedy algorithm performs very well in achieving a good privacy-utility tradeoff.

In Chapter 3, we considered a dynamic setting in which users continuously disclose their personal information over time resulting in an accumulated leakage of their private information. In the dynamic setting, we formulated a novel privacy-utility tradeoff problem capturing the dynamics of privacy leakage over a finite time period. Under our dynamic privacy-utility tradeoff model, we investigated different strategies that allow a user to maximize their net utility subject to certain future privacy requirements. We discussed challenges associated with finding optimal strategies for real world problems and motivated sub-optimal algorithms to solve the tradeoff problem. Via extensive performance evaluations on synthetic datasets, we demonstrated that despite being sub-optimal, the proposed algorithms perform extremely well in achieving a good privacy-utility tradeoff. We also formulated a simpler dynamic privacy problem that is computationally less intensive to solve but conserves the essence of the original problem.

In Chapter 4, we extended the dynamic setting to consider not only the leakage at a specific time in the future but the accumulated leakage over all finite future time steps. We developed a general dynamic privacy model designed to work in both static and various dynamic settings. Using a mix of randomization and compression techniques, we designed

a novel privacy mechanism that limits the accumulated privacy leakage over a finite time period while maximizing the cumulative utility of the shared information. Via experimental evaluations, we showed that our dynamic privacy mechanism is extremely effective in optimizing the privacy-utility tradeoff in various dynamic settings.

Bibliography

[1] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[2] Lalitha Sankar, S Raj Rajagopalan, and H Vincent Poor. Utility-privacy tradeoffs in databases: An information-theoretic approach. *IEEE Transactions on Information Forensics and Security*, 8(6):838–852, 2013.

[3] Xingze He, Xinwen Zhang, and C-C Jay Kuo. A distortion-based approach to privacy-preserving metering in smart grids. *IEEE Access*, 1:67–78, 2013.

[4] Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Transactions on Information Theory*, 62(2):925–951, 2015.

[5] Weina Wang, Lei Ying, and Junshan Zhang. On the relation between identifiability, differential privacy, and mutual-information privacy. *IEEE Transactions on Information Theory*, 62(9):5018–5029, 2016.

[6] Kousha Kalantari, Lalitha Sankar, and Oliver Kosut. On information-theoretic privacy with general distortion cost functions. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2865–2869. IEEE, 2017.

[7] Jiachun Liao, Oliver Kosut, Lalitha Sankar, and Flavio P Calmon. Privacy under hard distortion constraints. In *2018 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2018.

[8] Kousha Kalantari, Lalitha Sankar, and Anand D Sarwate. Robust privacy-utility tradeoffs under differential privacy and hamming distortion. *IEEE Transactions on Information Forensics and Security*, 13(11):2816–2830, 2018.

[9] Konstantinos Diamantaras and Sun-Yuan Kung. Data privacy protection by kernel subspace projection and generalized eigenvalue decomposition. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.

[10] Sun-Yuan Kung. Compressive privacy: From information\/estimation theory to machine learning [lecture notes]. *IEEE Signal Processing Magazine*, 34(1):94–112, 2017.

[11] Sun-Yuan Kung, Thee Chanyaswad, J Morris Chang, and Peiyuan Wu. Collaborative pca/dca learning methods for compressive privacy. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(3):1–18, 2017.

[12] Sun-Yuan Kung. A compressive privacy approach to generalized information bottleneck and privacy funnel problems. *Journal of the Franklin Institute*, 355(4):1846–1872, 2018.

[13] Yang Song, Chong Xiao Wang, and Wee Peng Tay. Compressive privacy for a linear dynamical system. *IEEE Transactions on Information Forensics and Security*, 15:895–910, 2020. doi: 10.1109/TIFS.2019.2930366.

[14] Mário S Alvim, Miguel E Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: on the trade-off between utility and information leakage. In *International Workshop on Formal Aspects in Security and Trust*, pages 39–54. Springer, 2011.

[15] Chunyong Yin, Jinwen Xi, Ruxia Sun, and Jin Wang. Location privacy protection based on differential privacy strategy for big data in industrial internet of things. *IEEE Transactions on Industrial Informatics*, 14(8):3628–3636, 2017.

[16] Yin Yang, Zhenjie Zhang, Gerome Miklau, Marianne Winslett, and Xiaokui Xiao. Differential privacy in data publication and analysis. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 601–606. ACM, 2012.

[17] Tianqing Zhu, Ping Xiong, Gang Li, and Wanlei Zhou. Correlated differential privacy: Hiding information in non-iid data set. *IEEE Transactions on Information Forensics and Security*, 10(2):229–242, 2014.

[18] Xiaotong Wu, Taotao Wu, Maqbool Khan, Qiang Ni, and Wanchun Dou. Game theory based correlated privacy preserving analysis in big data. *IEEE Transactions on Big Data*, 2017.

[19] Ali Makhdoumi and Nadia Fawaz. Privacy-utility tradeoff under statistical uncertainty. In *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1627–1634. IEEE, 2013.

[20] S Raj Rajagopalan, Lalitha Sankar, Soheil Mohajer, and H Vincent Poor. Smart meter privacy: A utility-privacy framework. In *2011 IEEE international conference on smart grid communications (SmartGridComm)*, pages 190–195. IEEE, 2011.

[21] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. *Electronic Notes in Theoretical Computer Science*, 249:75–91, 2009.

[22] Geoffrey Smith. On the foundations of quantitative information flow. In *International Conference on Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.

[23] Farhad Farokhi and Henrik Sandberg. Fisher information as a measure of privacy: Preserving privacy of households with smart meters using batteries. *IEEE Transactions on Smart Grid*, 9(5):4726–4734, 2017.

[24] Farhad Farokhi and Henrik Sandberg. Optimal privacy-preserving policy using constrained additive noise to minimize the fisher information. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2692–2697. IEEE, 2017.

[25] Chong Xiao Wang, Yang Song, and Wee Peng Tay. Preserving parameter privacy in sensor networks. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1316–1320. IEEE, 2018.

[26] Ibrahim Issa, Sudeep Kamath, and Aaron B Wagner. An operational measure of information leakage. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 234–239. IEEE, 2016.

[27] Jiachun Liao, Lalitha Sankar, Flavio P Calmon, and Vincent YF Tan. Hypothesis testing under maximal leakage privacy constraints. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 779–783. IEEE, 2017.

[28] Cheuk Ting Li and Abbas El Gamal. Maximal correlation secrecy. *IEEE Transactions on Information Theory*, 64(5):3916–3926, 2018.

[29] Shahab Asoodeh, Fady Alajaji, and Tamás Linder. On maximal correlation, mutual information and data privacy. In *2015 IEEE 14th Canadian Workshop on Information Theory (CWIT)*, pages 27–31. IEEE, 2015.

[30] Murat A Erdogdu and Nadia Fawaz. Privacy-utility trade-off under continual observation. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1801–1805. IEEE, 2015.

[31] Tiancheng Li and Ninghui Li. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–526. ACM, 2009.

[32] Lalitha Sankar, S Raj Rajagopalan, and H Vincent Poor. A theory of utility and privacy of data sources. In *2010 IEEE International Symposium on Information Theory*, pages 2642–2646. IEEE, 2010.

[33] Flávio du Pin Calmon and Nadia Fawaz. Privacy against statistical inference. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1401–1408. IEEE, 2012.

[34] Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Médard. From the information bottleneck to the privacy funnel. In *2014 IEEE Information Theory Workshop (ITW 2014)*, pages 501–505. IEEE, 2014.

[35] Bin Zhou, Yi Han, Jian Pei, Bin Jiang, Yufei Tao, and Yan Jia. Continuous privacy preserving publishing of data streams. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 648–659, 2009.

[36] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[37] Yang Song, Chong Xiao Wang, and Wee Peng Tay. Privacy-aware kalman filtering. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4434–4438, 2018. doi: 10.1109/ICASSP.2018.8462600.

[38] Chandra Sharma, Bishwas Mandal, and George Amariucai. A practical approach to navigating the tradeoff between privacy and precise utility. In *ICC 2021-IEEE International Conference on Communications*, pages 1–6. IEEE, 2021.

[39] Louis Leung. Generational differences in content generation in social media: The roles of the gratifications sought and of narcissism. *Computers in Human Behavior*, 29(3): 997–1006, 2013.

[40] Yongjun Sung, Jung-Ah Lee, Eunice Kim, and Sejung Marina Choi. Why we post selfies: Understanding motivations for posting pictures of oneself. *Personality and Individual Differences*, 97:260–265, 2016.

[41] Dong Liu and Roy F Baumeister. Social networking online and personality of self-worth: A meta-analysis. *Journal of Research in Personality*, 64:79–89, 2016.

[42] Lalitha Sankar, S Raj Rajagopalan, Soheil Mohajer, and H Vincent Poor. Smart meter privacy: A theoretical framework. *IEEE Transactions on Smart Grid*, 4(2):837–846, 2012.

[43] Farhad Farokhi, Henrik Sandberg, Iman Shames, and Michael Cantoni. Quadratic gaussian privacy games. In *2015 54th IEEE conference on decision and control (CDC)*, pages 4505–4510. IEEE, 2015.

[44] Emrah Akyol, Cédric Langbort, and Tamer Başar. Privacy constrained information processing. In *2015 54th IEEE conference on decision and control (CDC)*, pages 4511–4516. IEEE, 2015.

[45] Hsiang Hsu, Shahab Asoodeh, and Flavio P Calmon. Information-theoretic privacy watchdogs. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 552–556. IEEE, 2019.

[46] Borzoo Rassouli, Fernando E Rosas, and Deniz Gündüz. Data disclosure under perfect sample privacy. *IEEE Transactions on Information Forensics and Security*, 15:2012–2025, 2019.

[47] Gaia Franceschini and Sandro Macchietto. Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, 63(19):4846–4872, 2008.

[48] US Census Demographic Data. https://www.kaggle.com/muonneutrino/\us-census-demographic-data#acs2017_census_tract_data.csv, Last accessed on 2020-4-24.

[49] Fragkiskos Koufogiannis and George J Pappas. Differential privacy for dynamical sensitive data. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1118–1125. IEEE, 2017.

[50] Shuo Han and George J Pappas. Privacy in control and dynamical systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:309–332, 2018.

[51] Genki Sugiura, Kaito Ito, and Kenji Kashima. Bayesian differential privacy for linear dynamical systems. *IEEE Control Systems Letters*, 2021.

[52] Gregory F Welch. Kalman filter. *Computer Vision: A Reference Guide*, pages 1–3, 2020.

[53] William TB Uther and Manuela M Veloso. Tree based discretization for continuous state space reinforcement learning. *Aaai/iaai*, 98:769–774, 1998.

[54] John P Rust. A comparison of policy iteration methods for solving continuous-state, infinite-horizon markovian decision problems using random, quasi-random, and deterministic discretizations. *Infinite-Horizon Markovian Decision Problems Using Random, Quasi-random, and Deterministic Discretizations (April 1997)*, 1997.

[55] Sven Koenig, Reid Simmons, et al. Xavier: A robot navigation architecture based on partially observable markov decision process models. *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, (partially):91–122, 1998.

[56] Emil B Iversen, Juan M Morales, and Henrik Madsen. Optimal charging of an electric vehicle using a markov decision process. *Applied Energy*, 123:1–12, 2014.

[57] Zhengzhu Feng, Richard Dearden, Nicolas Meuleau, and Richard Washington. Dynamic programming for structured continuous markov decision problems. *arXiv preprint arXiv:1207.4115*, 2012.

[58] Geoffrey Smith. Quantifying information flow using min-entropy. In *2011 Eighth International Conference on Quantitative Evaluation of SysTems*, pages 159–167. IEEE, 2011.

[59] S Alvim M'rio, Kostas Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith.

Measuring information leakage using generalized gain functions. In *2012 IEEE 25th Computer Security Foundations Symposium*, pages 265–279. IEEE, 2012.

[60] Tom Chothia, Yusuke Kawamoto, and Chris Novakovic. A tool for estimating information leakage. In *International Conference on Computer Aided Verification*, pages 690–695. Springer, 2013.

[61] James Melbourne and Tomasz Tkocz. Reversal of rényi entropy inequalities under log-concavity. *IEEE Transactions on Information Theory*, 67(1):45–51, 2020.

[62] Allen L Barker, Donald E Brown, and Worthy N Martin. Bayesian estimation and the kalman filter. *Computers & Mathematics with Applications*, 30(10):55–77, 1995.