Privacy or Utility?
How to Preserve Both in Outlier Analysis

By

Hafiz Salman Asif

A dissertation submitted to the

Graduate School-Newark

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Management

Written under the direction of

Dr. Jaideep Vaidya and Dr. Periklis A. Papakonstantinou

And approved by

_____

_____

_____

_____

_____

Newark, New Jersey

January, 2021

ABSTRACT OF THE DISSERTATION

Privacy or Utility?
How to Preserve Both in Outlier Analysis

By Hafiz Salman Asif

Dissertation Directors:

Dr. Jaideep Vaidya and Dr. Periklis A. Papakonstantinou

Data analysts use outlier analysis to discover non-conforming patterns in data to generate actionable insights. It is an incredibly useful approach, but like all data-driven approaches, it raises privacy-related serious ethical and legal concerns when data is about peoples' information. *Is it possible to accurately analyze data for outliers while protecting the privacy of people whose data we analyze?* In this dissertation, we explicate methods to answer this question for the most practically relevant case, where outliers are defined in a data-dependent way and current privacy methods such as differential privacy fail to achieve practically meaningful utility.

To define what it means to protect privacy in outlier analysis, we conceptualize *sensitive privacy* — it not only admits efficient algorithmic constructions but is also amenable to analysis. We introduce novel constructions to develop sensitively private mechanisms to accurately identify outliers, and to compile low-accuracy differentially private mechanisms into high-accuracy sensitively private mechanisms. Furthermore, to address the lack of a principled approach to private outlier analysis, we provide a framework to help a data analyst identify the right problem-specification and a practical solution for her application.

Finally, we develop mechanisms — which guarantee privacy and practically meaningful utility — to identify (β, r)-anomalies as well as covid-19 hotspots (an outlying event). An extensive empirical evaluation of these private mechanisms over a range of real-world datasets and use cases overwhelmingly supports the effectiveness of our approach.

امی، ابو، شکریہ۔

Thank you, mom and dad!

Acknowledgements

It has been a pleasure, an honor, and a great fluke to work with Jaideep Vaidya and Periklis A. Papakonstantinou. I am very thankful to Jaideep for his kindness, his skillful guidance as my adviser, and his unmatched patience with me. As a mentor, he devoted countless hours to our discussions, paid close attention to my work, and taught me to stay focused on the bigger picture. I am evermore grateful to Jaideep for his continual guidance and support over the years. Periklis, my co-adviser, immensely contributed to my development as a researcher. As a mentor, he provided me with crucial mathematical training, dedicated innumerable hours to our discussions, and suggested new ideas and directions to pursue whenever I got stuck. I am forever indebted for all he has done for me. Both Periklis and Jaideep have contributed to my development as a scientist in more ways than I can possibly enumerate. I thank both of them for their guidance and contribution to my personal and intellectual growth.

I thank the committee members for reading my thesis and providing useful comments and feedback. A thanks is also due for the administrative staff, especially, Monnique Desilva, Goncalo Filipe, and Luz Kosar, whom I always found friendly and super-helpful. In addition, I thank NSF and NIH for their support.

I owe special thanks to my friends: Arslan Anjum, Lisa Buckley, Nathaniel Hobbs, Waqas Khan, Jay Shah, Kruti Shah, Tulika Tripathi, and Abdul Wasay, for their support and camaraderie over the years. A huge thanks to my younger siblings, Urooj Fatima and Usman Asif, who assumed many of my responsibilities at home while I worked thousands of miles away.

My utmost love and gratitude for my mom, Fareena Jabeen, and my dad, Asif Aziz, who always supported me in my endeavors and never lost faith in me — without their love and support, none of this would have been possible.

Table of Contents

List of Figures

List of Tables

Part I

FOUNDATIONS

CHAPTER 1

Introduction

Today, data and algorithms together are seamlessly curating our lives. They decide: whether you should be tested for covid-19; what movies or products should be recommended to you; if your tumor is malignant; or whether the recent transaction on your account is fraudulent. However, when the data is related to peoples' information, the same algorithms raise serious ethical and legal concerns related to privacy [2, 3, 4, 5]. Consequently, privacy concerns and the new privacy legislation [4, 5] are increasingly restricting our ability to analyze the data to solve the problems that data analysis made tractable.

Can we lift these restrictions, i.e. accurately analyze data without hurting the privacy of those who contribute their data? Here, we explicate methods and frameworks to answer this question for *outlier analysis*.

1.1 Motivation

Outlier analysis is a fundamental data analysis task and is extremely useful in practice with critical applications in medicine, finance, and national security. Yet it has only been analyzed for a few specialized cases of data privacy, and our understanding of this problem and its utility-privacy trade-off is limited. We call this problem of analyzing data for outliers and protecting privacy of the data contributors as *private outlier analysis*.

We use "private outlier analysis" as an umbrella term to refer to a class of problems that analyze data for outliers — these are among the core problems in statistics, data mining, and machine learning [6]. Data analysts use outlier analysis to discover complex

patterns in the data to generate actionable insights. We think of outliers as the non-conforming patterns in the data, e.g. the image of a malignant tumor compared to that of normal skin or a benign tumors.

The ability to identify outliers is an essential prerequisite to numerous applications in various domains [7, 8, 9, 6, 10]. To treat cancer, we must tell if a tumor is malignant; to counter email scams, we must filter spam; and to stop bank fraud, we must flag the suspicious transactions — and the de facto approach to solve these problems is outlier analysis.

However, outlier analysis like all data analyses is a double-edged sword. While it is crucial to solve challenging problems [11, 12, 13], it creates a risk to our privacy. Here is one way to think about the risk to your privacy when you share your data to be used in a data analysis task. From the result of a data analysis, an attacker will infer your identity and link it to your other anonymous data i.e. the data that does not contain your personally identifiable information such as email, name, phone number etc. Thus, the attacker will reidentify you and breach your privacy from data where the identity was kept hidden.

It is a well-established fact that an attacker can infer the identities of people from the analysis of their data, even if the data seems benign and does not contain any personally identifiable information. For example, an attacker can identify people by using quasi-identifiers (e.g. ZIP code, gender, and age) in anonymized data [14, 15], carrying out an inference attack on data repositories [16], genome analysis [17, 18], or their anonymous movie reviews [19]. Alarmingly, an attacker can even use a summary statistic to infer if a particular person's information was used to compute the statistic [20, 21].

So, how can one safeguard privacy in outlier analysis? To answer this, we must first define what we mean by "privacy" in data analysis. In this dissertation, we use an algorithmic notion of privacy that has been widely adopted in academia and industry. We say that an algorithm (for data analysis) protects the privacy of a person if, from the output of the algorithm, no attacker can tell (statistically) whether that person's information was recorded in the input database.

This notion of privacy was first formulated as Differential privacy, and it protects the privacy of all regardless of whether any particular person's information is in the database or not [22, 23]. Thus, it is an elegant solution to a difficult problem. A differentially private algorithm works by introducing carefully calibrated randomness to the true answers: either by using internal coin tosses to make decisions, or by adding noise to the true answers.

Since differential privacy works work well for doing statistics and other aggregate tasks [22, 23], it is natural to consider developing differentially private algorithms to protect privacy in outlier analysis. However, it becomes clear that differentially private algorithms are inherently unable to identify outliers records accurately [24, 25, 26, 27, 28].

The few variants of differential privacy [26, 27, 28] that are relevant for outlier analysis are either limited in their application or are unable to deal with the most practical case, when outliers are defined in a data-dependent fashion, for example, the outlier models that label a record as outlier based on its dissimilarity from the other records in the data.

Furthermore, the problem of private outlier analysis has only been studied for a few specialized cases and is often tackled with some ad-hoc approaches. Thus, there is no principled approach that analysts can use to attack this problem in practice. This dissertation reduces the gaps in our understanding of private outlier analysis and its various trade-offs and provides methods to solve this problem in practice.

1.2 Contributions

Below, we give a summary of the contributions of this dissertation.

- We conceptualize the notion of *sensitive privacy* to formalize what it means to protect privacy in outlier analysis. Sensitively private mechanisms protect privacy of most of the records, and simultaneously, achieve practically meaningful utility. In particular, sensitive privacy is well-suited for outliers that are defined in a data-dependent way. Our notion of privacy is not only computationally realizable

but is also amenable to analysis.

- We provide constructions to develop mechanisms to identify outliers, which provably guarantee sensitive privacy. These constructions are not tied to any specific definition of outlier and can even be used to develop differentially private mechanisms.

- We also develop a compiler construction that can compile a (less accurate) differentially private mechanism into a (more accurate) sensitively private mechanism that relative to the differentially private mechanism errs exponentially small.

- We propose a privacy-oriented taxonomy for outlier queries, which we use to develop a framework for private outlier analysis. This framework helps analysts in choosing the correct problem-specification for private outlier analysis. For instance, an analyst can use it to decide whether she should choose differential privacy or sensitive privacy for her application.

- We instantiate our proposed construction to develop sensitively private mechanism for the widely used notion of (β, r)-anomaly. We use this mechanism to establish the effectiveness of sensitive privacy through an extensive empirical evaluation over a diverse set of real-world data.

- Finally, we show how to design and develop a privacy-protecting crowdsensing based system to track covid-19 (corona virus disease 2019) pandemic (e.g. hotspots and outbreak, which are outlying events). We develop novel spatial and temporal data partitioning mechanisms to achieve practically meaningful utility (in tracking covid-19) while guaranteeing differential privacy for all the data contributors. We use real-world data on covid-19 confirmed cases to show that our approach is as effective as its non-private counterpart.

1.3 Organization

In Chapter 2, we review the state of art in data privacy relevant to outlier analysis, present the necessary background for the concept of outliers. In this chapter, we also

present important definitions and notions. In Chapter 3, we present the novel notion of sensitive privacy, its relationship with other relevant data privacy definitions, and some important properties of sensitive privacy. In Chapter 4, we present our constructions to achieve sensitive privacy; in this regards, we also give an impossibility result for n-step lookahead mechanism. In Chapter 5, we give a privacy-oriented taxonomy for outlier queries and an approach for analysts to tackle the problem of private outlier analysis. In Chapter 6, we instantiate our construction (given in Chapter 4) to develop sensitively private mechanism for (β, r)-anomaly, and present the results of its empirical evaluation. In Chapter 7, we present how to design and develop a privacy-protecting crowdsensing based system to track covid-19 pandemic. Finally, in Chapter 8, we present the future directions for research and open problems in private outlier analysis.

CHAPTER 2

Private Outlier Analysis: Background and Preliminaries

We use "private outlier analysis" as an umbrella term for a class of data analytics problems where privacy is to be protected. Basically, the differences in the problems in private outlier analyses arise due to the following three reasons:

(i) We need different outlier models, i.e. a description of what is an "outlier", for different problems. For example, the model best suited to identify fraudulent transactions is inappropriate to detect an epidemic as they are two very different problems.

(ii) There are subtle but crucial differences in what information we seek from an outlier analysis. For instance, the analysts look for very different information when they use outlier analysis to find all the outliers in the data compared to when they aim to identify an outlying event, e.g. a pandemic.

(iii) What information we want to protect, i.e. the definition of privacy. If we change what constitutes as "protection" for our particular application and setting, the problem of private outlier analysis also changes — yes, even if we have fixed the above two. For instance, depending upon the application and legislative requirements [29, 4, 5], you may want to protect the information of all or only a subgroup, each requiring a different treatment and solutions.

Thus, for a given private outlier analysis problem, an analyst has to choose the right problem-specification: (i) an appropriate outlier model, (ii) a query, that is, what information she seeks to obtain about the outlier(s), and (iii) the definition of privacy.

In our analysis and discussion, we combine the first two choices (i.e. (i) and (ii)) as the choice of an outlier query. Thus, in this context, the notion of outlier query and the notion of privacy form the two fundamental constituents of the problem of private outlier analysis.

Lastly and importantly, we note that privacy cannot be considered alone: every private data analysis task *must* take both privacy and utility (e.g. accuracy) into account. This is the case since we can always protect privacy by giving arbitrary answers and making data analysis useless. Hence, characterizing the right balance between the two (i.e. privacy and utility) is at the heart of private outlier analysis — explicating this and showing how to achieve this balance in practice is the key contribution of this dissertation.

In the following two sections, we review the important concepts and developments related to outlier analysis and data privacy.

2.1 Outliers: the 1st Constituent of the Problem

Outlier analysis is among the core problems in statistics, data mining, and machine learning [6, 30]. Intuitively, we think of outliers as the non-conforming patterns in the data, e.g. an image of a malignant tumor compared to that of normal skin or a benign tumor, and it either corresponds to a record, called outlier record, or an event, called an outlying event. See Figure 2.1 for the examples of outlier records, wherein o_1, o_2 and o_3 are three outlier records, and O_4 is a collection of outlier records.

We note that an event is a notion that depends on many records and is not associated with a single record, e.g. an epidemic (which can be identified, for example, by using peoples' health and demographic information [31]). Here, we briefly review the notion of an outlier and some of the models that we use in practice to analyze data for outliers. For further details you may refer to the surveys and books on outlier analysis [6, 32, 10, 33].

In practice, outlier analysis is an effective approach to generate actionable insights to solve numerous problems in various domains [6, 33]. The ability to identify outliers is an essential prerequisite to numerous critical applications in medicine, finance, and

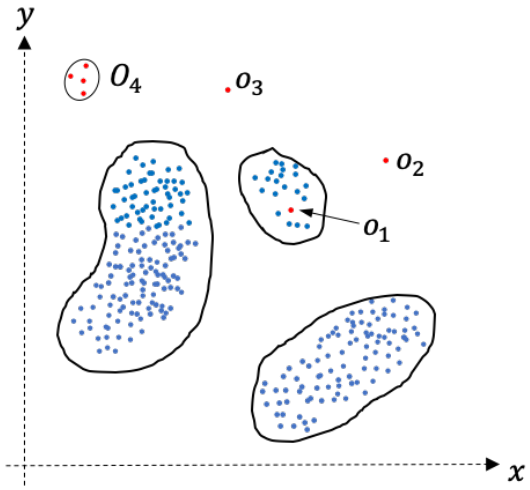

Figure 2.1: An example of outliers in data. Points (i.e. records) shown in red are outliers, and the ones in blue are non-outliers.

national security [7, 8, 9, 6, 10].

Outliers have been a subject of investigation for over 240 years [32], dating back to Bernoulli who discussed them in the context of "discarding discordant observation" [34]. In the literature, depending upon the application domain, outliers as non-conforming patterns have been referred to as discordant observations, exceptions, aberrations, surprises, peculiarities, or anomalies [33]. However outlier and anomaly are the two most popular terminologies used for this notion in computer science — and we adhere to using only these two terminologies, i.e. *outlier* and *anomaly*.

There are many descriptions of outliers that try to spell out what constitutes "non-conforming pattern" in a way that we can use to develop models to characterize outliers and build algorithms to analyze the data for outliers. But the following two description, one by Edgeworth and the other by Grubbs, are not only popular but also very useful.

> **(1887):** Discordant observations may be defined as those which present the appearance of differing in respect of their law of frequency from other observations with which they are combined [35].

> **(1969):** An outlying observation, or outlier, is one that appears to deviate markedly from the other members of the sample in which it occurs [36].

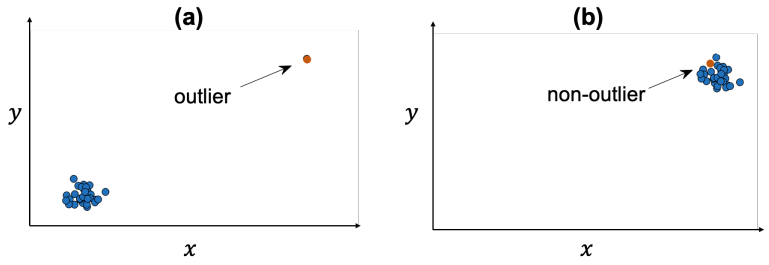

Figure 2.2: An example of data-dependent nature of outliers

Thus, at best, *an outlier is a data specific concept*. For example, see Figure 2.2, wherein the record shown as an orange-colored point is an outlier in Figure 2.2(a) but becomes a non-outlier with respect to the data in Figure 2.2(b).

To define what constitutes an outlier in practice is a challenging task. Because the nature of the non-conforming pattern varies across applications, domains, and data types. There is no single model that appropriately characterizes outliers in all environments [6, 32, 10, 33]. Hence, due to the complexity, diversity, and specificity to the context, the existing methods to analyze outliers solve a specific formulation of the problem.

Most of the outlier models use a measure or a scoring function to characterize how much a record (or an event) "deviates", and thus, assign a degree of outlyingness to each record. Now, given these scores, these models use a threshold to distinguish the outliers from the rest of the records in the data (or non-outlying events). We can divide these outlier models into two basic categories: non-parametric models and parametric models.

The non-parametric methods use a scoring function over the set of records. Such scoring functions are used to characterize the degree to which a record deviates from the other records. The basic idea is to compute the score for each of the subsets obtained by removing records from the given data. Now, the set of records whose removal results in the maximum deviation in the score are considered to markedly deviate from the others records in the database and are labeled as outliers [37, 38, 32, 10, 39]. These models are among the most fundamental ones, and we will use this fundamental notion of characterizing outliers to define *sensitive privacy*, a novel notion of privacy for outlier

analysis (Chapter 3).

The popular parametric outlier models use a range of criteria to characterize outliers. For instance, many statistical approaches use the deviation of a record (i.e. data point) from the population mean as the criterion to identify outliers — this is a well accepted and prevalent definition of outlier [40]. (β, r)-anomaly, k-nearest neighbor and clustering based outlier models use a distance based approach [41, 42, 43], local outlier factor [44] uses a density based approach, and AVF-outlier model uses attribute value frequency [45].

Once, we fix an outlier model suited for our analysis, depending upon what information we seek, there are different type of outlier queries. However, the two most useful queries in practice are: (1) *outlier identification*, for example a query that identifies if a given record is an outlier or a non-outlier, and (2) *outlier detection*, which find all the records in the data that are outliers. Once, we have finalized our outlier query, we can choose from a variety of algorithms, provided in the literature, to computer the query; for further details on this, see [6, 32, 10, 33].

The key takeaway here is that outliers are a data-dependent concept and the methods to identify them vary based on the nature of outliers, the context, and the data. However the overall notion remains the same: outliers are the records that considerably deviate from the pattern, distribution, or structure followed by a majority of the records in the data.

2.2 Privacy: the 2nd Constituent of the Problem

In data analysis, what does it mean to "protect privacy"? "Privacy" refers to the information disclosed about an individual via the results of data analysis — the lesser the disclosure, the higher the privacy. The information disclosed by data analysis can be used to infer the identities of individuals and breach their privacy. So, to protect privacy, ideally one needs to eliminate the information disclosure about the individuals.

However, protecting privacy in a way that *eliminates* the information disclosure about the individuals, that is, access to data does not enable one to learn anything about

an individual that cannot not be learned without access, is impossible [46]. Hence, for data analysis (where we do provide access to the data) to be useful, we will have to disclose some information.

Differential privacy (DP) [46, 22] was the first notion that mathematically defines privacy for data analysis to limit the information disclosure about the individuals — and it has shaped the field of private data analysis. It relaxes the constraint of the above privacy notion and explicitly specifies the information disclosure when a persons' data is used in data analysis [46]. It guarantees that no attacker can use a differentially private result (of data analysis) to find out with certainty if a particular person's data was used in the analysis. Thus, it affords "plausible deniability" to people, that is, any person can claim that her data was not used in the data analysis even if her data was used.

Differential privacy is an algorithmic definition of privacy, and it requires that the probability for any output of a privacy-protecting mechanism (i.e. an algorithm that takes a database as input) "should not change much" by adding or removing any one record in the input database. Thus, to define DP, we consider every pair of databases x and y that differ by one record (i.e. we can obtain y by adding or removing one record from x and vice versa) and call them *neighboring databases* or simply *neighbors*. Then, we say a mechanism M with domain $\mathcal{D}$ (the set of all databases) is ε-*differentially private* (DP) if for every pair of neighboring databases x and y and every set $R \subseteq \texttt{Range}(M)$, $\Pr[M(x) \in R] \leq e^{\varepsilon} \Pr[M(y) \in R]$ — we refer to this inequality on any pair of neighbors (x, y) as the *privacy constraint*.

Therefore, in the context of differential privacy, "should not change much" means the change in the probability is within a multiplicative factor of e^{ε}, where $\varepsilon > 0$ is the privacy parameter. The smaller the value of value of ε, the higher the privacy. Typically, to achieve differential privacy, a mechanism probabilistically perturbs the correct answer using noise (or randomness) from a carefully calibrated distribution(s).

Differential privacy works well for many classes of aggregate and statistical data analysis tasks as long as we can give an appropriate DP mechanism. However, there are various data analysis problems, e.g. outlier analysis, where it is inherently unable

to assure practically meaningful privacy (i.e. reasonably large values of ε) and utility [47, 48, 49, 24, 25]. This has lead researchers to develop variants of differential privacy to address important practical challenges in data analysis. Many of the variants of differential privacy either generalize the notion of neighboring databases or redefine what is meant by "the output should not change much" for neighboring databases (see [50] for a survey for different generalizations and variants of differential privacy).

Below, we review some of the important variants of differential privacy and identify and discuss the gaps in the context of private outlier analysis that still exist.

Pufferfish [48] and Blowfish [49] are two frameworks to give generalized versions of differential privacy. Both of them provide a way to redefine neighboring databases based on what secret (i.e. the kind of information disclosure) we want to protect. These frameworks add to our theoretical understanding of private data analysis and are useful for the applications where it is clear what secret we need to protect. However these frameworks do not provide any method or direction to deal with outlier detection or identification, especially, when the outliers are defined in a data-dependent fashion. We solve this problem in Chapter 3 by conceptualizing and formally defining what secret one must protect in outlier analysis.

Protected (differential) privacy [26], which was proposed for analyzing networks, divides the set of all possible records into two categories: one is protected and the other is not protected. It is possible to use protected privacy (instead of differential privacy) to boost the accuracy for some outlier analysis. However, this is not possible for all outlier analysis — especially, when outliers are defined in a data-dependent way. This shortcoming of protected privacy is due to the fixed and data-independent categorization of records into protected group and unprotected group, which is not possible when outliers are defined in data-dependent way. This is a critical limitation since without seeing the database it is not possible to say if a record is outlier or not, and additionally, by changing (or adding/removing) records in the given data can also affect the outlying status of a record as per the specification of the chosen outlier model. Thus, the privacy guarantee cannot be quantified in the order specified in protected privacy. This is one of the main problems that we tackle when defining sensitive privacy.

One-sided (differential) privacy [27] uses a similar approach as in protected privacy. Similar to protected privacy, it is useful for the cases where outliers are defined in data-independent fashion as it also defines the records to be protected independent of the database. Additionally, it further relaxes the definition by only considering a subset of pair of neighbors who must satisfy the privacy constraint. This leaves one-sided privacy open to attacks that can infer if a particular record — that was to be protected — is present in the data or not, while sensitive privacy is immune to such an attack.

Another way to generalize differential privacy is to have different levels of privacy (i.e. the value of ε) for different records, which Personalized (differential) privacy adopts [51]. Personalized privacy requires that the level of privacy be pre-specified for each record. For example, when sharing their data, people can specify the level of privacy they want. However, when the outliers are defined in a data-dependent fashion, and we want to provide privacy as per the degree of outlyingness of each record (which is required to make the analysis useful), this notion of privacy (for similar reasons discussed above) is also not applicable.

As opposed to Personalized privacy, Tailored (differential) privacy quantifies a level of privacy for each record as a function of the record and the database [29]. Thus it allows one to tailor the privacy guarantee across all records. Outlier privacy, an instantiation of tailored privacy, defines privacy in the presence of outliers, however, the problem that [29] focuses on is orthogonal to ours as it aims to protect outliers with higher privacy guarantee compared to rest of the records in the data. Below, we discuss some limiting features of outlier privacy to highlight the problems it presents in carrying out an accurate private outlier analysis.

Outlier privacy — wherein the definition of outlier is mechanism dependent — affords a stronger privacy guarantee to outliers (depending upon their degree of outlyingness) compared to the other records in the data. The notion of outlier that [29] uses is equivalent to that of $(\beta \geq 1, 0)$-anomaly for histogram-releasing mechanisms. However, this notion of outlier is too simple to work in practice for many tasks. In most practical cases the outlyingness of a record i also depends upon other records in the data that are different from i, and this nature of data-dependence must be taken

into account. Furthermore, when we provide more privacy to outliers, the utility of the outlier analysis degrades, even more than when we use plain differential privacy. Lastly, the mechanisms introduced in [29] do not address the problem of identifying outliers in the data. Thus, the applications of this work ([29]) are also limited. We propose the notion of sensitive privacy to address the above mentioned shortcomings. Additionally, we consider outlier models that are more general and develop constructions and a compiler to give mechanisms to identify outliers in the data — these constructions yield mechanisms that preserve utility and protect privacy.

Finally, we look at anomaly-restricted (differential) privacy [28] that does take into account the data-dependent nature of anomalies (i.e. outlier). However, it does so in a rather restricted setting: in [28] the input databases are guaranteed to have only one outlier, a structure not present in typically available databases, which is in addition to other restrictions on the input database. Although it has some theoretical value, anomaly-restricted privacy is inapplicable for most practical settings for outlier analysis. Sensitive privacy does not make such restricting assumptions. Furthermore, it is easy to interpret, amenable to analysis, and efficiently realizable in practice.

Although outlier analysis is a fundamental data analysis task with numerous applications in various domains, it has only been analyzed for a few specialized cases of data privacy [31, 29, 52, 53, 28]. For instance, one can use differential privacy to find the number of outliers in the data [52], to discover all the outliers in distributed data (where different parties have different set of records of the data) [54, 54], or to find if there is an epidemic outbreak [31]. However, to find an outlier in the data and identify the targeted population in a network, we need to use variants of differential privacy so that we can guarantee privacy without making the analysis futile [29, 53, 28]. In Chapter 5, we provide a general framework for data analysts, which can be used to carry out outlier analysis while preserving both privacy and accuracy.

2.3 Preliminaries: Notations, Definitions, and Settings

2.3.1 Databases

We consider a database as a multiset of elements from an arbitrary finite set $\mathcal{X}$, the set of possible values of records. For instance, when a record consists of m attributes, $\mathcal{X} = \prod_{t=1}^{m} A_t$, where A_t is the set of values of the t^{th}attribute. We assume that each record in any given database is associated with a distinct individual.

We represent a database x as a histogram in $\mathcal{D} = \{y \in \mathbb{N}^{\mathcal{X}} : \ ||y||_1 < \infty\}$, where $\mathcal{D}$ is the set of all possible databases, $\mathbb{N} = \{0, 1, 2, \dots\}$, and $||\cdot||_1$ represents ℓ_1 norm. For any database x and $i \in \mathcal{X}$, x_i is the number of records in x that are identical to i, $i \in x$ denotes a record i is present in a database x (i.e. $x_i \geq 1$); $i \notin x$ denotes no record of value i is present in the database (i.e. $x_i = 0$). For any $i \in \mathcal{X}$, we use $\mathbf{e}^i$ to denote the database consisting only of one record of value i, that is, $\mathbf{e}^i_i = 1$ and for all $j \neq i$, $\mathbf{e}^i_j = 0$.

2.3.2 Outliers (i.e. Anomalies)

We represent a given outlier model (i.e. characterization of what makes a record an outlier) as a predicate, F, that is, a function whose range is the set $\{0, 1\}$. The domain of the predicate is decided by the type of the outlier, that is, whether the outlier is a record or an event. Hence, we consider the two different types of outliers: (1) outlier is a record (e.g. a transaction) and (2) outlier is an event (e.g. epidemic) but the database consists of records (e.g. the health records). The main focus of this dissertation, except for Chapter 7, is on the the first type of outliers (where the outlier is a record).

For (1), we represent the given outlier model by a predicate, F, over the domain $\mathcal{X} \times \mathcal{D}$, i.e. $F : \mathcal{X} \times \mathcal{D} \to \{0, 1\}$. Now, for any database $x \in \mathcal{D}$ and record $i \in \mathcal{X}$, $F(i, x) = 1$ if a record of value i is an outlier with respect to the database x, otherwise $F(i, x) = 0$. We emphasize that for the predicate to be true, a record of value i need not present in x, namely, $F(i, x) = 1$ does not imply $i \in x$. Similarly, for (2), we define a predicate $\tilde{F}$ but over the domain $E \times \mathcal{D}$ for a given outlier model, where E is the set of events under consideration.

A related and important concept that we use in our analysis is of *normality property*. For a fixed outlier model F, a normality property is a predicate $p : \mathcal{X} \times \mathcal{D} \to \{0,1\}$ such that for every $x \in \mathcal{D}$ and every $i \in \mathcal{X}$, $p(i,x) = 1$ if $i \in x$ (i.e. i is present in x) and $F(i,x) = 0$ (i.e. i is a non-outlier, i.e. not an outlier, with respect to x). In this dissertation, we use this notion only for the case when an outlier is a record.

We must note that normality property and outlier model are two closely related concepts but are not the same. This is the case since normality is also predicated on the presence (or existence) of the record in the given database, which is not the case for outlier model. We also remark that whenever, in our analysis, we fix a normality property, we assume it corresponds to an arbitrarily fixed outlier model unless mentioned otherwise. Thus, for simplicity, when we talk about normality property, we omit the mention of the outlier model (i.e. F).

Finally, we use $\mathfrak{P} = \{\text{property} : \mathcal{X} \times \mathcal{D} \to \{0,1\}\}$ to denote the set that contains every property (i.e. function) from $\mathcal{X} \times \mathcal{D}$ to $\{0,1\}$, wherein the set of all normality properties makes a subset of $\mathfrak{P}$.

Anomaly identification function:

We now introduce the important notion of *anomaly identification function*, which tells us if a record is present in the database as an anomalous record. For a fixed outlier model, we say a boolean function $g : \mathcal{X} \times \mathcal{D} \to \{0,1\}$ is an anomaly identification function if for every record $i \in \mathcal{X}$ and database $x \in \mathcal{D}$, $g(i,x) = 1$ *if and only if* $i \in x$ and i is an anomalous record with respect to x — note that no change is made to x.

Our constructions (given in Chapter 4) will focus on the above formulation of identifying anomalies because it represents a fundamental and one of the most difficult cases of private outlier analysis, especially, for differential privacy like definitions of data privacy as we will see in Chapter 3 and Chapter 4. Alternatively, we could have defined g without predicating on the existence of i in x. However, when we drop the predicate on the existence of i, we in effect blur the distinction between the notion of a void spot (that in a different database could have been occupied by a record) in the database and

the notion of an anomaly. We, however, note that the above given formulation is extensible to the case where the database, over which anomaly identification is performed, is considered to include the record for which anomaly identification is desired. Here, for example, the anomaly identification for a record i over a data x can be computed over the database that consists of all the records in x as well as the record i.

Our focused anomaly (outlier) model:

We *focus on* (β, r)*-anomaly* as the notion of outlier for our discussions and to instantiate our constructions to developed sensitively private mechanism to identify anomalies. Not only is it is a prevalent outlier model in practice, it generalizes many statistical based outlier models, and it has many well-known variants and extensions [41, 55, 6] — and our work naturally extends to them. Under (β, r)-anomaly model, *we consider a record to be anomalous (i.e. outlier) if there are at most* β *records similar to it*, i.e. within distance r (see the definition given below). The parameters β and r are given by the domain experts [55] or found through exploratory analysis.

We use the following notation to give the definition of (β, r)-anomaly. For any database x, record $i \in \mathcal{X}$, $r \geq 0$, and a distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, $B_x(i, r) = \sum_{j \in \mathcal{X} : d(i,j) \leq r} x_j$.

Definition 2.1 ((β, r)-anomaly [55])**.** *An* anomaly *is defined for a database* $x \in \mathbb{N}^{\mathcal{X}}$ *and a record* $i \in \mathcal{X}$ *as follows. We say that* i *is a* (β, r)-anomaly *in the database* x *if* $i \in x$ *(i.e.* i *is present in* x*), and* $B_x(i, r) \leq \beta$ *(i.e. there are at most* β *records in* x *that are within distance* r *from* i*).*

Whenever we refer to a (β, r)-anomaly, we assume there is an arbitrarily fixed distance function d over $\mathcal{X} \times \mathcal{X}$.

Practical setting for outliers:

We consider that databases are (1) typically *sparse* and (2) contain a small number of outliers. By sparsity, we mean that the size ($|x|$) of the database x is much smaller than the size of $\mathcal{X}$, i.e. $|x| << |\mathcal{X}|$. The empirical evidence supports this belief — see

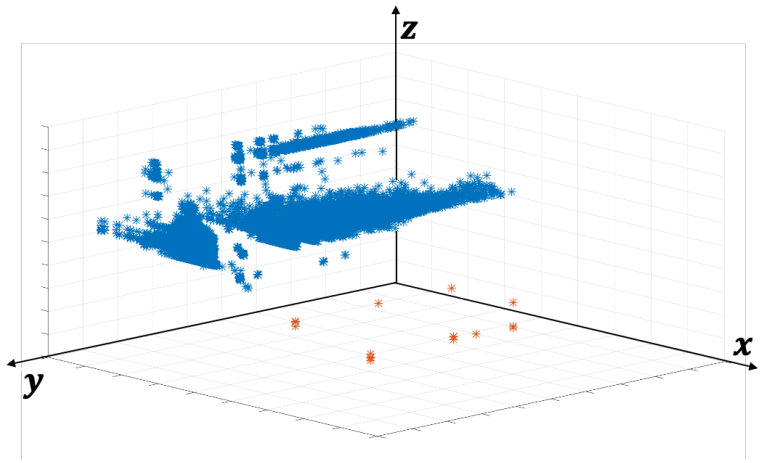

Figure 2.3: A real-world dataset with outliers (from ODDS [1]). Blue points are non-outliers. Orange points are outliers.

[24] and the analysis given below. Many outlier models, which are used in a wide range of practically important applications [6, 33], rely on these assumptions to characterize outlyingness.

We analyzed multivariate datasets from UCI data library [56] to validate our assumptions. First, we took categorical datasets with size at least 30, which were not artificially generated from a decision model; out of 30, this gave us 18 datasets. We fitted two linear model (via least squares), one for $|x|$ and the other for $\log_2(|\mathcal{X}|)$ for independent variable m, the number of attributes. This gave us $|x| \approx 14672.8m - 216430$ and $|\mathcal{X}| \approx 2^{0.9m+12}$. Second, we validated the assumption of sparsity on continuous data via discretization—out of 155 continuous datasets only 11 have $|x| < m$. For rest (i.e. 144) of the datasets, we followed [57] to discretize each of m continuous attributes into k intervals, where $k \approx \sqrt{|x|} \cdot \ell$ for a constant $\ell > 0$. The best fitted linear model for $|x|$ gave us $|x| \approx 2 \times 10^5 + 8m$, whereas $|\mathcal{X}| \approx \exp(m \ln(\sqrt{|x|}))$.

2.3.3 Privacy

All the privacy-protecting mechanisms (algorithms) that we consider here have domain $\mathcal{D}$. This is a standard practice and in no way does it weaken the results or the privacy guarantee.

Differential privacy:

Definition 2.2 (differential privacy [46, 22])**.** *For any given $\varepsilon > 0$ and mechanism M (with domain $\mathcal{D}$), we say M is ε-differentially private if for every $x, y \in \mathcal{D}$ such that $||x - y||_1 = 1$, and every $R \subseteq Range(M)$,*

$$\mathrm{P}\left(M(x) \in R\right) \leq e^{\varepsilon}\mathrm{P}\left(M(y) \in R\right).$$

We implicitly assume that the R's are chosen such that the events "$M(x) \in R$" are measurable.

Differential privacy enjoys important properties of composition and post-processing. These properties give us a way to determine an overall privacy guarantee for a data analysis that involves many mechanisms and differentially private answers.

Composition: There are two types of compositions: sequential composition and parallel compositions.

Sequential composition states that if we have an ε_1-DP mechanism $M_1 : \mathcal{D} \to R_1$ and an ε_2-DP mechanism $M_2 : \mathcal{D} \times R_1 \to R_2$, both with their independent sources of randomness, then the mechanism $M(\cdot) = (r = M_1(\cdot), M_2(\cdot, r))$ is $(\varepsilon_1 + \varepsilon_2)$-DP. For instance, when the same database is used to compute n results using n independent differentially private mechanisms, each with privacy ε, the analysis overall is $n\varepsilon$-DP.

Parallel composition, however, is useful when we have mechanisms that use disjoint fragments (i.e. a subset of records) of a database. Let us say we have an ε_1-DP mechanism $M_1 : \mathcal{D} \to R_1$ and an ε_2-DP mechanism $M_2 : \mathcal{D} \to R_2$ (both use independent sources of randomness). M_1 only uses its randomness and the records in the database that belong to the set $J \subseteq \mathcal{X}$, whereas M_1 uses its randomness and the records in the database that are in $\mathcal{X} \setminus J$. Then the mechanism $M(\cdot) = (M_1(\cdot), M_2(\cdot))$ is $\max(\varepsilon_1, \varepsilon_2)$-differentially private.

Post-processing: Differential privacy guarantee does not change by further processing a differentially private result. That is, if a mechanism, $M : \mathcal{D} \to R$, is ε-DP, then for every function $f : R \to K$, $f(M(\cdot))$ is also ε-DP.

Private anomaly identification query (AIQ):

Given an anomaly identification function, g, a query that asks for whether a particular record i is an anomaly is referred to as *anomaly identification query* (AIQ). Since, the input to the private mechanism is only the database, for AIQ, we consider the query is for a fixed record. Thus, we specify an AIQ by the pair (i, g), where i is a record and g is an anomaly identification function. To simplify the notation, we also depict an AIQ as g_i. Now, for a fixed AIQ, (i, g), a private anomaly identification mechanism, $M : \mathcal{D} \to \{0, 1\}$ can be represented by its distribution, where for every x, $\mathrm{P}\left(M(x) = g_i(x)\right)$ is the probability the M output correctly, and $\mathrm{P}\left(M(x) \neq g_i(x)\right)$ is the probability that M errs on x.

Privacy induced graphs:

We use a graph-based formulation to define privacy for outlier analysis, and give our constructions to identify outliers in a privacy-protecting way. These graphs are induced by differential privacy like definitions, and they play a central role in our analysis, for example, see [58, 48, 49]. We consider simple graphs over the databases and call them *neighborhood graphs*. The definition is as follows:

Definition 2.3 (neighborhood graph)**.** *A graph $G = (\mathcal{D}, E)$, where $\mathcal{D}$ is the set of all nodes, is called a* neighborhood graph *if the set of edges, E, is such that $E \subseteq \big\{\{x, y\} : \ x, y \in \mathcal{D}$ and $||x - y||_1 = 1\big\}$.*

For any given neighborhood graph, G, we define shortest path metric over each of the connected component, G' of G, i.e. $d_{G'}$, which gives the shortest path length between any two nodes of the connected component G', where the path length corresponding to any two nodes directly connected by an edge is 1 — we refer to this metric as the *shortest path metric*. For simplicity, we abuse the notation, and write d_G to depict the metric over all the components of the neighborhood graph G. We stress the fact that the d_G is only defined for the databases , i.e. nodes, that are connected in G. Furthermore every two database x and y that are connected by an edge in the neighborhood graph G (i.e. $d_G(x, y) = 1$) are called neighbors.

For any given neighborhood graph, G, we use $\mathcal{E}(G)$ to denote the set of edges in G. Furthermore, for any given neighborhood graphs G and G', we say G' is subgraph of G if $\mathcal{E}(G') \subseteq \mathcal{E}(G)$. As an example, let us look at the neighborhood graph induced by differential privacy, where neighbors in the graph (i.e. nodes connected by an edge) correspond to the neighboring databases in differential privacy. Thus, the neighborhood graph, $\mathbb{G}$, induced by DP, is such that for every $x, y \in \mathcal{D}$, $\{x, y\} \in \mathcal{E}(\mathbb{G})$ if and only if $||x - y||_1 = 1$. Note that $\mathbb{G}$ is a connected graph, i.e. every node in $\mathbb{G}$ is connected to every other node in $\mathbb{G}$. We can use this graph to re-state differential privacy by imposing the privacy constraints of differential privacy on every two nodes, x and y such that $d_{\mathbb{G}}(x, y) = 1$ (i.e. the neighboring nodes in $\mathbb{G}$). We call $\mathbb{G}$ the DP neighborhood graph. Note that $\mathbb{G}$ has an interesting and useful property, that is, every neighborhood graph is a subgraph of $\mathbb{G}$.

Another important concept in this context is that of *Lipschitz continuity*, a property of a function f with respect to the neighborhood graph G. We will use it to define a necessary constraint for privacy-protecting mechanisms to identify outliers. In our exposition we will consider f from $\mathcal{X} \times \mathcal{D}$ to $\mathbb{R}_{\geq 0} \cup \{\perp\}$. So we extend the standard notion of Lipschitz continuity, considered in privacy literature [58], to cover the non-real part (i.e. $\perp$) of the function as well.

Definition 2.4 (Lipschitz continuity)**.** *For any given neighborhood graph, G, $\alpha > 0$, and a function, $f : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0} \cup \{\perp\}$, we say f is* α-Lipschitz continuous *if for every $i \in \mathcal{X}$ and neighboring x and y in G, the following holds:*

if $f(i, x) = \perp$ *or* $f(i, y) = \perp$ *then* $f(i, x) = f(i, y) = \perp$

otherwise $|f(i, x) - f(i, y)| \leq \alpha$

Privacy setting:

We consider the trusted curator setting for privacy. The trusted curator is a fully trusted third party, who has access to the database. It receives the query (for example, an AIQ), uses a mechanism to compute the query, and sends the result back. Now, if

the curator uses a (differentially) private mechanism, then we are guaranteed that the query is answered in a differentially private fashion.

Part II

THEORETICAL DEVELOPMENTS

CHAPTER 3

Sensitive Privacy (SP) – An Intricate Balance of Privacy and Utility

Here, we conceptualize the notion of sensitive privacy — a data privacy definition for anomaly (i.e. outlier) analysis. Specifically, it is well-suited to detect and identify anomalies, the main focus of this work. Thus, here we solve the problem of accurate, private, and algorithmic anomaly identification, that is, labeling a record as anomalous or normal by an algorithm. In this chapter, we not only define the new notion of sensitive privacy, but also discuss why sensitive privacy is indispensable, what its strengths are, and how it is related to other important concepts of data privacy. In the next chapter, we will provide constructions to develop sensitively private mechanisms that can identify anomalies accurately.

Core privacy problem. Modern methods of anomaly identification label a record as anomalous (or normal) based on its degree of dissimilarity from the other existing records in the database [59, 60, 61, 55]. Consequently, the labeling of a record as anomalous is specific to a dataset, and knowing that a record is anomalous can leak a significant amount of information about the other records. This type of privacy leakage is the core obstacle that any privacy-preserving anomaly identification method must overcome. The current methods to protect privacy (which work well for doing statistics and other aggregate tasks [22, 23]), however, are inherently unable to deal with this problem and identify anomalous records accurately and privately.

This work is the first to develop methods (in a general setting where anomalies may be defined in a data-dependent way) to accurately identify if a record is anomalous while simultaneously guaranteeing privacy by making it statistically impossible to infer

if a non-anomalous record was included in the database.

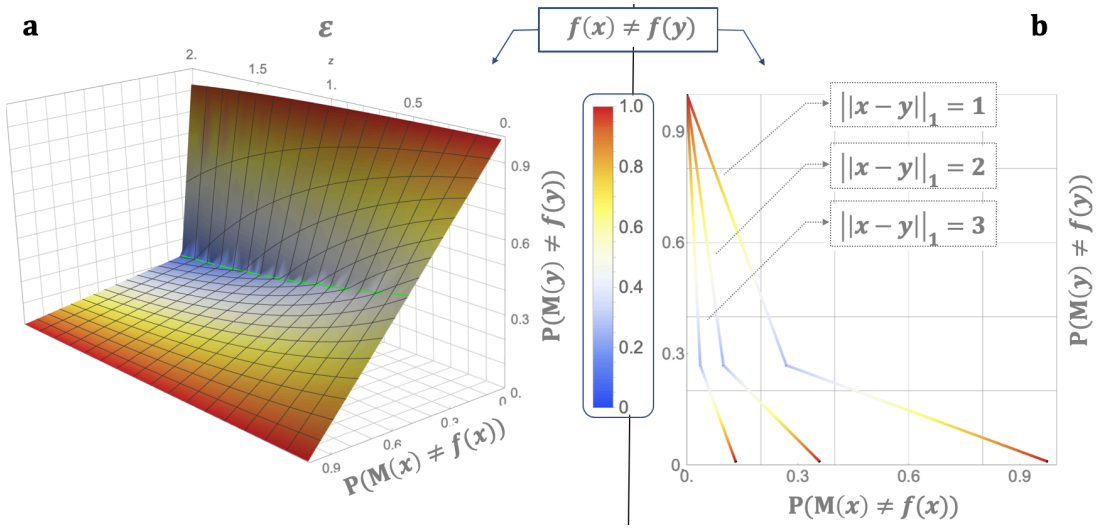

Figure 3.1: **(a)** x and y differ by one record, the "ε axis" is for the privacy parameter, the "$P(M(x) \neq f(x))$ axis" is for the minimum error over all ε-DP mechanisms M on x for a give error on y on the "$P(M(y) \neq f(y))$ axis". The graph depicts the trade-off between the errors committed on x and y. **(b)** this plot is for $\varepsilon = 1$ and otherwise is the same but for different x's and y's.

3.1 Why Do We Need a New Privacy Notion?

Consider the trusted curator setting for privacy. Recall that the trusted curator has access to the database, and it answers the anomaly identification queries using a mechanism. Next, consider the following notion of privacy: the privacy of an individual is protected if the output of an anomaly identification mechanism is unaffected by the presence or the absence of the individual's record in the database (which is the input to the mechanism). *This is the notion of privacy we consider here.* It protects individuals against any risk incurred due to the presence of their information in the database that is used to identify anomalies.

Recall from Section 2.2 that the above notion of privacy was first formalized in the seminal work of differential privacy [46, 22] (where privacy is quantified by a parameter $\varepsilon > 0$: the smaller the ε, the higher the privacy) and can informally be stated as follows: a randomized mechanism that takes a database as input is ε-differentially private if for any two input databases differing by one record, the probabilities (corresponding to the

two databases) of occurrence of any event are within a multiplicative factor e^{ε}.

Unfortunately, simply employing differential privacy does not address the need for both privacy and practically useful accuracy in our case (see Section 2.2 for why other definitions do not suffice for our case). For example, providing privacy equally to everyone severely degrades accuracy in identifying anomalies. This is because, for a database, the addition of a record in a region which is sparse in terms of data points creates an anomaly. Conversely, the removal of an anomalous record typically removes the anomaly altogether. Therefore, under differential privacy the accuracy achievable for anomaly identification is limited as explained below.

Differential privacy for any *binary* function, $f : \mathcal{D} \to \{0, 1\}$, such as anomaly identification, comes with inherent limitations. We highlight these limitations through the graph of Figure 3.1a.

Fix any ε-DP mechanism M that is supposed to compute f. The mere fact that f is binary and M is differentially private has the following effect.

For any two databases x and y that differ in one record, assume that $f(x) = 0$ and $f(y) = 1$. Now, a simple calculation shows that the differential privacy constraints create a trade-off: whenever M makes a small error in computing $f(x)$ then it is forced to err a lot when computing on its "neighbor" y and vice-versa. Moreover, the higher the privacy requirements are (i.e. for smaller ε) the stricter this trade-off is, as depicted on Figure 3.1a. Formally, we state this fact in Claim 3.1.

Claim 3.1. *Arbitrarily fix $\varepsilon > 0$, $f : \mathcal{D} \to \{0, 1\}$, and ε-DP $M : \mathcal{D} \to \{0, 1\}$. For every x and y, if $||x - y||_1 = 1$ and $f(x) \neq f(y)$, then*

$$\max\{P\big(M(x) \neq f(x)\big), P\big(M(y) \neq f(y)\big)\} \geq 1/(1 + e^{\varepsilon})$$

What happens to this inherent trade-off when x and y differ in more than one record? As shown on Figure 3.1b this trade-off is relaxed. Note in the proof of Claim 3.1 (given below) that to derive the trade-off, there is nothing specific to the ℓ_1 metric (used for differential privacy); and instead we could have used any metric over the space of databases. Other data privacy works that considered general metrics are e.g. [49, 48]. In this dissertation, we proposes a distance metric that is appropriate for anomaly

identification, in conjunction to an appropriate relaxation of differential privacy. This way we will lay out a practically meaningful (but also amenable to analysis) privacy setting for outlier analysis.

Of course, one way to obviate the low-utility problem of differential privacy is to set a higher value of ε. Although this approach solves the problem of low-utility, *it severely weakens the privacy guarantee for everyone.* Thus, it is desirable to have a notion of privacy that allows for practically meaningful utility while providing strong privacy guarantee for most of the records — sensitive privacy fulfills this requirement.

Proof of Claim 3.1. Arbitrarily fix $\varepsilon > 0$, $f : \mathcal{D} \to \{0, 1\}$, ε-differentially private mechanism $M : \mathcal{D} \to \{0, 1\}$, and $x, y \in \mathcal{D}$ such that $||x - y||_1 = 1$ and $f(x) \neq f(y)$; and let $b = f(x)$.

If $\mathrm{P}\big(M(y) \neq f(y)\big) = \mathrm{P}\big(M(y) = b\big) \leq 1/(1 + e^{\varepsilon})$ then, by differential privacy constraints, we get that $\mathrm{P}\big(M(x) = b\big) \leq e^{\varepsilon}/(1 + e^{\varepsilon})$. Hence, it follows that $\mathrm{P}\big(M(x) = 1-b\big) = \mathrm{P}\big(M(x) \neq f(x)\big) \geq 1/(1+e^{\varepsilon})$. Similarly, $\mathrm{P}\big(M(x) \neq f(x)\big) \leq 1/(1+e^{\varepsilon})$ implies that $\mathrm{P}\big(M(y) \neq f(y)\big) \geq 1/(1 + e^{\varepsilon})$. Hence, from the above, it follows that

$$\max\{\mathrm{P}(M(x) \neq f(x)), \mathrm{P}(M(y) \neq f(y))\} \geq 1/(1 + e^{\varepsilon}).$$

Since x, y were fixed arbitrarily, the claim follows. □

3.2 What Do We Want From the New Privacy Notion?

We want to relax differential privacy since affording the same level of protection to everyone severely degrades the accuracy for anomaly identification. One possible relaxation, suitable for the problem at hand, is providing protection only for a subset of the records. We note that such a relaxation is backed by privacy legislation, e.g. GDPR allows for giving up privacy for an illegal activity [4].

Although protecting a prefixed set of records, which is decided independent of the database, works when anomalies are defined independent of the other records, such a notion of privacy fails to protect the normal records when the anomalies are defined in a data-dependent way. Here the problem arises due to the *fixed* nature of the set that is

database-specific. In the case of a data-dependent definition of anomaly, if we wish to provide privacy guarantee to the normal – call them *sensitive* – records that are present in the database, then specifying the set of sensitive records itself leaks information and can lead to a privacy breach. Therefore, sensitive records must be defined based on a more fundamental premise to reduces such dependencies. This notion of sensitive record plays a pivotal role in defining our notion of *sensitive privacy* for outlier analysis.

Recall that although anomaly identification method provide binary labeling, they assign scores to represent how outlying a record is [60, 61]; thus these models (implicitly or explicitly) assign every record a degree of outlyingness with respect to the other records, which the following discussion takes into account.

An appropriate notion of privacy in our setting must allow a privacy mechanism to have the following two important properties. First, the more outlying (or non-outlying) a record is, the higher the accuracy the privacy mechanism can achieve for anomaly identification — this is in contrast to DP (Figure 3.2c). Second, all the sensitive records should have DP like privacy guarantee with the same value of the privacy parameter.

The mechanisms that are private under sensitive privacy achieve both the properties (see Figure 3.2, which gives the indicative experimental results on a synthetic dataset generated from the normal distribution; see the Chapter's endnote[1] for the details on the experiment and the values of the parameters). Furthermore, it has an important additional property: in the typical settings for outliers, the anomalies do not lose privacy altogether; instead the more outlying a record is the lesser privacy it has (Figure 3.2d).

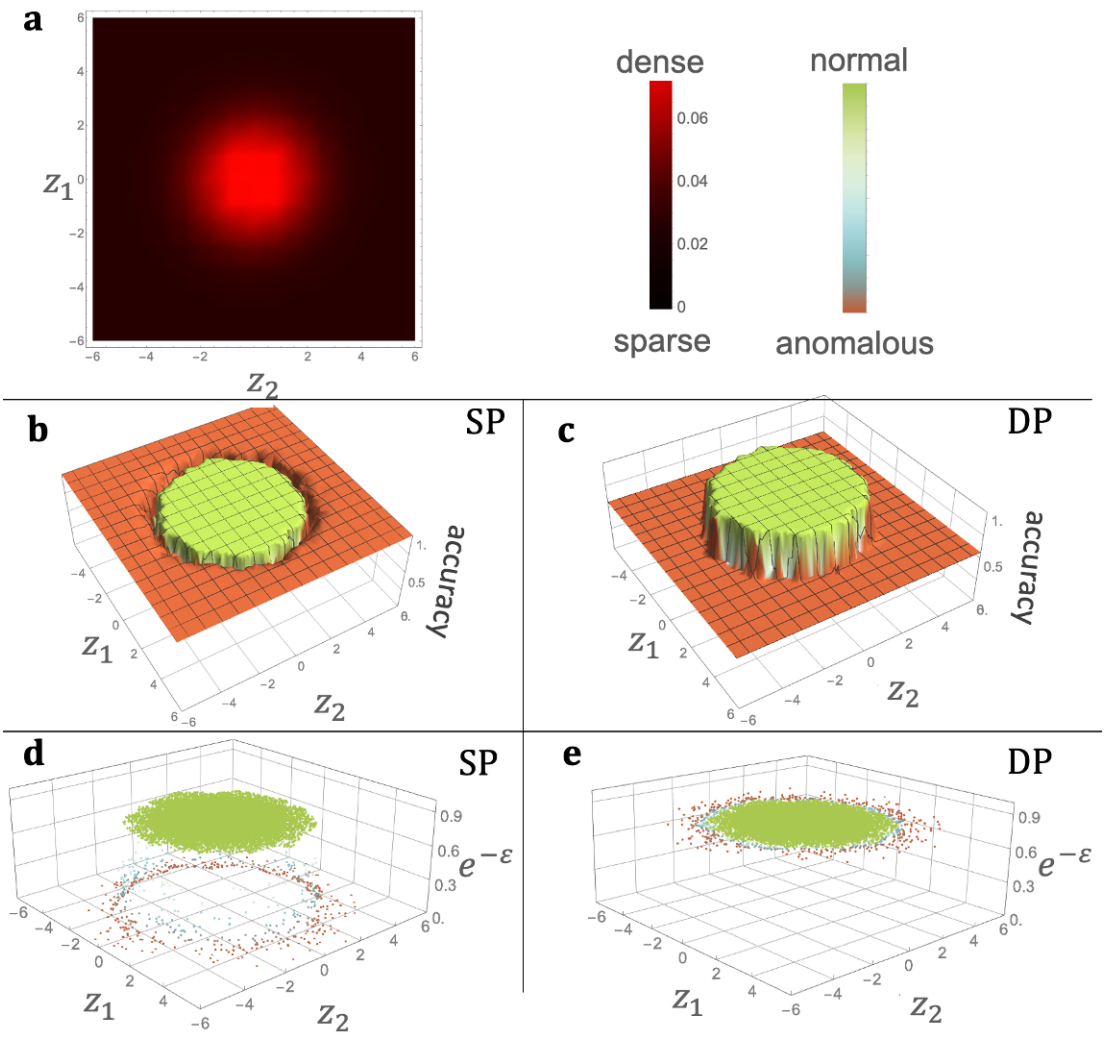

Figure 3.2: **(b)**, **(c)** is for the same data, and **(d)**, **(e)** is for the same data. **(a)** gives the density plot of the distribution of the example data. z_1 and z_2 axes give the coordinate of a point (record). **(b)** and **(c)** respectively show the accuracy (on vertical axis) for anomaly identification query (AIQ) via sensitively private (SP) and DP mechanisms for the data. The plots give the interpolated results to clarify the relationship of outlyingness and accuracy. **(d)** and **(e)** give the privacy (on vertical axis) for each record in the data for private AIQ. All the green (normal) points in **(d)** are at the same level as all the points in **(e)**.

3.3 How Do We Define the New Privacy Notion?

To define privacy, we need a metric space over the databases. This is because a private mechanism needs to statistically blur the distinction between databases that are close in the metric space. While differential privacy uses the ℓ_1-metric, we utilize a different metric over databases, which we defined using the notion of sensitive record. Informally, we say a record is *sensitive* with respect to a database if it is normal or becomes normal under a small change — we formalize this in Section 3.4.

We argue that our notion of sensitive record is quite natural as it is inspired from the existing literature on outlier analysis [60, 61]. Recall that, by definition, an anomalous record significantly diverges from other records in the database [60, 61]; hence, a small change in the database should not affect the label of an anomalous record.

Given the definition of sensitive record, we define the metric over the databases by considering a graph over the databases. The graph is defined over $\mathcal{D}$ (the set of all the databases), where each database is a node in the graph and there is an edge between every two nodes (i.e. databases) that differ by a sensitive record. The metric over the databases is now given by the shortest path length between the databases in this graph. This metric space has the property that databases differing by a sensitive record are closer compared to the databases differing in a non-sensitive record. We use the proposed metric space to define sensitive privacy, which enables us to fine-tune the trade-off between accuracy and privacy.

3.4 Sensitive Privacy (SP)

Our notion of *sensitive privacy* requires *privacy protection* of every record that may be *normal* under a *small change* in the database. We use the notion of *normality property* p to identify the normal records that exist in the database. Recall (from Section 2.3) that for a given anomaly (outlier) model, a normality property, $p : \mathcal{X} \times \mathcal{D} \to \{0, 1\}$, is such that for every record i and database x, $p(i, x) = 1$ if and only if i is present in x as a normal record.

We formalize the notion of "small change" in the database as the addition or removal

of k records from the database. We consider this change to be typical and want to protect the privacy of every record that may become normal under this small change in the database. We use this notion of small change in the database to define the key concept of *sensitive record*. Informally, for a fixed normality property, all the records whose privacy must be protected are termed as *sensitive records*.

Definition 3.1 (sensitive record)**.** *Arbitrarily fix $k \geq 1$ and a normality property p. For every record $i \in \mathcal{X}$ and every database $x \in \mathcal{D}$, we say i is* k-sensitive *with respect to x if, for a database $y \in \mathcal{D}$, $||x - y||_1 \leq k$ and $p(i, y) = 1$.*

Next, we give the important notion of *k-sensitive neighborhood graph*, which is central to our privacy definitions and constructions to develop sensitively private mechanisms. It generalizes the DP neighborhood graph through the concept of sensitive records. Recall that the DP *neighborhood graph*, $\mathbb{G} = (\mathcal{D}, E)$, is a neighborhood graph (see Definition 2.3) such that for every two databases x and y in $\mathcal{D}$, $\{x, y\} \in E \iff ||x - y||_1 = 1$. Throughout this work, we use $\mathbb{G}$ to depict the DP neighborhood graph.

Definition 3.2 (sensitive neighborhood graph)**.** *For a fixed $k \geq 1$ and a normality property, the* k-sensitive neighborhood graph*, G_S, is a subgraph of the DP neighborhood graph, $\mathbb{G}$, such that for every $\{x, y\} \in \mathcal{E}(\mathbb{G})$, $\{x, y\} \in \mathcal{E}(G_S)$ if and only if there is an $i \in \mathcal{X}$ such that: (1) $|x_i - y_i| = 1$ and (2) i is k-sensitive with respect to x or y.*

Both DP neighborhood graph and k-sensitive neighborhood graph are neighborhood graphs, and hence, the two databases connected by an edge in each graph are called neighbors. Furthermore, the k-sensitive neighborhood graph (as opposed to $\mathbb{G}$) is tied to the normality property, and hence, the anomaly definition. With this, we can state the notion of sensitive privacy.

Definition 3.3 (sensitive privacy)**.** *Arbitrarily fix $\varepsilon > 0$, $k \geq 1$, and a normality property p, and let G_S be the k-sensitive neighborhood graph for p. Then for any mechanism M with domain $\mathcal{D}$, we say M is* (ε, k)-sensitively private *if for every neighboring x and*

y in G_S, and every $R \subseteq Range(M)$,

$$\mathrm{P}\big(M(x) \in R\big) \leq e^{\varepsilon}\ \mathrm{P}\big(M(y) \in R\big)$$

We implicitly require that the events "$M(x) \in R$" are measurable. In our discussion, we omit k when it is clear from the context or when referring to the sensitive neighborhood graph in general. The privacy guarantee here is quantified through ε and the k-sensitive neighborhood graph. The smaller the value of ε, the higher the privacy guarantee. Furthermore, the more neighboring databases in the DP neighborhood graph are neighbors in the sensitive neighborhood graph, the wider the reach of the privacy guarantee in sensitive privacy.

3.4.1 Understanding sensitive privacy

Sensitive privacy guarantees that given the output of a sensitively private mechanism, an adversary cannot infer the presence or the absence of a sensitive record in the database. Similar to differential privacy, the privacy constraints of sensitive privacy necessitate that for every two neighbors, any test (i.e. event) one may be concerned about, should occur with "almost the same probability", that is, the presence or the absence of a sensitive record should not affect the likelihood of occurrence of any event. Here, "almost the same probability" means that the above probabilities are within a multiplicative factor e^{ε}.

The privacy parameter ε plays the same role in sensitive privacy as it does in differential privacy: the smaller its value, the higher the privacy. For neighboring databases in a sensitive neighborhood graph (G_S), the guarantee of sensitive privacy is exactly the same as that of differential privacy. However, if the two databases x and y, differ by one record that is non-sensitive (with respect to both the databases), then they are not neighbors in G_S, and the guarantee provided by sensitive privacy is weaker[1] than differential privacy, nevertheless, having the same form. So, intuitively, if we only consider the databases, where all the records are sensitive, then differential privacy and

[1] "Weaker" means that every mechanism which is ε-DP is also ε-SP, but in general not the other way around.

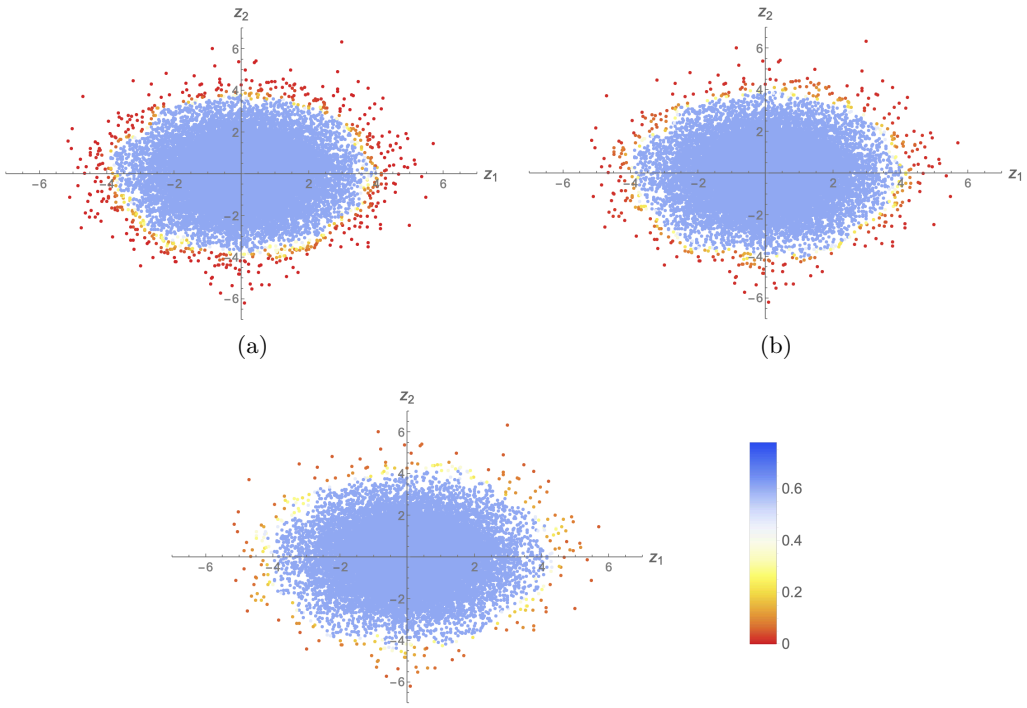

Figure 3.3: **(a)**-**(c)**, the plot is for the same data and SP mechanism as in Figure 3.2 but for varying k. The two axes give the coordinate of a point (record). The color gives the level of privacy in terms of the value $e^{-\varepsilon}$, for $(0.25, k)$-SP AIQ for every record. **(a)**, $k = 1$. **(b)**, $k = 7$. **(c)**, $k = 14$.

sensitive privacy provide exactly the same guarantee. In general, every (ε, k)-SP mechanism M for G_S satisfies $P(M(x) \in R) \leq P(M(y) \in R)\ e^{\varepsilon d_{G_S}(x,y)}$ for every $x, y \in \mathcal{D}$ that are connected and every R (recall that d_{G_S} is the shortest path metric over G_S, defined over the connected components of G_S, given in Section 2.3).

The parameter k ,which is associated with the sensitive neighborhood graph, provides a way to quantify what is deemed as a small change in the database, which varies from domain to domain. Nevertheless in many common cases it can be quantified over an appropriate metric space[2]. When we increase the value of k, we move the boundary between what is considered sensitive and what is non-sensitive: higher the value of k, the more records are considered sensitive, and therefore, must be protected. For an example, see Figure 3.3 (where the plots are for the same data, parameters, and method

[2]The metric space we are using for anomaly identification has a rather complicated structure, but it is induced by formalizing our intuition for sensitive records.

as for Figure 3.2d but for varying k). This is due to the fact that, for any $k \geq 1$, if a record is k-sensitive with respect to a database x, it is also $(k+1)$-sensitive with respect to x. For example, with respect to a database x, a 2-sensitive record, may not be 1-sensitive, but a 1-sensitive record will also be 2-sensitive.

In conceptualizing sensitive privacy, we prioritize protecting privacy of the normal records and the accuracy for anomaly identification. This results in the non-sensitive records receiving less stronger privacy guarantee compared to the normal records. This loss in privacy for the data-dependent definition of anomalies is inevitable as discussed in Section 3.1 and [29, 26, 27, 28]. However, we know that in practical settings (Section 2.1), the data contains only a small number of outlier records, most of which are non-sensitive records. Thus, this less stronger privacy guarantee is for a very small number of records, and most records enjoy a very strong privacy guarantee under sensitive privacy.

3.5 Composition

Our formalization of sensitive privacy enjoys the important properties of composition and post-processing, which a good privacy definition should have [62]. These properties help to quantify how much privacy may be lost (in terms of the value of ε) if one asks multiple queries or uses multiple sensitively private mechanisms in the analysis.

Here, we emphasize that sensitive privacy composes with respect to both, the privacy parameter (i.e. ε) and the k-sensitive neighborhood graph. This is because sensitive privacy is defined with respect to the k-sensitive neighborhood graph for the privacy parameter ε. For instance, if we answer multiple queries with sensitive privacy guarantee for different values of ε, but for the same k-sensitive neighborhood graph, then sensitive privacy composes similar to differential privacy, that is, the overall privacy loss is not more than sum of all the ε corresponding to each of the queries.

Furthermore, we note that all the mechanisms in the following discussion are assumed to have their own independent source of randomness.

3.5.1 Sequential composition

Sequential composition provides the privacy guarantee over multiple queries that use a common set of (one or more) records to compute the result. Consider two mechanisms, $M_1 : \mathcal{D} \to R$, which is ε_1-sensitively private for k_1-sensitive neighborhood graph $G_{S_1} = (\mathcal{D}, E_1)$, and $M_2 : \mathcal{D} \times R \to R'$, which is ε_2-sensitively private for k_2-sensitive neighborhood graph $G_{S_2} = (\mathcal{D}, E_2)$, with their independent sources of randomness. Recall (from Section 2.3) that in the context of anomaly identification query (AIQ), a private mechanisms for AIQ, (i, g), is fixed; thus M_1 and M_2 may correspond to different records and anomaly identification function. Now, $M_2(x, M_1(x))$ (for every database x) is $(\varepsilon_1 + \varepsilon_2)$-sensitively private for $G_S = (\mathcal{D}, E_1 \cap E_2)$ (Claim 3.2).

One application of this is that for a fixed G_S, even performing multiple queries interactively will lead to at most a linear loss (in terms of ε) in privacy in the number of queries — in an interactive query over a database x, one firstly gets the answer of M_1, i.e. $M_1(x)$, and based on this, one selects M_2.

Furthermore, for a fixed normality property, if $k_1 \leq k_2$ then G_{S_1} is a subgraph of G_{S_2}, and M_2 is $(\varepsilon_1 + \varepsilon_2)$-sensitively private for G_{S_1}.

Below, we state the claim for sequential composition of sensitive privacy. First we need to make an observation about a fact about the general neighborhood graphs, which is helpful in proving the claims about composition. Recall that for a graph G, $\mathcal{E}(G)$ gives the set of edges in G.

Fact 3.1. *For every neighborhood graph G and G', and every $x, y \in \mathcal{D}$, x and y are neighbors in the graph $H = (\mathcal{D}, \mathcal{E}(G) \cap \mathcal{E}(G'))$ if and only if they are neighbors in G and G'.*

Claim 3.2 (sequential composition)**.** *If mechanisms M_1 and M_2 are respectively ε_1-SP for G_{S_1} and ε_2-SP for G_{S_2}, then $M(x) := (M_1(x), M_2(x))$ for every x is $(\varepsilon_1 + \varepsilon_2)$-SP for $G_S = (\mathcal{D}, \mathcal{E}(G_{S_1}) \cap \mathcal{E}(G_{S_2}))$.*

Proof sketch. The claim follows from M_1 and M_2 being SP for ε_1 and ε_2, and Fact 3.1, which ensures that the privacy constraints will be met for neighbors in G_S. □

3.5.2 Parallel composition

Parallel composition deals with multiple queries, each of which only uses non-overlapping partition of the database. Let $\mathcal{X} = Y_1 \sqcup Y_2$ (i.e. $Y_1 \cap Y_2 = \emptyset$). Now, consider M_1 and M_2, each with domain $\mathcal{D}$, that are respectively ε_1-sensitively private for G_{S_1} and ε_2-sensitively private for G_{S_2}, where G_{S_1} is a subgraph of G_{S_2}. Further, M_1 and M_2 only depend on their randomness (each with its independent source) and records in Y_1 and Y_2 respectively. In this setting, the mechanism $M(x) = (M_1(x), M_2(x))$ is $\max(\varepsilon_1, \varepsilon_2)$-sensitively private for G_{S_1}, or in general case for sensitive neighborhood graph $(\mathcal{D}, E_1 \cap E_2)$ (Claim 3.3), where E_1 and E_2 are the sets of edges for G_{S_1} and G_{S_2} respectively.

To give the claim about parallel composition, we need to formalize the notion that a mechanism only depends upon a fragment of the data. We say, for $Y \subseteq \mathcal{X}$, a mechanism M is Y-dependent if and only if for every $r \in Range(M)$ and x and y such that $x_i = y_i$ for every $i \in Y$, $P(M(x) = r) = P(M(y) = r)$.

Claim 3.3. *For any partition of $\mathcal{X} = Y_1 \sqcup Y_2$, if mechanisms M_1 and M_2 are respectively Y_1-dependent ε_1-SP for G_{S_1} and Y_2-dependent ε_2-SP for G_{S_2}, then $M(x) := (M_1(x), M_2(x))$ for every x is $\max(\varepsilon_1, \varepsilon_2)$-SP for $G_S = (\mathcal{D}, \mathcal{E}(G_{S_1}) \cap \mathcal{E}(G_{S_2}))$.*

Proof sketch. Firstly, note that M_1 and M_2 being SP for ε_1 and ε_2 along with Fact 3.1, ensure that the privacy constraints will be met for neighbors in G_S for $\varepsilon = \max(\varepsilon_1, \varepsilon_2)$. Further, since every neighbor in G_S differ by one record and mechanisms M_1 and M_2 are respectively Y_1 and Y_2 dependent (for an arbitrarily fixed partition), every privacy constraint will hold for either ε_1 or ε_2. From here the claim follows. □

We also remark that privacy is maintained under post-processing for the same value of ε and the neighborhood graph.

3.5.3 Example for (β, r)-anomaly

Consider composition for the case of multiple (β, r)-AIQs. Let us say we answer anomaly identification queries for records $i_1, i_2, \ldots, i_n$ respectively for (β_1, r_1), (β_2, r_2),

$\ldots,(\beta_n, r_n)$ anomalies over the database x, while providing sensitive privacy.

Let the mechanism for answering (β_t, r_t)-AIQ for i_t be (ε_t, k_t)-SP for k_t-sensitive neighborhood graph for (β_t, r_t)-normality property. Further, assume that the mechanism for (β_t, r_t)-AIQ Y_t-dependent, where $Y_t = \{i \in \mathcal{X} : d(i, i_t) \leq r_t\}$, that is, the mechanism depends on the partition of the database that contains the records within distance r_t of i_t (because it suffices to compute (β_t, r_t)-AIQ) and its independent source of randomness. Finally, let $k = \min(k_1, \ldots, k_n)$, $\beta = \max(\beta_1, \ldots, \beta_n)$, and $r = \min(r_1, \ldots, r_n)$.

In this case, *answering all of the queries provides an overall guarantee of* $(m\varepsilon, k)$-*sensitive privacy for* (β, r)-*normality property*, where m is the maximum number of i_t's that are within any ball of radius $\max(r_1, \ldots, r_n)$ (Claim 3.4).

Thus, from the above, it follows that if we fix β, r and k and allow a querier to ask m many (β', r)-AIQ's (each may have a different value for β') such that $\beta' \leq \beta$, then we can answer all of the queries with sensitive privacy $(m\varepsilon, k)$ in the worst case for k-sensitive neighbor for to (β, r)-normality property. The same is true if the queries are for (β, r') with $r' \geq r$. Furthermore, for fixed β, r and k, answering (β, r)-AIQ for i and i' such that $d(i, i') > 2r$ still maintains (ε, k)-SP. One may employ this to query adaptively to carry out the analysis while providing sensitive privacy guarantees over analysis as a whole.

To prove Claim 3.4, we need the following lemma (this will also be useful for proving some other results related to private (β, r)-AIQ, presented in the following chapters). For (β, r)-anomaly, Lemma 3.1 gives the necessary and sufficient condition for a record to be sensitive.

Lemma 3.1. *Fix arbitrary values for* $k \geq 1$, $\beta \geq 1$ *and* $r \geq 0$, *and fix the normality property corresponding to* (β, r)*-anomaly. Then for every* $i \in \mathcal{X}$ *and every* $x \in \mathcal{D}$,

$$i \text{ is } k\text{-sensitive with respect to } x \iff B_x(i, r) \geq \beta + 1 - k.$$

Proof. Arbitrarily fix $k, \beta \geq 1$, $r \geq 0$, $i \in \mathcal{X}$, and $x \in \mathcal{D}$. Further, fix p to be the normality property corresponding to (β, r)-anomaly.

Firstly, we prove the "if" direction through its contrapositive. Assume $B_x(i, r) <$

$\beta + 1 - k$. Now, for every y such that $||x - y||_1 \leq k$, $B_y(i, r) \leq \beta$ since we can only add up to k records in x. Thus for each of the above y, $p(i, y) = 0$, which follows from the definition of (β, r)-anomaly, and i is not k-sensitive with respect to x. This completes the proof for "if" direction.

Next, we prove the "only if" direction. Let $B_x(i, r) \geq \beta + 1 - k$. Consider a y, obtained by adding k records, which are the same as i, to x. For this y, it holds that $||x - y||_1 = k$ and $p(i, y) = 0$ because $y_i \geq 1$ and $B_y(i, r) \geq \beta + 1$ (since $k \geq 1$). Hence, we conclude that i is k-sensitive for x. This completes the proof as k, β, r, i, and x were chosen arbitrarily. □

For any $i \in \mathcal{X}$ and $r \geq 0$, we write $Y(i, r)$ to denote the set $\{j \in \mathcal{X} : d(i, j) \leq r\}$.

Claim 3.4. *Fix an arbitrary value of $n \in \mathbb{N}$.*

- *For every $t = 1, \ldots, n$, arbitrarily fix $\varepsilon_t, r_t > 0$, $k_t, \beta_t \geq 1$, and a mechanism, $M_t : \mathcal{D} \to \{0, 1\}$, such that: (1) it is (ε_t, k_t)-SP for (β_t, r_t)-normality property, and (2) it is $Y(i_t, r_t)$-dependent.*
- *Let $\varepsilon = \max(\varepsilon_1, \ldots, \varepsilon_n)$, $k = \min(k_1, \ldots, k_n)$, $\beta = \max(\beta_1, \ldots, \beta_n)$, and $r = \min(r_1, \ldots, r_n)$.*

Then for $M(x) := (M_1(x), \ldots, M_n(x))$ for every x, M is $(m\varepsilon, k)$-sensitively private for (β, r)-normality property, where m be the maximum number of i_t's that are within any ball of radius $\max(r_1, \ldots, r_n)$.

Proof. Arbitrarily fix the values for all the symbols used in the claim above as per the specification in the lemma.

Firstly, we consider the guarantee with respect to the sensitive neighborhood graph. Here it is sufficient to show that the k-sensitive neighborhood graph, G_S, corresponding to (β, r)-normality property, is a subgraph of the k_t-sensitive neighborhood graph, G_S^t, corresponding to (β_t, r_t)-anomaly for every t. Thus we show that, for any t and two databases x and y, if x and y are neighbors in G_S, then they are neighbors in G_S^t.

Arbitrarily fix $t \in [n] = \{1, 2, \ldots, n\}$, and databases x and y that are neighbors in G_S. Since x and y are neighbors in G_S, there exists a record i that k-sensitive

with respect to x or y. Let i be k-sensitive with respect to x — this is without loss of generality since x and y are picked arbitrarily. Because $B_x(i, r_t) \geq B_x(i, r)$, from Lemma 3.1, we get that $B_x(i, r) \geq \beta - k + 1$. Since $\beta \geq \beta_t$ and $k \leq k_t$, $B_x(i, r) \geq \beta_t - k_t + 1$; this implies that i is k_t-sensitive with respect to x (Lemma 3.1), and thus, x and y are neighbors in G_S^t. Since t was picked arbitrarily, we conclude that G_S is a subgraph of G_S^t for every $t \in [n]$.

Next, we prove the bound on the divergence of probabilities to show that the loss in privacy is at max $m\varepsilon$.

For any $j \in \mathcal{X}$, let A_j be such that for every $t \in [n]$, $t \in A_j \iff d(j, i_t) \leq r'$, where $r' = \max(r_1, \ldots, r_n)$. And let $m = \max_{j \in \mathcal{X}} |A_j|$. Arbitrarily fix the neighboring databases x and y in G_S and $w \in \{0,1\}^n$. Let i be the record in which x and y differ, i.e. $|x_i - y_i| = 1$. Now it follows that

$$\begin{aligned}\frac{\mathrm{P}(M(x) = w)}{\mathrm{P}(M(y) = w)} &= \prod_{t \in A_i} \frac{\mathrm{P}(M_t(x) = w_t)}{\mathrm{P}(M_t(y) = w_t)} \times \prod_{l \in [n] \setminus A_i} \frac{\mathrm{P}(M_l(x) = w_l)}{\mathrm{P}(M_l(y) = w_l)} \\ &= \prod_{t \in A_i} \frac{\mathrm{P}(M_t(x) = w_t)}{\mathrm{P}(M_t(y) = w_t)} \leq \exp\left(\sum_{t \in A_i} \varepsilon_t\right) \leq \exp(m\varepsilon)\end{aligned}$$

Above, the first equality holds because each of the M_t has its independent source of randomness. The second equality holds because each M_t is $Y(i_t, r_t)$-dependent in addition to its randomness and $r_t \leq r'$. The first inequality follows from M_t being ε_t-SP for G_S, which is a subgraph of G_S^t. The last inequality follows from the fact that $\varepsilon \geq \varepsilon_t$ and $m \geq |A_i|$.

Lastly, we note that from the above it follows that for any $W \subseteq \{0,1\}^n$

$$\frac{\mathrm{P}(M(x) \in W)}{\mathrm{P}(M(y) \in W)} \leq \frac{\sum_{w \in W} \mathrm{P}(M(x) = w)}{\sum_{w \in W} \mathrm{P}(M(y) = w)} \leq \exp(m\varepsilon)$$

Thus, we conclude that the claim holds. □

3.6 Privacy Under a Regular Normality Property

Here, we consider a set of normality properties for which sensitive privacy guarantee is strong and practically meaningful — we refer to such properties as *regular properties*

(defined shortly). We also show that, for most practical setting, the normality property corresponding to $(\beta, r > 0)$-anomaly is regular.

Regular properties, P, make a subset of the set of all properties $\mathfrak{P}$ (Section 2.3) (i.e. $P \subsetneq \mathfrak{P}$). For every property in P, the sensitive neighborhood graph, G_S, is such that every two nodes (i.e. databases), each of which has at least one sensitive record, are connected.

To define the notion of regular property, we first need to clarify some notation and definitions. For arbitrarily fixed $k \geq 1$ and a property $p \in \mathfrak{P}$, let D^p be a subset of $\mathcal{D}$ such that $D^p = \{x \in \mathcal{D} : \exists i \in \mathcal{X} \text{ s.t. } i \text{ is } k\text{-sensitive w.r.t. } x\}$. And for a given D^p and a k-sensitive neighborhood graph, G_S, corresponding to p, let $G_S(D^p) = (D^p, E)$ be a subgraph of G_S such that for every $x, y \in D^p$, $\{x, y\} \in E$ if an only if $\{x, y\} \in \mathcal{E}(G_S)$. The definition follows.

Definition 3.4 (regular property)**.** *For any $p \in \mathfrak{P}$, we say p is* regular *if for every $k \geq 1$, $G_S(D^p)$ is connected, where G_S is the k-sensitive neighborhood graph for p.*

Therefore, for any regular property p, all the database that have at least one sensitive records are in one connected component, C, of the sensitive neighborhood graph, and the only databases that are not connected with this component (C) of the graph are the ones that do not have any sensitive record.

When a database x is such that there is no record that is sensitive with respect to x, there is no normal record in the database. However, we know that most of the records in real-world databases for outlier analysis are normal records, and only a small minority of records constitute anomalies. Thus, for the practical settings for outlier analysis, sensitive privacy guarantee is strong as well as meaningful.

3.6.1 Regular vs. non-regular (β, r)-normality properties

We use p to depict an arbitrary (β, r)-normality property. Recall that we refer to the normality property corresponding to (β, r)-anomaly as (β, r)-*normality property*.

(β, r)-normality properties for most practical settings are indeed regular, but not all. For instance, for any $\beta \geq 1$ and $k < \beta$, the $(\beta, 0)$-normality property is not regular.

To confirm this, arbitrarily fix values of $\beta \geq 1$ and $k < \beta$. Let x and y be two databases such that for some $i \neq j$, $x_i = \beta+1-k$ and $x_j = 0$, and $y_i = 0$ and $y_j = \beta+1-k$. Note that i is k-sensitive with respect to x but not y, and j is k-sensitive with respect to y but not x. This is since for any database z and record l, l is k-sensitive with respect to z if and only if $B_z(l, r = 0) = z_l \geq \beta + 1 - k$ (Lemma 3.1). Although both databases have at least one record that is k-sensitive with respect to each, they are not connected in the k-sensitive neighborhood graph. This holds because for every neighbor z of x, j is not k-sensitive with respect to z ($z_j + k \leq \beta$); in fact, for every neighbor z of x, $z_j = x_j = 0$. Furthermore, a simple inductive argument (on the databases at distance ℓ from x) shows that any database, z, that is connected to x has $z_j = 0$. Thus, we conclude that y is not connected to x, and the $(\beta, 0)$-normality property (for $k < \beta$) is not regular.

We use $k < \beta$ constraint because when $k \geq \beta$, for every $i \in \mathcal{X}$ and $x \in \mathcal{D}$, there is a neighbor y of x such that i is k-sensitive for x or y; and in this case, the k-sensitive neighborhood graph for (β, r)-normality property is the same as the DP neighborhood graph.

We now characterize the condition that makes a (β, r)-normality property regular. For this, we use radius parameter r. When r is *non-trivial*, the (β, r)-normality property is regular. To define what makes r non-trivial, we first need to clarify some notation. We use $\mathcal{S} : \mathcal{X} \times \mathbb{N} \to 2^{\mathcal{X}}$ (where $2^{\mathcal{X}}$ is the power set of $\mathcal{X}$) to define the sets of records that are reachable from a given record i. For any given r, every $\ell \in \mathbb{N}$ and every $i \in \mathcal{X}$, $\mathcal{S}(i, \ell) = \bigcup_{j \in \mathcal{S}(i,\ell-1)} X(j, r)$ if $\ell > 0$ otherwise $\mathcal{S}(i, \ell) = \{i\}$, whereas $X(i, r) = \{j \in \mathcal{X} : \ d(i, j) \leq r\}$ (recall that (β, r)-anomaly comes with a metric over $\mathcal{X} \times \mathcal{X}$, see Section 2.3).

For (β, r)-anomaly, we say the radius parameter $r \geq 0$ is *non-trivial* if there exists $s \in \mathbb{N}$ such that for every $i \in \mathcal{X}$, $\mathcal{S}(i, s) = \mathcal{X}$. Below, we give Claim 3.5 stating that if the parameter r is non-trivial then $\mathfrak{p}$ is a regular ((β, r)-normality) property.

Claim 3.5. *For arbitrarily fixed $\beta \geq 1$, and $r \geq 0$, if the parameter r is non-trivial for (β, r)-anomaly, then (β, r)-normality property, $\mathfrak{p}$, is regular.*

Proof. Arbitrarily fix a (finite) set $\mathcal{X}$ of size m and a (β, r)-normality property, $\boldsymbol{p}$, for arbitrarily fixed $\beta \geq 1$ and $r \geq 0$ that is non-trivial. Next fix an arbitrary value of $k \geq 1$, and let G_S be the k-sensitive neighborhood graph for $\boldsymbol{p}$.

We now recall that $D^{\boldsymbol{p}}$ is the maximal set such that $D^{\boldsymbol{p}} \subseteq \mathcal{D}$ and for every $x \in \mathcal{D}$, $x \in D^{\boldsymbol{p}} \iff$ there exists $i \in \mathcal{X}$ that is k-sensitive with respect to (w.r.t.) x.

We prove the claim in two steps through a reachability argument, where we show every database (node) in $G_S(D^{\boldsymbol{p}})$ is connected to a fixed database that has β mass in each coordinate. For this proof, we arbitrarily fix an $x \in D^{\boldsymbol{p}}$ and an $i \in \mathcal{X}$ such that i is k-sensitive w.r.t x, and hence, $B_x(i, r) \geq \beta + 1 - k$ (follows from Lemma 3.1).

Next, we define a function ω, which we will use to give databases that are same as x except they differ from x in the coordinate i. For every $a \in \mathbb{N}$, we define $\omega_x(a) = x + a \cdot \mathsf{sgn}(\beta - x_i)\mathbf{e}^i$, where sgn is the standard *signum* function that outputs 0 for the input 0. Thus, $\omega_x(a)$ is the same as x except for ith coordinate where its mass is more (less) by a than that of x if $x_i < \beta$ (respectively if $x_i > \beta$).

Part (a): Here we show that the database x is connected to a database in $G_S(D^{\boldsymbol{p}})$ that is same as x except for the coordinate i, where it has mass β (i.e. the database $\omega_x(|\beta - x_i|)$).

When $|\beta - x_i| = 0$, the claim holds trivially true (as x is reachable from itself). So, we consider the case, when $|\beta - x_i| > 0$. Here, for $a = 1, \ldots, |\beta - x_i|$, $\omega_x(a-1)$ is a neighbor of $\omega_x(a)$ in G_S, and both $\omega_x(a-1)$ and $\omega_x(a)$ belong to $D^{\boldsymbol{p}}$ because for every a (as given above), i is sensitive w.r.t. $\omega_x(a)$ as $B_{\omega_x(a)}(i, r) \geq \beta + 1 - k$.

From the above it follows that x is connected to the database $\omega_x(|\beta - x_i|) \in D^{\boldsymbol{p}}$ through the path $< x, \omega_x(1), \ldots, \omega_x(|\beta - x_j|) >$. Furthermore, since x and i were fixed arbitrarily, this claim holds for every i and and every $x \in D^{\boldsymbol{p}}$ such that i is sensitive with respect to x.

Part (b): Here, we will use an inductive argument to show that the database, which has β mass in each coordinate, is reachable from x.

For our fixed x and any $J \subseteq \mathcal{X}$, let y^J be a database such that for every $j \in J$,

$y^J_j = \beta$, and for every $j \in \mathcal{X} \setminus J$, $y^J_j = x_j$. And let s be the smallest integer such that $\mathcal{S}(i,s) = \mathcal{X}$ — this holds because r is non-trivial. We want to show that $y^{\mathcal{S}(i,s)}$ reachable from x.

Firstly, note that $y^{\mathcal{S}(i,0)}$ can be reached from x (Part (a)) — this proves the base case. Next, assume that for some $\ell \geq 0$, $y^{\mathcal{S}(i,\ell)}$ is reachable from x (inductive hypothesis). We will show that $y^{\mathcal{S}(i,\ell+1)}$ is also reachable from x.

If $\ell \geq s$, then the claim holds trivially; hence, in the following we assume that $\ell < s$.

Let $J = \mathcal{S}(i,\ell) \cup \big(\mathcal{S}(i,\ell+1) \setminus \mathcal{S}(i,\ell)\big) = \mathcal{S}(i,\ell) \cup \{i_1, i_2, \ldots, i_n\}$ for some n, and for every t in $\{0,1,2,\ldots,n\}$, $J_t = \mathcal{S}(i,\ell) \cup \{i_1, i_2, \ldots, i_t\}$, where $J_0 = \mathcal{S}(i,\ell)$.

Note that every $j \in J$ is k-sensitive with respect to $y^{\mathcal{S}(i,\ell)}$. To confirm this, arbitrarily fix a $j \in J$. If $j \in \mathcal{S}(i,\ell)$ then j is sensitive with respect to $y^{\mathcal{S}(i,\ell)}$ (follows from Lemma 3.1 as $y^{\mathcal{S}(i,\ell)}_j = \beta$). If, however, $j \in J \setminus \mathcal{S}(i,\ell)$, then there exists a $j' \in \mathcal{S}(i,\ell)$ such that $d(j,j') \leq r$ (follows from the definition of $\mathcal{S}$), and for $z = y^{\mathcal{S}(i,\ell)}$, $B_z(j,r) \geq \beta$; thus, j is sensitive with respect to $z = y^{\mathcal{S}(i,\ell)}$ (Lemma 3.1).

Now, from Part (a), it follows that, y^{J_1} is reachable from y^{J_0}, and y^{J_2} is reachable from y^{J_1}, and so on. Thus, y^J is reachable from y^{J_0}, and the inductive hypothesis implies that y^J is also reachable from x.

Finally, we conclude from the above that $y^{\mathcal{S}(i,s)}$ can be reached from x. Since $x \in D^p$ was fixed arbitrarily, the claim holds true for every $x \in D^p$. Because we fixed k arbitrarily, the claim holds for all $k \geq 1$ — this completes the proof.

□

3.7 Relation of Sensitive Privacy with Other Privacy Definitions

In this section, we show how sensitive privacy relates to the other related data privacy concepts in the literature.

3.7.1 Differential privacy

We begin by restating the definition of differential privacy in our language of the neighborhood graph.

Definition 3.5. *For any $\varepsilon > 0$ and mechanism, M, with domain $\mathcal{D}$, we say M is ε-differentially private if for every neighboring x and y in the DP neighborhood graph, and every $R \subseteq Range(M)$,*

$$\mathrm{P}\left(M(x) \in R\right) \leq e^{\varepsilon}\mathrm{P}\left(M(y) \in R\right).$$

Firstly, note that for every $x, y \in \mathcal{D}$, $||x - y||_1 = 1 \iff d_{\mathbb{G}}(x, y) = 1$ (Lemma 3.2), where $d_{\mathbb{G}}$ is the shortest path metric over the DP neighborhood graph $\mathbb{G}$. Thus Definition 3.5 is equivalent to Definition 2.2.

Lemma 3.2. *If $\mathbb{G}$ is the DP neighborhood graph and $d_{\mathbb{G}}$ is the shortest path metric over $\mathbb{G}$, then for every $x, y \in \mathcal{D}$, $d_{\mathbb{G}}(x, y) = ||x - y||_1$.*

Proof. Let $\mathbb{G}$ be the DP neighborhood graph over $\mathcal{D}$ and $d_{\mathbb{G}}$ be the shortest path metric over $\mathbb{G}$. Since for $x = y$, $d_{\mathbb{G}}(x, y) = ||x - y||_1 = 0$, arbitrarily fix $x, y \in \mathcal{D}$ such that $x \neq y$. Since each edge, from one databases to an other, represents an addition or removal of a record, the shortest path between x and y represents the minimum number of such changes required to modify x into y or vice versa, which is exactly the quantity measured by ℓ_1-distance. Thus, we conclude $d_{\mathbb{G}}(x, y) = ||x - y||_1$. □

The restatement of differential privacy makes it easy to see that if a k-sensitive neighborhood graph, G_S, is the same as the DP neighborhood graph, $\mathbb{G}$ (i.e., $G_S = \mathbb{G}$), then a mechanism is (ε, k)-sensitively private if and only if it is ε-differentially private. One can easily confirm this by: (1) comparing Definition 3.3 with Definition 3.5, and (2) using the fact that when $G_S = \mathbb{G}$.

Furthermore, if a mechanism is ε-differentially private then it is also (ε, k)-sensitively private for all $k \geq 1$ and normality properties. This follows from the fact that every k-sensitive neighborhood graph is a subgraph of DP neighborhood graph (i.e. any two neighboring databases in k-sensitive neighborhood graph are also neighbors in the DP neighborhood graph).

Lastly, we note that if a mechanism, M, is (ε, k)-sensitively private, then there exists an $\varepsilon' \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ such that the M is ε'-differentially private (Claim 3.6). Note that when $\varepsilon' = 0$, we achieve the maximum level of privacy, that is, for all neighboring

databases x and y in $\mathbb{G}$ and every $R \subseteq Range(M)$, $\mathrm{P}\big(M(x) \in R\big) = \mathrm{P}\big(M(y) \in R\big)$. But when $\varepsilon' = \infty$, there is no privacy at all. In this case, M can output the correct answer (of the function it is computing) for every input.

Claim 3.6 only applies to the connected k-sensitive neighborhood graph, G_S. If the G_S is not connected, the guarantee under differential privacy will be for $\varepsilon' = \infty$. This is because, for a non-connected G_S, there are two databases differing by one record (i.e. neighbors as per DP neighborhood graph) that are not connected. For these databases, a mechanism, which is sensitively private for G_S, can arbitrarily diverge in terms of its distributions.

Claim 3.6. *Arbitrarily fix $\varepsilon > 0$, a connected k-sensitive neighborhood graph, G_S, for a normality property, and let $\alpha = \max_{||x-y||_1=1} d_{G_S}(x, y)$. For any mechanism M, if M is (ε, k)-sensitively private then it is $(\varepsilon \cdot \alpha)$-differentially private.*

The above claim follows from the fact that for an ε-sensitively private mechanism, the probabilities corresponding to x and y for any of the output are within a multiplicative factor of $e^{\varepsilon d_{G_S}(x,y)}$ (Claim 3.7).

Claim 3.7. *Fix any $\varepsilon > 0$, $k \geq 1$, and a normality property p such that G_S is the k-sensitive neighborhood graph for p and is connected. For any mechanism, M, with domain $\mathcal{D}$, if M is (ε, k)-sensitively private, then for every $x, y \in \mathcal{D}$ and $R \subseteq Range(M)$,*

$$P(M(x) \in R) \leq e^{\varepsilon \cdot d_{G_S}(x,y)} P(M(y) \in R)$$

The above claim immediately follows from the definition of sensitive privacy, that is, (on any of the shortest paths from x to y) each of the pair of neighbors, z and z', in the sensitive neighborhood graph satisfies $\mathrm{P}(M(z) \in R) \leq e^{\varepsilon} \mathrm{P}(M(z') \in R)$.

3.7.2 Blowfish & Pufferfish privacy

Blowfish and Pufferfish privacy [49, 63, 48] are general frameworks for generating privacy definitions. Basically, any definition of privacy, that is, based on "indistinguishability" of neighboring databases (in term of privacy constraints, e.g. as given in Definition 2.2 and Definition 3.3) is covered under one of these frameworks. Under these frameworks,

the main problem to solve is to decide what secret one wants to protect and then define which databases are neighbors so as to protect the secret.

In Blowfish privacy [63], one needs to define what secret we want to protect, which then results in the neighboring databases. In this context, our proposed notion of k-sensitive record plays a vital role as it induces the k-sensitive neighborhood graph. Now, if we define the secret to be the presence or absence of a sensitive record (for a prefixed value of k), then the resultant neighbors would be the same as given by k-sensitive neighborhood graph. In this case, a mechanism will be sensitively private if and only if it is private under Blowfish privacy.

Pufferfish privacy assumes that there is an attacker (or a set of attackers) against whom we want to protect privacy. It makes the attacker's information explicit, that is, from what distribution (or set of distributions) the databases will be drawn and what it is that we are trying to protect. In the case of sensitive privacy, the information to be protected is whether or not a sensitive record is present in the database. In this setting, if the presence and absence of every record is independent from the presence and absence of the other records, then for the secrets to be protected (specified by the sensitive neighborhood graph), a mechanism is sensitively private if and only if it is private under Pufferfish privacy [63, 48].

3.7.3 Protected differential privacy

In [26], the authors present a definition of privacy and private algorithms for a targeted search in social networks, which can be used to search for anomalies (e.g., terrorists) as well. Their algorithms are private under protected differential privacy, their proposed notion.

However, as discussed in Section 2.2, the definition of anomaly (or the normality property in our context) that we can consider under their definition cannot be data-dependent. Namely, for a pre-specified $X \subseteq \mathcal{X}$ (the set of the protected population), we can define the corresponding normality property, p, where p is such that for every $i \in \mathcal{X}$ and $x \in \mathcal{D}$, $p(i, x) = 1 \iff i \in X$. Thus, protected differential privacy deals with a subclass of definitions from sensitive privacy. Furthermore, in this context, solving the

problem for the data-dependent definition of anomalies is an open question [26] that sensitive privacy addresses.

3.7.4 Tailored differential privacy

Tailored differential privacy [29] generalizes differential privacy, wherein the privacy parameter, ε, is a function of a record and a database (i.e. $\epsilon : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$). Hence, it allows for having different levels of privacy (i.e. the value of ε) for different records.

Definition (TDP). For any given $\epsilon : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ and mechanism M (with domain $\mathcal{D}$), M is called $\epsilon(\cdot)$-*tailored differentially private* if for every $x \in \mathcal{D}$ and every $i \in \mathcal{X}$ such that $x_i \geq 1$, the following holds

$$M(x) \leq e^{\epsilon(i,x)} M(x - \mathbf{e}^i) \text{ and } M(x - \mathbf{e}^i) \leq e^{\epsilon(i,x)} M(x).$$

Sensitive privacy deals with a subclass of mechanisms that are private under tailored differential privacy. If a mechanism is sensitively private then there exists a function ϵ such that the mechanism is also $\epsilon(\cdot)$-tailored differentially private. Alternatively, we can say that for specific function ϵ, a mechanism is sensitively private if and only if it is tailored differentially private (Claim 3.8).

We define the *privacy-functions*, ϵ_α, that we need to formally state the claim. For any given k-sensitive neighborhood graph G_S (for an arbitrary $p \in \mathfrak{P}$) and a fixed $\alpha > 0$, we say $\epsilon_\alpha : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a *privacy-function* if for every $i \in \mathcal{X}$ and every $x \in \mathcal{D}$, $\epsilon_\alpha(i, x) = \alpha \cdot d_{G_S}(x, \xi(x - \mathbf{e}^i))$ if x and $\xi(x - \mathbf{e}^i)$ are connected in G_S otherwise $\epsilon_\alpha(i, x) = \infty$. Recall that ξ, for a given input, replaces the negative value of each coordinate with zero, and $\mathbf{e}^i$ is a database that consists of only one record that is of value i (see Section 2.3 for details).

Claim 3.8. *Arbitrarily fix $k \geq 1$, $\alpha > 0$, a normality property p, and let G_S be the k-sensitive neighborhood graph for p. If ϵ_α is the privacy-function for G_S (as given above), then for every mechanism M with domain $\mathcal{D}$, M is (α, k)-sensitively private if and only if it is $\epsilon_\alpha(\cdot)$-tailored differentially private.*

Proof. Let k, p, α, and G_S be as given above. Let ϵ_α be the privacy-function for G_S, and M be a mechanism with domain $\mathcal{D}$.

We first prove the "only if" direction. Let M be (α, k)-sensitively private (SP). Arbitrarily pick $x \in \mathcal{D}$ and $i \in \mathcal{X}$ such that $x_i \geq 1$. Let $y = x - \mathbf{e}^i$. Thus, by definition of G_S, y is a node in G_S. Now, if x and y are connected in G_S, then for an arbitrarily fixed $R \subseteq Range(M)$, it follows that

$$\begin{aligned}
\mathrm{P}\big(M(x) \in R\big) \leq & e^{\alpha d_{G_S}(x,y)} \mathrm{P}\big(M(y) \in R\big) \\
\mathrm{P}\big(M(x) \in R\big) \leq & e^{\epsilon_\alpha(i,x)} \mathrm{P}\big(M(y) \in R\big)
\end{aligned}$$

The first inequality follows from the definition of sensitive privacy, whereas the second one follows from the definition of ϵ_α. Thus one of the constraints of tailored differential privacy (TDP) holds for x and i. Similarly, from the symmetry of d_{G_S} and the second constraint imposed by SP, the other privacy constraint of TDP follows.

On the other hand, if x and y are not connected in G_S, then under sensitive privacy, M's distributions corresponding to x and y are allowed to arbitrarily diverge from each other. The same is the case for TDP under our ϵ_α, which is equal to ∞ here. Thus, in this case, the claim holds as well. Since x, i, and R were picked arbitrarily, the constraints hold for every x and i (such that $x_i \geq 1$) and R. Hence by the definition of TDP, M is $\epsilon_\alpha(\cdot)$-TDP. This completes the proof for "only if" direction.

Next, we prove the "if" direction. Let M be $\epsilon_\alpha(\cdot)$-TDP. Arbitrary pick neighboring database x and y in G_S, and $R \subseteq Range(M)$. Since x and y are neighbors, there exists $i \in \mathcal{X}$ such that $||x - y||_1 = |x_i - y_i| = 1$; let us fix this i. Now, if $x_i > y_i$, then it follows that

$$\begin{aligned}
\mathrm{P}(M(x) \in R) \leq & e^{\epsilon_\alpha(i,x)} \mathrm{P}(M(y) \in R) \\
\mathrm{P}(M(x) \in R) \leq & e^{\alpha d_{G_S}(x,y)} \mathrm{P}(M(y) \in R) \\
\mathrm{P}(M(x) \in R) \leq & e^{\alpha} \mathrm{P}(M(y) \in R)
\end{aligned}$$

The first inequality holds because M is $\epsilon_\alpha(\cdot)$-TDP; the second one follows by the definition of ϵ_α, and the third one holds because $d_{G_S}(x, y) = 1$ for neighbors. Similarly, the second privacy constraint for SP follows from the other privacy constraint of TDP.

In the case, $x_i < y_i$, we get $\mathrm{P}(M(x) \in R) \leq e^{\epsilon_\alpha(i,y)}\mathrm{P}(M(y) \in R)$; here again we can show, in a similar fashion as above, that both the privacy constraints for SP hold. Since we picked the neighboring databases and R arbitrarily, the privacy constraints for privacy parameter α and k-sensitive neighborhood graph G_S hold for all the neighboring databases and R. Hence, M is (α, k)-SP. This completes the proof. □

3.8 Key Takeaways

This work is the first to lay out the foundations of the privacy-preserving study of data-dependent outliers. Our formalization and conceptual development are independent of any particular definition of outlier. Indeed, the definition of sensitive privacy (Definitions 3.1 and 3.3) applies to an arbitrary definition of anomaly. And in the next chapter, we will present our constructions that can be used to develop sensitively private mechanisms for an arbitrary definition of anomaly.

Sensitive privacy generalizes differential privacy, and thus, the guarantees provided by sensitive privacy are similar to that of differential privacy (DP). In general, however, the sensitive privacy guarantee for any two databases differing by one record could be correspondingly weaker than that offered by differential privacy depending on the distance between the two databases in the sensitive neighborhood graph. Furthermore, in contrast to DP that composes with respect to the privacy parameter ε, sensitive privacy composes with respect to ε and the k-sensitive neighborhood graph.

Nevertheless, sensitive privacy provides a strong DP-like guarantee for most of the records in the database without sacrificing the utility. This is in direct contrast to differential privacy, where to achieve the same utility (as in the case of sensitive privacy), one has to weaken the privacy guarantee for all.

Notes

[1]If the data is from one dimensional normal distribution with mean μ and standard deviation σ then a record i is anomalous (or equivalently an outlier) if $|i-\mu| \geq 3\sigma$, and is statistically equivalent to $(\beta = 1.2 \times 10^{-3}n,\ r = 0.13\sigma)$-anomaly [55], where n is the size of the database.

To adapt this result for 2D normal distribution in Figure 3.2, set $r = 0.13\sqrt{\sigma_1^2 + \sigma_2^2}$ and compute β in a similar fashion as above. Next, take 30 samples of size $20K$, i.e. $n = 20,000$, from the 2D normal distribution, $N(\mu, \Sigma)$, where $\mu = (0,0)$ and $\Sigma = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$, and run SP-mechanism (in Section 6.2) and DP-mechanism (in Section 6.1) for (β, r)-anomaly identification query to compute accuracy, which is measured by the probability of outputting the correct answer by the private mechanism, and average the results over the samples for each query. We then plot the average accuracy and interpolate the results using one-degree polynomial in the two coordinates (Figure 3.2b-c). We used the "ListPlot3D" function of Mathematica with the argument "InterpolationOrder" set to 1.

In Figure 3.2d-e and Figure 3.3, we plot the level of privacy (in term of ε) that each record (point) has under private anomaly identification query. Here, the level of privacy for a record in a given database is measured by the maximum divergence in the probability of outputting a label when we add or remove the record from the database. For ε-SP-mechanism, U, to compute the value of the privacy parameter, ε, for a record i in a given database x, consider databases y and z. y and z are same as x except for y has one more record of value i and z has one less record of value i—if there is no record of value i in x then z will be the same as x. Now we can calculate e^{ε} for record i be by (3.1).

$$e^{\varepsilon} = \max_{w \in \{y,z\}} \max_{b \in \{0,1\}} \left(\frac{P(U(x) = b)}{P(U(w) = b)}, \frac{P(U(w) = b)}{P(U(x) = b)} \right) \tag{3.1}$$

CHAPTER 4

Sensitive Privacy and Mechanism Design

How can we achieve sensitive privacy for identifying anomalies? In this chapter, we answer this question. We will first look at the simple case, where we analyze data for $(\beta, r = 0)$-anomalies with the guarantee of sensitive privacy. To do this, we develop a sensitively private mechanism, named n-step lookahead mechanism, which works well for this case. However, when r is non-trivial (defined in Section 3.6), we show that n-step lookahead mechanism cannot achieve sensitive privacy.

We will then introduce a construction to give SP mechanism for an arbitrary anomaly identification query (AIQ). We will also show how this construction can be used to give optimal mechanisms for AIQ. Our construction has an added benefit: since sensitive privacy generalizes differential privacy, our constructions can also be used to give a DP mechanism. Lastly, we will present a compiler construction that can compile a "bad" DP mechanism for AIQ to a "good" SP mechanism — here good and bad are indicative of utility.

4.1 n-Step Lookahead Mechanisms

Here, we give a sensitively private mechanism, named *n-step lookahead mechanism*, to compute a given query, f, for outlier analysis. For a fixed normality property p and $n \in \mathbb{N}$, n-step lookahead mechanism, $M_{p,n}$ adds noise to each coordinate i of the given database x if i is n-sensitive with respect to x, that is, if in n-steps (i.e. adding or removing n records) from x gives a database y such that $p(i, y) = 1$.

n-Step lookahead mechanism works well for $(\beta, r = 0)$-anomalies. However when r is non-trivial (defined in Section 3.6), it cannot guarantee sensitive privacy unless n is

very large. Note that for large values of n all the records become sensitive with respect to all the databases, and in such a case, n-step lookahead mechanism will perturb all the coordinates of the given database to guarantee privacy. Hence, this mechanism will have very low utility. Thus, we cannot gain in utility by using sensitive privacy instead of differential privacy.

We now clarify the notion and definitions we need to give n-step lookahead mechanism. Let $f : \mathcal{D} \to \mathcal{R}$ be a query function for analyzing data for anomalies, e.g. an anomaly identification function — we assume f is computable. We use $\texttt{Lap}(1/\varepsilon)$ to denote an independent sample from Laplace distribution of mean 0 and scale $1/\varepsilon$ — for these parameters, the probability density function for Laplace distribution is $\varepsilon e^{-\varepsilon|a|}/2$ for $a \in \mathbb{R}$. Also, recall that the function ξ replaces the negative value of every coordinate with zero. For given n and p (normality property) the mechanism $M_{p,n}$ also uses `isSensitive` function, which for given record i and database x returns $True$ if there is a y such that $||x - y||_1 \leq n$ and $p(i, y) = 1$, and $False$ otherwise.

n-Step lookahead mechanism $M_{p,n}$:

1. Input: database x
2. For each $i \in \mathcal{X}$:
3. If $\texttt{isSensitive}(i, x, p, n) = True$:
4. set $x = x + \texttt{Lap}(1/\varepsilon)\ \mathbf{e}^i$
5. Return $f(\xi(x))$.

For $(\beta, 0)$-normality property, denoted as p_0, the n-step lookahead mechanism $M_{p,n}$ is (ε, k)-SP if $n = k + 1$ (Claim 4.1). For this, `isSensitive` is simple and easily computable. For given i, x, p_0, and n, $\texttt{isSensitive}(i, x, p_0, n) = (x_i \geq \beta + 1 - n)$.

$M_{p_0,n}$ works by first perturbing the input database and then computing the query f over the perturbed database. Hence, for the practical settings (i.e. $\beta \geq k+1$), $M_{p_0,n}$ only has an additional linear (in size of the input database) computational overhead. Note that for any given database x, we only need to perturb the coordinates that have non-zero mass (i.e. $i \in \mathcal{X}$ such that $x_i \geq 1$) as others will not be k-sensitive.

Claim 4.1. *Arbitrarily fix $\varepsilon > 0$, $k \geq 1$, and $\beta \geq 1$. Let f be the query for outlier analysis, p_0 be the $(\beta, 0)$-normality property, and let* `isSensitive`$(i, x, p_0, n) = (x_i \geq \beta + 1 - n)$ *for $i \in \mathcal{X}, x \in \mathcal{D}$, and $n \in \mathbb{N}$. Then, $(k+1)$-step lookahead mechanism (i.e. $M_{p_0, n=k+1}$ as given above) is (ε, k)-sensitively private.*

Proof. Arbitrary fix $\mathcal{X}$ (a finite set), ε, k, β, and f as per the description in the claim. Let p be the $(\beta, 0)$-normality property and $n = k + 1$. Let `isSensitive`$(i, x, p_0, n) = (x_i \geq \beta + 1 - n)$ for $i \in \mathcal{X}$ and $x \in \mathcal{D}$, and let $M_{p_0,n}$ be the n-step lookahead mechanism.

For the proof, we note that the mechanism $(M_{p_0,n})$ perturbs the coordinate using Laplace distribution of mean zero and scale $1/\varepsilon$. Thus, the perturbed database it generates will be guaranteed to be sensitively private if, for every two neighbors x and y, (1) the mechanism perturbs the same set of coordinates $J \subseteq \mathcal{X}$ in x and in y, and (2) J is a super set of all coordinates that are either sensitive in one neighbor or the other.

We first prove (1). Arbitrarily fix an $i \in \mathcal{X}$ and two neighboring databases x and y in G_S. Since x and y are neighbors, $||x - y||_1 = |x_i - y_i| = 1$, and they have the same value in each coordinate except for the coordinate i. Thus, $M_{p_0,n}$ will perturb the same coordinates in both the databases except for i.

So, we next show that $M_{p_0,n}$ perturbs the coordinate i for both the database.

When i is k-sensitive w.r.t. (with respect to) to both the databases, the mechanism perturbs the i coordinate of both x and y. Note that when i is k-sensitive w.r.t. to z, $B_z(i, r = 0) = z_i \geq \beta + 1 - k \geq \beta + 1 - n$ (follows from Lemma 3.1 and $n = k + 1$). Thus, without loss of generality, let i be k-sensitive w.r.t x, but not w.r.t y.

Since i is k-sensitive w.r.t x, $x_i \geq \beta + 1 - k$ and $y_i \geq \beta - k$ (as $|x_i - y_i| = 1$). Thus, $M_{p_0,n}$ will perturb the coordinate i in both x and y.

Lastly, note that, for every database z and every coordinate j, $M_{p_0,n}$ perturbs the coordinate j if $z_j \geq \beta - k$. Hence, the set of coordinate that $M_{p_0,n}$ perturbs for both x and y is a super set of the coordinates that are sensitive either with respect to x or y. Since, the x, y, i were chosen arbitrarily, the claim holds for all the neighbors in G_S.

Now, given that the perturbed database is guaranteed to be sensitively private, from

post-processing property, we conclude the claim holds. This completes the proof.

□

4.1.1 Impossibility result

One may think that n-step lookahead mechanism will work for (β, r)-normality properties in general. But this is not the case, especially, when the normality property is regular. We prove this *impossibility result*, that is, for regular (β, r)-normality properties, it is impossible for n-step lookahead mechanism to achieve (ε, k)-SP when $n, k \leq \beta$ (Theorem 4.1).

In Theorem 4.1, we only consider $n, k \leq \beta$. Since for every $k > \beta$, every $i \in \mathcal{X}$ is k-sensitive with respect to every database $x \in \mathcal{D}$. Thus, in such a case, the k-sensitive neighborhood graph (G_S) is the same as DP neighborhood graph (i.e. $G_S = \mathbb{G}$). Now, using sensitive privacy instead of differential privacy will not result in any gain in utility.

Theorem 4.1. *Arbitrarily fix $\varepsilon > 0$, (β, r)-normality property, p, with a non-trivial r, and let g be the (β, r)-anomaly identification function. If there exist $i, j \in \mathcal{X}$ such that $d(i, j) > 2r$ then for every k and n such that $1 \leq k, n \leq \beta$, n-step lookahead mechanism for (β, r)-AIQ, (j, g), is* not *(ε, k)-sensitively private.*

Proof. Arbitrarily fix $\mathcal{X}$ and d (distance function over $\mathcal{X}$) such that there exist $i, j \in \mathcal{X}$ such that $d(i, j) > 2r$. Next arbitrarily fix $\varepsilon > 0$, $\beta \geq 1$, and a (β, r)-normality property, p with a non-trivial r. Fix an arbitrary value of k such that $1 \leq k \leq \beta$, and let G_S be the k-sensitive neighborhood graph for p.

To prove the claim, we consider two databases that are connected in G_S. Note that there exist two database x and y in $\mathcal{D}$ such that $x_i = x_j = \beta$, $||x||_1 = 2\beta$, $y_i = \beta$, and $||y||_1 = \beta$. Thus, i and j both are k-sensitive with respect to (w.r.t.) x, and i is k-sensitive w.r.t. y, but j is not (Lemma 3.1). Now, since r is non-trivial, p is regular, and $x, y \in D^p$ are connected (recall from Definition 3.4 that D^p is the set of all databases, each of which has at least one $l \in \mathcal{X}$ is k-sensitive).

Now, let (j, g) be the (β, r)-AIQ, and let $f = g_j$. Note that $f(x) = 1$ and $f(y) = 0$. Next, arbitrarily fix n such that $1 \leq n \leq \beta$, and let $M_{p,n}$ be the n-step lookahead

mechanism for the query f.

Now, note that j is n-sensitive w.r.t. to x. But j is not n-sensitive w.r.t. y because $B_y(j,r) + n \leq \beta$ (Lemma 3.1). Therefore, $M_{p,n}$ will perturb x_j but not y_j. Thus, $\mathrm{P}\big(M_{p,n}(x) \in \{1\}\big) > 0$ but $\mathrm{P}\big(M_{p,n}(y) \in \{1\}\big) = 0$. Hence, from the above and Lemma 4.1, we conclude that $M_{p,n}$ is not (ε, k)-SP. Since, n and k were fixed arbitrarily the claim holds.

□

Lemma 4.1. *Arbitrarily fix $\varepsilon > 0$, $k \geq 1$, and $p \in \mathfrak{P}$, and let G_S be the k-sensitive neighborhood graph for p. Now, for any mechanism M, if M is (ε, k)-SP then for every $x, y \in \mathcal{D}$ that are connected in G_S and every $R \subseteq Range(M)$,*

$$P\big(M(x) \in R\big) = 0 \iff P\big(M(y) \in R\big) = 0$$

Proof. Arbitrarily fix $\varepsilon > 0$, $k \geq 1$, and $p \in \mathfrak{P}$, and let G_S be the k-sensitive neighborhood graph for p. Let $M : \mathcal{D} \to \mathcal{R}$ be an arbitrarily picked mechanism that is (ε, k)-SP.

We prove the claim by contradiction. Arbitrarily fix $x, y \in \mathcal{D}$ and $R \subseteq \mathcal{R}$ such that for some $\ell > 0$, $d_{G_S}(x, y) = \ell$, and $\mathrm{P}\big(M(x) \in R\big) = 0$ and $\mathrm{P}\big(M(y) \in R\big) > 0$.

Because, M is (ε, k)-SP, it must hold that $\mathrm{P}\big(M(y) \in R\big) \leq e^{\varepsilon \ell} \mathrm{P}\big(M(x) \in R\big)$. But no value of $\ell > 0$ and $\varepsilon > 0$ satisfies the above constraint. This, implies that M is not sensitively private, which contradicts our assumption. Thus, the claim holds.

□

4.2 Private Mechanism Construction for AIQ

We now present the constructions to develop sensitively private mechanism for anomaly identification query (AIQ). In Chapter 6, we will instantiate our construction for (β, r)-anomaly [55] (a widely prevalent model), and evaluate its performance on a range of real-world datasets. Although we instantiate the constructions for (β, r)-anomaly, it is not tied to any specific anomaly definition (or model), and hence, generally applicable for other anomaly (outlier) definition.

We define the notion of *minimum discrepant distance* (mdd) over a sensitive neighborhood graph (G_S), which plays the central role in our construction. Roughly speaking, for a given G_S and a function f, mdd of a database x is the minimum distance in G_S at which there is a database y such that $f(y) \neq f(x)$. We use minimum discrepant distance to give Construction 4.1, which we use to develop privacy-preserving mechanisms for arbitrary AIQs — these mechanisms are highly accurate in practice (Theorem 6.2). We will also show how to use the same construction to give optimal SP mechanisms or even differentially private mechanisms.

Mdd encodes the privacy-utility tradeoff of the given problem. Let us see how. But, firstly, recall that our privacy mechanism, $M : \mathcal{D} \to \{0, 1\}$, for a fixed AIQ, (i, g), will output the label of i for the given database, where g is an anomaly identification function, i is a record, and the label 1 means the i is an anomalous record in x.

Fix a normality property p, and let G_S be the k-sensitive neighborhood graph corresponding to it. For a database x, let y be the closest (connected) database such that $g_i(y) \neq g_i(x)$. For these x and y, any (ε, k)-sensitively private mechanism M must satisfy: $e^{-\varepsilon\ d_{G_S}(x,y)} \leq \mathrm{P}(M(x) = g_i(x))/\mathrm{P}(M(y) = g_i(x)) \leq e^{\varepsilon\ d_{G_S}(x,y)}$ (d_{G_S} is well-defined for x and y, connected in G_S). Thus, for x, the greater the distance to the closest y such that $g_i(y) \neq g_i(x)$, the higher accuracy an SP mechanism can achieve on the input x for answering $g_i(x)$ and vice versa. We capture this metric-based property by the *minimum discrepant distance* (mdd) function, defined below. Note that it is possible that y (as given above) does not exist or is not connected to x. Thus, we must tackle these special cases in the definition of mdd-function.

Definition 4.1 (mdd-function)**.** *Arbitrarily fix a normality property p and an anomaly identification function g, and let G_S be the k-sensitive neighborhood graph for p. Then, for any function $\Delta_{G_S} : \mathcal{X} \times \mathcal{D} \to \mathbb{N} \cup \{\perp\}$, we say Δ_{G_S} is the* minimum discrepant distance function *for g, if for every $i \in \mathcal{X}$ and $x \in \mathcal{D}$, the following holds:*
If *there is a database $y \in \mathcal{D}$ such that y is connected to x and $g_i(y) \neq g_i(x)$, then*

$$\Delta_{G_S}(i, x) = \min_{z \in \mathcal{D} : g_i(z) \neq g_i(x)} d_{G_S}(x, z)$$

otherwise, $\Delta_{G_S}(i, x) = \perp$.

For a sensitive neighborhood graph, G_S, (which is a neighborhood graph) and an anomaly identification function g (both corresponding to the same anomaly definition), a simple and efficient mechanism for anomaly identification — which is both accurate and sensitively private — can be given if g and the corresponding mdd-function, Δ_{G_S}, can be computed efficiently.

However, for an arbitrary normality property, computing the mdd-function efficiently is a non-trivial task. This is because the metric, d_{G_S}, which gives rise to the metric-based property captured by the mdd-function, is induced by (a) the definition of anomaly (e.g. specific values of β and r) and (b) the metric over the records. Thus, making it exceedingly difficult to analyze it in general.

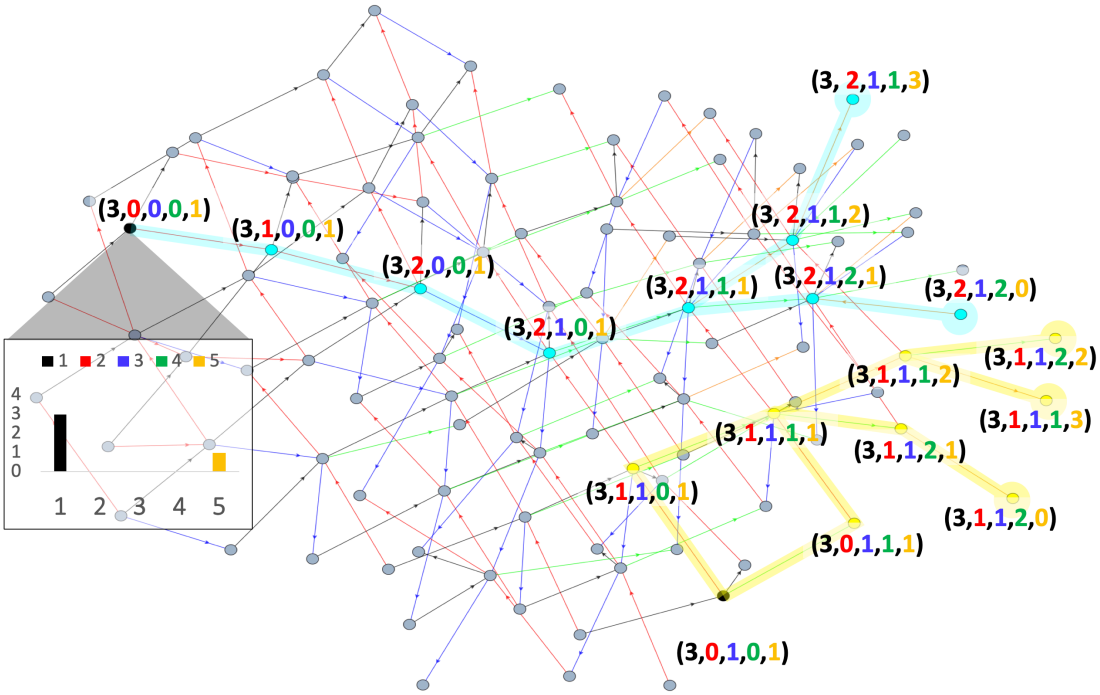

Figure 4.1: **Sensitive neighborhood graph.** A simple example of a 1-sensitive neighborhood graph, G_S, for $(\beta = 3, r = 1)$-normality property with $\mathcal{X} = \{1, 2, 3, 4, 5\}$ and ℓ_1-metric over $\mathcal{X} \times \mathcal{X}$. Note that G_S is an undirected graph; arrowheads indicates the record is added at the end node; the color of the edge corresponds (as per the given color code) to the value of the record added. Further, each database x is represented as a 5-tuple with x_i representing the number of records in x that have value i.

We use the example given in Figure 4.1 to explain the above mentioned relationships of mdd-function. This figure depicts a subgraph of 1-sensitive neighborhood graph for $(\beta = 3, r = 1)$-anomaly with ℓ_1-metric.

One can appreciate the conceptual difficulty in calculating mdd-function, Δ_{G_S} (for this setting) by for example thinking the value of $\Delta_{G_S}(5,(3,0,0,0,1))$ — and recall that this is just a 1-sensitive neighborhood graph. Next, note that for a given database x and a record i, the shorter is the distance of the closest sensitive record from i, the smaller the value of $\Delta_{G_S}(i,x)$, e.g. $\Delta_{G_S}(5,(3,0,0,0,1)) > \Delta_{G_S}(5,(3,2,1,0,1))$.

Lastly, observe that the presence of non-sensitive records can also influence the value of the mdd-function, e.g. $\Delta_{G_S}(5,(3,0,0,0,1)) > \Delta_{G_S}(5,(3,0,1,0,1))$ although the closest sensitive record to 5 is the same in both the databases. In addition, the values of β and r also affect the value of mdd-function, and in most realistic settings, the size of $\mathcal{X}$ is large, and the sensitive neighborhood graph is quite complex.

In the next section, we present our constructions that uses a lower bound on the mdd-function to give sensitively private mechanism.

4.2.1 Construction 4.1: SP-mechanism via lower bounding mdd

Here, we show how to construct an SP mechanism to identify anomalies by using a lower bound, λ, for the minimum discrepant distance (mdd). Our construction (Construction 4.1) is parameterized by λ, which is associated with a sensitive neighborhood graph. Since the sensitive neighborhood graph is tied to an anomaly definition, it will become concrete once we give the definition of anomaly (e.g. see Sections 6.2 and 6.1).

For any fixed AIQ, (i,g), and given λ, Construction 4.1 provably gives an SP mechanism as long as λ is an *acceptable lower bound* on the mdd-function (Theorem 4.2).

Below, we define the notion of acceptable lower bound on an mdd-function. Arbitrarily fix a neighborhood graph G and an anomaly identification function g (for the definitions given below). We say $\lambda : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0} \cup \{\perp\}$ is a ***lower bound*** on the mdd-function Δ_G for g if for every $i \in \mathcal{X}$ and $x \in \mathcal{D}$, the following holds: if $\Delta_G(i,x) \in \mathbb{N}$, then $\lambda(i,x) \in \mathbb{R}_{\geq 0}$ (otherwise $\lambda(i,x) = \perp$ or $\lambda(i,x) \in \mathbb{R}_{\geq 0}$). For any given $\ell, \alpha \geq 0$, we say λ is (ℓ,α)-***acceptable*** if the following holds: (1) for every i and x, if $\lambda(i,x) \in \mathbb{R}_{\geq 0}$, then $\lambda(i,x) \geq \ell$, and (2) λ is α-Lipschitz continuous over G (defined in Section 2.3).

Definition 4.2 (acceptable lower bound)**.** *Arbitrarily fix a neighborhood graph G and*

anomaly identification function g, and let Δ_G is the mdd-function for g. For any given $\ell, \alpha \geq 0$, we say $\lambda : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0} \cup \{\perp\}$ is an (ℓ, α)-acceptable lower bound *on Δ_G, if it is (ℓ, α)-acceptable and a lower bound on Δ_G.*

We remark that although at first it appears that the *Lipschitz continuity condition* is some side technicality, in fact bounding its value constitute the main part of our argument for privacy of our mechanisms. Thus giving an SP mechanism for (i, g) via Construction 4.1 reduces to giving a Lipschitz continuous lower bound for the mdd-function corresponding to g.

Construction 4.1. U_λ

1. *Input $x \in \mathcal{D}$.*
2. *If $\lambda(i, x) = \perp$, set $t = 0$*
3. *Else, set $t = e^{-\varepsilon(\lambda(i,x)-1)}/(1 + e^{\varepsilon})$*
4. *Sample b from $\{0, 1\}$ such that $\mathrm{P}(b \neq g(i, x)) = t$.*
5. *Return b.*

Note that the above is a family of constructions parameterized by λ, i.e. one construction, U_λ, for each λ. This construction is very efficiently realizable as long as we can efficiently compute g and λ. Furthermore, the error of the mechanism, yielded by the construction, for any input is exponentially small in λ (Theorem 4.2).

Theorem 4.2 (U_λ is SP)**.** *Arbitrarily fix $\varepsilon > 0$, $k, \alpha \geq 1$, and a normality property p. Let G_S be the k-sensitive neighborhood graph for p, and (i, g) be an arbitrary AIQ, where g and p are for the same anomaly definition.*

If λ is $(1, \alpha)$-acceptable lower bound on the mdd-function, Δ_{G_S}, for g, then Construction 4.1 yields an $(\varepsilon\alpha, k)$-SP mechanism, U_λ, such that for every $x \in \mathcal{D}$ and $\lambda(i, x) \in \mathbb{R}_{\geq 0}$,

$$P(U_\lambda(x) \neq g(i, x)) = e^{-\varepsilon(\lambda(i,x)-1)}/(1 + e^{\varepsilon})$$

In order to show that the theorem holds, it suffices to verify that for an arbitrary i and two arbitrary neighboring x and y in G_S, the privacy constraints hold. For any AIQ, (i, g), it is immediate when $g(i, x) = g(i, y)$ because λ is α-Lipschitz continuous.

When $g(i,x) \neq g(i,y)$, $\lambda(i,x) = \lambda(i,y) = 1$ because $\Delta_{G_S}(i,x) = \Delta_{G_S}(i,y) = 1$ and $\lambda \geq 1$. Thus, the constraints are satisfied in this case as well. The complete proof is given below.

Proof of Theorem 4.2. Arbitrarily fix $\varepsilon > 0$, $k \geq 1$, $\alpha \geq 1$, and a normality property p. Let G_S be the k-sensitive neighborhood graph for p, and g be the anomaly identification function for the anomaly definition for p.

Fix λ to be an $(1,\alpha)$-acceptable lower bound on the mdd-function, Δ_{G_S}, for g. Arbitrarily fix an anomaly identification query, (i,g), and let U_λ be as given by Construction 4.1. Finally, choose arbitrary $x, y \in \mathcal{D}$ that are neighbors in G_S (i.e. $d_{G_S}(x,y) = 1$).

Below, we show that U_λ satisfies the privacy constraint for both the outputs, i.e. 0 and 1.

If $\lambda(i,x) = \perp$ then $\lambda(i,y) = \perp$ (and vice versa) because λ is Lipschitz continuous (follows from λ being $(1,\alpha)$-acceptable). Since λ is a lower bound on Δ_{G_S}, $\lambda(i,x) = \perp$ implies that either (a) there is no z in G_S such that $g(i,z) \neq g(i,x)$, or (b) every z' that is connected with x is such that $g(i,z') = g(i,x)$. In both the scenario, we get that $g(i,x) = g(i,y)$ as there is no z connected to x, and hence to y, such that $g(i,z) \neq g(i,x)$. And in this case, the privacy constraint holds trivially. So below we consider the case, where $\lambda(i,x), \lambda(i,y) \in \mathbb{R}_{\geq 0}$.

Firstly, consider the case, when $g(i,x) = g(i,y) = b$ for some $b \in \{0,1\}$. Here, from the α-Lipschitz continuity the following holds.

$$\frac{\mathrm{P}\left(U_\lambda(x) = 1-b\right)}{\mathrm{P}\left(U_\lambda(y) = 1-b\right)} = e^{\varepsilon(-\lambda(i,x)+\lambda(i,y))} \leq e^{\alpha\varepsilon}$$

Now for the other constraint, we have the following:

$$\begin{aligned}\frac{\mathrm{P}\left(U_\lambda(x) = b\right)}{\mathrm{P}\left(U_\lambda(y) = b\right)} &= \frac{1-\mathrm{P}\left(U_\lambda(x) = 1-b\right)}{1-\mathrm{P}\left(U_\lambda(y) = 1-b\right)} \\ &= \frac{1+e^{\varepsilon}-e^{-\varepsilon(\lambda(i,x)-1)}}{1+e^{\varepsilon}-e^{-\varepsilon(\lambda(i,y)-1)}} \end{aligned} \tag{4.1}$$

Next, we show that $e^{\varepsilon\alpha}$ is indeed an upper bound for the expression given in (4.1).

Since $\varepsilon > 0$ and $\lambda(i, y)$ and α are at least 1, we get the following:

$$
\begin{aligned}
& e^{\varepsilon}(e^{\varepsilon\alpha} + 1) \leq e^{\varepsilon(\alpha+\lambda(i,y))}(1 + e^{\varepsilon}) \\
\iff & e^{\varepsilon}(e^{2\varepsilon\alpha} - 1) \leq e^{\varepsilon(\alpha+\lambda(i,y))}(1 + e^{\varepsilon})(e^{\varepsilon\alpha} - 1) \\
\iff & e^{\varepsilon}(e^{\varepsilon\alpha} - e^{-\varepsilon\alpha}) \leq e^{\varepsilon\lambda(i,y)}(1 + e^{\varepsilon})(e^{\varepsilon\alpha} - 1) \\
\iff & e^{\varepsilon\lambda(i,y)}(1 + e^{\varepsilon}) - e^{\varepsilon}e^{-\varepsilon\alpha} \leq e^{\varepsilon\alpha}\left[e^{\varepsilon\lambda(i,y)}(1 + e^{\varepsilon}) - e^{\varepsilon}\right] \\
\iff & \frac{e^{\varepsilon\lambda(i,y)}(1 + e^{\varepsilon}) - e^{\varepsilon}e^{-\varepsilon\alpha}}{e^{\varepsilon\lambda(i,y)}(1 + e^{\varepsilon}) - e^{\varepsilon}} \leq e^{\varepsilon\alpha} \\
& \text{since } -\alpha \leq \lambda(i, y) - \lambda(i, x), \text{ the following holds} \\
& \frac{e^{\varepsilon\lambda(i,y)}(1 + e^{\varepsilon}) - e^{\varepsilon}e^{\varepsilon(\lambda(i,y)-\lambda(i,x))}}{e^{\varepsilon\lambda(i,y)}(1 + e^{\varepsilon}) - e^{\varepsilon}} \leq e^{\varepsilon\alpha} \\
\iff & \frac{1 + e^{\varepsilon} - e^{-\varepsilon(\lambda(i,x)-1)}}{1 + e^{\varepsilon} - e^{-\varepsilon(\lambda(i,y)-1)}} \leq e^{\varepsilon\alpha}
\end{aligned}
$$

Now consider the case of $g(i, x) \neq g(i, y)$. Here, $\lambda(i, x) = \lambda(i, y) = 1$. To confirm this, note that $d_{G_S}(x, y) = 1$, and $\Delta_{G_S}(j, z) \geq \lambda(j, z) \geq 1$ for every $j \in \mathcal{X}$ and $z \in \mathcal{D}$ (because λ is $(1, \alpha)$-acceptable lower bound on Δ_{G_S}). Thus, $\Delta_{G_S}(i, x) = \Delta_{G_S}(i, y) = 1$ implies $\lambda(i, x) = \lambda(i, y) = 1$. Therefore, the privacy constraints hold trivially for this case. Since, x and y were picked arbitrarily, the above shows that all the privacy constraints hold for all the neighbors. This concludes the formal argument. □

4.2.2 Optimal SP mechanism via Construction 4.1

For $\lambda = \Delta_{G_S}$, Construction 4.1 gives a pareto optimal sensitively private mechanism (Theorem 4.3).

For arbitrarily fixed $\varepsilon > 0$, $k \geq 1$, and a normality property p, any mechanism U is called ***pareto optimal*** (ε, k)-sensitively private if (1) it is (ε, k)-SP and (2) for every (ε, k)-SP mechanism $M : \mathcal{D} \to \{0, 1\}$ and every database $x \in \mathcal{D}$, $P(U(x) = g_i(x)) \geq P(M(x) = g_i(x))$. Particularly, this implies that of all the SP mechanisms yielded by Construction 4.1, each corresponding to a different λ, *the "best" mechanism is for* $\lambda = \Delta_{G_S}$.

Theorem 4.3. *Arbitrarily fix $\varepsilon > 0$, $k \geq 1$, and a normality property p. Let G_S be the k-sensitive neighborhood graph for p, and (i, g) be an arbitrary AIQ, where g and p are for the same anomaly definition.*

If Δ_{G_S} is the mdd-function for g, then $U_{\Delta_{G_S}}$ (Construction 4.1) is pareto optimal (ε, k)-sensitively private.

Proof. Let ε, k, p, g, and G_S be as given above. Arbitrarily fix $i \in \mathcal{X}$, and let Δ_{G_S} be the mdd-function for g, and $U_{\Delta_{G_S}}$ be as given by Construction 4.1.

Firstly, note that $U_{\lambda=\Delta_{G_S}}$ is (ε, k)-SP. This follows from Theorem 4.2 and the fact that, for $\lambda = \Delta_{G_S}$, λ is $(1, 1)$-acceptable lower bound on Δ_{G_S} (Lemma 4.2).

Next, we prove the optimality claim by contradiction. Assume that $U_{\Delta_{G_S}}$ is not pareto optimal. That is, there exits an (ε, k)-SP mechanism M (for p) such that

- for every x, $P\big(M(x) = g_i(x)\big) \geq P\big(U_{\Delta_{G_S}}(x) = g_i(x)\big)$ and
- for a database y, $P\big(M(y) = g_i(y)\big) > P\big(U_{\Delta_{G_S}}(y) = g_i(y)\big)$

Let us fix the y given above. Using y and our assumption about it, we show that there is an input database z, where M does worse than $U_{\Delta_{G_S}}$.

Let z be a database such that $d_{G_S}(y, z) = \Delta_{G_S}(i, y)$ and $g_i(z) \neq g_i(y)$. Note that if there is no such z, then $\Delta_{G_S}(i, y) = \perp$. In this case, our assumption about the database y, and hence, M cannot hold because, in this case, $P\big(U_{\Delta_{G_S}}(y) = g_i(y)\big) = 1$. Furthermore, similar would be the case if z is not connected to y (i.e. $\Delta_{G_S}(i, y) = \perp$). Thus, if the assumptions about M holds, then $d_{G_S}(y, z) \in \mathbb{N}$ (i.e. y and z are connected).

If we let w be a neighbor of z on the shortest path from y to z such that $g_i(w) \neq g_i(z)$, then $g_i(w) = g_i(y)$ and $d_{G_S}(y, w) = \Delta_{G_S}(i, y) - 1$. Since M (ε, k)-SP, for $b = g_i(w)$, it follows that

$$\begin{aligned} P\big(M(w) \neq b\big) \leq & e^{\varepsilon d_{G_S}(y,w)} P\big(M(y) \neq b\big) \\ = & e^{\varepsilon(\Delta_{G_S}(i,y)-1)} P\big(M(y) \neq b\big) \\ < & e^{\varepsilon(\Delta_{G_S}(i,y)-1)} P\big(U_{\Delta_{G_S}}(y) \neq b\big) \\ = & 1/(1 + e^{\varepsilon}) \end{aligned} \tag{4.2}$$

The first inequality is due to the SP constrains on M. The second inequality is due to the fact that M is strictly better than $U_{\Delta_{G_S}}$ on y. The last equality holds because $P\big(U_{\Delta_{G_S}}(y) \neq g_i(y)\big) = e^{-\varepsilon(\Delta_{G_S}(i,y)-1)}/(1+e^{\varepsilon})$ (follows from Construction 4.1).

Since M is assumed to (ε, k)-SP, we get the following:

$$\begin{aligned} P\big(M(z) \neq 1-b\big) \geq & e^{-\varepsilon} P\big(M(w) \neq 1-b\big) \\ = & e^{-\varepsilon} P\big(M(w) = b\big) \\ = & e^{-\varepsilon}\big(1 - P\big(M(w) \neq b\big)\big) \\ > & 1/(1+e^{\varepsilon}) \end{aligned} \tag{4.3}$$

The first inequality is due to M being SP. The first equality is due to the fact that there are only two possible outputs. The last inequality holds because $P\big(M(w) \neq b\big) < 1/(1+e^{\varepsilon})$ (which follows from the inequality given by (4.2)).

Finally, for $b = g_i(w)$, it follows that

$$\begin{aligned} P\big(M(z) = g_i(z)\big) = & P\big(M(z) = 1-b\big) \\ = & 1 - P\big(M(z) \neq 1-b\big) \\ < & e^{\varepsilon}/(1+e^{\varepsilon}) \\ = & P\big(U_{\Delta_{G_S}}(z) = g_i(z)\big) \end{aligned}$$

The first equality is due to the fact that $b = g_i(w) \neq g_i(z)$. The first inequality is due to the inequality given by (4.3). The last equality holds due to the following facts: (1) $P\big(U_{\Delta_{G_S}}(z) \neq g_i(z)\big) = e^{-\varepsilon(\Delta_{G_S}(i,z)-1)}/(1+e^{\varepsilon})$, and (2) $\Delta_{G_S}(i,z) = 1$ because $d_{G_S}(z,w) = 1$ and $g_i(z) \neq g_i(w)$.

From the above, we reach a conclusion that contradicts our assumption that M is strictly "better" than $U_{\Delta_{G_S}}$. Thus, we conclude the $U_{\Delta_{G_S}}$ is pareto optimal. □

Lemma 4.2. *Arbitrarily fix a neighborhood graph G and an anomaly identification function g, and let Δ_G be the mdd-function for g. Then for $\lambda = \Delta_G$, λ is $(1,1)$-acceptable lower bound on Δ_G.*

Proof. Let G, g, Δ_G, and λ be as given above. We show that λ is $(1,1)$-acceptable and is a lower bound on Δ_G.

To show that λ is $(1,1)$-acceptable, we first prove that Δ_G is 1-Lipschitz continuous. For this, arbitrarily fix an $i \in \mathcal{X}$ and two neighbors x and y in G, i.e. $d_G(x,y) = 1$.

Firstly, consider the case when $\Delta_G(i,x) = \perp$ (this is without loss of generality as x and y were picked arbitrarily). $\Delta_G(i,x) = \perp$ implies that either (a) there is no z in G such that $g(i,z) \neq g(i,x)$, or (b) every z' that is connected with x is such that $g(i,z') = g(i,x)$. In both the scenario, we get that $g(i,x) = g(i,y)$ as there is no z connected to x, and hence to y, such that $g(i,z) \neq g(i,x)$; thus, $\Delta_G(i,y) = \perp$. Hence, for this case, the Lipschitz continuity constraints hold.

Next, we consider the case where $\Delta_G(i,x), \Delta_G(i,y) \in \mathbb{N}$ (i.e. there is a z connected to x, and hence to y, such that $g(i,z) \neq g(i,x)$). Firstly note that by triangular inequality we get that for every database $z \in \mathcal{D}$, $d_G(x,z) \leq d_G(x,y) + d_G(y,z) = 1 + d_G(y,z)$. Thus, $\Delta_G(i,x) = \min_{z \in \mathcal{D}: g_i(z) \neq g_i(x)} d_G(x,z) \leq 1 + \Delta_G(i,y)$. Since x and y were chosen arbitrarily, swapping x and y gives $\Delta_G(i,y) \leq 1 + \Delta_G(i,x)$. Thus, 1-Lipschitz continuity constraints hold for x and y. Since we arbitrarily picked, i, and neighbors x, and y, the claim holds for all the neighbors and every $i \in \mathcal{X}$. Hence, Δ_G is 1-Lipschitz continuous.

Next, we show that if, for any i and x, $\Delta_G(i,x) \in \mathbb{N}$, then $\Delta_G(i,x) \geq 1$. For this, arbitrarily fix $i \in \mathcal{X}$ and x such that $\Delta_G(i,x) \in \mathbb{N}$. This implies that there is a database z at distance $d_G(x,y)$ from x such that $g(i,z) \neq g(i,x)$. For this to hold, $z \neq x$, and hence, $d_G(z,x) \geq 1$. Thus, by definition of mdd-function, $\Delta_G(i,x) \geq 1$. Since i and x were picked arbitrarily, the claim holds. Thus, the above shows that Δ_G is $(1,1)$-acceptable.

Since for every i and x, $\lambda(i,x) = \Delta_G(i,x)$, λ is indeed a lower bound on Δ_G. This completes the proof. □

4.2.3 DP mechanism via Construction 4.1

For Construction 4.1, if we use a λ that is an $(1,\alpha)$-acceptable lower bound on the mdd-function of the DP neighborhood graph, $\Delta_{\mathbb{G}}$, then the construction yields a differentially private mechanism for AIQ (Corollary 4.1).

Corollary 4.1. *Arbitrarily fix $\varepsilon > 0$, $\alpha \geq 1$, and an AIQ (i, g). For every λ such that it is $(1, \alpha)$-acceptable lower bound on $\Delta_{\mathbb{G}}$ for g, U_λ (given by Construction 4.1) is an $\varepsilon\alpha$-differentially private mechanism.*

To confirm above claim, note that from Definition 3.3 (of sensitive privacy) and Definition 3.5, it follows that differential privacy is a special case of sensitive privacy, when the k-sensitive neighborhood graphs, G_S, is the same as neighborhood graph, $\mathbb{G}$, i.e., $G_S = \mathbb{G}$ (for more details see Section 3.7). Thus, for $G_S = \mathbb{G}$, a mechanism is ε-differentially private if and only if it is ε-sensitively private. Hence, Corollary 4.1 follows from Theorem 4.2.

4.3 DP to SP Mechanism Compiler

In this section, we present a construction to compile a differentially private mechanism for an anomaly identification query into a sensitively private one. The DP mechanism, which the compiler takes, is given in terms of its distribution over the outputs for every input. The compiled SP mechanism comparatively has much better accuracy for the non-sensitive records; however, for the sensitive records, the SP and the input DP mechanism err by the same amount.

It is noteworthy that for many problems, we already know the distributions given by differentially private mechanisms [23, 22, 46]. Thus, our construction can be employed using these mechanism as long as the distributions given by the differentially private mechanism are not too "wild". For example, the probability of the wrong answer for any input is not too high (we formalize this below), which is typically true.

The compiler construction (Construction 4.2) is parameterized by δ, a $(0, 2)$-acceptable lower bound on $\Delta_{G_S} - \Delta_{\mathbb{G}}$ (defined below, Definition 4.3). Here Δ_{G_S} and $\Delta_{\mathbb{G}}$ are the mdd-functions for an arbitrarily a fixed anomaly identification function g, where G_S is the k-sensitive neighborhood graph for the normality property corresponding to the anomaly definition for g.

Definition 4.3. *Arbitrarily fix a normality property p and a non-constant AIQ, (i, g),*

both corresponding to the same anomaly definition, and let Δ_{G_S} *and* $\Delta_{\mathbb{G}}$ *be the mdd-functions for* g *such that* G_S *is the* k*-sensitive neighborhood graph for* p *and* $\mathbb{G}$ *is the DP neighborhood graph. Then for every* $x \in \mathcal{D}$*,*

$$[\Delta_{G_S} - \Delta_{\mathbb{G}}](i,x) = \begin{cases} \perp & \text{if } \Delta_{G_S}(i,x) = \perp \\ \Delta_{G_S}(i,x) - \Delta_{\mathbb{G}}(i,x) & \text{otherwise} \end{cases} \tag{4.4}$$

Note that the DP neighborhood graph is always connected. Now, since the AIQ, (i,g), is non-constant (i.e. for some $x, y \in \mathcal{D}$, $g_i(x) \neq g_i(y)$), we have that for every database $x \in \mathcal{D}$, $\Delta_{\mathbb{G}}(i,x) \in \mathbb{N}$. Therefore, $\Delta_{G_S} - \Delta_{\mathbb{G}}$ is a well-defined notion. Given below is our construction for the compiler (Construction 4.2), and it is parameterized by δ, a lower bound on $\Delta_{G_S} - \Delta_{\mathbb{G}}$. It is useful when obtaining δ is easier than λ (the lower bound on mdd we used in previous sections), and we already know the distributions of a DP mechanism for the problem.

Construction 4.2. U_δ

1. *Input* $x \in \mathcal{D}$.
2. *Set* $t = \mathrm{P}\left(M(x) \neq g(i,x)\right) / e^{\frac{\varepsilon}{4}\delta(i,x)}$.
3. *Sample* b *from* $\{0,1\}$ *such that* $\mathrm{P}\left(b \neq g(i,x)\right) = t$.
4. *Return* b.

A differentially private mechanism (in terms of its distributions) that can be transformed (with provable guarantees) through our compiler is termed as a *valid* mechanism. For $\varepsilon > 0$ and any fixed AIQ, (i,g), we say an ε-DP mechanism, $M : \mathcal{D} \to \{0,1\}$, is valid if, for every two neighbors x and y in the DP neighborhood graph such that $g(i,x) = g(i,y) = b$ for $b \in \{0,1\}$, the following holds

$$1 - \mathrm{P}\left(M(x) \neq b\right) e^{-\varepsilon} \leq e^{2\varepsilon} \left(1 - \mathrm{P}\left(M(y) \neq b\right)\right).$$

Note that any ε-differentially private mechanism, M, for a fixed AIQ, (i,g), that satisfies $\mathrm{P}\left(M(x) \neq g(i,x)\right) \leq e^{2\varepsilon}/(1+e^{2\varepsilon})$ for every x is valid – this is shown below for $\varepsilon > 0$ and two arbitrary neighbors x and y such that $b = g(i,x) = g(i,y)$; hence the

notion of valid differentially private mechanism is well defined.

$$\begin{aligned}
\mathrm{P}\left(M(y) \neq b\right) \leq & \frac{e^{2\varepsilon}}{e^{2\varepsilon}+1} \Longrightarrow \\
\mathrm{P}\left(M(y) \neq b\right) e^{4\varepsilon} - \mathrm{P}\left(M(y) \neq b\right) \leq & e^{2\varepsilon}(e^{2\varepsilon}-1) \\
\text{since } M \text{ is } \varepsilon\text{-DP, it follows from the above} & \\
\mathrm{P}\left(M(y) \neq b\right) e^{4\varepsilon} - \mathrm{P}\left(M(x) \neq b\right) e^{\varepsilon} \leq & e^{2\varepsilon}(e^{2\varepsilon}-1) \Longrightarrow \\
1 - \mathrm{P}\left(M(x) \neq b\right) e^{-\varepsilon} \leq & e^{2\varepsilon}\left(1 - \mathrm{P}\left(M(y) \neq b\right)\right)
\end{aligned}$$

We claim that for a given valid differentially private mechanism, M, for a fixed AIQ, (i, g), and a $(0, 2)$-acceptable lower bound, δ, on $\Delta_{G_S} - \Delta_{\mathbb{G}}$, Construction 4.2 compiles M into a sensitively private mechanism, U_δ (Theorem 4.4).

We stress that for the compiled SP mechanism, the probability of error can be exponentially smaller compared to the input DP mechanism. This is especially true for the non-sensitive records. This leads to an improvement in accuracy. Furthermore, as the input mechanism, M, to the compiler becomes better (i.e. has lower error) so does the compiled sensitively private mechanism, U_δ, since the error of U_δ, is never more than that of M.

We highlight the effectiveness of the compiler by instantiating it for $\delta(i, x) = \lambda_1(i, x) - \Delta_{\mathbb{G}}(i, x)$ for every i and x for (β, r)-anomaly ($\lambda_{k=1}$ and $\Delta_{\mathbb{G}}$ are given in Chapter 6). The proof of δ satisfying the constraints is given in the Chapter's end-note[2].

Figure 4.2 shows the compilation of two DP mechanisms for (β, r)-AIQ, which widely differ in their performance. As expected, the compiled SP-mechanism outperforms the input DP-mechanism.

In Figure 4.2a, the input DP mechanism, M, is trivial and has a constant error for every input database, that is, $1/(1+e^{\varepsilon})$ for fixed $\varepsilon = 0.25$. Clearly, this mechanism has extremely bad accuracy. This is a difficult case even for the compiled mechanism, which nevertheless, attains exponential gain in accuracy for non-sensitive records. However, when we input the DP-mechanism given in Section 6.1, which is much better than the

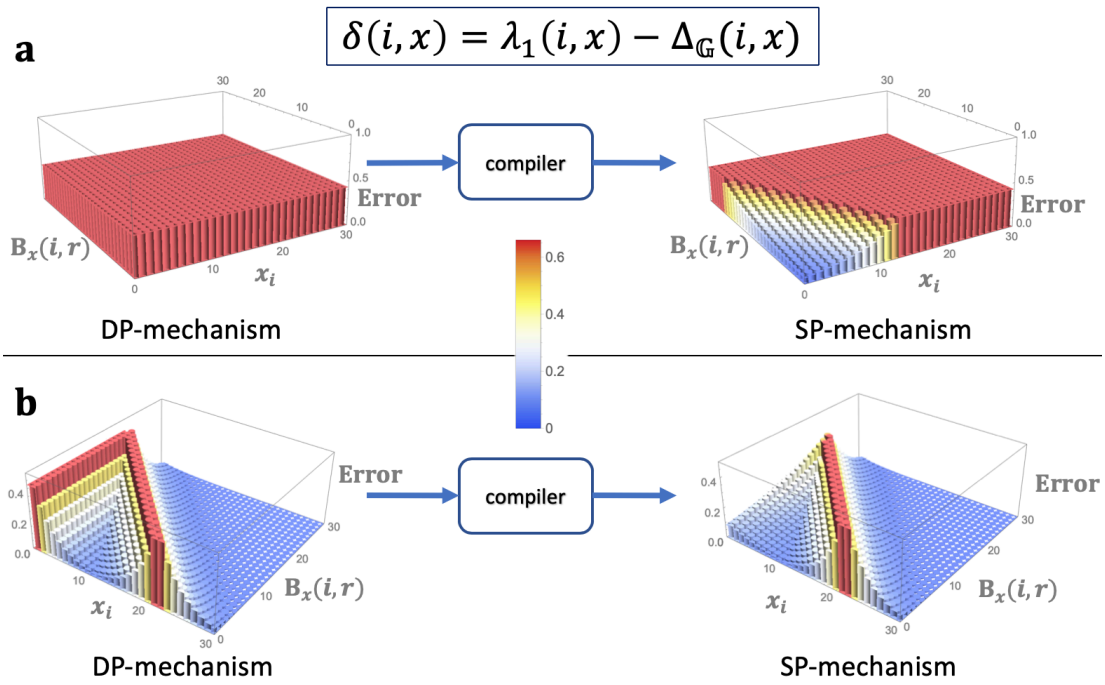

Figure 4.2: **Compilation of DP-mechanism for (β, r)-AIQ into SP-mechanism.** In both **(a)** and **(b)**, the input mechanism is 0.25-DP for a fixed record i and δ (given in the figure). Each database x is given by $(x_i, B_x(i,r))$ since (β, r)-anomaly identification function only depends upon x_i and $B_x(i,r)$. Each mechanism is depicted by its error over databases i.e. $P(M(x) \neq g(i,x))$. **(a)**, DP-mechanism has constant error ≈ 0.44. **(b)**, DP-mechanism has error $\approx 0.56/e^{0.25\Delta_{\mathbb{G}}(i,x)}$.

one in Figure 4.2a, the compiled mechanism is clearly superior compared to the one in Figure 4.2a (Figure 4.2b).

Theorem 4.4. *Arbitrarily fix $\varepsilon > 0$, $k \geq 1$ and a normality property p, and let G_S be the k-sensitive neighborhood graph for p, and (i,g) be a non-constant AIQ such that g and p are for the same anomaly definition, and $\Delta_{G_S} - \Delta_{\mathbb{G}}$ be as per Definition 4.3 for g.*

Then, for any given valid $\varepsilon/2$-DP mechanism, M, for the AIQ, (i,g), and $(0,2)$-acceptable lower bound, $\delta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0}$, on $\Delta_{G_S} - \Delta_{\mathbb{G}}$, Construction 4.2 yields an (ε, k)-SP mechanism, U_δ, such that for every $x \in \mathcal{D}$,

$$P\left(U_\delta(x) \neq g(i,x)\right) = P\left(M(x) \neq g(i,x)\right) e^{-\frac{\varepsilon}{4}\delta(i,x)}.$$

Proof. Arbitrarily fix ε, k, p, (i,g), and G_S as given above. Let M be any valid $\varepsilon/2$-differentially private mechanism for the AIQ, (i,g), and $\Delta_{G_S} - \Delta_{\mathbb{G}}$ be as given above.

Next, let $\delta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}_{\geq 0}$ be a $(0, 2)$-acceptable lower bound on $\Delta_{G_S} - \Delta_{\mathbb{G}}$. Finally, let U_δ be the mechanism that Construction 4.2 yields.

We prove the claim by showing that for arbitrarily picked x and y that are neighbors in G_S, all the privacy constraint hold.

When $\delta(i, x) = \delta(i, y) = 0$, $\mathrm{P}\left(U_\delta(z) = b\right) = \mathrm{P}\left(M(z) = b\right)$ for every database z and every b in $\{0, 1\}$. In this case, the privacy constraints are trivially satisfied because M is $\varepsilon/2$-DP. Now, assume that $\delta(i, x) > \delta(i, y) \geq 0$ — this is without loss of generality as x and y are picked arbitrarily and δ is $(0, 2)$-acceptable. We first show that in this case, $g(i, x) = g(i, y)$.

Firstly, note that if $\Delta_{G_S}(i, x) = \perp$ then $\Delta_{G_S}(i, y) = \perp$ (an vice versa), and in this case, $g(i, x) = g(i, y)$. Secondly, consider the case when $\Delta_{G_S}(i, x), \Delta_{G_S}(i, y) \in \mathbb{N}$. For this, we show that for $\delta(i, x) > \delta(i, y) \geq 0$ to hold, $\Delta_{G_S}(i, x)$ must be at least 2, which implies that $g(i, x) = g(i, y)$ as x and y are neighbors (i.e. $d_{G_S}(x, y) = 1$).

Since $\mathcal{E}(G_S) \subseteq \mathcal{E}(\mathbb{G})$, we get $\Delta_{G_S}(i, z) \geq \Delta_{\mathbb{G}}(i, z) \geq 1$ (follows from Lemma 4.2 and the definition of neighborhood graph). From here it follows that $\Delta_{G_S}(i, x) = 1$ implies $\Delta_{\mathbb{G}}(i, x) = 1$. But if $\Delta_{G_S}(i, x) = 1$, then $\Delta_{G_S}(i, y) = 1$, and hence, $\Delta_{\mathbb{G}}(i, y) = 1$ as well. Thus, $[\Delta_{G_S} - \Delta_{\mathbb{G}}](i, x) = [\Delta_{G_S} - \Delta_{\mathbb{G}}](i, y) = 0$, and $\delta(i, x) > \delta(i, y)$ cannot hold. Therefore, we conclude that $\Delta_{G_S}(i, x) \geq 2$.

Since M is valid $\varepsilon/2$-differentially private, we get the following for $b = g(i, x)$

$$
\begin{aligned}
& 1-\mathrm{P}\left(M(x) \neq b\right) e^{-\varepsilon/2} \leq e^{\varepsilon}\left(1-\mathrm{P}\left(M(y) \neq b\right)\right) \\
\Longrightarrow & 1-e^{\varepsilon} \leq \mathrm{P}\left(M(x) \neq b\right) e^{-\varepsilon/2}-\mathrm{P}\left(M(y) \neq b\right) e^{\varepsilon} \\
& \text{since } \delta \text{ is 2-Lipschitz continuous, we get} \\
& 1-e^{\varepsilon} \leq \frac{\mathrm{P}\left(M(x) \neq b\right)}{e^{\frac{\varepsilon}{4}(\delta(i,x)-\delta(i,y))}}-\mathrm{P}\left(M(y) \neq b\right) e^{\varepsilon} \\
& \text{since LHS is negative, and } \delta \geq 0, \text{ the following holds} \\
& 1-e^{\varepsilon} \leq e^{-\frac{\varepsilon}{4}\delta(i,y)}\left(\frac{\mathrm{P}\left(M(x) \neq b\right)}{e^{\frac{\varepsilon}{4}(\delta(i,x)-\delta(i,y))}}-\mathrm{P}\left(M(y) \neq b\right) e^{\varepsilon}\right) \\
\Longrightarrow & 1-\frac{\mathrm{P}\left(M(x) \neq b\right)}{e^{\frac{\varepsilon}{4}\delta(i,x)}} \leq e^{\varepsilon}\left(1-\frac{\mathrm{P}\left(M(y) \neq b\right)}{e^{\frac{\varepsilon}{4}\delta(i,y)}}\right) \\
\Longrightarrow & \mathrm{P}\left(U_{\delta}(x)=b\right) \leq e^{\varepsilon} \mathrm{P}\left(U_{\delta}(y)=b\right)
\end{aligned}
$$

In a similar fashion, by swapping x and y, one can show that the privacy constraint $\mathrm{P}\left(U_{\delta}(y)=b\right) \leq e^{\varepsilon} \mathrm{P}\left(U_{\delta}(x)=b\right)$ also holds. Below we show that the other constraints are also satisfied.

$$
\frac{\mathrm{P}\left(U_{\delta}(x) \neq b\right)}{\mathrm{P}\left(U_{\delta}(y) \neq b\right)}=\frac{\mathrm{P}\left(M(x) \neq b\right) e^{-\frac{\varepsilon}{4}\delta(i,x)}}{\mathrm{P}\left(M(y) \neq b\right) e^{-\frac{\varepsilon}{4}\delta(i,y)}} \leq e^{\varepsilon}
$$

The above inequality holds because M is $\varepsilon/2$-DP and δ is 2-Lipschitz continuous.

Since all the privacy constraints hold for arbitrarily picked neighbors and δ (which satisfies the conditions specified in the claim), and a valid $\varepsilon/2$-differentially private M for an anomaly identification query, the claim holds in general.

As for the claim of accuracy, it is a direct implication from the Construction 4.2. This completes the proof. □

Remark:

We emphasize that both of our constructions are not tied to any specific definition of anomaly, and even the requirement of Lipschitz continuity is due to privacy constraints.

Notes

[2]Note that the δ *in Figure 4.2 is a* $(0, 2)$*-acceptable lower bound on* $\Delta_{G_S} - \Delta_{\mathbb{G}}$ (as required by Theorem 4.4), where λ_1 is given by (6.3) for $k = 1$ and $\Delta_{\mathbb{G}}$ is given by (6.1). $\delta = \lambda_1 - \Delta_{\mathbb{G}} \geq 0$ follows because $\Delta_{G_S} \geq \lambda_1 \geq \Delta_{\mathbb{G}}$. The first inequality follows from Lemma 6.2. The second one trivially holds true for all the cases except for $x_i \geq 1$ and $B_x(i, r) < \beta$, where $\lambda_1(i, x) = \beta + 1 - B_x(i, r)$ and $\Delta_{\mathbb{G}}(i, x) = \min(x_i, \beta + 1 - B_x(i, r))$; thus, even in this case, we get $\delta(i, x) = \max(\beta + 1 - B_x(i, r) - x_i, 0) \geq 0$.

The 2-Lipschitz continuity of δ follows from the λ_1 and $\Delta_{\mathbb{G}}$ being 1-Lipschitz continuous (Lemma 6.2 and Lemma 6.1). Thus, for any i and two neighbors x and y in G_S (1-sensitive neighborhood graph),

$$|\delta(i, x) - \delta(i, y)| \leq |\lambda_1(i, x) - \lambda_1(i, y)| + |\Delta_{\mathbb{G}}(i, x) - \Delta_{\mathbb{G}}(i, y)| \leq 2.$$

CHAPTER 5

Private Outlier Analysis – A Principled Approach

Private outlier analysis consists of a class of data analytics problems for analyzing outliers, where privacy is to be protected. These problems differ depending upon the notion of outlier we use, the information we seek about the outliers, and the type of data privacy we want to guarantee. Thus, for a given private outlier analysis problem, an analyst has to choose the right problem-specification: (i) an appropriate outlier model, (ii) a query, and (iii) the definition of privacy. We combine the first two choices (i.e. (i) and (ii)) as the choice of an *outlier query*.

We must note that privacy cannot be considered in a vacuum and any private analytics task needs to take both privacy and accuracy into account. Thus, in this context, the notion of privacy and the notion of outlier query form the two fundamental constituents of the problem of private outlier analysis. We use these two fundamental constituents to provide a general framework to help the analyst choose the right problem-specification for private outlier analysis. The framework provides a two-step process. First, we show how to identify the relevant problem-specifications and then provide a practical solution that formally meets these specifications.

To protect privacy while computing outlier queries, an analyst may prefer differential privacy (DP) due to its strong guarantees. However, as shown before, DP does not work well for all the settings for outlier analysis (see Section 2.2). So the natural question is:

What makes a setting unsuitable for differential privacy?

Here, we answer this question. We first present a novel privacy-oriented taxonomy for outlier queries, which we conceptualize to isolate the suitable settings for differential privacy from the unsuitable ones. Under this taxonomy, we *formally* characterize the

privacy-utility trade-off of a class of queries that is comprised of the unsuitable settings for differential privacy. The formal treatment, here, is necessary since our analysis will not be confined to a specific outlier model or a specific outlier query.

Afterwards, we will present a framework that an analyst can use to pick the right problem-specification for private outlier analysis. The privacy-oriented taxonomy for outlier queries plays a central role in developing this framework.

Let use begin with the following three examples of outlier queries that cover a wide range of outlier analyses of practical interest. Here, at an intuitive level, for any given database, we either wish to identify which records are potentially outliers, or find if there is an outlying event (recall that an event is a notion that depends on many records; for more details, see Section 2.1).

Q_1: *Is the following transaction potentially a fraudulent one?*
"transfer 1 million dollars to acc# 123-4567, Cayman Isl. Bank"

Q_2: *Which of today's transactions are potentially fraudulent?*

Q_3: *Is there a COVID-19 outbreak in New York City in the last 14 days?*

We now highlight the heart of the matter by comparing Q_1, Q_2, and Q_3.

Recall (from Section 2.1) that in the case of an COVID-19 (corona virus disease 2019) outbreak, the notion of "COVID-19" and "outbreak" are universally defined concepts, and further, a "COVID-19 outbreak" is a property of the entire database (a predicate over its records). This sets Q_3 apart from Q_1 and Q_2. Because Q_1 and Q_2 regard individual records, where we tag a transaction as "potentially fraudulent" relative to other records in the database.

Furthermore, we note that the query Q_1 pertains to information that is not necessarily an entry in the database. Q_2, however, pertains to the information about the records that *exist* in the database. This seemingly unimportant issue not only plays a pivotal role in specifying the right privacy-outlier problem, but it serves as a crucial technical distinction when solving the problem.

5.1 Outlier Queries and the Notion of Existence-Independence

Here we give the important notion of *existence-independence* that naturally separates the settings that are suitable for DP from the unsuitable ones. We will use this notion to define the classes of outlier queries under our taxonomy. This notion helps characterize if an outlier query depends on the existence of any particular record(s) in the database — a property of the queries that is crucial to DP performing well in practice (as we will see shortly). To this end, let us first formalize what is an outlier query.

Outlier query:

An outlier query is a function, f, over $\mathcal{D}$, whose output (directly or indirectly) gives some information about the outlier(s). We assume, for each outlier query, an outlier model is prefixed.

For clarification, consider our example queries. The output of Q_1 is *yes* or *no* (to indicate if the record is an outlier). The output of Q_2 is a subset of records in the database, which are outliers. Next, consider a variant of Q_3 that is used by CDC[1]. For this variant, an outlier query f outputs the number of COVID-19 confirmed cases, which is then used to check for an outbreak, an outlying event.

Existence-independence:

Existence-independence is a constraint on a database for a given outlier query. Namely, a database meets this constraint if the output of the outlier query corresponding to an input database that has a record i can be obtained for another database that does not contain the record i.

Formally, for a given outlier query, f, and $\mathcal{R}^f \subseteq \texttt{Range}(f)$, we say a database x such that $f(x) \in \mathcal{R}^f$ satisfies the *existence-independence constraint* if for every record $i \in x$ (i.e. $x_i \geq 1$), there exists a database y such that $i \notin y$ (i.e. $y_i = 0$) and $f(x) = f(y)$.

We call $\mathcal{R}^f$ the *admissible range* of f. We use it to characterize a subset of range

[1]Hospitals report total cases of certain diseases to Center of Disease Control (CDC), which CDC uses to check for an epidemic outbreak [64].

of f that is of practical interest. Thus, whenever we talk about existence-independence and its related concepts, we assume that an admissible range ($\mathcal{R}^f$) has been prefixed. In practice, a a domain expert decides the admissible range.

For instance, consider an outlier query, f, that outputs the number of outliers in the given database. As per the definition of outlier corresponding to f, it is possible that there exists a database $x \in \mathcal{D}$ in which all the records are outliers. Such a situation, however, does not occur in practice (see practical setting for outliers in Section 2.3). Thus, to consider only the practically possible cases, we need admissible range, $\mathcal{R}^f$.

Now, for a fixed database x (such that $f(x) \in \mathcal{R}^f$), if for every $i \in x$, we have another database y (i.e. $y \neq x$) that has the same number of outliers as in x (i.e. $f(y) = f(x)$), but it does not contain i, then x satisfies the existence-independence constraint. This holds for most of the typical settings in practice, wherein the number of outliers in a database is independent of the existence of any one particular record in the database.

Note that we choose to formulate existence-independence as oppose to its inverse because it is intuitively easier to understand.

5.2 Privacy Oriented Taxonomy

In this section, we use the existence-independence constraint to define three classes of outlier queries, and in the next section, we show how achieving DP in the unsuitable settings leads to poor accuracy in practice.

5.2.1 Existence-independent query.

An outlier query,f, is *existence-independent* if all the databases (in $f^{-1}(\mathcal{R}^f)$) satisfy the existence-independence constraint — here f^{-1} is the inverse function of f is such that $f^{-1}(\mathcal{R}^f) = \{x \in \mathcal{D} : \ f(x) \in \mathcal{R}^f\}$.

Examples are: (i) How many outliers are in the given database? This is existence-independent as explained above. (ii) Q_3 is also existence-independent since the occurrence of COVID-19 outbreak is independent of any single individual contracting the disease.

We know that differential privacy protects the existence of any record in the given database. The existence-independent queries, in a sense, are independent of this information, and hence for such queries, DP mechanisms can achieve both privacy and utility that is practically meaningful, e.g. [64, 54, 65, 52]. Many outlier queries of practical interest, however, are not existence-independent — this is a major reason for DP to result in poor utility. Such outlier queries are *existence-dependent*.

5.2.2 Existence-dependent query.

For a given outlier query, f, and an admissible range, $\mathcal{R}^f$, f is *existence-dependent* if it is not existence-independent, namely, there exists a database (in $f^{-1}(\mathcal{R}^f)$) that violates the existence-independence constraint. For example, Q_2 is an existence-dependent outlier query because a record that is not present in the database, cannot be in the output.

5.2.3 Typical existence-dependent query.

Many existence-dependent outlier queries of practical significance are such that there is not one but many databases (encountered in practice) that violate the existence-independence constraint. We call such a query *typical* existence-dependent query.

The examples of typical outlier queries are: (i) What are the outlier records in the database? and (ii) What are the top-k outlier records in the database?

We first clarify some notation and then define typical outlier queries. For any $x \in \mathcal{D}$ and $i \in \mathcal{X}$, $x^{(+i)}$ gives the database that has exactly one record of value i and is otherwise identical to x. For any outlier query, f, and $i \in \mathcal{X}$, let $\mathcal{H}_f(i) = \{f(y) : \ i \notin y\}$.

We say an existence-dependent outlier query, f, is *typical* if there exists a non-empty set $X_f \subseteq \mathcal{X}$ such that for every $i \in X_f$ there are *many* databases x such that

$f(x^{(+i)}) \notin \mathcal{H}_f(i)$, i.e. $x^{(+i)}$ violates the existence-independence constraint.

We call X_f the *discriminative set*. Note that for Q_2, the discriminative set is equal to $\mathcal{X}$. As for what constitutes *many*, it depends on the particular outlier model, its parameters' values, the maximum size of the database, and the size of $\mathcal{X}$. In general, however, we assume it is at least linear in $|\mathcal{X}|$.

In the next section, we show, via an upper bound on accuracy, how DP leads to poor accuracy for typical outlier queries.

5.3 Low Utility Under DP for Typical Outlier queries

We first present a high-level example to give an intuition of why DP does not seem to solve the right problem for the practical settings for typical outlier queries.

Consider a database where the landscape of the database is such that in various places the removal or addition of a record changes the output of an outlier (detection) query which outputs all the outlier records in the given database. See Figure 5.1, where if you add a record in the empty region, it will be an outlier and will be in the output of the outlier query; and if you remove an outlier record, it will be removed from the output.

However, the addition or removal of a record is what defines a neighboring database, and thus, for many common definitions of outliers any DP mechanism will irrevocably introduce excessive levels of noise to protect privacy, which leads to poor utility.

The poor utility for typical queries is due to the lower bound on the error of any DP mechanism, which we give in Claim 5.1. Henceforth, we refer to a typical existence-dependent outlier query as *typical outlier query*.

To state Claim 5.1, we need the following notion. For any typical outlier query, f, M_f is a randomized mechanism with domain $\texttt{Domain}(f)$ and range $\texttt{Range}(f)$. For every x and $i \in \mathcal{X}$, $x^{(-i)}$ denotes the database that has no record of value i and is otherwise identical to x.

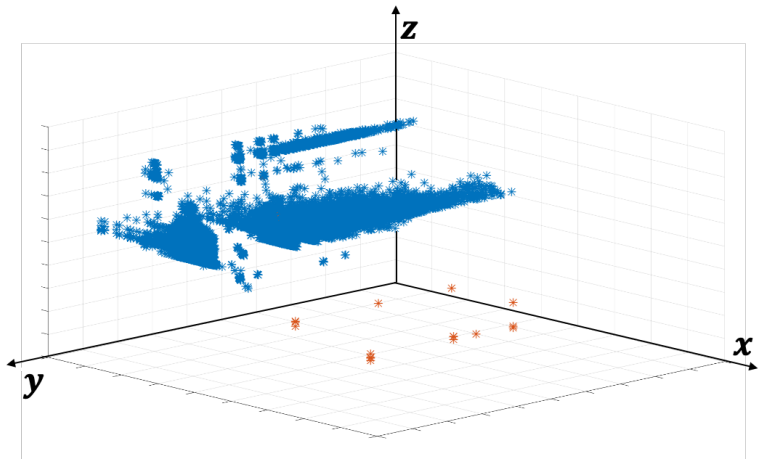

Figure 5.1: A real-world dataset with outliers from ODDS [1]. Blue points are non-outliers. Orange points are outliers.

Claim 5.1. *Fix $\varepsilon > 0$. For every typical outlier query, f, ε-DP mechanism, M_f, $i \in X_f$, and database x such that $f(x^{(+i)}) \notin \mathcal{H}_f(i)$,*

$$\max \left\{ \begin{array}{l} \mathrm{P}\big(M_f\big(x^{(-i)}\big) \in \texttt{Range}(f) \setminus \mathcal{H}_f(i)\big) \ , \\ \mathrm{P}\big(M_f\big(x^{(+i)}\big) \in \mathcal{H}_f(i)\big) \end{array} \right\} \geq \frac{1}{1+e^{\varepsilon}}$$

Proof. Arbitrarily fix $\varepsilon > 0$, a typical outlier query f, and an ε-differentially private mechanism M_f. Let X_f be the discriminative set for f. Lastly, fix arbitrary $i \in X_f$ and $x \in \mathcal{D}$ such that $f(x^{(+i)}) \in W(i) = \texttt{Range}(f) \setminus \mathcal{H}_f(i)$.

Let $P\big(M_f(x^{(-i)}) \in W(i)\big) \leq 1/(1+e^{\varepsilon})$, and note that $x^{(+i)}$ and $x^{(-i)}$ are neighboring databases under differential privacy. Then, from differential privacy constraints, it follows that

$$\begin{aligned} P\big(M_f(x^{(+i)}) \in W(i)\big) \leq & e^{\varepsilon} P\big(M_f(x^{(-i)}) \in W(i)\big) \\ P\big(M_f(x^{(+i)}) \in W(i)\big) \leq & e^{\varepsilon}/(1+e^{\varepsilon}) \end{aligned}$$

The last inequality holds due to our assumption. Since $\mathcal{H}_f(i)$ and $W(i)$ partition $\texttt{Range}(f)$, it follows from the above that

$$\begin{aligned} P\big(M_f(x^{(+i)}) \in \mathcal{H}_f(i)\big) = & 1 - P\big(M_f(x^{(+i)}) \in W(i)\big) \\ \geq & 1/(1+e^{\varepsilon}) \end{aligned}$$

When we let $P\big(M_f(x^{(+i)}) \in \mathcal{H}_f(i)\big) \leq 1/(1+e^{\varepsilon})$, in a similar fashion as above, we

can show that $P\big(M_f(x^{(-i)}) \in W(i)\big) \geq 1/(1+e^{\varepsilon})$. Since $\varepsilon, f, M_f, i \in X_f$, and x were fixed arbitrarily, the claim follows from the above. □

For many typical setting, to achieve a good enough privacy guarantees, ε is set to be small, i.e. closer to 0, for example, $\varepsilon = 0.1$. In such cases, the error would be closer to 1/2; namely, the chance of getting the correct answer from a DP mechanism with only two possible outputs (i.e. $M_f(x) = f(x)$) is almost the same as deciding the answer by the toss of a fair coin.

What are the implications of Claim 5.1 in typical real-world settings? To understand this, consider the query described in Q_2, and let f represent the query described in Q_2. Furthermore, let F be the outlier model (recall that F is the predicate, see Section 2.3 for complete description) such that f and F both corresponds to the same definition of anomaly.

Firstly, note for Q_2, $X_f = \mathcal{X}$. Then, see that from Claim 5.1, we get that for any database x and any record i such that $F(i, x) = 1$,

(a) if $x = x^{(+i)}$, $i \notin M_f(x)$ with probability $\geq \frac{1}{1+e^{\varepsilon}}$ or

(b) if $x = x^{(-i)}$, $i \in M_f(x)$ with probability $\geq \frac{1}{1+e^{\varepsilon}}$.

Now recall that real-world databases are such that:

(1) Many of the outliers are unique records in the database, i.e. for many i's in $f(x)$, i does not belong to $f(x - \mathbf{e}^i)$, and

(2) There are many possible records that are not present in the database and if we add any one of them to the database, it will be considered an outlier, i.e. there is a large set $I \subseteq \mathcal{X}$ such that for every $i \in I$, $x_i = 0$ and $F(i, x^{(+i)}) = 1$.

Thus due to (1), the antecedent of (a) holds for many outliers in x, and that of (b) holds for a large subset of $\mathcal{X}$ due to (2) (recall $|x| << |\mathcal{X}|$, and see practical settings for outlier in Section 2.3 for empirical evidence). Hence, if we try to reduce the error in the case of (a) (where $i \in f(x)$), the error for (b) will increase (where $i \notin x$ and thus

$i \notin f(x)$) and vice versa. Therefore, achieving DP for Q_2 or typical existence-dependent outlier queries, in general, will result in little utility in practice.

5.4 The Framework

Here we present an approach to analyze data for outliers to preserve both privacy and utility. Although this approach may seem rudimentary, it covers the most critical factors that an analyst must considered to achieve both privacy and accuracy that is practically meaningful.

We first provide an overview (also depicted in Figure 5.2), which is followed by the detailed account.

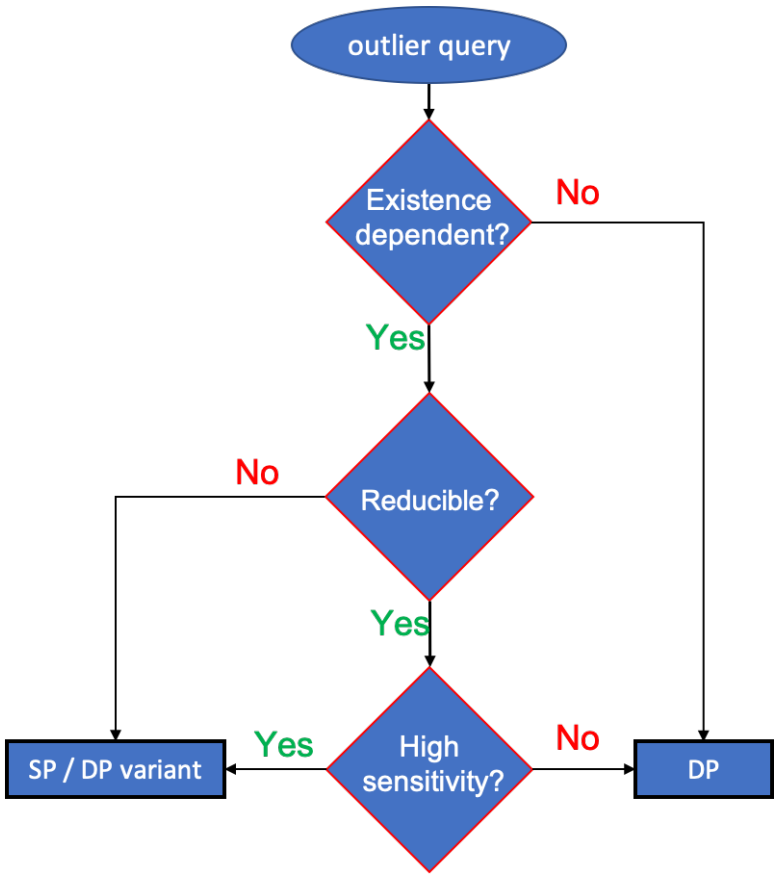

Figure 5.2: A Framework for Private Outlier Analytics

Given an outlier query, an analyst should first check if the query is existence-independent. If it is, then DP is the appropriate choice. If the query, however, is existence-dependent, then the analyst should try to reduce the query to a set of existence-independent queries, and compute the original query via the new queries. The above *reduction* (defined shortly) must take into account the various utility and information leakage constraints imposed by the analyst for her application. If there is no reduction that meets the imposed constraints, then *sensitive privacy* (or another variant of differential privacy) may be used — we refer to the variants of DP as *DP variant*. If a reduction, however, meets the constraints, then the next step is to analyze the accuracy in relation to achieving DP. If the accuracy achievable under DP is not practically meaningful for the application, then using sensitive privacy (or a DP variant) is the appropriate choice.

The detailed account of the approach follows, where we begin with the assumption that the given outlier query is existence-dependent as otherwise the analyst should use differential privacy.

5.4.1 Reducibility analysis

Let us say, after analyzing the given query, the analyst establishes that the query is an existence-dependent outlier (*endo*) query.

The first step for the analyst is to analyze if the query can be computed via a set of new queries such that none of the new queries is existence-dependent — here, we will refer to an existence-independent outlier query as *nendo*-query (as in not endo). We call such a set of new queries a *reduction* (defined shortly). This is particularly useful when data that we are to analyze is distributed among multiple parties as it is often the case for many practical settings; for example, the transaction data is distributed among banks and merchants, and health related data is distributed among doctors' offices, hospitals, and pharmacies.

We now formalize our notion of reduction. We do this in a multiparty setting, where different parties have different parts of a complete database. This complete database, however, is available to the trusted curator to whom the the parties can send

the queries to be computed (see privacy setting in Section 2.3 for details about the trusted curator setting). The multiparty setting is indeed more general, and hence, it makes the application of our framework wider. Furthermore, translating the concepts from multiparty setting to the single party setting is simple. Later, we will also give an example of reduction to compute an outlier detection query — an endo-query — to achieve both privacy and accuracy.

Say n parties, namely, parties $1, 2, \ldots, n$, are to compute an endo-query, f, on the database x that is distributed among these parties. For each party $t = 1, 2, \ldots, n$, $\mathcal{L}_t(f)$ represents the information that t-party receives during the computation of f. One can think of $\mathcal{L}_t(f)$ as the leakage similar to the one considered in secure computation [66, 67]. So, we call $\mathcal{L}_t(f)$ the *leakage* for t-party, which contains the output (i.e. the part of $f(x)$) of t-party and any other information that t-party receives about any other party's input. With this we are ready to give the definition of reduction.

Definition 5.1 (reduction)**.** *For a given existence-dependent outlier (i.e. endo) query, f, a* reduction*, $\mathcal{T}_f$, of f is $\{(f_{t,1}, f_{t,2}, \ldots, f_{t,m_t})\}_{t\in[n]}$ with leakage $\mathcal{L}_t(\mathcal{T}_f)$ for each t-party such that for every $t = 1, \ldots, n$:*

- *Each $f_{t,l}$, for $l = 1, \ldots, m_t$ is a nendo-query or a functionality that t-party can compute locally (i.e. without relying on any other party) given the outputs of $f_{t,1}, f_{t,2}, \ldots, f_{t,l-1}$, and*
- *Each $\mathcal{L}_t(\mathcal{T}_f)$ contains the output of t-party (and it may also contain some other information about the inputs of the other parties).*

Note that it is possible to have multiple reductions for a given endo-query, but not all of them may be acceptable to the analyst as per the requirements for her application. Thus, we say a reduction is *acceptable* if it satisfies the utility and information leakage constraints imposed by the analyst.

The utility constraints may include a bound on error as well as computation and communication power. The error we consider here may be inherent to one of the reduced queries (see the example below) or due to the internal coin tosses of the algorithm that the analyst uses to compute a reduced query (e.g. randomness may be needed for

efficiency) [68]. We remark that, at this stage, we do not consider the error due to achieving privacy (e.g. sensitive privacy or differential privacy).

The information leakage constraints impose a restriction on the information that a party can receive about other parties inputs. In some cases, to reduce an endo-query to nendo-queries it may be necessary to reveal some extra information to the parties so that they may compute the original endo-query. However, the analyst can still protect the secretes that are critical for her application, by imposing information leakage constraints.

Finally, if the provided reduction is acceptable, the next step is to analyze whether it is possible to use differential privacy to compute all the nendo-queries (given by the reduction) and achieve the accuracy required by the analyst for her application (we discuss this in the next section).

However, if an acceptable reduction is not possible, then sensitive privacy may be employed.

Example $\mathcal{T}_f$ from [54, 65].

As an illustrative example of reduction, consider the following situation. Assume n parties and $\ell = 20$. Each t-party has a database x^t such that $|x^t| \geq \ell$. Let $x = \sum_{t\in[n]} x^t$. Next consider AVF outlier model [69], F, that is known to all the parties (recall, F is a predicate over $\mathcal{X} \times \mathcal{D}$). Let f be the outlier query that to each party outputs its records (in x^t) that are among the top-ℓ outliers in x, i.e. ℓ records with the smallest AVF score[2]. Clearly, f is an endo-query as its output depend upon the records preset in the database x. The privacy and utility constraints for the reduction are as follows.

Privacy constraint: For each party, besides its output, the leakage may contain the frequency of each attribute value, as per x, ℓ-th smallest AVF score, λ_ℓ, corresponding to a record in x, but nothing else.

Utility constraint: For every $i \in x$ such that it is among the top-ℓ AVF based outliers, i must detected as an outlier, and should be in the output of t-party if $i \in x^t$ for every t.

[2]Roughly speaking, AVF score of a record is the normalized sum of the frequencies of its attribute values.

Reduction: Each t-party proceeds as follows. First, it collaboratively computes $f_{t,1}, f_{t,2}, \ldots, f_{t,m}$ on x, where $f_{t,l}$ gives the frequency of each value of l-th attribute as per x, and m is the total number of attributes. Second, using the output of the above queries, it computes $f_{t,m+1}$ on x^t, which gives $O_t = \{i_{t,1}, \ldots, i_{t,\ell}\}$, the top-$\ell$ outliers in x_t. Third, it computes $f_{t,m+2}$, which gives λ_ℓ (ℓ-th smallest AVF score). Lastly, for $l = 1, \ldots, \ell$, it computes $f_{t,m+2+l}(x) = AVFS(i_{t,l}, x) \leq \lambda_\ell$ to find the outliers in its database, where $AVFS$ gives the AVF score of $i_{t,l}$ as per the database x.

All of the above queries are nendo-queries except for $f_{t,m+1}$, which can be computed locally given AVF's. Furthermore, this reduction meets the given privacy constraint. As for the utility constraint, note that if F (the outlier model) is such that for every $z \in \mathcal{D}$, and every y that consists of subset of records from z, and every $i \in y$, $\quad F(i, z) = 1$ implies $F(i, y) = 1$, then the utility constraint will be satisfied. This property is satisfied by AVF outlier model as well as many other models, for example (β, r)-anomaly.

5.4.2 Sensitivity (i.e. utility) analysis under DP

Given an acceptable reduction, the analyst should check if computing the query via the reduction gives the required utility (e.g. desired accuracy or tolerable error) under an appropriate differential privacy guarantee, that is, the value of ε. Note that the analyst needs to consider the sensitivity of each query in the reduction, and its overall effect on the utility.

Recall that in differential privacy, there is a trade-off between the value of the ε and the accuracy one can achieve: smaller the value of ε, the higher the privacy but with lower utility. For example, one can set ε to be very high to achieve desired accuracy. However such an action will sacrifice the privacy of everyone whose data is used in the analysis.

Therefore, an analyst must use the best methods available in the literature to achieve the highest possible utility for an appropriate value of ε.

We must note that so far there is no standard way to set the value of ε. Furthermore, what is considered an acceptable or appropriate value of ε changes from one application to the other. In the literature, the value of ε ranges from 0.01 to 10 [70, 71].

Nevertheless, an analyst must strive to develop mechanisms that use a smaller value of ε for differential privacy to achieve the desired accuracy. This is important since using a large value of ε weakens the privacy guarantee for everyone. Thus, for now, it is the analyst's or the domain expert's prerogative to decide what is an appropriate value of ε. In this regard, we say an outlier query has *high sensitivity* if for typical values of ε (for the application or domain under consideration), the required accuracy is unachievable under differential privacy. In such a case, sensitive privacy (or a DP variant) may be employed to provide a stronger privacy guarantee for most of the records in the database as opposed using DP and weakening the privacy guarantee for all.

Methods to boost utility under DP for fixed ε:

Here we briefly discuss some of the ways to improve accuracy under DP. If the range of a reduced query is real numbers (or integers), then one can calculate the global sensitivity[3] and use Laplace mechanism[4] [23, 22] to achieve DP.

Count queries (which count the number of records in a database, which satisfy a given predicate) can be computed via Laplace mechanism with very reasonable accuracy guarantees [23]. However, if the global sensitivity of the query is too high, and the Laplace mechanism cannot produce the required accuracy, the analyst can opt for the mechanisms that are designed using a smooth upper bound on the local sensitivity of the query [72]. Such differentially private mechanisms (compared to the DP mechanism that use global sensitivity) have been shown to improve utility considerably for many problems of practical interest, for example, to compute median, k-means clustering [72], and the number of outliers in the data [52]

When an analyst has to compute many queries to carryout a private outlier analysis (e.g. in the case of reduction of an endo-query), she can use mechanisms optimized for

[3]Global sensitivity, i.e. $\Delta(f) = \max_{x \sim y} ||f(x) - f(y)||_1$, where $x \sim y$ means that the databases x and y are neighbors as per DP [23, 22, 46].

[4]Laplace mechanism achieves ε-DP by adding independent noise from Laplace distribution (of mean zero and scale $\Delta(f)/\varepsilon$) to each coordinate of the output.

a batch of queries [73] to boost utility. Exponential mechanism [23] is another way to design DP mechanisms to meet the accuracy constraint specific to an application.

Sensitive privacy (or other DP variants)

By this point the analyst has established that differential privacy is not the appropriate choice to compute the existence-dependent outlier query she is dealing with. In this case, sensitive privacy comes to the rescue — it can protect privacy of most of records with a very strong guarantee as well as achieve high utility.

Although the analyst can use some other variant of differential privacy, we note that sensitive privacy is particularly defined for private outlier analysis and is well-suited for this task, especially for identifying anomalies; see Section 2.2 for the DP variants relevant to outlier analysis and [50] for a survey of DP variants.

Sensitive privacy is able to handle data-independent as well as data-dependent notions of outliers, where the second case is the hardest to deal with. In Chapter 6, we will show that on real-world datasets for outlier analysis, sensitively private mechanisms commit error that is practically zero when the dataset sizes are large enough. Furthermore, sensitive privacy generalizes many notions of privacy and overcomes their limitations (see Section 3.7).

Part III

APPLICATIONS

CHAPTER 6

Private (β, r)-Anomaly Identification

Here, we show how to use our constructions (from Chapter 4) to develop sensitively private mechanisms for anomaly identification queries (AIQs). In particular, we instantiate Construction 4.1 for (β, r)-anomaly model to identify anomalies in data.

We use (β, r)-anomaly model to characterize outliers because of its prevalence in practice and its generalizability to many other outlier models. Furthermore, it has many well-known variants and extensions [41, 55, 6] — and our work naturally extends to them. On typical inputs, our sensitively private (SP) mechanism errs with an exponentially small probability (Theorem 6.2). The empirical evaluation of the SP mechanism over a range of real-words datasets also supports this result — for large enough datasets, the relative error in identifying anomalous (i.e. outlier) records is about 10^{-10}.

Recall that anomaly identification queries are existence-dependent outlier queries, and we consider the trusted curator setting (where the curator has access to all the data). We pose our AIQs to the trusted curator, who computes these queries via a privacy-protecting mechanism and answers 1 (i.e. yes) or 0 (i.e. no). Furthermore, as per our framework for private outlier analysis (in Chapter 5), we consider the information leakage constraint to be such that the querier must only receive the answer to the AIQ (i.e. 1 or 0) and nothing else. Thus, AIQ is not reducible (see Chapter 5 for details), and sensitive privacy is the appropriate choice here. In Section 6.3, we will see that even the 'best' DP mechanism performs poorly for this setting.

We will first use Construction 4.1 to develop an optimal differentially private (DP) mechanism for (β, r)-AIQ (Theorem 6.1). We do this for two reasons. First, we will use

this optimal DP mechanism as a baseline to compare our SP mechanisms' performance. Second, and importantly, to develop the optimal DP mechanism, we give the minimum discrepant distance (mdd) function ($\Delta_{\mathbb{G}}$) for the DP neighborhood graph ($\mathbb{G}$) and (β, r)-anomaly identification function, which we will use to develop our SP mechanism. Specifically, we will use $\Delta_{\mathbb{G}}$ to give a lower bound for the mdd-function for k-sensitive neighborhood graph (for (β, r)-normality property) and (β, r)-anomaly identification function, which can be used to develop SP mechanism via Construction 4.1.

6.1 Optimal DP-mechanism for (β, r)-AIQ

For $\lambda = \Delta_{\mathbb{G}}$ (in Construction 4.1) gives an pareto optimal DP mechanism for (β, r)-AIQ (Theorem 6.1). Below, we give $\Delta_{\mathbb{G}}$ for arbitrary $\beta \geq 1$ and $r \geq 0$, $i \in \mathcal{X}$, and $x \in \mathcal{D}$ — which is indeed the mdd-function for (β, r)-AIQ (Lemma 6.1). Recall that $B_x(i, r) = \sum_{j \in \mathcal{X}: d(i,j) \leq r} x_j$.

$$\Delta_{\mathbb{G}}(i, x) = \begin{cases} 1 & x_i = 0 \wedge B_x(i, r) < \beta \\ 2 + B_x(i, r) - \beta & x_i = 0 \wedge B_x(i, r) \geq \beta \\ \min\left(x_i, \beta + 1 - B_x(i, r)\right) & x_i \geq 1 \wedge B_x(i, r) \leq \beta \\ B_x(i, r) - \beta & x_i \geq 1 \wedge B_x(i, r) > \beta \end{cases} \tag{6.1}$$

Lemma 6.1. *For every $\beta \geq 1$ and $r \geq 0$, $\Delta_{\mathbb{G}}$ (given by (6.1)) is the mdd-function for (β, r)-anomaly identification function, where $\mathbb{G}$ is the DP neighborhood graph.*

Theorem 6.1 ($U_{\Delta_{\mathbb{G}}}$ is optimal and DP)**.** *Arbitrarily fix $\varepsilon > 0$, $\beta \geq 1$ and $r \geq 0$, and let $\Delta_{\mathbb{G}}$ (given by (6.1)) for (β, r)-anomaly identification function, g, where $\mathbb{G}$ is the DP neighborhood graph. Then, for any fixed (β, r)-AIQ, (i, g), $U_{\Delta_{\mathbb{G}}}$ (Construction 4.1) is pareto optimal ε-DP mechanism.*

The claim that the $U_{\lambda=\Delta_{\mathbb{G}}}$ is differentially private follows from Corollary 4.1. Because $\Delta_{\mathbb{G}}$ is the mdd-function for (β, r)-anomaly identification function (Lemma 6.1), and thus, for $\lambda = \Delta_{\mathbb{G}}$, λ is $(1, 1)$-acceptable lower bound on $\Delta_{\mathbb{G}}$ (follows from Lemma 4.2 since $\mathbb{G}$ is a neighborhood graph). Furthermore, for $G_S = \mathbb{G}$, Theorem 4.3 establishes the optimality claim of $U_{\Delta_{\mathbb{G}}}$.

Proof of Lemma 6.1. Let $\mathbb{G}$ be the neighborhood graph over $\mathcal{D}$, d be the distance metric over $\mathcal{X} \times \mathcal{X}$, $d_{\mathbb{G}}$ be the shortest path metric over $\mathbb{G}$, and g be the anomaly identification function for (β, r)-anomaly for arbitrarily fixed values of $\beta \geq 1$ and $r \geq 0$. Lastly, arbitrarily fix an $i \in \mathcal{X}$ and a database $x \in \mathcal{D}$.

We know that the value of $g(i, x)$ only depends upon x_i and $B_x(i, r)$ — recall that $g(i, x) = 1 \iff x_i \geq 1$ and $B_x(i, r) \leq \beta$. Further, $d_{\mathbb{G}}(x, y) = ||x - y||_1$ (Lemma 3.2). Hence, from the above, it follows that for $X(i, r) = \{j \in \mathcal{X} : \ d(i, j) \leq r\}$,

$$\Delta_{\mathbb{G}}(i, x) = \min_{y: g(i,y) \neq g(i,x)} ||x - y||_1 = \min_{y: g(i,y) \neq g(i,x)} \sum_{j \in X(i,r)} |x_j - y_j|. \tag{6.2}$$

We will consider four cases based on the condition (given in the $\Delta_{\mathbb{G}}$) that x satisfies. From (6.2), we know that $\Delta_{\mathbb{G}}(i, x)$ is the same as the minimum number of records by which a database y differs such that $g(i, x) \neq g(i, y)$. Thus in the proof we will modify the database x by adding or (and) removing records from x, and show that minimum number of changes required in x to change the output of g is given by $\Delta_{\mathbb{G}}$.

Case 1 $[x_i = 0 \wedge B_x(i, r) < \beta]$**:** Here, for any database y such that $g(i, y) = 1$, it must hold that $y_i \geq 1$ and $B_y(i, r) \leq \beta$. So we obtain a y by adding one record of value i to x. Thus $\Delta_{\mathbb{G}}(i, x) = 1$.

Case 2 $[x_i = 0 \wedge B_x(i, r) \geq \beta]$**:** Here, similar to the case above, $g(i, x) = 0$, and for any database y such that $g(i, y) = 1$, it must hold that $y_i \geq 1$ and $B_y(i, r) \leq \beta$. So we will have to add one record of value i to x to obtain a database y', but now $B_{y'}(i, r) \geq \beta + 1$. Thus, to obtain a y, we will have to remove $B_{y'}(i, r) - \beta = B_x(i, r) + 1 - \beta$ records of values in $X(i, r) \setminus \{i\}$ from y' (or x). Thus, $\Delta_{\mathbb{G}}(i, x) = 1 + B_x(i, r) + 1 - \beta$.

Case 3 $[x_i \geq 1 \wedge B_x(i, r) \leq \beta]$**:** In this case, $g(i, x) = 1$. For a y such that $g(i, y) = 0$, either $y_i = 0$ or $B_y(i, r) \geq \beta + 1$. Thus $\Delta_{\mathbb{G}}(i, x)$ will be the minimum of x_i (which corresponds to the number of records of value i present in x that we will have to remove) and $\beta + 1 - B_x(i, r)$ (which corresponds to the number of records of values in $X(i, r)$ that we will have to add to x).

Case 4 $[x_i \geq 1 \wedge B_x(i,r) > \beta]$**:** In this case, $g(i,x) = 0$ because $B_x(i,r) > \beta$. Thus, we will have to remove $B_x(i,r) - \beta$ records of values in $X(i,r)$ from x such that there is at least on record of value i in the modified x. Hence, $\Delta_{\mathbb{G}}(i,x) = B_x(i,r) - \beta$.

Further, in all the cases, $\Delta_{\mathbb{G}}(i,x) \geq 1$. Therefore, we conclude the $\Delta_{\mathbb{G}}$ is the mdd-function for g (i.e. (β,r)-AIQ).

□

6.2 SP-mechanism for (β,r)-AIQ

Below, we give λ_k is $(1,1)$-acceptable lower bound on the mdd-function for the k-sensitive neighborhood graph for (β,r)-normality property (Lemma 6.2). For this λ_k, Construction 4.1 yields (ε,k)-SP mechanism, U_{λ_k}, for (β,r)-AIQ such that, for non-sensitive records, U_{λ_k} can have exponentially small error in β (Theorem 6.2). Below, we give λ_k for arbitrary $k,\beta \geq 1$, $r \geq 0$, $i \in \mathcal{X}$, and $x \in \mathcal{D}$.

$$\lambda_k(i,x) = \begin{cases} \Delta_{\mathbb{G}}(i,x) & B_x(i,r) \geq \beta + 1 - k \\ \beta + 1 - B_x(i,r) & \\ + \min(0, x_i - k) & B_x(i,r) < \beta + 1 - k \end{cases} \tag{6.3}$$

Lemma 6.2. *Arbitrarily fix $k,\beta \geq 1$ and $r \geq 0$. Let g be (β,r)-anomaly identification function and Δ_{G_S} be the mdd-function for g, where G_S is the k-sensitive neighborhood graph for (β,r)-normality property. Then, λ_k (given by (6.3)) is $(1,1)$-acceptable lower bound on Δ_{G_S}.*

It is clear from the definition of λ_k (given by (6.3)) that when a record, i, is k-sensitive with respect to x, $\lambda_k(i,x) = \Delta_{\mathbb{G}}(i,x)$, which implies that there is no gain in utility (i.e. accuracy) compared to the optimal DP mechanism (in Section 6.1). However, when a record is not sensitive, $\lambda(i,x) > \Delta_{\mathbb{G}}(i,x)$, our SP mechanism achieves much higher utility compared to the optimal DP mechanism, which is especially true

for the records that are (β, r)-anomalous with a higher degree of outlyingness (i.e. the records that lie in a very sparse region).

Theorem 6.2 (accuracy and privacy of U_{λ_k})**.** *Arbitrarily fix $\varepsilon > 0$, $k \geq 1$, and a (β, r)-AIQ, (i, g). Let λ_k be as given by (6.3) and G_S be the k-sensitive neighborhood graph for (β, r)-normality property. Then, the mechanism, U_{λ_k} (Construction 4.1) is (ε, k)-SP such that for every $i \in \mathcal{X}$ and $x \in \mathcal{D}$ if i not k-sensitive for x, then*

$$P(U_{\lambda_k}(x) \neq g(i, x)) \leq e^{-\varepsilon|\beta+1-k-B_x(i,r)|}$$

The privacy claim follows from Lemma 6.2 and Theorem 4.2, while the error bound follows from Theorem 4.2 and the definition of λ_k — note that $B_x(i, r) < \beta + 1 - k$ implies that i is not sensitive for x (Lemma 3.1).

We give an example to show that U_{λ_k} achieves high accuracy in typical settings. Fix $k \leq \beta/10$. Now for any record i in a database x, satisfying $B_x(i, r) \leq \beta/2$ is an outlier for which U_{λ_k} *will err with probability less that* $e^{-2\varepsilon\beta/5}$.

Proof of Lemma 6.2

We need the following lemma to prove the main lemma (Lemma 6.2).

Lemma 6.3. *Let $\mathbb{G}$ be the DP neighborhood graph. Then, for every neighborhood graph G, $X \subseteq \mathcal{X}$, and every connected x, y in G, it follows that*

$$d_G(x, y) \geq d_{\mathbb{G}}(x, y) = ||x - y||_1 \geq \sum_{j \in X} |x_j - y_j| \geq \left| \sum_{j \in X} (x_j - y_j) \right|.$$

Proof. Let $\mathbb{G}$ be the neighborhood graph over $\mathcal{D}$. Arbitrarily fix a neighborhood graph G, $X \subseteq \mathcal{X}$, and $x, y \in \mathcal{D}$ that are connected in G.

Since $\mathcal{E}(G) \subseteq \mathcal{E}(\mathbb{G})$, we get that $d_G(x, y) \geq d_{\mathbb{G}}(x, y)$, where $\mathcal{E}(G)$ gives the set of edges of G. Furthermore, $d_{\mathbb{G}}$ is the same as ℓ_1-metric over the databases (Lemma 3.2). Hence, it follows that $d_G(x, y) \geq d_{\mathbb{G}}(x, y) = ||x - y||_1$.

The second inequality (in the lemma) holds because X is a subset of $\mathcal{X}$, and thus, $||x - y||_1 = \sum_{j \in \mathcal{X}} |x_j - y_j|$. The third inequality follows from the reverse triangle inequality. This completes the proof. □

Proof of Lemma 6.2. Arbitrarily fix $\beta, k \geq 1$, and $r \geq 0$. Let g be the (β, r)-anomaly identification function and λ_k be as given by (6.3). Let $\Delta_{\mathbb{G}}$ and Δ_{G_S} be the mdd-functions for g, where $\mathbb{G}$ is the DP neighborhood graph and G_S is the k-sensitive neighborhood graph for (β, r)-normality property (i.e. the normality property corresponding to (β, r)-anomaly model).

We will fist show that $\lambda_k \geq 1$ and is a lower bound on Δ_{G_S}, and then prove that λ_k is 1-Lipschitz continuous — this together suffices as a proof of λ_k being $(1, 1)$-acceptable lower bound on Δ_{G_S}.

Part 1, $\Delta_{G_S} \geq \lambda_k \geq 1$: Arbitrarily fix a record i and a node (database) x in G_S. Firstly, note that $\lambda_k(i, x) \geq 1$. We can confirm this as follows. When $B_x(i, r) \geq \beta + 1 - k$, $\lambda_k(i, x) = \Delta_{\mathbb{G}}(i, x) \geq 1$ (Lemma 4.2 and Lemma 6.1). And when $B_x(i, r) < \beta + 1 - k$, both $\beta + 1 - B_x(i, r) \geq 1$ and $\beta + 1 - B_x(i, r) + x_i - k \geq 1$ hold. So, below we show that $\Delta_{G_S}(i, x) \geq \lambda(i, x)$.

If $\Delta_{G_S}(i, x) = \perp$, then the claim holds trivially as for all $j \in \mathcal{X}$ and $y \in \mathcal{D}$, $\lambda_k(j, y) \in \mathbb{N}$ (see Definition 4.2). Thus, in the following, we assume that $\Delta_{G_S}(i, x) \in \mathbb{N}$. That is, there is a database y such that it is connected to x and $g(i, y) \neq g(i, x)$. We prove the rest of Part-1 by considering two sub-cases, each corresponding to a condition given in the definition of λ_k.

Sub-case $[B_x(i, r) \geq \beta + 1 - k]$: Here, we show that $\Delta_{G_S}(i, x) \geq \Delta_{\mathbb{G}}(i, x)$.

$$\begin{aligned}\Delta_{G_S}(i, x) &= \min_{y \in \mathcal{D}: g(i,x) \neq g(i,y)} d_{G_S}(x, y) \\ &\geq \min_{y \in \mathcal{D}: g(i,x) \neq g(i,y)} d_{\mathbb{G}}(x, y) = \Delta_{\mathbb{G}}(i, x)\end{aligned}$$

The above inequality follows from Lemma 6.3. Hence, from the above, we conclude that if $B_x(i, r) \geq \beta + 1 - k$ then $\Delta_{G_S}(i, x) \geq \lambda_k(i, x)$.

Sub-case $[B_x(i, r) < \beta + 1 - k]$: Let $b = g(i, x)$ and fix any y in G_S such that $g(i, y) \neq b$ and $\Delta_{G_S}(i, x) = d_{G_S}(x, y)$ — note that such a y exists because $\Delta_{G_S}(i, x) \in \mathbb{N}$. For this case, we divide the argument into two sub-cases: $x_i = 0$ and $x_i \geq 1$.

When $x_i = 0$, it must hold that $y_i \geq 1$ and $B_y(i, r) \leq \beta$. Now, on any of the

shortest path from x to y, we will first reach a database z for which i is k-sensitive, otherwise $y_i \geq 1$ cannot hold as we can only add (or remove) sensitive records. Hence, $B_z(i,r) \geq \beta + 1 - k$ (from Lemma 3.1). Thus, for this z, we get

$$\begin{aligned}\Delta_{G_S}(i,x) =& d_{G_S}(x,z) + d_{G_S}(z,y)\\ \geq& d_{G_S}(x,z)\\ \geq& B_z(i,r) - B_x(i,r)\\ \geq& \beta + 1 - k - B_x(i,r) = \lambda_k(i,x).\end{aligned}$$

The second inequality follows from Lemma 6.3 and the fact that $B_z(i,r) \geq B_x(i,r)$, while the third one follows because $B_z(i,r) \geq \beta + 1 - k$.

When $x_i \geq 1$, it must hold that either $y_i = 0$ or $B_y(i,r) \geq \beta + 1$ (or both). If $y_i = 0$, then on any of the shortest path from x to y, we will first reach a database z, where i becomes k-sensitive, i.e. $B_z(i,r) \geq \beta + 1 - k$ (from Lemma 3.1). If z is the first such database, then $z_i \geq x_i$. Thus, we get the following.

$$\begin{aligned}d_{G_S}(x,y) =& d_{G_S}(x,z) + d_{G_S}(z,y)\\ \geq& \big(B_z(i,r) - B_x(i,r)\big) + |z_i - y_i|\\ \geq& 1 + \beta - k - B_x(i,r) + x_i. \end{aligned} \tag{6.4}$$

The first inequality follows from Lemma 6.3, and the second one follows from the fact that $B_z(i,r) \geq \beta + 1 - k$ and $x_i \leq z_i$. But if $B_y(i,r) \geq \beta + 1$, then

$$d_{G_S}(x,y) = d_{G_S}(x,y) \geq B_y(i,r) - B_x(i,r) \geq 1 + \beta - B_x(i,r) \tag{6.5}$$

From (6.4) and (6.5), we get the following, which establishes that λ_k is a lower bound on the Δ_{G_S}.

$$\Delta_{G_S}(i,x) \geq 1 + \beta - B_x(i,r) + \min(0, x_i - k) = \lambda_k(i,x)$$

Part 2, 1-Lipschitz continuity of λ_k: Arbitrarily fix $i \in \mathcal{X}$, and $x, y \in \mathcal{D}$ that are neighbors in G_S. Let $\lambda_k(i,x) \neq \lambda_k(i,y)$, otherwise the continuity condition is trivially satisfied (since for every z and j, $\lambda_k(j,z) \in \mathbb{N}$).

If both x and y satisfy the first condition of λ_k (i.e. $B_x(i,r), B_y(i,r) \geq \beta+1-k$), then the 1-Lipschitz continuity condition is satisfied because $\Delta_{\mathbb{G}}$ is 1-Lipschitz continuous (Lemma 4.2 and Lemma 6.1).

If x and y both satisfy the second condition of λ_k (i.e. $B_x(i,r), B_y(i,r) < \beta+1-k$), then $x_i = y_i$. This holds because: (1) i is not k-sensitive with respect to both x and y (Lemma 3.1) and we can only add or remove sensitive records to create neighbors, and (2) $|B_x(i,r) - B_y(i,r)| \leq 1$. Thus, the 1-Lipschitz continuity condition holds in this case as well.

Lastly, consider the case, where y and x respectively satisfy the first and the second condition of λ_k — note that this is without loss of generality due to symmetry (i.e. $|\lambda_k(i,x)-\lambda_k(i,y)| = |\lambda_k(i,y)-\lambda_k(i,x)|$). This case is only possible if $B_y(i,r) = \beta-k+1$ and $B_x(i,r) = \beta - k$. Since $||x-y||_1 = 1$, the following holds: $x_i \leq y_i \leq x_i + 1$. We divide our argument into two cases for x: $x_i = 0$ and $x_i \geq 1$.

When $x_i = 0$, $\lambda_k(i,x) = 1$ and y_i is either 0 or 1 (since $x_i \leq y_i \leq x_i + 1$). If $y_i = 0$, then from the definition of $\Delta_{\mathbb{G}}$, given by (6.1), we get the following:

$$\lambda_k(i,y) = \Delta_{\mathbb{G}}(i,y) = 2 \text{ for } k = 1, \text{ and } \lambda_k(i,y) = \Delta_{\mathbb{G}}(i,y) = 1 \text{ for } k > 1$$

But if $y_i = 1$, then $\lambda_k(i,y) = \Delta_{\mathbb{G}}(i,y) = \min(y_i, k) = 1$ as $k \geq 1$. Hence, the 1-Lipschitz continuity condition is satisfied for $x_i = 0$.

Next, consider the case, when $x_i \geq 1$. Here, it follows that

$$\begin{aligned}
\lambda_k(i,x) &= 1 + k + \min(0, x_i - k) = 1 + \min(x_i, k) \\
\lambda_k(i,y) &= \Delta_{\mathbb{G}}(i,y) = \min(y_i, k) \qquad (\text{since } x_i \leq y_i \text{ and } B_y(i,r) = \beta - k + 1)
\end{aligned}$$

Clearly, if $x_i < k$, then $\lambda_k(i,x) = 1 + x_i$ and $x_i \leq \lambda_k(i,y) \leq x_i + 1$; and if $x_i \geq k$, then $\lambda_k(i,x) = 1 + k$ and $\lambda_k(i,y) = k$ since $x_i \leq y_i \leq x_i + 1$; hence the 1-Lipschitz continuity condition is fulfilled in this subcase as well.

Thus, we conclude that in all of the above case, $|\lambda_k(i,x) - \lambda_k(i,y)| \leq 1$. Since i, x, and y (neighbor of x) were picker arbitrarily, we conclude that λ_k is 1-Lipschitz continuous lower bond on the Δ_{G_S}. This completes the proof. □

Dataset	**size**	**dim**	$(\boldsymbol{\beta}, \boldsymbol{r})$	**true** (β, r)**-anomalies**
Credit Fraud	$284,807$	28	$(1022, 6.7)$	103
APS Trucks	$60,000$	170	$(282, 16.2)$	677
Synthetic	$20,000$	200	$(97, 3.8)$	201
Mammography	$11,183$	6	$(55, 1.7)$	75
Thyroid	$3,772$	6	$(18, 0.1)$	61

Table 6.1: **Dataset specifications and parameter values.**

6.3 Empirical Evaluation

To evaluate the performance of the SP-mechanism for (β, r)-anomaly identification queries (given in Section 6.2), we carry out several experiments on synthetic dataset and real-world datasets from diverse domains: Credit Fraud [74] (available at Kaggle [13]), Mammography and Thyroid (available at Outlier Detection DataSets Library [1]), and APS Trucks (APS Failure at Scania Trucks, available at UCI machine learning repository [56]). Table 6.1 provides the datasets specifications.

We compare the performance of our SP mechanism with that of the optimal DP mechanism for (β, r)-AIQ (given in Section 6.1).

To generate the synthetic data, we followed the strategy of Dong et al. [75], which is standard in the literature. The synthetic data was generated from a d-dimensional mixed Gaussian distribution, given below, where $\mathbf{I}$ is the identity matrix of dimension $d \times d$, $\sigma << 1$, and e_{i_t} is a standard base. In our experiments, we used $\rho = .01$, $d = 200$, and $a = 5$, and chose a standard bases uniformly at random.

$$(1-\rho)\mathcal{N}(\mathbf{0}, \mathbf{I}) + \sum_{t=1}^{a} (\rho/a)[\frac{1}{2}\mathcal{N}(\sqrt{d/\rho}\ \mathbf{e}_{i_t}, \sigma^2\mathbf{I}) + \frac{1}{2}\mathcal{N}(-\sqrt{d/\rho}\ \mathbf{e}_{i_t}, \sigma^2\mathbf{I})]$$

The aim of this work is to study the effect of privacy in identifying anomalies. So we keep the focus on evaluating the proposed approach for achieving privacy for this problem, and how it compares to differential privacy in *real world settings*. Our experiments make use of (popular) (β, r) notion of anomaly.

Following the standard practice for identifying outliers in the data with higher dimension [60, 76], we carried out the principal component analysis (PCA) to reduce the dimension of the three datasets with higher dimension. We chose, top $6, 9$, and 12

features for the Credit Fraud, Synthetic, and APS Trucks datasets respectively. Next, we obtain the values of β and r, which typically are provided by the domain experts [55]. Here, we employed the protocol outlined in the Chapter's endnote[3] to find β and r; this protocol follows the basic idea of parameter selection presented in the work [55] that proposed the notion of (β, r)-anomaly. Table 6.1 gives the values of β and r, which we found through the protocol, along with the number of true (β, r)-anomalies (true anomalies identifiable by (β, r)-anomaly method for the given parameter values).

6.3.1 Results

To measure the performance of our mechanisms, we use the following definition of error.

Error: We measure the error of a private mechanism (which is a randomized algorithm) as its probability of outputting the wrong answer — recall that in the case of AIQ, there are only two possible answers, i.e. 0 and 1. For each AIQ for a *fixed record*, we estimate the error as the fraction of mistakes (a private mechanism makes) over m trials. So for our experiments we choose m to be 10000.

For each dataset, we find all the true (β, r)-anomalies and for each of them perform private anomaly identification query using SP-mechanism (given in Section 6.2) and DP-mechanism (given in Section 6.1) for $\varepsilon = 0.01,\ 0.1$, and 1 and compute the error, which we give by the box plot in Figure 6.1. The reason we only considered our DP mechanism for this part is that it is the best among the baselines (see Table 6.3) and it also has strong accuracy guarantees (Theorem 6.1).

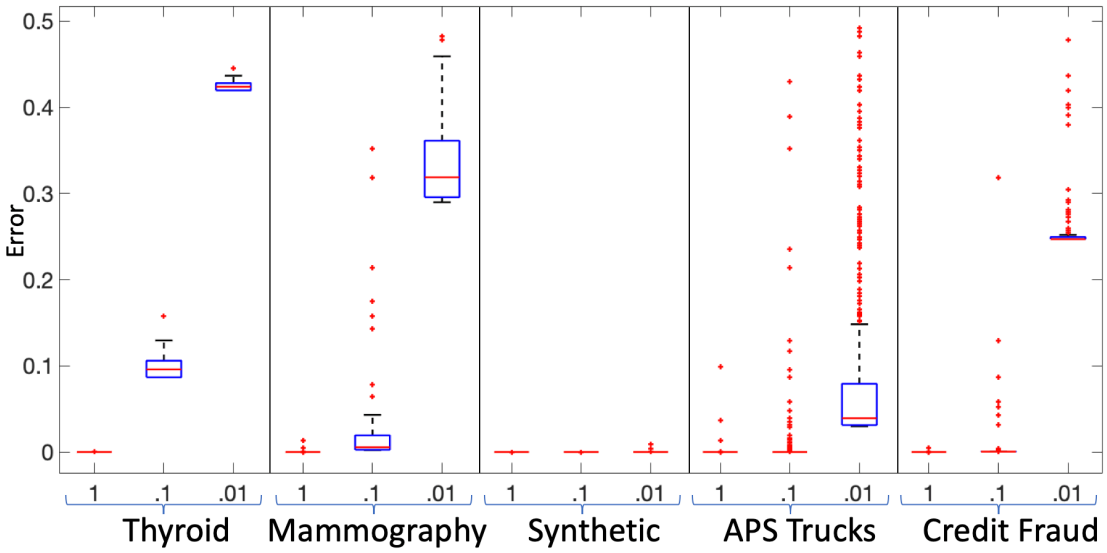

Figure 6.1: Box plots of the errors of the SP mechanism for (β, r)-AIQ over the true (β, r)-anomalies for $\varepsilon = \{.01, .1, 1\}$.

The error of SP-mechanism, in many cases, is so small (e.g. of the order 10^{-15} or even smaller for larger values of ε) that it can be considered zero for all practical purposes. Furthermore, as the data size increases (and correspondingly the value of β), the error of SP-mechanism reduces. However, in the case of anomalies, the error of DP-mechanism is consistently close to that of random coin flip (i.e. selecting 0 or 1 with probability close to 1/2) except for a few anomalous records in some cases — we will shortly explain the reason for this. The error of the SP-mechanism was overwhelmingly concentrated about zero (Figure 6.1), which is also true for the smaller values of ε.

Thus, *we can have higher privacy guarantee for sensitive records, while still being able to accurately identify anomalies.* Also, note that as the size of the dataset increases, not only does the error of SP-mechanism reduces (for anomalies), but also its divergence. Thus, it indicates that our methodology is even more appropriate for big data settings. On the other hand, for anomalies, the errors of DP-mechanism are concentrated about $1/(1 + e^{\varepsilon})$ (Figure 6.2). This is in accordance with our theoretical results and the assumption that the databases are typically sparse.

Next, we evaluated the performance over the normal records. Here, both the SP and the DP mechanisms performed equally (Figure 6.3). For the same value of ε, every

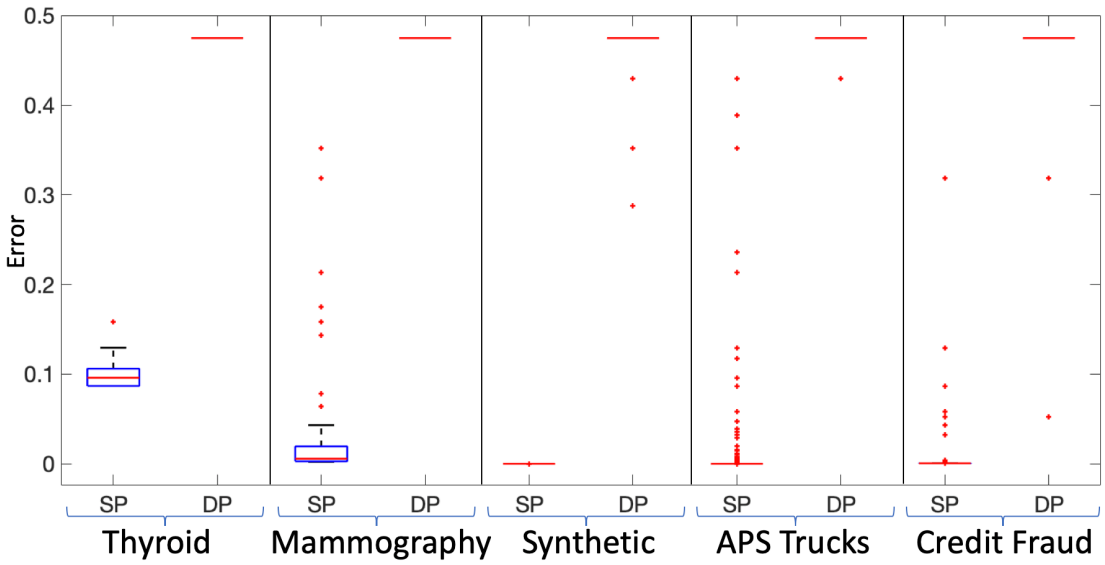

Figure 6.2: Box plots of the error of the SP and the DP mechanisms for (β, r)-AIQ over the true (β, r)-anomalies for $\varepsilon = 0.1$.

sensitive record in the database has the same level of privacy under sensitive privacy as all the records under differential privacy; thus the same level of accuracy should be achievable under both the privacy notions. Here we see again that datasets with larger sizes exhibit very small error.

To evaluate the performance over future queries, we picked n records uniformly at random from the space of possible (values of) records for each dataset, where n was set to be 20% of the size of the dataset. Here too the SP-mechanism outperforms the DP-mechanism significantly (Table 6.2). This is because most of the randomly picked records are anomalous as per the (β, r)-anomaly, which is due to the sparsity of the databases. This fact becomes very clear when we compare the mean error over the random records to the mean error over the anomalous records in the randomly picked records (see the second and the last column of Table 6.2). Since the probability of observing a mistake is extremely small (e.g., 1 in 10^{10} trials) , in Table 6.2, the mean is computed over the actual probability of error of the mechanism instead of the estimated error.

We know that by increasing k we move the boundary between sensitive and non-sensitive records (for example, we saw this in Figure 3.3). So to observe the effect of varying values of k on real world datasets, we carried out experiments for $k =$

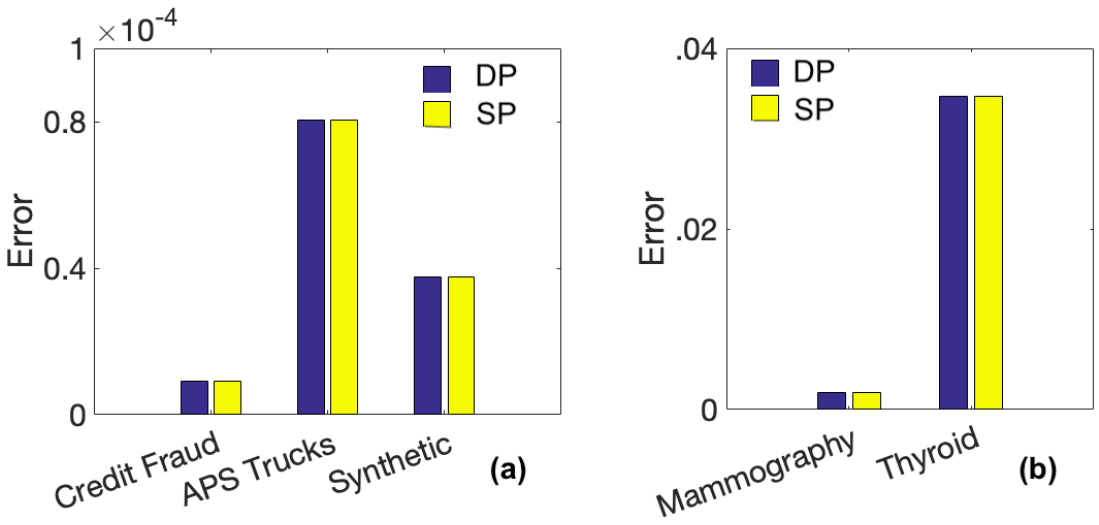

Figure 6.3: **Evaluation over normal records. (a),(b),** give the average error of SP and DP mechanism for AIQ over all the normal records from each data set; $\varepsilon = 0.1$.

Dataset	**mean error**		**mean error (anomalies)**
	SP	**DP**	**SP**
Credit Fraud	1.1127E−21	0.4750	1.1127E−21
APS Trucks	2.9719E−13	0.4750	2.9719E−13
Synthetic	3.2173E−5	0.4750	3.2173E−5
Mammography	0.0022	0.4749	0.0021
Thyroid	0.0870	0.4750	0.0867

Table 6.2: **Effect of sparsity of databases.** "mean error" is over the randomly picked n records from the possible values of the records for each dataset for SP and DP mechanisms for (β, r)-AIQ. "mean error (anomalies)" is only over the anomalous records in the n picked records. Here, n is 20% of the size of the dataset, $\varepsilon = 0.1$, and E$-n = 10^{-n}$.

$\lfloor 0.1\beta \rfloor$, $\lfloor 0.2\beta \rfloor$, and $\lfloor 0.3\beta \rfloor$ — recall that a record is considered k-sensitive with respect to a database if the record is normal or becomes normal under the addition and (or) deletion of at most k records from the database. Note that if $k \geq \beta+1$ then every record will be sensitive regardless of the database. The results are provided in Figure 6.4. Here we conclude that even for the higher values of k SP-mechanism performs reasonably well. Further, if the size of dataset is large enough, then the loss in accuracy for most of the records is negligible.

Recall that for Credit Fraud and APS Trucks datasets, differentially private AIQ for some of the anomalous records gave smaller error. The deviation using the Credit Fraud

dataset as an example. The above mentioned deviation in the error occurs whenever the anomalous record is not unique (Figure 6.5a-b), which is typically rare (Figure 6.5c). The reason DP-mechanism's error remains constant in most cases is that the anomalies lie in a very sparse region of space and mostly do not have any duplicates (i.e. other records with the same value).

Finally, to evaluate the overall performance of our SP-mechanism for (β, r)-AIQ, we computed precision, recall, and F_1-score [60]. We also provide a comparison with two different baseline mechanisms, B_1, B_2 in addition to pareto optimal DP mechanism (see Table 6.3).

B_1 and B_2 are the *best* performing mechanisms (i.e., with the highest F_1-score) from two families of mechanisms. This mechanisms serve as the naive base lines. Each mechanism in each of the family is identified by a threshold t, where $0 \leq t \leq 1$. Below, we describe the mechanisms from both the families for fixed ε, threshold t, record $i \in \mathcal{X}$, and database $x \in \mathcal{D}$.

The mechanism in the first family is given as follows. $B^i_{1,t}(x) = 1$ if and only if $\mathcal{O}(x) + \texttt{Lap}(1/\varepsilon) > t \times (||x||_1 + \texttt{Lap}(1/\varepsilon))$; here $\mathcal{O}(x)$ gives the number of anomalies in x and $\texttt{Lap}(1/\varepsilon)$ is independent noise from Laplace distribution of mean zero and scale $1/\varepsilon$.

The mechanism in the second family is given as follows. $B^i_{2,t}(x) = 1$ if and only if $\mathcal{O}(x) + \texttt{Lap}(\beta/\varepsilon) > t \times (||x||_1 + \texttt{Lap}(1/\varepsilon))$.

Note that, the mechanism from the first family are ε_1-DP, where $\varepsilon_1 \geq \beta\varepsilon$. This is due to the fact that $\max_{x,y \in \mathcal{D}:||x-y||_1=1} |\mathcal{O}(x) - \mathcal{O}(y)| = \beta$ [46]. However the mechanism from the second family are ε_2-DP, where $\varepsilon_2 \geq \varepsilon$.

Our SP mechanism for (β, r)-anomalies outperforms all the baselines. Furthermore, our DP-mechanism largely outperforms rest of the baselines.

Dataset	Precision				Recall				F_1-score			
	B_1	B_2	DP	SP	B_1	B_2	DP	SP	B_1	B_2	DP	SP
Credit Fraud	0.0101	0.0230	0.9930	0.9963	1.0000	0.0498	0.5250	0.9968	0.0199	0.0315	0.6868	0.9966
APS Trucks	0.0115	0.0165	0.9870	0.9931	1.0000	0.0753	0.5263	0.9954	0.0227	0.0271	0.6865	0.9943
Synthetic	0.0101	0.0114	0.9930	0.9963	1.0000	0.1189	0.5250	0.9968	0.0199	0.0208	0.6868	0.9966
Mammography	0.0070	0.0081	0.0211	0.2004	0.8244	0.1000	0.5250	0.9977	0.0138	0.0149	0.0435	0.3337
Thyroid	0.0174	0.0191	0.1427	0.3100	0.6656	0.2918	0.5250	0.8993	0.0339	0.0358	0.2244	0.4610

Table 6.3: B_1 and B_2 are the best mechanisms from two families of mechanism. DP and SP are the mechanisms from Section 6.1 and Section 6.2 respectively. Going from red to blue the value decreases. For our SP and DP mechanisms, $\varepsilon = 0.1$

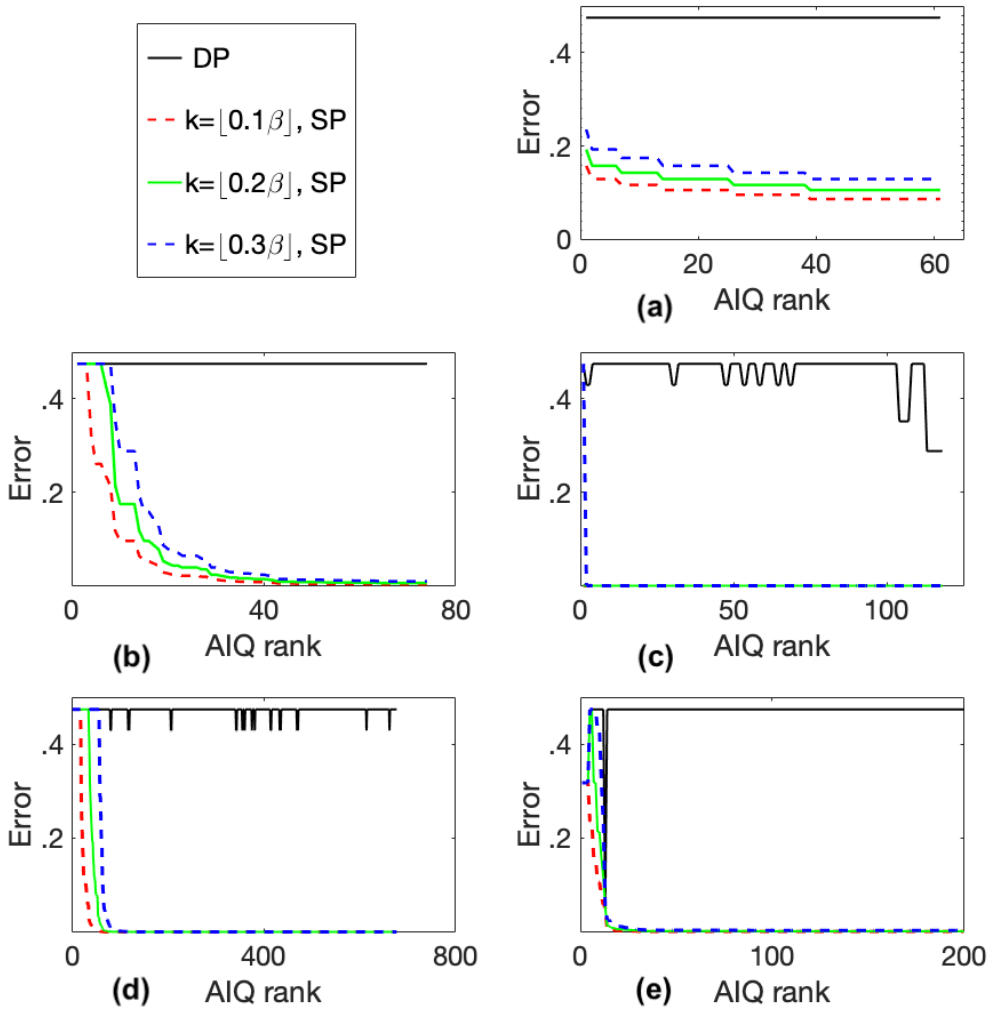

Figure 6.4: **Evaluation over true (β, r)-anomalies for varying k.** **(a)-(e)**, give the errors of SP and DP mechanisms. AIQ rank is given by the error of SP-mechanism for each anomaly: the higher the rank, the lower the error given by the SP mechanism. In all the figures, $\varepsilon = 0.1$. **(a)**, Thyroid, **(b)**, Mammography, **(c)**, Credit Fraud, **(d)**, APS Trucks, **(e)**, Synthetic data.

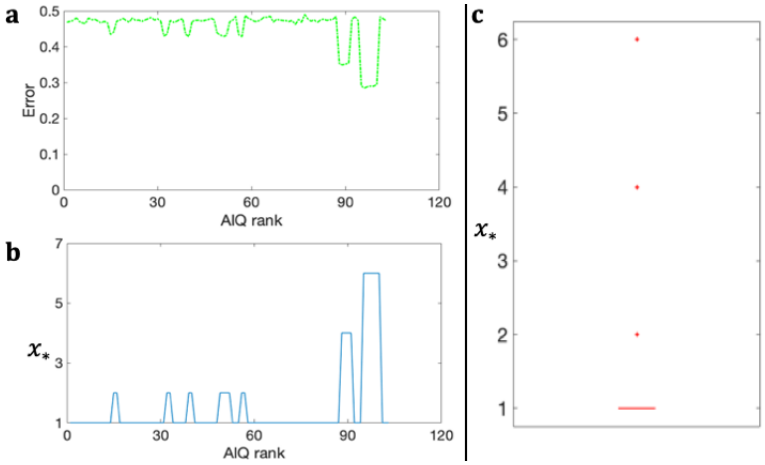

Figure 6.5: **Why DP mechanism's error deviates.** In **(a)**, the plot is the same as given in Figure 6.4c for the DP-mechanism. In **(b)** and **(c)**, x_* for each record is the number of records in the database x that have the same value. **(c)**, shows the box plot for the data.

Notes

[3]The main idea is to fix a value of β, for a dataset of size n, as $(1-p)\times n$, where p is close to 1, and then search for an appropriate value of r. It is recommended [55] that for the datasets of sizes 10^3 and 10^6, β be $(1-0.995)\times 10^3$ and $(1-0.99995)\times 10^6$. By assuming that p is linearly related to n, we can use the provided values to find the value of β for any given dataset.

For a fixed value of β, we search for r that maximize the F_1-score (also known as balanced F-measure), which is a popular performance metric for imbalanced datasets [77], and it is the harmonic mean of precision and recall. We use the following protocol to select the value of r.

Initialize $r_{\min} = .001$, $r_{\max} = 40$ (or the value that is not smaller than the maximum distance between any two points in the given dataset), $r = 0$, and $S = 0$. Next, set $r_1 = r_{\min} + (r_{\max} - r_{\min})/4$, $r_2 = r_{\min} + 3(r_{\max} - r_{\min})/4$, pick α from $[0, 1]$ uniformly at random and set $\mathrm{r}_3 = \alpha\ r_1 + (1-\alpha)\, r_2$. Compute F_1-score for each of the r's, i.e. S_{r_1}, S_{r_2}, and S_{r_3}. Let S_{r_t} be the maximum of the computed scores. Now, if S_{r_t} is greater than S then set $S = S_{r_t}$ and $r = r_t$; further, if $S_{r_2} < S$ and $r_2 > r$ then set $r_{\max} = r_2$ but if it is not the case and $S_{r_1} < S$ and $r_1 < r$ then set $r_{\min} = r_1$, otherwise do nothing. Repeat this process, except for the initialization step, until the improvement in S becomes insignificant. In our experiments, repeating the process for ten iterations generally sufficed.

CHAPTER 7

Private Hotspot and Epidemic Outbreak Detection

The coronavirus disease 2019 (covid-19) broke out in China, and in two months turned into a global pandemic [78]. While the disease spread rapidly, the population testing for covid-19 remained inadequate [79]. The result? Public health officials were unable to effectively track the spread of the covid-19 pandemic in real-time and the countermeasures lagged the actual spread of the pandemic until the countries were placed under lockdown.

So, to gain real-time insights about the spread of covid-19 and track its outbreaks in communities, we developed COVID Nearby — a privacy-preserving symptoms-tracking system for covid-19 [80]. In the absence of wide-spread covid-19 testing, this system crowdsenses covid-19 related information such as covid-19 related symptoms, demographic information, health history, and location from people. This data provides a useful alternative to track the pandemic and generate insights about existing and emerging covid-19 hotspots and outbreaks — which are outlying events [81, 82].

The surging pandemic coupled with inadequate covid-19 testing led to the development of many such symptoms-tracking systems across the world [83]. These systems, however, have one major problem: *the lack of privacy safeguards for the people who share their sensitive information* [83, 84]. One could argue that these systems do safeguard privacy as they only release anonymized and generalized aggregates. But such "safeguards" fail to protect privacy of everyone as people can still be identified using anonymized aggregates or data [14, 15, 16, 17, 18, 19, 20, 21].

In contrast, COVID Nearby guarantees differential privacy for everyone who share their data, and additionally, it allows people to query the information for any region

in the USA. Particularly, for a prefixed value of ε, it can answer an arbitrary number of spatiotemporal range queries with ε-differential privacy (DP) guarantee overall. For a given database of reported symptoms (with the time and location for each report), a spatiotemporal range query asks for the number of reports in a given region that were recorded within a specified time duration, for example, "How many people reported experiencing covid-19 symptoms in New York City within the past 14 days?" is a spatiotemporal range query.

Spatiotemporal range queries are the key to tracking covid-19. To find if an area is an emerging hotspot of covid-19, we count the people in this area who reported experiencing covid-19 symptoms within a specified time duration — which we compute via a spatiotemporal range query. Now, given that we can compute these counts (i.e. spatiotemporal range queries) for any region over time with a DP guarantee, it is straightforward to identify outbreaks, see [31] for details on how to identify outbreaks using these counts. Therefore, in this work we focus on computing these counts via spatiotemporal range queries.

Why DP is the appropriate choice here?

As per our framework for private outlier analysis (Chapter 5), differential privacy is the appropriate choice to compute spatiotemporal range queries privately and accurately. This is because spatiotemporal range queries are existence-independent outlier queries.

In this chapter, we show how to design and build a crowdsensing system to track covid-19, which guarantee privacy as well as practically meaningful accuracy.

7.1 System and Setting

Our proposed system has three basic components, which are given below. Figure 7.1 shows these components and their interaction.

1. **Crowdsensing:** It enables users to securely report data such as covid-19 symptoms, demographic information, and location; once the data is reported, it stores this information in a secure database.

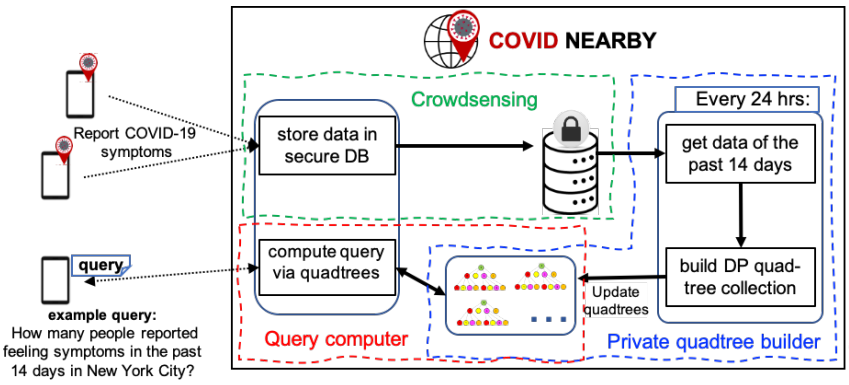

Figure 7.1: System architecture

2. **Private quadtree builder:** This component uses the reported data to create a collection of differentially private (DP) quadtrees — a data structure suited to efficiently store and query spatial data.

3. **Query computer:** It is responsible for answering users' queries using the collection of DP quadtrees.

Our approach allows one to compute differentially private answers to an arbitrary number of spatiotemporal range queries over time and on a dynamically changing database. Additionally, it has the following important properties:

- The risk to privacy is bounded and does not increase as users ask more queries.
- The query computation is efficient as the queries have to be answered in real-time.
- The accuracy is practically meaningful. For example, we are able to accurately rank covid-19 hotspots, generate heatmaps, and calculate moving average in terms of the numbers of symptom reports.

Setting

To present our approach, we focus on a specific type of spatiotemporal range query, named *SR query*: for a given database x of reports and a region R, SR query gives the number of reports in x that: (i) were reported in the past 14 days, (ii) lie in R, and

(iii) contain at least one COVID-19 symptom[1].

Furthermore, we consider the following setting:

- The space (of possible locations) is two-dimensional (2D) and bounded with north-south and east-west as two perpendicular axes. One can achieve this, for example, by taking Mercator projection of Earth [85].
- The region given by an SR query is an axis parallel rectangle in the 2D space.
- The database contains the reports that are from within the USA, and are tagged with location coordinates and the time they were reported at.
- We only consider one report per user since the DP guarantee is for each record, and hence, for each individual.
- The database is with a trusted curator who will answer SR queries using a differentially private mechanism.

7.2 Approach Overview

We create differentially private spatially indexed partitions of the space. For this, we use a hybrid of data agnostic and data dependent approaches. Firstly, we partition the space, based on legislatively defined administrative units such as country, state, or county — we call them *divisions*. Then, for any given day, we partition the past 14 consecutive days into N groups, where each group consists of distinct consecutive days. Finally, we partition each of the divisions by a building differentially private quadtree over the data reported (from within the division) for each group of days. The private quadtree builder is responsible for this space and (temporal) data partitioning.

To answer an SR query, we compute the query over all the quadtrees whose corresponding division overlaps with the query region and contain data of the past 14 days. Lastly, we add the answers from all the quadtrees to compute the final output and return it to the user. The Query computer is responsible for this part.

[1]This can be extended to more fine grained information (for example, in terms of the counts of people with particular symptoms) but with a corresponding trade-off of privacy.

In the following sections, we present the details of our approach.

7.3 Spatial Partitioning — A Hybrid Approach

In this section, we discuss how to create spatial partitions for any given day using the data of the past d days — think of $d = 14$. In the next section, we will show how to create temporal partitions of the data of the past d days.

For data agnostic partitioning, we defined the *division* to be the county/district. Then, on any given day, we build a DP quadtree over each division using the data of the past d days from the division.

A quadtree is a hierarchical spatial data structure. For a given rectangular space and the data lying in it, the quadtree recursively, level by level, partitions the space into rectangular regions by bisecting their sides — the rectangular regions are called quadrants (see Figure 7.2(a)-(b)). At each level, every node in a quadtree corresponds to a quadrant and contains the total number of data points (i.e. reports) lying in that quadrant. We control the granularity of partitions that a quadtree creates using the following two parameters: (i) the *max height* (i.e. the maximum number of levels), and (ii) the *min count* threshold, that is, the minimum number of data points that have to be in a quadrant for it be further partitioned.

For the division (i.e. the county) to be partitioned by a quadtree, we use the bounding box of the division (which we get from Google Map's API), and use the length w (in kilometers) of its longest side to compute the max height as $h = \lfloor log_2(w) \rfloor$. This ensures that even in the smallest partition one of the sides will be 1 km long. For min count threshold, we use a value of 10. This threshold serves an important purpose. It helps stop the partitioning of the empty quadrants or the quadrants with very little data and improves the accuracy. As for the data, we use the past 14 days reports from the division, each of which contains at least one covid-19 symptom.

Now, we show how to build an ε-DP quadtree. Here, we will use Laplace mechanism to compute private counts. For a given count c, Laplace mechanism perturbs c by adding some noise. This noise is sampled from Laplace distribution of mean zero and scale λ

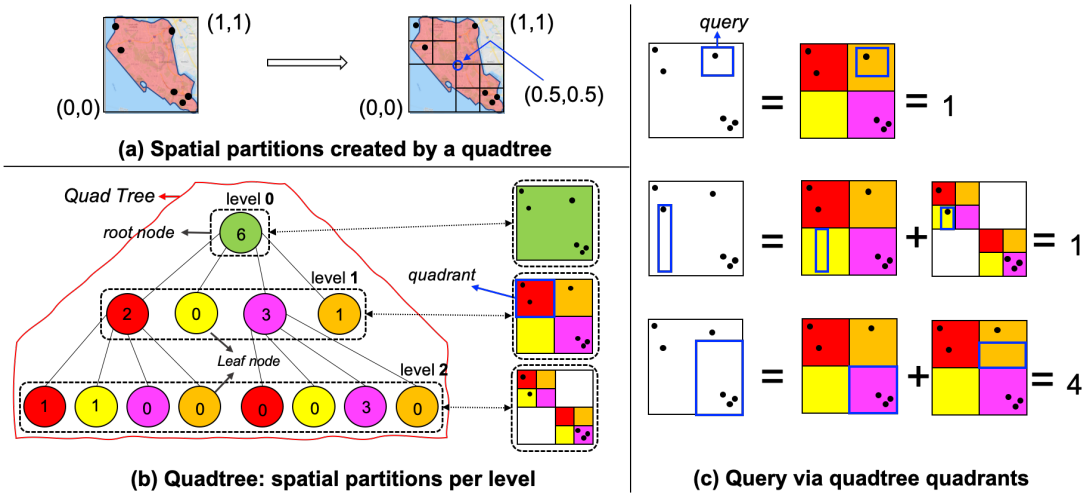

Figure 7.2: **(a)** shows a partition of a space (bounding box of a division (Richmond County, NY) shown in red shade) created by a quadtree of height 2 and min count 2; each square in the panel **(a)**, left to the arrow, represents a quadrant, while all the black points together represent the data. **(b)** gives the quadtree that created the partitions shown in **(a)**; it also depicts the quadrants (or partition) at every level. **(c)** shows how queries are computed by splitting them across quadrants and using the count in the node of the quadtree.

to guarantee $(1/\lambda)$-DP.

We begin by fixing the value of ε — this is our privacy budget, and when building a tree, we must not exceed this budget. When building a quadtree, at every level ℓ, we compute the count (the number of reports) in every quadrant by using Laplace mechanism for $\lambda_\ell = d/\varepsilon_\ell$ (why the factor d? We will explain this shortly). When $\ell = 0$ (i.e. at the root), we use $\varepsilon_0 = 1$ — this level corresponds to the division level count — and this fixed budgeting strategy boosts the accuracy of the queries with larger region that consists of multiple divisions; for example, a query that asks for the count in a state. For $\ell \geq 1$, we distribute rest of the privacy budget, i.e. $\varepsilon' = \varepsilon - \varepsilon_0$, geometrically such that for all $\ell \geq 2$, $\varepsilon_{\ell+1} = 2^{1/3}\varepsilon_\ell$ and $\varepsilon_1 = \varepsilon'(2^{1/3} - 1)/(2^{h/3} - 1)$. This allocation provably improves the accuracy [86]. Note that in building a quadtree, the privacy budget does not exceed $\varepsilon/d = (\varepsilon_0 + \cdots + \varepsilon_h)/d$ (due to sequential composition) because any record in the data is used to compute at most h counts, one count per each level from the root to a leaf.

Why the factor d?

Above, in all the scales, λ_ℓ, the factor d is due to sequential composition: because each quadtree is built over the data of past d days, and hence, each person's data is used to build d quadtrees, one for each day. Thus, the overall privacy risk due to building quadtrees over time does not exceed ε.

7.3.1 Why use the hybrid approach?

We use a hybrid of data agnostic and data dependent approaches because relying solely on either one leads to severe loss of accuracy in computing the queries. For example, if you use a data agnostic approach — which we refer to as *naive approach* — then to get the right level of granularity you will have to consider divisions of much smaller sizes. Now, to achieve differential privacy, it is necessary to add noise to the count in each division. Firstly, this will create a huge number of partitions. Secondly, though this gives a good estimate for the division, the answers to queries that consist of many divisions, for example, a state, are highly erroneous. In Section 7.6, we compare the naive approach with our proposed approach.

On the other hand, if you use quadtrees (or any other data dependent partitioning) approach alone, then you will have to build one quadtree over the whole of the USA. Now, to get the same level of granularity, the max height would have to be higher, which will lead to poor accuracy since at every level ℓ the value of ε_ℓ would now be much smaller.

We also note that one could use other approaches such as k-d trees to create better (i.e. well-balanced) partitions. But to create partitions such an approach uses data (e.g. to find median). So, one will have to allocate some privacy budget for this task as well; hence, in this case, compared to our approach, the values of all ε_ℓ's (except for the root) will have to be decreased, which will degrade the accuracy. In contrast, the quadtree partitions the quadrant without using any reported data, and hence, it does not incur any privacy loss in this step.

7.4 Temporal Partitioning

We saw in the previous section that the scale of the noise sampling distribution had to be increased by a factor of d — this indeed increases the magnitude of the noise and reduces the accuracy.

One way to obviate this problem is to build a new quadtree over each division every day by using data of that day. In this new approach, each report will be used to build only one tree, and we will get rid of the factor d. But this approach degrades the quality of spatial partitions created by the quadtree because the data per day varies and can be too little to create sufficiently granular partitions. Thus, we expect that the sweet spot lies somewhere in between one 1 and d, and here we will show how to find it.

In order to find the sweet spot, we first need a way to partition every d consecutive days into N groups of consecutive days, where $1 \leq N \leq d$ — we call this partitioning of days into groups *temporal partitioning*. Then, we can build an independent quadtree over the combined data of all the days in each group of the given temporal partition.

We stress that the partitioning here refers to dividing days into groups and is different from spatial partitions that we discussed earlier.

To save the privacy budget when building a quadtree over a group of days, we use the following strategy. For two given partitions, each of d consecutive days, if a group is the same across the two partition (i.e. the two groups consist of exactly the same days), then we only build the tree once, and reuse it without incurring any cost in terms of the privacy budget. Thus, we develop a novel partitioning scheme that gives reasonably small number of unique groups across all the partitions, and hence, improves the accuracy.

We define the problem of temporal partitioning as follows. Let us say we are given d and n such that $n \leq d$. For every day $t \geq d$, we want to partition the days, $t-(d-1), t-(d-2), \ldots, d$, into $N = \lceil d/n \rceil$ groups of consecutive days such that every day must be in exactly one group and: if n divides d, there are N groups of size n, otherwise $N-1$ groups of size n and one group of size $m = \text{MOD}(d, n)$, where MOD gives the remainder when d is divided by n. Furthermore, we want the partitions to be such that

every day must be present in at most n unique groups across all the partitions.

7.4.1 Notation and definitions

We now present some notation and definitions that we need to describe our algorithm for temporal partitioning.

Let $\mathbb{N}^+ = \{1, 2, 3, \dots\}$. We use contiguous intervals over $\mathbb{N}^+$ to denote the sets of consecutive days (i.e. groups of consecutive days). Thus, for every $t \in \mathbb{N}^+$ and $n \in \mathbb{N}$, we use $I_n(t) = \{t, t+1, \dots, t+n-1\}$, i.e. a contiguous interval that starts at t, ends at $t+n-1$, and has a length (or size) $n = |I_n(t)|$. For $n = 0$, we use the following convention: for every $t \in \mathbb{N}^+$, $I_n(t) = \emptyset$.

To give all the sets of d consecutive days that we want to partition, we use a sequence of intervals. Thus, for any given positive integer d, we use $I(d) = I^{(1)}, I^{(2)}, \dots$ to denote a sequence of intervals such that for every $i = 1, 2, \dots$, $I^{(i)} = I_d(i)$: we call $I(d)$ the *sliding interval-sequence* of length d. We use $\mathcal{P}$ to denote the set of all contiguous intervals, i.e. $\mathcal{P} = \{I_n(t) : \ t \in \mathbb{N}^+ \text{ and } n \in \mathbb{N}\}$.

We use $C \subseteq \mathcal{P}$ to denote a partition of any d consecutive days. For any $d, t \geq 1$, we say a set $C \subseteq \mathcal{P}$ is a *cover* of $I_d(t)$ if $I_d(t) = \mathcal{U}(C)$, where $\mathcal{U}(C) = \cup_{I \in C} \ I$.

Next, we define the important notion of cover-sequence, which gives all the covers for the sliding interval-sequence. Let us say we are given a sequence $\mathcal{C} = C_1, C_2, \dots$ such that $C_j \subseteq \mathcal{P}$ for every $j = 1, 2, \dots$. Then, for a given $d \geq 1$, we say $\mathcal{C} = C_1, C_2, \dots$ is a *cover-sequence* of $I(d) = I^{(1)}, I^{(2)}, \dots$ if for every $i = 1, 2, \dots$, C_i is a cover of $I^{(i)}$.

In our algorithm, NEXT-COVER (Algorithm 1), to generate a cover-sequence of the sliding interval-sequence of length d, we use the three operations: SHIFT, CIRCULAR-SHIFT, MOD, which we define below.

Let us say we are given a cover $C \in \mathcal{P}$ of N pairwise disjoint intervals such that $C = \{I_{n_1}(t - \sum_{l=1}^{N} n_l), I_{n_2}(t - \sum_{l=2}^{N} n_l), \dots, I_{n_N}(t - n_N)\}$, where $t > \sum_{l=1}^{N} n_l$ and $n_l \in \mathbb{N}^+$ for every $l = 1, 2, \dots, N$. Then, for input C, SHIFT increments each element of every interval in C by 1; and CIRCULAR-SHIFT rearranges the intervals in C by

performing a cyclic shift, that is, it outputs the following:

$$\{I_{n_2}(t - n_1 - \sum_{l=2}^{N} n_l), I_{n_3}(t - n_1 - \sum_{l=3}^{N} n_l), \ldots, I_{n_N}(t - n_1 - n_N), I_{n_1}(t - n_1)\}$$

For any given non-negative integers t, n, MOD(t, n) gives the remainder when t is divided by n.

7.4.2 Temporal partitioning algorithm

Our algorithm takes three parameters: (1) t, the day which we are building the partitions of the past d days, (2) C, the cover of the previous day, and (3) M, the period after which the algorithm performs CIRCULAR-SHIFT operation.

NEXT-COVER, takes C and shifts all the intervals in C by 1; furthermore, if it has been M days since the last CIRCULAR-SHIFT, it additionally performs the CIRCULAR-SHIFT operation. Below we give the definition of what it means to generate a cover-sequence using our NEXT-COVER algorithm.

Definition 7.1. *For a given cover C_1 and an integer $M \geq 1$, we say a cover-sequence, $\mathcal{C} = C_1, C_2, \ldots,$ is* generated by NEXT-COVER *(Algorithm 1) if for every $i > 1$, $C_i =$ NEXT-COVER$(C_{i-1}, d + i - 1, M)$.*

Note that if C_1 consists of contiguous and disjoint intervals, then all the covers, $C_2, C_3, \ldots,$ generated by NEXT-COVER, will consist of contiguous and pairwise disjoint intervals. Furthermore, for any given d and n such that $1 \leq n \leq d$, $M = n$ and an appropriate value of the first cover, the cover sequence, $\mathcal{C}$, generated by NEXT-COVER is such that for every $t \geq d$, $\mathcal{S}_t(\mathcal{C}) \leq n$ — here $\mathcal{S}_t(\mathcal{C}) = |\{J \in C_i : \ i \in \mathbb{N}^+ \text{ and } t \in C_i\}|$ (Theorem 7.1).

Theorem 7.1. *Arbitrarily choose integers d and n such that $d \geq n \geq 1$, and let $q = \lfloor d/n \rfloor$ and $m = MOD(d, n)$. Now, let $\mathcal{C} = C_1, C_2, \ldots$ be the cover-sequence generate by NEXT-COVER (Algorithm 1) for $C_1 = \{I_n(1), \ldots, I_n(1 + (q-1)n), I_m(1 + qn)\}$ and $M = n$. Then $\mathcal{C}$ is a cover-sequence of $I(d)$ such that for every $t \geq d$, $\mathcal{S}_t(\mathcal{C}) \leq n$.*

The proof of the theorem is provided in the last section.

ALGORITHM 1: NEXT-COVER

Input: C$\in \mathcal{P}$, $t \in \mathbb{N}^+$, and $M \in \mathbb{N}^+$

Output: C$'$ (new cover)

$d = |\mathcal{U}(C)|$ and T = C

if $MOD(t-d, M) = 0$ **then**
| T = CIRCULAR-SHIFT(C)

end

C$'$ = SHIFT(T)

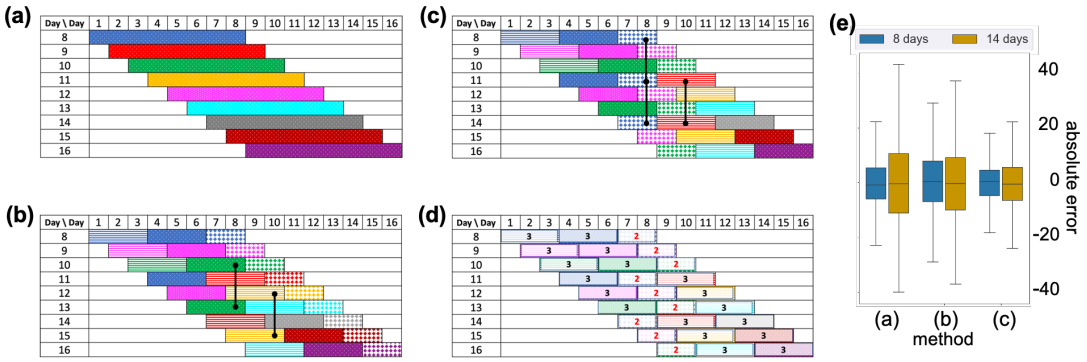

Figure 7.3: This figure shows different possible groupings of days into partitions used for building the quadtrees. In all the figures, both rows and columns show the progression of time, and for each day, data of past $d = 8$ days is aggregated to answer queries. The colored rectangles on each row show the partition built on the day given in the left-most column in the same row. Each rectangle shows the days that are grouped together, and it is uniquely identifiable by its color and pattern — therefore, a rectangle with the same color and pattern refers to the same group of days. **(a)** shows the naive approach that groups all 8 days into one group. **(b)** shows a simple partitioning with two groups of size $n = 3$ and one group of size $m = 2$. The vertical connections explicitly show the same group across different partitions. This has limited repeatability of the groups compared to our approach. **(c)** shows the partitioning via our approach where the groups of different sizes are ordered in a particular way. **(d)** explicitly shows the repeating pattern that would be obtained through a circular shift of the groups after every 3 days (i.e. $M = 3$). **(e)** shows the boxplot of the noise that Laplace mechanism adds (over 100 iterations) for each case ((a), (b), and (c)) for $\lambda = s$, where $s = d$ for (a), $s = n + m$ for (b), and $s = n$ for (c).

The temporal partitioning is an important step that helps to gain in utility, otherwise no single quadtree can be used to compute two SR queries at different days, see Figure 7.3(a) for an example of $d = 8$. Furthermore, by applying CIRCULAR-SHIFT

operation, we can reduce the number of unique groups containing any day compared to the case when we do not, for example, see Figure 7.3(b)-(c), where $n = 3$ and m =MOD(d, n). Figure 7.3(e) gives the boxplot of the noise Laplace mechanism will add in each of the case for $d = 8$ (for the example given in the figure) and $d = 14$, our focused case. Clearly, our approach introduce lesser noise compared to the other two.

7.4.3 How to choose n over time?

Finally, we give a heuristic to pick the value of n over time for the case at hand, i.e. $d = 14$. By choosing appropriate values of n over time further improves the accuracy, which is due to two reasons. First, when the infection rate is really low, there is too little data to create partitions; thus, no need to use privacy budget for this task. Second, the divisions that have larger size, also have higher value of the scale (for the noise sampling distribution) for each quadrant of the corresponding quadtree, leading to higher magnitude of noise. Thus, picking a higher value of n will reduce the accuracy.

We set $n = 1$ if the number of reports, $\#R$, from the division is less than 20 (we use differentially private values to compute $\#R$). When $\#R$ exceeds 20, we pick n based on $\#R$ and the max height of the quadtree, h, for the division. When $h \leq 4$ and $20 \leq \#R \leq 4^4$, we set $n = 7$. When $h = 6$ and $20 \leq \#R \leq 4^4$, we set $n = 3$. For all the remaining cases, we set $n = 2$.

To develop this heuristic, we used the data (of confirmed covid-19 cases described in Section 7.6) over five months (from March, 2019 to July, 2020). We picked four counties varying in their sizes, and hence, the maximum height of their quadtrees. We used this data to build the DP quadtree collection for each of the 14 consecutive days over the five months for each of the four counties. Then, for each county and day, we uniformly picked 900 queries (i.e. query regions), with constraint on the count (of the query region) such that: 300 had count between 50 and 150, another 300 with count between 151 and 250, and 300 with count 251 or greater. Lastly, we grouped days together based on their total count of the last 14 days for each day, and computed the average of the relative error for each group of days — Figure 7.4 plots this error for each county. We used these plots to devise our heuristics.

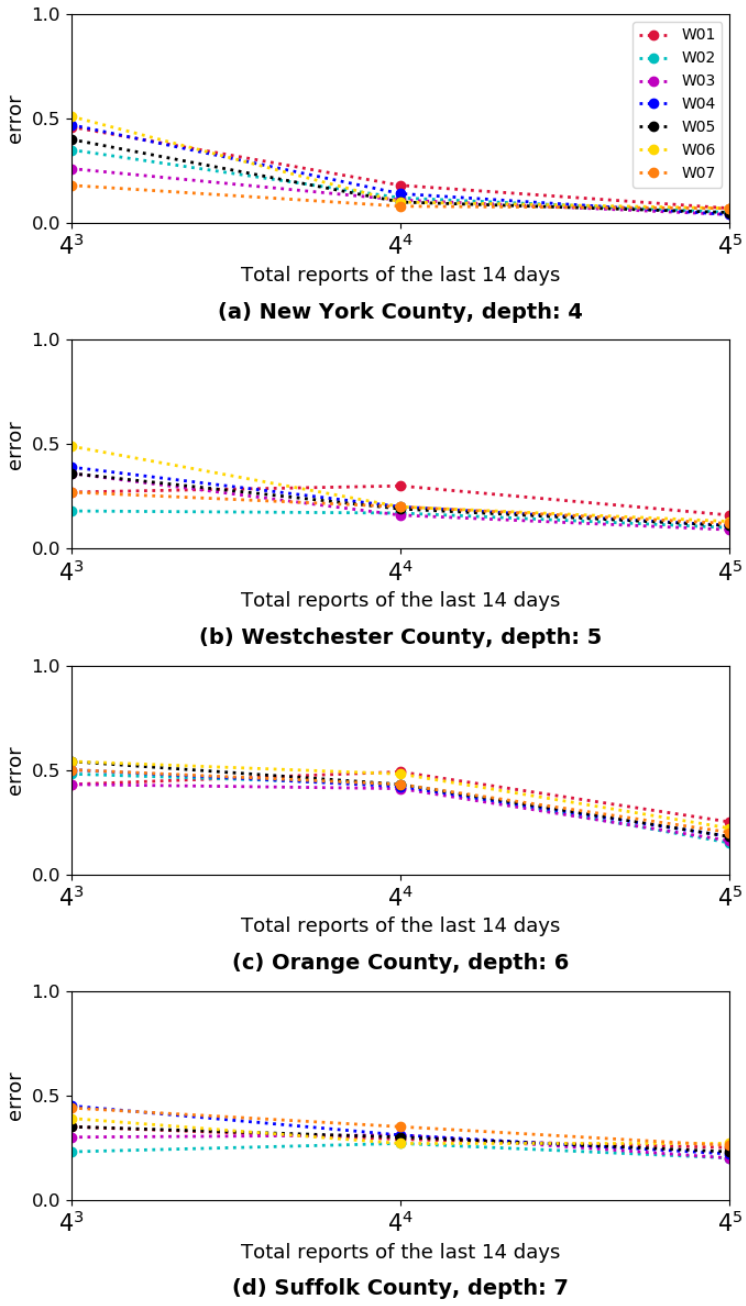

Figure 7.4: Average relative error for our approach over uniformly picked queries for each day over a five month period.

7.5 Query Computation

To compute an SR query, we find all the divisions that intersect with the query region, then compute the query over the all the quadtrees (identified by the group of days given by our algorithm for the past d days under consideration) for each such division and aggregate their outputs to compute the final answer.

To compute a query over a quadtree, we traverse the tree to find all the quadrants that intersect with the query region and sum their counts to compute the final result. In this step, we further improve the accuracy by using the count of the parent node if all of its children nodes are selected (see Figure 7.2(b)-(c)) — for details on how to compute a query over a quadtree see [86].

Since the differentially private quadtrees only store the count in each quadrant, we get the count for the whole quadrant even when the query region partially intersects with the quadrant (see the third query in Figure 7.2(c)). In such cases, we employ uniformity assumption (i.e. the data in a quadrant is distributed uniformly) to get a better estimate [86]. So, instead of the count of the quadrant, we use the count proportional to the area of the query region R that intersects with the region (R_Q) of a quadrant Q, i.e. $c_Q \times A(R \cap R_Q))/A(R_Q)$, where c_Q is the count for the quadrant Q, and A gives the area of the provided region.

However, there are many instances when the actual region from a division makes a small part of the region of a quadrant. This is because we build quadtrees over the bounding boxes of the divisions. For example, see the bounding box and the actual region given in Figure 7.2(a). Thus, in such cases, the area of the quadrant is much larger than the actual area of the region of the division it contains — this leads to a lower estimate of the count. We solve this problem by employing a polygonal (shape) approximations of divisions. We use the intersection of the polygon with a quadrant as the area of the quadrant, and the intersection of the query region with the polygon as the area of the query region in the quadrant.

7.6 Empirical Evaluation

In this section, we empirically evaluate of our approach to validate its effectiveness. For this evaluation, we used real-world data on confirmed covid-19 cases for the USA, which were given at the county level [87].

Data

The data we used gives the aggregate counts (of confirmed covid-19 cases) at the county level for each day. Therefore, we first disaggregated the data for each county for each day. To do this, we estimated the radius of each county, and used it to parameterize the scale of exponential distribution, which we used to sample the distance r from the center of the county for each point. Then, for each point at distance r, we picked a location uniformly on the circle of radius r, centered at the county's center coordinate, and assigned the data point this location.

Results

We use $\varepsilon = 6$, which is the same value that US Census Bureau used for 2020 census [88]. Below, Figures 7.5 and 7.6 plot the crucial results for our approach.

Our approach gives accurate cumulative counts over time as well as ranking to identify hotspots at the state level (Figures 7.5(a)-(c)) as well as at the county level and even within the county (Figures 7.6(a)-(b)).

The relative error for the cumulative counts in the start is higher (Figures 7.5(b)). This is because in the beginning of the pandemic the counts are very small for most of the states; that is why the average error is lower for the 25 states with the most cases as compared to the average error over all the states in the USA. We also computed the 14-days moving average of the new cases; for this, our approach incurs a very low error. Even in the case of moving average, if the count is sufficiently high, the error is negligible (Figures 7.5(d)-(e)).

We also compare our approach with the *naive* approach that only uses data agnostic partitioning (discussed in Section 7.3). The naive approach creates fixed partitions for

each state such that each of the partition is of size 1 km^2. For the naive approach, we perturbed the count in each partition using Laplace mechanism for scale 1/6. Thus, compared to our approach, the naive approach adds much less noise to the count of each partition. Yet, our approach outperform the naive one because the naive approach creates a huge number of partitions.

Overall, the empirical evaluation shows that our approach can effectively track covid-19 pandemic while preserving privacy of those who contribute their data.

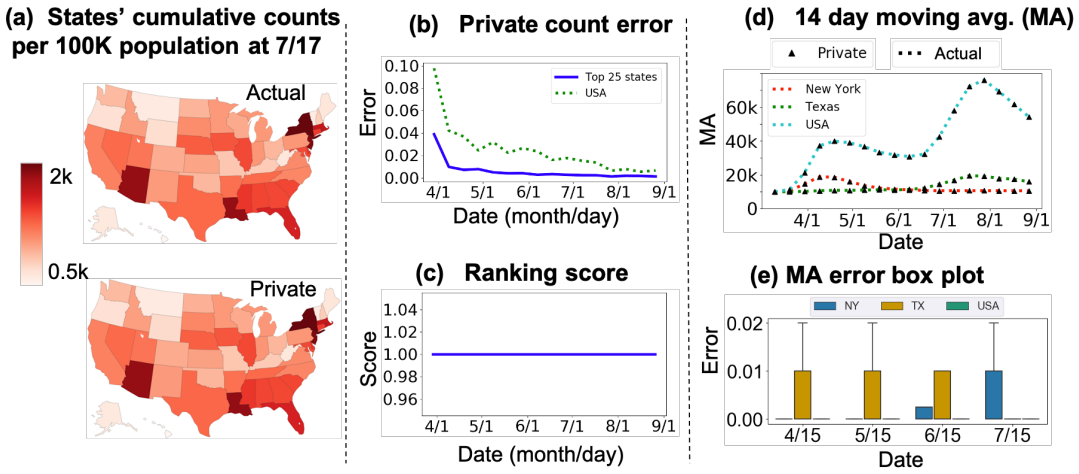

Figure 7.5: Private counts refer to the counts computed by our method from the actual covid-19 case count. **(a)** juxtaposes the heatmaps of the cumulative counts at the state level, both actual and private on 7/17/2020. **(b)** plots the average of the relative error in cumulative counts for the top 25 states (by case count) and the entire US over the period from 3/20/2020 – 9/1/2020. **(c)** plots the Kendall's τ (Kendall rank correlation coefficient [89]) of the two ranked lists of states obtained from the private and the actual counts — when two list have the same ordering, the score is 1 **(d)** plots the 14 day moving average of both the private and the actual counts for New York, Texas, and the entire US over the period from 3/20/2020 – 9/1/2020. Since our method is probabilistic, the private counts shown are the average over 100 iterations. **(e)** shows the boxplot of the relative error of the moving average over these 100 iterations.

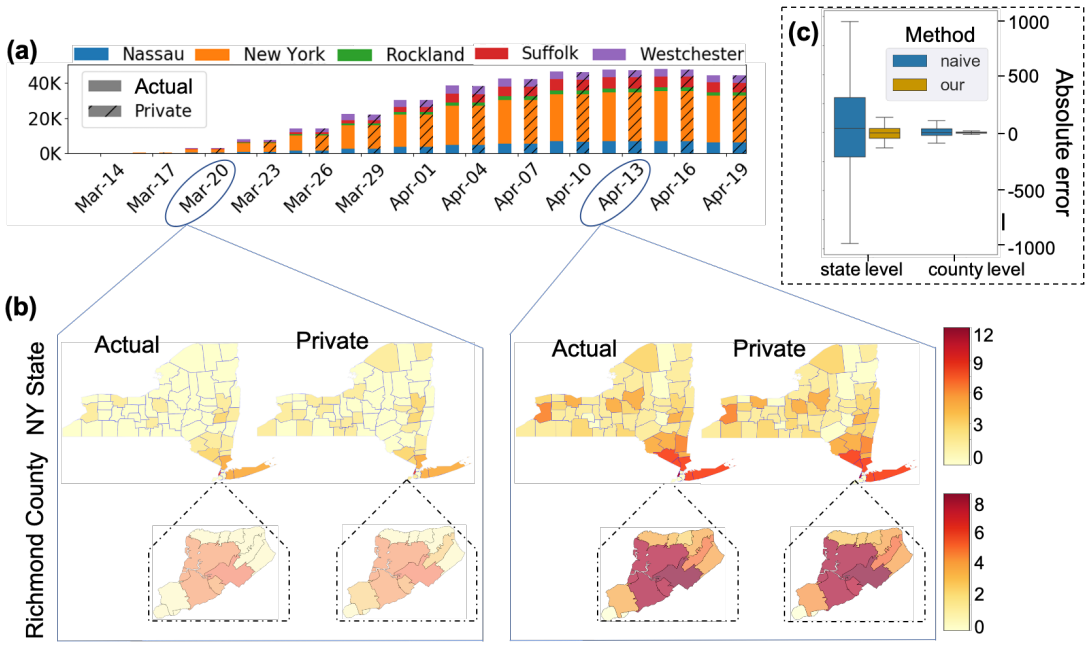

Figure 7.6: Private counts refer to the counts computed by our method from the actual covid-19 case count. **(a)** depicts the stacked barchart of the case counts of the 5 counties in NY State with the most covid-19 cases. For each day: (i) two stacked bars are given, the first for the actual counts, and the second for the private counts; and (ii) each bar gives the total covid-19 cases for the 14 day period leading upto that day. **(b)** juxtaposes the heatmaps of the actual and private 14-days case counts (on a log scale) for NY state and for Richmond county, on 3/20/2020 as well as on 04/13/2020. **(c)** compares our approach to the naive approach that also guarantees the same level of privacy in terms of their absolute error at the state and county level. Both the methods are probabilistic, and therefore the boxplots are computed over 100 iterations.

Proof of Theorem 7.1

Theorem (7.1 restated)**.** *Arbitrarily choose integers d and n such that $d \geq n \geq 1$, and let $q = \lfloor d/n \rfloor$ and $m = MOD(d,n)$. Now, let $\mathcal{C} = C_1, C_2, \ldots$ be the cover-sequence generate by NEXT-COVER (Algorithm 1) for $C_1 = \{I_n(1), \ldots, I_n(1+(q-1)n), I_m(1+qn)\}$ and $M = n$. Then $\mathcal{C}$ is a cover-sequence of $I(d)$ such that for every $t \geq d$, $\mathcal{S}_t(\mathcal{C}) \leq n$.*

To prove the theorem, we need to define some notation and note some observations about the cover-sequence generated by NEXT-COVER, which are provided below. We note that the main part of the proof focuses on the case when t is included in at least one interval of length $m \geq 1$ belonging to one of the covers given by $\mathcal{C}$.

Let d, n, m, and q be as given above, and assume that $m \geq 1$. In this case, for every i, $|C_i| = q + 1$; let $N = q + 1$. For every $t \in \mathbb{N}^+$, let $k_t = N - \text{MOD}(\lfloor (t-d)/M \rfloor, N)$ — recall $M = n$ in Theorem 7.1. Henceforth, we assume $M = n$. For any $t \geq d$, we define $C^{[k_t]}(t)$ to be $\{I^{(1)}, \ldots, I^{(N)}\}$ such that for every $l \neq k_t$, $|I^{(l)}| = n$ and $I^{(l)} = I_n(1+t-\sum_{r=l}^{N} |I^{(r)}|)$, and for k_t, $|I^{(k_t)}| = m$ and $I^{(k_t)} = I_m(1+t-m-(N-k_t)n)$. Note that for $C^{[k_t]}(t)$, k_t gives the position of the interval of length m (from the left) when all the intervals in $C^{[k_t]}(t)$ are placed on the number line.

Now, note that for C_1 given in Theorem 7.1, the cover-sequence, $\mathcal{C} = C_1, C_2, \ldots$, generated by NEXT-COVER for C_1 and $M = n$, is such that not only $C_1 = C^{[N]}(d)$, but the similar relationship holds for all the covers (Fact 7.1). Furthermore, for every $i \geq 1$, C_{i+nN} is the same cover as C_i with every interval in C_i shifted to right by nN because for $t = d+i-1$, $k_t = k_{t+nN}$ — the intuition provided by this observation plays an important role in our proofs.

Fact 7.1. *For every $i \geq 1$ and $t = d + i - 1$, $C_i = C^{[k_t]}(t)$.*

For the above setting (also given in Theorem 7.1), every interval of length m (which belongs to any of the cover given by $\mathcal{C}$) is present in N distinct covers in $\mathcal{C}$ (Observation 7.1).

Observation 7.1. *For every $i \geq 1$ and $t \geq d$, if $I_m(t) \in C_i$, then $I_m(t)$ belongs to N distinct covers, namely, $I_m(t) \in \cap_{a=0}^{N-1} C_{j+an}$ for some $j \geq 1$.*

To confirm the above observation, arbitrarily fix an integer $b \geq 0$, and let $t_s = d + bnN$. Now, for $t = t_s, \ldots, (t_s + nN - 1)$, the interval $I_m(1+t-m-(N-k_t)n) \in C^{[k_t]}(t)$. Finally, note that for $t_s \leq t < t_s + (n-1)$, $k_t = N$, and for every a such that $1 \leq a < N$,

$$I_m(1 + t - m - (N - k_t)n) = I_m(1 + t + an - m - (N - k_{t+an})n) \in C^{[k_{t+an}]}(t + an)$$

The above relation holds because $k_{t+an} = N - a$ for $1 \leq a < N$.

Observation 7.1 also implies that for any $t \geq d$, all the covers that contain t consist of at most n unique intervals of length m. This holds because for any given t, there are at most d covers in $\mathcal{C}$ that can contain t, each of which contains only one interval of length m, and each of these interval is present in N distinct covers; therefore, $d/N =$

$(qn+m)/N \leq n$. We will use this result to confirm our next observation that gives us the values of t that can be contained in an interval of length m in the cover-sequence $\mathcal{C}$.

Observation 7.2. *For every $t \geq d$ such that t is contained in an interval of length m given by the $\mathcal{C}$, there exists an integer $b \geq 0$ such that either $d+bnN \leq t < d+(bN+1)n$ or $d+(b+1)nN-m < t < d+(b+1)nN$.*

To confirm the above observation, arbitrarily fix t satisfying the condition given above. Note that there exists a non-negative integer b such that $d+bnN \leq t < d+(b+1)nN$ — let us fix this b. Let $t_s = d+bnN$, and note that for $t' = t_s, t_s+1, \ldots, t_s+n-1$, $C_{t'-d+1} = C^{[N]}(t')$, and $I_m(t'-m+1) \in C^{[N]}(t')$ — these intervals are the n unique intervals of length m in the covers $C_{t_s-d+1}, \ldots, C_{t_s+nN-d}$ (an implication of Observation 7.1 discussed above). Furthermore, for $t' = t_s+n-1$, t' belongs to $I_m(t'-m+1)$; furthermore, this interval contains the maximum value of $l \in \mathbb{N}^+$ compared to all n uniquer intervals of length m. Since $t'+1$ does not belong to $I_m(t'-m+1)$, we conclude that if t belongs to any of the intervals given here, then $d+bnN \leq t < d+(bN+1)n$.

The next interval of length m, which is not present in any of the previous covers, is at $t' = d+(b+1)nN$, and here, $I_m(t'-m+1) \in C_{t'-d+1} = C^{[N]}(t')$. Thus, thus there is no interval of length m that can contain t when $d+(bN+1)n \leq t \leq d+(b+1)nN-m$. Now, from a similar argument as for the case above, the claim in Observation 7.2 follows.

Observation 7.3. *For every $t_1, t_2 \geq d$ such that $t_1 \neq t_2$, if there exists t such that $I_m(t) \in C^{[k_{t_1}]}(t_1) \cap C^{[k_{t_2}]}(t_2)$, then it follows that:*

(1) $I_n(t-n) \in C^{[k_{t_1}]}(t_1) \cap C^{[k_{t_2}]}(t_2)$ *if* $k_{t_1}, k_{t_2} > 1$,

(2) $I_n(t+m) \in C^{[k_{t_1}]}(t_1) \cap C^{[k_{t_2}]}(t_2)$ *if* $k_{t_1}, k_{t_2} < N$.

The above claim follows from the fact that every covers in the cover-sequence, $\mathcal{C}$, contains one intervals of length m and at least one interval of length n — recall that we assumed that $1 \leq m < n$.

Proof of Theorem 7.1. Let q, m, and $\mathcal{C}$ be as described in Theorem 7.1 for arbitrary values of d and n such that $d \geq n \geq 1$.

When n divides d, C_1 will have an empty interval, i.e. the interval with length $m = 0$. In this case, we will disregard this interval and assume that C_1 only consists of d/n intervals, each of length n. Note that this does not affect the correctness of our arguments because the intervals we discard is an empty set; furthermore, this assumption will not be applicable when $m \geq 1$.

We divide our argument into two parts: the first one establishes that $\mathcal{C}$ is a cover-sequence of $I(d)$, and the second shows that $\mathcal{S}_t(\mathcal{C}) \leq n$ for every $t \geq d$.

Part 1, $\mathcal{C}$ is a cover-sequence of $I(d)$:

We use an inductive argument to show that $\mathcal{C} = C_1, C_2, C_3, \ldots$ is a cover-sequence of $I(d) = I^{(1)}, I^{(2)}, I^{(3)}, \ldots$, that is, for every $t \geq d$ and $i = t - d + 1$, $I^{(i)} = I_d(i) = \mathcal{U}(C_i)$. Note that $I_d(1) = \mathcal{U}(C_1)$ and all the intervals in C_1 are contiguous and pairwise disjoint (base case). Assume $I_d(i) = \mathcal{U}(C_i)$ for some $i \geq 1$ and all the intervals in C_i are contiguous and pairwise disjoint (inductive hypothesis).

Because C_i is a cover of $I_d(i)$ and all the intervals in it are contiguous and pairwise disjoint, it follows that $\mathcal{U}(\text{SHIFT}(C_i)) = \mathcal{U}(\text{SHIFT}(\text{CIRCULAR-SHIFT}(C_i)))$. Now, from the above we get: $I_d(i+1) = \mathcal{U}(\text{SHIFT}(\{I_d(i)\})) = \mathcal{U}(\text{SHIFT}(C_i)) = \mathcal{U}(C_{i+1})$. From here, the claim follows.

Part 2, $\mathcal{S}_t(\mathcal{C}) \leq n$:

Now we prove that for every $t \geq d$, $\mathcal{S}_t(\mathcal{C}) \leq n$ (recall that $\mathcal{S}_t(\mathcal{C}) = |\{J \in C_i : \ i \in \mathbb{N}^+ \text{ and } t \in C_i\}|$). Fix an arbitrary value of $t \geq d$.

First, consider the case, when all intervals in the cover-sequence containing t are of length n. That is, for every C_i that contains t, i.e. $t \in J \in C_i$, $|J| = n$. In this case, $\mathcal{S}_t(\mathcal{C}) \leq n$ holds because there are only n unique contiguous intervals of length n that can contain t, namely, $I_n(t-n+1), \ldots, I_n(t)$. This also covers the case when n divides d; thus, for rest of the proof, we assume that $1 \leq m < n$ — thus, all the observations we made above are applicable here.

Now, we consider the case, where t belongs to at least one interval of length m in one of the covers: that is, for some C_i, $t \in J \in C_i$ and $|J| = m$. Let there be ℓ unique intervals of length m that contain t — note that ℓ cannot be more than m. This gives use two cases for t (Observation 7.2). Let b be an integer such that $d+bnN \leq t < (b+1)nN$. Let $t_r = t + r$ for $r = 0, 1, 2, \ldots$ (recall that t was arbitrarily picked).

Case 1 $\left[d + bnN \leq t < d + (bN+1)n\right]$**:** Firstly, note that for every t' such that $d + bnN \leq t' < d + bnN + n$, $I_m(t' - m + 1) \in C_{t'-d+1} = C^{[N]}(t')$, all of these intervals are different from each other. Next, note that when $d + bnN \leq t \leq d + (bN+1)n - m$, $\ell = m$, otherwise $\ell < m$. This holds because for $t_s = d + anN$, the intervals of length m in the covers $C^{[k_{t_s}=N]}(t_s), C^{[k_{t_s+1}=N]}(t_s + 1), \ldots, C^{[k_{t_s+(n-1)}=N]}(t_s + (n-1))$ are different, where each interval in the next cover is obtained by shifting a corresponding interval in the previous cover by 1.

Case 1.a $[d + bnN \leq t \leq d + (bN+1)n - m$ **and** $\ell = m]$**:**

For $\ell \leq r \leq n-1$, $t \in I_n(t_r - n - \alpha m + 1) \in C_{t_r-d+1} = C^{[k_{t_r}]}(t_r)$ for $\alpha = 0$ or $\alpha = 1$; and each of these $n - \ell$ distinct intervals of length n is present in $N - 1$ distinct covers. This holds because, here, k_{t_r} is either N (giving $\alpha = 1$) or $N-1$ (giving $\alpha = 0$), and for both of these cases, each $I_n(t_r - n - \alpha m + 1)$ is present in $N - 1$ distinct covers. This follows from Observation 7.1 (i.e. each interval of length m is present in N distinct covers) and Observation 7.3.

Hence, for this sub-case (i.e. $\ell = m$), we conclude that $\mathcal{S}_t(\mathcal{C}) \leq n$. Because t is contained in m unique intervals of length m, each of which is present in N distinct covers, as well as $n - m$ unique intervals of length n, each of which is present in $N - 1$ distinct cover, and thus, $mN + (n - m)(N - 1) = d$.

Case 1.b $[d + (bN+1)n - m < t < d + (bN+1)n$ **and** $1 \leq \ell < m]$**:**

If we let $t_s = d+bnN$, then we get $t = t_s+n-\ell$. This holds because for $t = t_s+n-m$, there are m intervals of length m, which contain t, and each of the n distinct intervals of length m, namely, $I_m(t_s - m + 1), \ldots, I_m(t_s - m + n)$ (which respectively belong to covers $C^{[N]}(t_s), \ldots, C^{[N]}(t_s + n - 1)$), can be obtained by shifting the preceding interval by 1 (except for the first one).

Next, we show that in this sub-case, there are additional covers that contain t, and were not considered in Case 1.a. For $w \in \mathbb{N}$, we let $t'_w = t_s + nN + w$ and $\hat{t}_w = t_s + 2n - m + w$. Now, for $t = t_s + n - \ell$ and every $w = 0, 1, \ldots, m - \ell - 1$, we have that $t \in I_n(t'_w - d + 1) = I_n(\hat{t}_w - n + 1)$, where $I_n(t'_w - d + 1) \in C^{[k_{t'_w}=N]}(t'_w)$ and $I_n(\hat{t}_w - n + 1) \in C^{[k_{\hat{t}_w}=N-1]}(\hat{t}_w)$. However, we have considered the covers corresponding to $\hat{t}_w$ (i.e. $C^{[k_{\hat{t}_w}=N-1]}(\hat{t}_w)$) in Case 1.a. Thus, we conclude that in this sub-case as well, we have $n - \ell$ unique intervals of length n that contain t. This completes the claim for Case 1.

Case 2 $\big[d + (1+b)nN - m + 1 \leq t \leq d + (1+b)nN - 1\big]$**:** Here, $1 \leq \ell \leq m - 1$ because for $t' = d + (1+b)nN$, $C_{t'-d+1} = C^{[N]}(t')$ is the first cover (in the cover-sequence) that contains J such that $|J| = m$ and $t \in J$, and the interval $J \in C^{[N]}(t' + (m-1))$ that contains t has length n.

Next, note that $t \in I_n(t_r - n + 1) \in C_{t_r-d+1} = C^{[1]}(t_r)$ for $0 \leq r \leq m - \ell - 1$. Furthermore, $C_{t_r+m-d+1} = C^{[N]}(t_r + m)$ and $I_n((t_r + m) - n - m + 1) \in C^{[N]}(t_r + m)$. Now, using Observations 7.1 and 7.3, a similar argument as in Case 1, shows that there are at most n unique intervals that contain t, i.e., $\mathcal{S}_t(\mathcal{C}) \leq n$.

Since we picked t arbitrarily, the claim hods for every t. This completes the proof.

□

CHAPTER 8

Conclusion and Future Directions

We began with the question, "Is it possible to accurately analyze data for outliers while protecting privacy of the people whose data we analyze?", which this dissertation answers in the affirmative. It is the first to lay out the foundations of privacy-preserving outlier analysis. In it, we showed that practically-relevant settings for outlier analysis reduce to two settings: one is suitable for differential privacy, and the other is not. For this "other" setting, we conceptualized the novel notion of sensitive privacy, which makes it possible to answer outlier queries accurately and privately. We then developed constructions to give sensitively private mechanisms to identify outliers — which have very high accuracy in practice. We also proposed a framework to distinguish between the two settings for outlier analysis (described above). This framework is useful for data analysts who can use it to identify the right problem-specification for their applications, e.g., whether they should use differential privacy or sensitive privacy for their application and setting. The theoretical analysis as well as an extensive empirical evaluation of our approach strongly support its effectiveness.

8.1 Future Work

This dissertation opens new directions in data privacy research for outlier analysis. Below, we provide an account of important and practically relevant future directions for further research.

8.1.1 Characterization of privacy-utility trade offs

We want to completely analyze the privacy-utility trade offs for private mechanisms for boolean queries (recall that anomaly identification is one such query). For instance, how can one characterize the utility of private mechanisms for boolean queries, and how can one build the optimal mechanism under differential privacy and sensitive privacy (we only looked at the pareto optimality).

We showed that for existence-dependent outlier queries, the utility achievable by differentially private mechanism is restrictive. We established an upper bound on the utility of any differentially private mechanism for boolean outlier queries, e.g. anomaly identification, a fundamental query in outlier analysis. However, the corresponding non-trivial bounds for other outlier analysis tasks such as computing the number of outliers or all the outliers in data are still unknown. Deriving these bounds is essential to completely understand and solve the problem of private outlier analysis. Furthermore, deriving similar bounds on the utility of sensitively private mechanisms is another important research problem to improve our understanding of private outlier analysis.

Sensitive neighborhood graph is one of the fundamental objects in conceptualizing sensitive privacy. It plays a pivotal role in developing our constructions to achieve sensitive privacy. We want to further explore and analyze its properties and its relation to the accuracy/performance of sensitively private mechanisms, for example, how its connectivity relates to the sensitive privacy guarantee and accuracy in anomaly identification and outlier detection.

We used minimum discrepant distance (mdd) function over a neighborhood graph to characterize the utility of sensitively private mechanisms for boolean queries. Since sensitive privacy generalizes differential privacy, the utility characterization is also applicable to differential privacy. However, as discussed in Section 4.2, analyzing and computing mdd-function over neighborhood graph (which are over the set of databases) is a non-trivial task due to the complexity of the neighborhood graphs. It is possible to reduce this complexity by considering mdd-functions for graphs over a space (R) that is intermediate between the output space (i.e. $\{0, 1\}$) of the query f and the

database space ($\mathcal{D}$), where for query $f : \mathcal{D} \to \{0, 1\}$, we use functions $h : \mathcal{D} \to R$ and $w : R \to \{0, 1\}$ such that for every $x \in \mathcal{D}$, $w(h(x)) = f(x)$. Analyzing mdd-function in this intermediate space will help in developing high-utility mechanisms for differential privacy as well as sensitive privacy. Thus, it is an important research direction to understand the privacy-utility trade off for boolean queries.

8.1.2 Private mechanism constructions

We presented three general construction to develop privacy-preserving mechanisms for outlier analysis. We used n-Step lookahead mechanism, which works for non-regular normality properties, to develop a mechanism to analyze $(\beta, r = 0)$-anomalies. Furthermore, we developed Construction 4.1 and Construction 4.2 to give sensitively private mechanisms for anomaly identification or boolean queries in general. We established the feasibility and practicality of sensitive privacy for anomaly identification in the real-world by instantiating our technical developments for (β, r)-anomaly definition. We will expand this analysis by developing mechanisms for other anomaly definitions such as (β, r)-anomaly variants, Local outlier factor, AVF based outliers, and neural network based outlier models.

To increase the practical applicability of our constructions, we also plan to develop a framework to give and compute non-trivial lower bounds for mdd-function over a sensitive neighborhood graph for a class of distance-based and density-based outlier definitions. This framework will also be useful in preserving utility and privacy in computing boolean queries in general.

The privacy-preserving mechanism based on a smooth upper bound on local sensitivity[1] [72] can help achieve sensitive privacy for outlier analysis. This is due to the fact that neighboring databases in sensitive privacy are data-defined and are restricted to a subset of neighboring databases considered under differential privacy. However, to develop a sensitively private mechanism via this approach, we need to extend the technical developments (for computing these bounds) from DP neighborhood graphs to

[1]Under bounded differential privacy, local sensitivity of a query $f : \mathcal{D} \to \mathbb{R}$, for any $x \in \mathcal{D}$, is given as $LS_f(x) = \max_{y:H(x,y)} |f(x) - f(y)|$, where H gives the hamming distance between two databases.

sensitive neighborhood graphs and develop methods to compute smooth upper bounds on local sensitivity under sensitive privacy. We plan to pursue this research direction in the future.

8.1.3 Temporal partitioning of data

In Chapter 7, we presented an algorithm to partition temporal data to boost accuracy by increasing the reuse of already computed differentially private results. In particular, we considered the problem of partitioning each interval in the sliding interval-sequence by a cover-sequence consisting of contiguous and pairwise disjoint intervals. This approach has useful applications in differentially private streaming computation as well as sliding-window based algorithms. In this context, an important question is of optimality, i.e. how to generate a cover-sequence of a sliding interval-sequence that minimizes the number of unique intervals in the cover-sequence that contain any fixed element from sliding interval-sequence. Furthermore, here, we only considered covers with contiguous intervals, however, we can build better algorithms by generating covers that consist of non-contiguous intervals, where 'better' means that the number of unique intervals (mentioned above) can be further reduced. We plan to analyze this problem and explicate methods to solve it optimally and practically.

References

[1] Shebuti Rayana. ODDS library, 2016. Available at http://odds.cs.stonybrook.edu.

[2] Martin Bobrow. Balancing privacy with public benefit. *Nature News*, 500(7461):123, 2013.

[3] Yaniv Erlich and Arvind Narayanan. Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics*, 15(6):409, 2014.

[4] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, May 2016.

[5] Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/, 1996.

[6] Charu C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[7] Seppo Karrila, Julian Hock Ean Lee, and Greg Tucker-Kellogg. A comparison of methods for data-driven cancer outlier discovery, and an application scheme to semisupervised predictive biomarker discovery. *Cancer informatics*, 10:CIN–S6868, 2011.

[8] Soumi Ray, Dustin S McEvoy, Skye Aaron, Thu-Trang Hickman, and Adam Wright. Using statistical anomaly detection models to find clinical decision support malfunctions. *Journal of the American Medical Informatics Association*, 2018.

[9] Gordon D Schiff, Lynn A Volk, Mayya Volodarskaya, Deborah H Williams, Lake Walsh, Sara G Myers, David W Bates, and Ronen Rozenblum. Screening for medication errors using an outlier detection system. *Journal of the American Medical Informatics Association*, 24(2):281–287, 2017.

[10] Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. John Wiley and Sons, 3rd edition, 1994.

[11] Alison M Darcy, Alan K Louie, and Laura Weiss Roberts. Machine learning and the profession of medicine. *Jama*, 315(6):551–552, 2016.

[12] Ziad Obermeyer and Ezekiel J Emanuel. Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216, 2016.

[13] Machine Learning Group. Credit card fraud detection, Mar 2018.

[14] Latanya Sweeney. Uniqueness of simple demographics in the us population. lidap-wp4. laboratory for international data privacy, 2000.

[15] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 77–80. ACM, 2006.

[16] Jaideep Vaidya, Basit Shafiq, Xiaoqian Jiang, and Lucila Ohno-Machado. Identifying inference attacks against healthcare data repositories. *AMIA Summits on Translational Science Proceedings*, 2013:262, 2013.

[17] Melissa Gymrek, Amy L McGuire, David Golan, Eran Halperin, and Yaniv Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.

[18] D Luquetti, P Claes, DK Liberton, K Daniels, KM Rosana, EE Quillen, LN Pearson, B McEvoy, M Bauchet, AA Zaidi, et al. Modeling 3d facial shape from dna. *PLoS Genetics*, 10(3):e1004224, 2014.

[19] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[20] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[21] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 650–669. IEEE, 2015.

[22] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.

[23] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[24] Hafiz Asif, Periklis A. Papakonstantinou, and Jaideep Vaidya. How to accurately and privately identify anomalies. In *SIGSAC CCS*. ACM, 2019.

[25] Hafiz Asif, Periklis A Papakonstantinou, and Jaideep Vaidya. A guide for private outlier analysis. *IEEE Letters of the Computer Society*, 3(1):29–33, 2020.

[26] Michael Kearns, Aaron Roth, Zhiwei Steven Wu, and Grigory Yaroslavtsev. Private algorithms for the protected in social network search. *Proceedings of the National Academy of Sciences*, 113(4):913–918, 2016.

[27] Stelios Doudalis, Ios Kotsogiannis, Samuel Haney, Ashwin Machanavajjhala, and Sharad Mehrotra. One-sided differential privacy. *arXiv preprint arXiv:1712.05888*, 2017.

[28] Daniel M Bittner, Anand D Sarwate, and Rebecca N Wright. Using noisy binary search for differentially private anomaly detection. In *International Symposium on Cyber Security Cryptography and Machine Learning*, pages 20–37. Springer, 2018.

[29] Edward Lui and Rafael Pass. Outlier privacy. In *TCC*, pages 277–305. Springer, 2015.

[30] S. U. Nabar, K. Kenthapadi, N. Mishra, and R. Motwani. A survey of query auditing techniques for data privacy. In *In Privacy-Preserving Data Mining: Models and Algorithms, Springer, 2008 (to appear)*.

[31] Liyue Fan and Li Xiong. Differentially private anomaly detection with a case study on epidemic outbreak detection. In *2013 IEEE 13th ICDM Workshops*, pages 833–840. IEEE, 2013.

[32] R. J. Beckman and R. D. Cook. Outlier.....s. *Technometrics*, 25(2):119–149, 1983.

[33] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[34] Daniel Bernoulli and CG Allen. The most probable choice between several discrepant observations and the formation therefrom of the most likely induction. *Biometrika*, 48(1-2):3–18, 1961.

[35] F.Y. Edgeworth M.A. Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(143):364–375, 1887.

[36] Frank E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[37] F. J. Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, 1960.

[38] J. A. Hartigan. Note on discordant observations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(3):545–550, 1968.

[39] Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Wei, Sang-Hee Lee, and John Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14(1):99–129, 2007.

[40] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proc. of the Conf. on Very Large DataBases (VLDB)*, 2002.

[41] Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, 2000.

[42] Mon-Fong Jiang, Shian-Shyong Tseng, and Chih-Ming Su. Two-phase clustering process for outliers detection. *Pattern recognition letters*, 22(6-7):691–700, 2001.

[43] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, California Univ Berkeley, 1951.

[44] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD*, volume 29, pages 93–104. ACM, 2000.

[45] Anna Koufakou and Michael Georgiopoulos. A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. *Data Min. Knowl. Discov.*, 20(2):259–289, March 2010.

[46] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[47] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD*, pages 193–204. ACM, 2011.

[48] Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):3, 2014.

[49] Xi He, Ashwin Machanavajjhala, and Bolin Ding. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the 2014 ACM SIGMOD*, pages 1447–1458. ACM, 2014.

[50] Damien Desfontaines and Balázs Pejó. Sok: Differential privacies. *Proceedings on Privacy Enhancing Technologies*, 2020(2):288–313, 2020.

[51] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31st International Conference on Data Engineering (ICDE)*, pages 1023–1034. IEEE, 2015.

[52] Rina Okada, Kazuto Fukuchi, and Jun Sakuma. Differentially private analysis of outliers. In *ECML PKDD*, pages 458–473. Springer, 2015.

[53] Jonas Böhler, Daniel Bernau, and Florian Kerschbaum. Privacy-preserving outlier detection for data streams. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 225–238. Springer, 2017.

[54] Hafiz Asif, Tanay Talukdar, Jaideep Vaidya, Basit Shafiq, and Nabil Adam. Collaborative differentially private outlier detection for categorical data. In *IEEE CIC*, pages 92–101. IEEE, 2016.

[55] Edwin M Knorr and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the 1998 VLDB*, pages 392–403. Citeseer, 1998.

[56] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[57] Ying Yang, Geoffrey I Webb, and Xindong Wu. Discretization methods. In *Data mining and knowledge discovery handbook*, pages 101–116. Springer, 2009.

[58] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 82–102. Springer, 2013.

[59] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[60] Charu C Aggarwal. Outlier analysis. In *Data mining*, pages 237–263. Springer, 2015.

[61] Vic Barnett and Toby Lewis. *Outliers in statistical data*. Wiley, 2000.

[62] Daniel Kifer and Bing-Rong Lin. An axiomatic view of statistical privacy and utility. *Journal of Privacy and Confidentiality*, 4(1):5–49, 2012.

[63] Samuel Haney, Ashwin Machanavajjhala, and Bolin Ding. Design of policy-aware differentially private algorithms. *Proceedings of the 2015 VLDB Endowment*, 9(4):264–275, 2015.

[64] Liyue Fan and Li Xiong. Differentially private anomaly detection with a case study on epidemic outbreak detection. In *2013 IEEE 13th ICDM Workshops*, pages 833–840. IEEE, 2013.

[65] Hafiz Asif, Tanay Talukdar, Jaideep Vaidya, Basit Shafiq, and Nabil Adam. Differentially private outlier detection in a collaborative environment. *IJCIS*, 27(03):1850005, 2018.

[66] Andrew C. Yao. Protocols for secure computation. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[67] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218–229. ACM, 1987.

[68] Joseph I Choi and Kevin RB Butler. Secure multiparty computation and trusted hardware: Examining adoption challenges and opportunities. *Security and Communication Networks*, 2019, 2019.

[69] Anna Koufakou, Enrique G Ortiz, Michael Georgiopoulos, Georgios C Anagnostopoulos, and Kenneth M Reynolds. A scalable and efficient outlier detection strategy for categorical data. In *ICTAI 2007*, volume 2, pages 210–217. IEEE, 2007.

[70] Justin Hsu Marco Gaboardi Andreas Haeberlen and Sanjeev Khanna. Differential privacy: An economic method for choosing epsilon. 2014.

[71] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410. IEEE, 2014.

[72] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84. ACM, 2007.

[73] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *PODS*, pages 123–134. ACM, 2010.

[74] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 159–166. IEEE, 2015.

[75] Yihe Dong, Samuel B Hopkins, and Jerry Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. *arXiv preprint arXiv:1906.11366*, 2019.

[76] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[77] Ye Nan, Kian Ming Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing f-measure: A tale of two approaches. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 1, 06 2012.

[78] Listings of who's response to covid-19. Available at https://www.who.int/news-room/detail/29-06-2020-covidtimeline.

[79] The covid-19 testing debacle, Jun 2020. Available at https://www.nature.com/articles/s41587-020-0575-3.

[80] Nsf sponsored initiative by rutgers university. Available at https://covidnearby.org.

[81] Cristina Menni, Ana M Valdes, Maxim B Freidin, Carole H Sudre, Long H Nguyen, David A Drew, Sajaysurya Ganesh, Thomas Varsavsky, M Jorge Cardoso, Julia S El-Sayed Moustafa, et al. Real-time tracking of self-reported symptoms to predict potential covid-19. *Nature medicine*, pages 1–4, 2020.

[82] David A Drew, Long H Nguyen, Claire J Steves, Cristina Menni, Maxim Freydin, Thomas Varsavsky, Carole H Sudre, M Jorge Cardoso, Sebastien Ourselin, Jonathan Wolf, et al. Rapid implementation of mobile technology for real-time epidemiology of covid-19. *Science*, 2020.

[83] Tanusree Sharma and Masooda Bashir. Use of apps in the covid-19 response and the loss of privacy protection. *Nature Medicine*, pages 1–2, 2020.

[84] Apps and covid-19. Available at https://privacyinternational.org/examples/apps-and-covid-19.

[85] Derek Hylton Maling. *Coordinate systems and map projections*. Elsevier, 2013.

[86] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. Differentially private spatial decompositions. In *2012 IEEE 28th International Conference on Data Engineering*, pages 20–31. IEEE, 2012.

[87] Dong E, Du H, and Gardner L. An interactive web-based dashboard to track covid-19 in real time. lancet inf dis. 20(5):533-534. Available at doi:10.1016/S1473-3099(20)30120-1.

[88] US Census Bureau. Memorandum 2019.25: Design parameters and global privacy-loss budget, Oct 2019. Available at https://www.census.gov/programs-surveys/decennial-census/2020-census/planning-management/memo-series/2020-memo-2019_25.html.

[89] Guy Lebanon and John Lafferty. Cranking: Combining rankings using conditional probability models on permutations. In *ICML*, volume 2, pages 363–370. Citeseer, 2002.