Differential Privacy: An Exploration of the Privacy-Utility Landscape

BY Darakhshan J. Mir

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Rebecca N. Wright

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2013

© 2013

Darakhshan J. Mir

ALL RIGHTS RESERVED

ABSTRACT OF THE DISSERTATION

Differential Privacy: An Exploration of the Privacy-Utility Landscape

by Darakhshan J. Mir

Dissertation Director: Rebecca N. Wright

Facilitating use of sensitive data for research or commercial purposes, in a manner that preserves the privacy of participating entities, is an active area of study. Differential privacy is a popular, relatively recent, framework that formalizes data privacy. In this dissertation, I examine the often conflicting goals of privacy and utility within the framework of differential privacy. The contributions of this dissertation fall into two main categories:

1) We propose differentially private algorithms for several tasks that could potentially involve sensitive data, such as synthetic graph modeling, human mobility modeling using cellular phone data, regression, and computing statistics on online data.

We study the tradeoff between privacy and utility for these analyses—theoretically in some cases, and experimentally in others. We show that for each of these tasks, both privacy and utility can be successfully achieved by considering a meaningful tradeoff between the two.

2) We also examine connections between information theory and differential privacy, demonstrating how differential privacy arises out of a tradeoff between information leakage and utility. We show that differentially private mechanisms arise out of minimizing the information leakage (measured using mutual information) under the constraint of achieving a given level of utility. Further, we establish a connection between differentially private learning and PAC-Bayesian bounds.

Preface

Portions of this dissertation are based on work previously published or submitted for publication by the author. Chapter 3 is joint work with Rebecca N. Wright [Mir and Wright, 2012, 2009]. Chapter 4 is joint work with Ramón Cáceres, Sibren Isaacman and Maragaret Martonosi [Mir et al., 2013]. Chapter 5 is joint work with S. Muthukrishnan, Aleksandar Nikolov and Rebecca N. Wright [Mir et al., 2011]. Chapters 6 [Mir, 2013] and 7 [Mir, 2012] have also been published.

Acknowledgments

Like most other dissertations, this one would not have been possible without the support of several outstanding colleagues, friends, teachers, and family members. Taking the opportunity to thank them has been the best part of writing this dissertation. I would like to thank my advisor Rebecca Wright for her constant support through the years and for giving me the freedom to explore and work on my interests. I really appreciate that she always found ways of supporting my interests. Rebecca has an extraordinary ability to help one pull out the gist of their work, and I have greatly benefited from her comments and feedback. I am also very grateful for her help with my writing and presentation skills, and I hope I have improved on them over the years; I might just remember to keep the spacing and titles consistent and to spell-check!

This writing of this dissertation was supported in part by the National Science Foundation's grant numbers CCF-0728937 and CCF-1018445.

If there is one person without whom none of this would have been possible, who is really responsible for all of this, it is Poorvi Vora. Poorvi introduced me to research, when it first dawned on me that I could consider doing a Ph.D. She has been an extraordinary teacher and mentor to me. She encouraged me to challenge myself, to push my boundaries, to learn and question without fear, and to be comfortable not knowing; she believed in me, often at times when I myself did not. She continued watching out for me and believing in me through the ups and downs of graduate school and has supported me through the years in more than one way. I have been truly fortunate to have had met her. Many many thanks to her for humoring me,

checking on me and harassing me through the writing of this dissertation! I can only hope to be as inspiring and sincere a teacher and mentor as she is.

I would like to thank other members of my committee, David Cash, S. Muthukrishnan, and Magda Procopiuc for serving on my committee with enthusiasm and for helpful comments and feedback. I would also like to thank Tina Eliassi-Rad and Michael Littman for serving on my qualifying examination committee and for their encouragement.

Thanks to my first advisor at Rutgers, Joe Kilian, for his advice, encouragement and support in my first year. I came to Rutgers because of Joe, and it has been a productive, exciting and challenging journey! Even after Joe left Rutgers, his grant supported me for a year in graduate school.

I would like to thank Eric Allender, Michael Saks and William Steiger for their support at various stages of my graduate school career.

Many thanks to my coauthors and collaborators. Muthu is an extraordinarily inspiring researcher; thank you Muthu for your infectious enthusiasm and for your encouragement throughout graduate school, it has meant a lot to me. Alex Nikolov is not only a very bright researcher and great to work with, but also very generous with his ideas and time. Some of the fondest grad school memories I have are Alex and I scrambling to catch the 9:00 am train or bus into New York City to meet Muthu on a Sunday morning! Thanks to Ramon Caceres, Sibren Isaacman and Margaret Martonosi for their collaboration on the Human Mobility project. This was the first "real experimental" project I worked on, and many thanks to Sibren for his help with the experiments. Not only did I use his vast code repository but he also helped me deal with the uncertainty and lack of control that is inherent to such experimental work.

Thanks to Claire Monteleoni and Alantha Newman for a very pleasant collaboration. Though, unfortunately, the work Alex and I did with Claire and Alantha did not

make it to this dissertation (or any other publication), we spent many productive (in my opinion!) hours thinking, meeting, and eating cookies at Rutgers and Columbia! Claire and Alantha are a lot of fun to work with!

I am very grateful to Tong Zhang for pointing me to the PAC-Bayes literature. Without that, Chapters 7 and 8 would not have been possible.

I would like to thank other fellow graduate students at Rutgers for their support in different ways. Brian Thompson and I have been here in the program together through the years, he has always been tremendously helpful, supportive and encouraging. Brian and I both spent many (un)productive evenings in our first year solving Joe's problem sets. Thank you Brian, for your sharp and insightful comments, they have helped me improve a lot. Thanks to Crystal for being such a warm friend and well-wisher and for being part of my wonderful first year memories at Rutgers.

I would also like to thank Mangesh Gupte, Geetha Jagannathan, Aaron Jaggard, Neil Lutz, Sai Lu, Alex Nikolov, Jason Perry, Joe Wegehaupt, and Brian Thompson for sitting through several of my presentations, they have helped me improve as a presenter and researcher tremendously. Mangesh, in particular, was a tremendously helpful source of advice during the first years of grad school. Many thanks to Carlos Diuk, Priya Govindan, Monica Babes-Vroman, and Lars Sorensen for their encouragement.

My stay in graduate school would not have been the same without the strong support circle I fortunately found myself in. Several friends have cooked for me and provided me with support and nourishment when I most needed it: Aatish, Anindya, Chuck, Deepak, Kshitij, Sushmita, Vandana and Vijay, thank you! Huge thanks to my physicist and mathematician friends, Aatish, Deepak, and Vijay for sitting through my presentations and for asking me really good questions. A big thanks to Aatish and Deepak for being so enthusiastic about Science, in general, to Anindya for all the conversations on life and everything, to Kshitij for being my silliness partner and for

being so caring, to Sushmita for being delightfully weird, to Senia and Sinisa for the wonderful times and for making "Slavic" biryani on an American holiday, and to Vijay for sharing my interests in poetry.

Thanks also to Adina, Chioun, Daniel, George, John, Lizzie, Nida, Rebecca Flint and Shambhavi for all the good times, the dinners, and for the hanging out.

Thanks to D. for being in this together with me, for encouraging me through challenging moments, for always believing in me and for his love, companionship and support.

Thanks to my childhood friend, Asma for her constant love, support and undying belief in me.

Many thanks to the CS department and DIMACS staff for their extraordinary help in navigating the Rutgers bureaucracy. Thanks to the Douglass Project and affiliates, to Beth, Elaine and Laura for a wonderful last year at Rutgers.

I would also like to thank my family: my mother, my father, my sister and brother-in-law for all their support. The landscape of my extended family changed over the years it took me to write this dissertation. Many people who had a constant presence in my childhood passed away, and my nephew and one of my nieces were born. My late grandmother's and my late *nanaji*'s wishes and hopes have stayed with me. My nieces and nephew helped me maintain perspective throughout graduate school. My grandfather's poetry nourished my soul and provided inspiration during difficult circumstances.

I would like to end this by acknowledging my mother's fundamental contributions to all of this. My mother provided me with an environment where I could develop a love of knowledge and learning. In a time and place of surrounding violence and uncertainty, she created an environment where I could reflect, think and ask. I am deeply grateful for her support through the years.

Dedication

To

Zoon, my late, unlettered grandmother, who would often tell me that if she had gone to school—even primary school— she would have been a "bigger" *Science-daan* (Scientist) than me. I have never doubted that claim (except for the primary school part)

and,

to

Poorvi.

Table of Contents

Abstract ii

Preface iv

Acknowledgments v

Dedication ix

List of Tables xv

List of Figures xvi

List of Algorithms xviii

1. Introduction 1

1.1. Overview of thesis contributions 3

1.1.1. Differentially private algorithms and the utility-privacy tradeoff 4

1.1.2. Differential Privacy and Information theory 6

2. Background and Related Work 9

2.1. Why quantify and formalize privacy? 9

2.2. The case for differential privacy 12

2.2.1. Advantages 16

2.3. How to achieve differential privacy? 19

2.4. Pan-privacy 23

2.5. Related work in differential privacy and pan-privacy 24

2.5.1. Beyond differential privacy 27

3. Differentially Private Estimation of Random Graph Models 30

3.1. Introduction 30

3.2. Related work in privacy and anonymization 32

3.3. Parametric models and estimation 34

3.3.1. Kronecker graph model 35

3.3.2. Stochastic Kronecker graph model 37

3.3.3. Parameter estimation in the SKG Model 38

3.3.4. Moment based estimation of SKGs 39

3.4. A differentially private graph estimator 41

3.4.1. Differential privacy for graphs 41

3.4.2. Experimental results 46

3.5. Conclusions and future work 49

4. Differentially Private Modeling of Human Mobility at Metropolitan Scales 55

4.1. Introduction 55

4.2. Related work 57

4.3. Background 59

4.3.1. WHERE 59

Home and *Work* 60

CommuteDistance 60

CallsPerDay 61

ClassProb and *CallTime* 61

HourlyLocs 61

4.3.2. Differential privacy for Call Detail Records databases 62

4.4. Differentially private WHERE 63
4.4.1. Pre-processing 63
4.4.2. Distributions 63
Home and Work 63
Commute distance 65
Calls per day per user 67
Call times per user class 69
Hourly calls per location 72
4.4.3. DP-WHERE: putting it all together 75
4.5. Experimental evaluation 75
4.5.1. Datasets and methodology 76
4.5.2. Earth Mover's Distance 76
4.5.3. Daily range 79
4.6. Conclusions and future work 80

5. Differentially Private Gaussian Regression 83
5.1. Related Work 84
5.1.1. Differential Privacy 85
5.2. A relaxed exponential mechanism 85
5.3. Linear Regression 88
5.3.1. Gaussian Regression 89
5.4. Differentially Private Gaussian Regression 91
Connections to ridge regression 91
5.4.1. A probabilistic upper bound on the global sensitivity of the score function in Gaussian regression 93

5.4.2. Computationally efficient exponential mechanism for Bayesian regression . . . 96
5.5. Risk Bounds . . . 97
5.5.1. Risk bounds for differentially private regression . . . 98
5.6. Experiments . . . 100
5.7. Conclusions and future work . . . 102

6. Pan-private Algorithms via Statistics on Sketches . . . 104
6.1. Introduction . . . 104
6.1.1. Our contributions . . . 106
6.2. Background . . . 109
6.2.1. Model and notation . . . 109
6.2.2. Pan-privacy . . . 109
6.2.3. Sketches and stable distributions . . . 111
6.3. Pan-private algorithms for fully dynamic data . . . 113
6.3.1. Distinct count . . . 113
6.3.2. A general noise-calibrating technique for sketches. . . . 117
6.4. Discussion . . . 118

7. Information Theoretic Foundations of Differential Privacy . . . 121
7.1. Introduction . . . 121
7.2. Definitions and background . . . 122
7.2.1. Related work . . . 124
7.3. Differentially private mechanisms in a rate-distortion framework . . . 124
7.3.1. An information channel . . . 126
7.3.2. Connection to the rate-distortion framework . . . 127

7.4. Differential privacy arising out of the Maximum Entropy principle or Minimum Discrimination Information principle 129
7.5. Conclusion and future work 131

8. Differentially Private Learning and PAC-Bayesian Bounds 133
8.1. Introduction 133
8.1.1. Differentially private learning 134
8.2. PAC-Bayesian bounds and differentially private learning 135
8.3. Conclusion and future work 139

9. Conclusion 141

Bibliography 142

List of Tables

3.1. Comparison of parameter estimation for $\varepsilon = 0.2, \delta = 0.01$ 47

4.1. Average EMD error for WHERE using CDRs and DP-WHERE using various ε, as the commute-grid cell size changes. 79

List of Figures

3.1. Overlayed patterns of real network for CA-GrQC ($N = 5,242, E = 28,980$) and the estimated synthetic Kronecker graph using the three different estimators. 50

3.2. Overlayed patterns of real network for AS20 ($N = 6,474,\ E = 26,467$) and the estimated synthetic Kronecker graphs using the three different estimators. 51

3.3. Overlayed patterns of real network for CA-HepTh ($N = 9,877,\ E = 51,971$) and the estimated synthetic Kronecker graph using the three different estimators. 52

3.4. Overlayed patterns of a synthetic source Kronecker network and the estimated sythetic Kronecker graph using the three different estimators. 53

4.1. Overview of DP-WHERE, which modifies WHERE by adding noise to achieve differentially private versions of the input probability distributions. The rest of WHERE remains unchanged. 57

4.2. CDF of *Home* distribution for different values of $\varepsilon_{\texttt{home}}$. 65

4.3. CDF of *CallsPerDay* 69

4.4. Comparison of distribution of call times for two classes of users as determined by Algorithm 5 to the non-private clustering. 72

4.5. *HourlyLocs* Distribution for 5:00pm to 6:00pm 74

4.6. EMD error for DP-WHERE using different values of ε and a fixed commute-grid cell size of $0.01^\circ \times 0.01^\circ$, as compared to WHERE using CDRs and WHERE using public data. 77

4.7. EMD error for DP-WHERE using different sizes of commute-grid cell side and a fixed ε of 0.23. 78

4.8. Daily range for DP-WHERE ($\varepsilon = 0.23$, commute-grid size $= 0.01^\circ \times 0.01^\circ$), WHERE from CDRs, and the real CDR dataset. 80

5.1. Mean Squared Error (MSE) of the predictor for the Boston Housing data set, averaged over 100 instances of 5-fold cross validation 101

5.2. Mean Squared Error (MSE) of the predictor for the Boston Housing data set, for lower values of ε 102

7.1. Information theoretic model of differentially private channel 126

7.2. Risk-distortion curve 129

8.1. Information theoretic model of differentially private learning 139

List of Algorithms

1. Differentially private estimation of $\tilde{\Theta}$ 46
2. Algorithm to compute an $\varepsilon_{\texttt{home}}$-differentially private CDF of the *Home* distribution. 64
3. Algorithm to compute $\varepsilon_{\texttt{commute}}$-differentially private CDFs of the Commute distributions. 66
4. Algorithm to compute an $\varepsilon_{\texttt{cpday}}$-differentially private CDF of the *CallsPerDay* distribution 68
5. Differentially private k-means algorithm 70
6. Algorithm to compute $\varepsilon_{\texttt{hrlocs}}$-differentially private CDFs of the *HourlyLocs* distributions. 73
7. Pan-private approximation of $D^{(t)}$ 117

1

Introduction

> *"Indeed, we appear to be in the midst of a massive collision between unprecedented increases in data production and availability about individuals and the privacy rights of human beings worldwide, most of whom are also effectively research subjects."*
>
> – Gary King [71].

Data contributed by individuals is becoming increasingly central to scientific inquiry. Privacy concerns, however, prevent the fullest use of this data. Debates about tension between individual privacy and use of data for research and analysis, in fields as varied as social science [71], genomics [49], and library studies [98] are already underway. While use of individuals' data raises ethical and technical questions about privacy, reluctance towards contributing such data creates hurdles in scientific progress and sharing of knowledge. Scientific, social, governmental and legal institutions, therefore, have a sustained interest in examining questions of potential use of data while providing privacy to the participating individuals. For example, a report published by the National Academy of Sciences [120] that aims at examining "privacy in the information age", in a "deep, comprehensive and multidisciplinary" manner, raises the following questions:

> "How are the threats to privacy evolving, how can privacy be protected, and how can society balance the interests of individuals, businesses, and government in ways that promote privacy reasonably and effectively?"

Such discussions are not just academic; several business decisions have been affected by this concern. In 2006, for example, the online movie rental and streaming

service Netflix announced a contest with a million dollar prize for a movie prediction algorithm that would improve their in-house prediction algorithm by at least 10%. To encourage wider participation, Netflix published an "anonymized" subset of customers' movie rating data. The contest was a commercial and scientific success with a wide array of researchers making use of the large dataset in several ways. Two winning teams were announced [13] whose prediction algorithms showed a significant improvement in the accuracy of predictions. The contest also succeeded in spurring other research in the area of recommender systems; see Korolova's Ph.D. thesis [72] for a detailed discussion. However, in 2008, Narayanan and Shmatikov [94] showed how to de-anoymize users in the published Netflix dataset by correlating it with non-anonymous movie reviews on the Internet Movie Database (IMDb). They showed that with 8 movie ratings, 99% of records in the published database could be uniquely identified and with two ratings as many as 68%. This led to widespread concern among privacy advocates and society, in general, leading to a Jane Doe lawsuit—one in which Narayanan and Shmatikov's work [94] was extensively cited—by a closeted lesbian who claimed that she did not "want her sexuality nor interests in gay and lesbian themed films broadcast to the world" and that Netflix has violated her (and other subscribers') privacy. As a result of the lawsuit, Netflix decided to cancel the second phase of this competition [57].

The Netflix case illustrates both the scientific potential of such datasets containing individuals' sensitive data as well as the fallouts of not carefully considering privacy issues. The fundamental question we have to contend with is whether such uses of data can still be compatible with privacy? In view of such widely known privacy debacles, how can we persuade data owners and stakeholders to contribute their data to analyses? What risks do individuals face when their data are used? How can privacy-preserving solutions mitigate these risks? What kinds of data analyses can be done

privately?

To answer these questions, one must first quantify and define privacy, and develop a formal framework that facilitates its deployment. In this thesis we will use *differential privacy* [33] as our notion of privacy. We will explain this notion and our motivations for using it in more detail in Chapter 2. The fundamental question raised by the Netflix case—of whether data utility can be achieved along with privacy— is a central question that pervades this thesis.

We briefly note that privacy means different things in different contexts. Throughout the thesis, we will assume a setting where information is released to the intended party. The privacy questions that concern us are *inferential* in nature, that is, what (other) sensitive information about participating individuals can be inferred from this legitimate piece of information, often by possibly combining it with arbitrary *auxiliary information*?

1.1 Overview of thesis contributions

Using differential privacy as our notion of privacy, we examine how privacy and utility relate to each other and whether in various contexts of data analyses there can be a meaningful tradeoff between the two. Specifically, our contributions fall into two categories. In Section 1.1.1 we summarize the first part of the thesis—of studying the privacy-utility tradeoff for differentially private algorithms in various settings. In Section 1.1.2 we summarize the second part—of locating the meaning of this privacy-utility tradeoff in an information theoretic framework

1.1.1 Differentially private algorithms and the utility-privacy tradeoff

We examine the privacy-utility tradeoff for data analyses that include both *interactive* and *non-interactive* settings. Chapters 3, 4 and 5 consider the question in a non-interactive setting, where statistical information about a dataset is published once. Chapter 6 considers the interactive setting in which a user poses queries to a data curator who responds with answers to such queries.

In Chapter 3 we examine the privacy-utility tradeoff in the case of graph data—data that have associations between entities, such as social networks. Here the non-interactive setting is particularly appealing. For example, access to a suitably "transformed" social network may help researchers track the spread of an epidemic or a sexually-transmitted disease in a community. We take recourse to techniques in the random graph modeling literature to generate representative synthetic graphs that also achieve a given level of differential privacy. Using tools from statistical inference, we assume that an observed graph is generated from an underlying, but unknown, probability distribution. Given a graph, that is treated as a sequence of observations in such a model, our goal is to infer the (parametrized) distribution itself. The choice of a model is typically guided by empirical and theoretical considerations of how well the model captures key properties of real-world graphs. For our purpose, we use Leskovec et al.'s stochastic Kronecker graph model [77, 78] that effectively models salient features of real-world graphs. We empirically show that in the stochastic Kroencker graph model, one can achieve a differentially private estimation of the generating probability distribution that retains the utility of the original estimation proposed by Leskovec et al. [77] in terms of matching several statistics of the original graph. We demonstrate experimentally that for a meaningful level of privacy, we can still approximate statistics that the original Kronecker model did with high accuracy. This is joint work with Rebecca N. Wright [93, 93].

In Chapter 4 we turn our attention to study the utility-privacy tradeoff in the construction of models of human mobility in metropolitan areas. Models of human mobility have wide applicability to areas such as infrastructure and resource planning, analysis of infectious disease dynamics and ecology. We demonstrate that using an earlier mobility model named WHERE (Work and Home Extracted REgions) [62] drawn from real-world and large-scale (cellular phone) location data, we can construct a differentially private model of human mobility (called DP-WHERE) while still retaining the good utility of the original WHERE model. We study the privacy-utility tradeoff in this setting showing that differential privacy can be achieved for a modest reduction in accuracy. We do this by generating commuting patterns for synthetic users in a geographical area over a period of time. In particular, across a wide array of experiments involving 10,000 synthetic users moving across more than 14,000 square miles, the distance between synthetic and real population density distributions for various levels of privacy in Differentially Private-WHERE (DP-WHERE) differ by only 0.17–2.2 miles from those of the original WHERE approach. This is joint work with Ramón Cáceres, Sibren Isaacman, Margaret Martonosi and Rebecca N. Wright [91].

In Chapter 5 we study the problem of differentially private regression, a supervised learning task concerned with the prediction of continuous quantities. The *training set* for this task consists of individuals' sensitive data. For example, consider a database that contains individuals' data on their smoking frequencies and associated risk of lung cancer. Using this data—the training data—we would like to determine a function that predicts an individual's risk of lung cancer given her smoking frequency. This function is learned by using individuals' private data, but clearly has great social benefit. Adapting techniques from Gaussian regression, we propose a differentially private mechanism for linear regression. We achieve this by introducing a novel "relaxed" *exponential mechanism* that may be of independent interest. The utility of differentially

private regression is measured using the *expected risk* of the predictor. Exploiting a connection between Gaussian regression and ridge regression helps us achieve utility bounds that, unlike previous work, do not always depend on the dimensionality of the predictor or feature space. This makes our technique useful in high dimension problems—where the norm of the "true" predictor is known to be small. We also experimentally demonstrate the performance of our privacy preserving scheme on real-world data.

In Chapter 6, we turn to the online interactive setting. Consider online data, where we track data as it gets inserted and deleted. There are well developed notions of private data analyses with dynamic data using differential privacy. We want to go beyond privacy, and consider privacy together with security, formulated as *pan-privacy* by Dwork et al. [34]. Informally, pan-privacy preserves differential privacy while computing desired statistics on the data, even if the internal memory of the algorithm is compromised (say, by a malicious breakin or insider curiosity or by fiat by the government or law). We study pan-private algorithms for estimating the *distinct count* statistic on dynamic data with both insertions and deletions. We present the first known pan-private algorithm for approximating the distinct count statistic, where both positive and negative updates are made. Our algorithm relies on a sketching technique popular in streaming, to which we add suitable noise, using a novel approach of calibrating noise to the underlying problem structure and the projection matrix of the sketch. This is joint work with Aleksandar Nikolov, S. Muthukrishnan, and Rebecca N. Wright [92].

1.1.2 Differential Privacy and Information theory

Any algorithm that releases a data statistic or any useful approximation of it leaks some *information* (in the information-theoretic sense) about participating individuals.

How can this information leakage be traded against the utility that the algorithm provides, and what is the relationship of this framework to differential privacy?

Chapter 7 characterizes differentially private mechanisms in terms of this utility-leakage tradeoff. We observe that differentially private mechanisms arise out of minimizing the information leakage (measured using the information-theoretic notion of *mutual information*) while trying to maximize "utility". The notion of utility is captured by an abstract distortion function that measures the "distortion" between the input and the output of the mechanism. This also helps us relate differentially private mechanisms to the *rate-distortion* framework in information theory.

Chapter 8 examines similar tradeoffs in the specific context of differentially private learning. Such tradeoffs help us establish a connection between differentially private learning and the field of *PAC-Bayesian learning*. In machine learning, *generalization bounds* provide an upper bound on the *true risk* of a predictor θ in terms of a) its *empirical risk* on the training data $\hat{\boldsymbol{Z}}$, b) some function of a measure of the complexity of the predictors, and c) a confidence term $\tau \in [0, 1]$. Given such a (hopefully tight) upper bound, one can then compute the predictor that minimizes it. In bounds such as the *VC-Dimension bounds*, (see, for example, [7]) the data-dependencies only come from the empirical risk of the predictor on the training set. This allows the difference between the empirical risk and the true risk to be bounded uniformly for all predictors in this class. As a result such bounds are often loose. For "data-dependent" bounds, on the other hand, the difference between the true risk and the empirical risk depends on the training set $\hat{\boldsymbol{Z}}$. In data-dependent bounds such as PAC-Bayesian bounds possible, prior knowledge is incorporated into a model that places a prior distribution on the space of possible predictors. This is then updated to a posterior distribution after observing the data. We show that under certain assumptions, the randomized predictor that optimizes the PAC-Bayesian bounds is differentially private thus establishing the

equivalence between PAC-Bayesian learning and differentially private learning. This helps us propose generalization bounds for different differentially private machine learning tasks. For example, the utility bounds in Chapter 5 can also be achieved in this manner.

2

Background and Related Work

"I'm in the database but nobody knows."

– Cynthia Dwork[1]

In this chapter we briefly survey the history of quantifying privacy and the development of the notion of differential privacy. In Section 2.2 we formally define differential privacy and survey some of the techniques known to achieve differential privacy which we will use in the rest of the thesis. In Section 2.4 we review an extension of differential privacy called *pan-privacy*. We briefly summarize related work on differential privacy in Section 2.5, but describe it in more detail in the relevant chapters.

2.1 Why quantify and formalize privacy?

The Netflix fallout discussed in Chapter 1 shows that ad-hoc, even well-intentioned, methodologies employed with the hope of preserving individuals' privacy do not always work. Organizations that use (and especially seek to publish) individuals' sensitive data need to carefully consider and think through privacy issues. There is need for a rigorous, robust definition of privacy that provides clear and quantifiable guarantees.

Even before the ubiquity of datasets containing individuals' sensitive information

[1] `http://cyber.law.harvard.edu/interactive/events/luncheon/2010/09/dwork`

analysts, researchers, and policy makers had been grappling with the issue of data privacy. Modern technology that enables massive generation and storage of individuals' data, and networking between sources of data, exacerbates this condition in ubiquity and scale, but some of the fundamental issues remain the same. For example, federal and state agencies have been collecting data, called *microdata*, on individuals for many decades. These collections are critical for social planning, research and development. The United States Census Bureau collects a variety of information on individuals and businesses—information that is used by federal, state and local governments to make important social and economic decisions. This information is intended to be disseminated either as microdata or summary data to both the public and interested researchers. Some of the earliest work on privacy stems from complex ethical, legal, and scientific needs of these organizations (see [28], for example). One of the motivations of this body of research was that if potential participants (whose data is being collected) are not assured of confidentiality, their responses might not be candid. Consider, for example, a local government planning an AIDS clinic in a specific neighborhood based on responses to a survey that asks sensitive questions. As early as 1965, Warner [121] proposed ways to allow participants to respond to questions on sensitive issues (such as a sexually transmitted disease or use of drugs) while maintaining confidentiality

Such concerns in the statistical community have given rise to a body of literature in *Statistical Disclosure Control* (SDC) methods [28, 56]. In 1977 Dalenius formulated *disclosure* in the following manner [25]:

> "If the release of the statistic S makes it possible to determine the value [of confidential statistical data] more accurately than is possible without access to S, a *disclosure* has taken place."

We will revisit this formulation in Section 2.2, see why it is an unachievable notion of privacy, and how it can be suitably modified to define "achievable privacy".

SDC techniques are defined in the glossary on Statistical Disclosure Control [38] as:

> "the set of methods to reduce the risk of disclosing information on individuals, businesses or other organizations. Such methods are only related to the dissemination step and are usually based on restricting the amount of or modifying the data released."

While this stream of work seeks to define and identify privacy and "privacy breaches", a salient shortcoming is the lack of a unifying principle. Different kinds of disclosure are identified and sought to be mitigated by making various assumptions, that may not always hold in reality. Often, models of background knowledge assumed are short-sighted leading to subsequent attacks on these definitions.

One of the first such attacks was by Latanya Sweeney [114] who carried out a *linkage attack* on "sanitized" published data. The US Census Bureau routinely released data stripped of directly identifying information (such as social security numbers), leaving fields such as the 5-digit zip code, the gender and date of birth intact. These were believed to be innocuous attributes that did not directly identify an individual in a database. Using the 1990 US census data [114] Sweeney showed that for 87% of the US population, a combination of these three attributes was enough to uniquely identify them. Further, she carried out a linkage attack on supposedly anonymized data that consisted of these three attributes (among others). She started with anonymized data released by the Massachusetts based Group Insurance Commission (GIC) responsible for purchasing health insurance for state employees. She then correlated this database with a voter registration list that included the name, address, ZIP code, birth date, and gender of each voter. By linking the two databases she was able to identify the medical records of William Weld, then the governor of Massachusetts.

This attack led to a subsequent formalization of privacy that seeks to mitigate

the risks of re-identification because of such linkage attacks. The notion of *quasi-identifiers* [115] was used to capture the fact that a combination of information other than direct identifiers (such as name or social security numbers) may help in uniquely identifying an individual among a population. For example, for the US population the combination of ZIP code, birth date and gender constitutes a quasi-identifier. Sweeney developed a notion of privacy called k-anonymity [115] to prevent the kinds of linkage attacks she used to identify William Weld's medical records. Informally, a published data set is k-anonymous if every tuple in the dataset looks exactly like $k-1$ others with respect to their quasi-identifiers.

Even though k-anonymity thwarts linkage attacks that Sweeney carried out, it does not prevent other kinds of attacks. If because of some background information, an adversary knows that a particular individual appears in a database, then they might be able to learn some sensitive piece of information about them. For example, if because of background knowledge an adversary can locate an individual among a group of k-anonymous tuples, each of whom suffer from AIDS, then something sensitive about the individual has been learned. Machanavajjhala et al. [82] study some of the shortcomings of k-anonymity using *homogeneity* and *background knowledge* attacks. Subsequently, other notions of privacy in the literature such as ℓ-diversity [82] and t-closeness [79] were developed, each addressing shortcomings identified in the preceding definition.

2.2 The case for differential privacy

The history of privacy research is rife with formalizations of notions of privacy, whose weaknesses and inadequacies are revealed in subsequent attacks. While these definitions such as k-anonymity [115], ℓ-diversity [82] and t-closeness [79] themselves have not been successful in the sense of being robust to different assumptions, they have

taught us that modeling the background knowledge an adversary may possess is very hard. It has become clear that basing a privacy definition on assumptions about the background information of an adversary is a perilous path. It would, therefore, be worthwhile to strive for a notion of privacy that is independent of such assumptions. This is particularly important in the case of privacy because, unlike cryptographical settings, there is no real distinction between a legitimate recipient and an attacker, in terms of what information about the database is communicated to them.

With this in mind, let us revisit Dalenius's formalization of privacy. Dalenius reasoned that if, because of a released statistic, something sensitive has been learned about an individual that would not have been possible without access to this statistic, then a privacy breach has occurred. While at first sight, this sounds like a reasonable expectation of privacy, Dwork [37] presents the following scenario to illustrate why this notion is unachievable.

> "Suppose we have a statistical database that teaches average heights of population subgroups, and suppose further that it is infeasible to learn this information (perhaps for financial reasons) in any other way (say, by conducting a new study). Finally, suppose that ones true height is considered sensitive. Given the auxiliary information "Turing is two inches taller than the average Lithuanian woman," access to the statistical database teaches Turing's height. In contrast, anyone without access to the database, knowing only the auxiliary information, learns much less about Turing's height."

An important thing to note is that Turing does not have to be in the database of Lithuanian women for this "privacy breach" to occur. Since the purpose of a statistic release is for recipients to learn something useful about the underlying population, this can be combined with other background information to learn something sensitive about an individual—even one who is not in the database.

This example shows that absolute claims about what one can learn from a released are impossible to make. But, what about relative risks? Can we still retain focus on the individual and compare the risk of learning something about an individual from

a released statistic when they are in a database compared to when they are not in the database.

This *"I'm in the database but nobody knows"* intuition is precisely what differential privacy [33] formalizes. It compares the probability distribution on the range of possible outputs of an algorithm when an individual is present in a database, to the probability distribution (on the same set of possible outputs) when the individual is absent from the database. Informally, if the two distributions are guaranteed to be "almost the same" then the effect of one individual on any possible output (and its consequences) is negligible. Such a guarantee incentivizes participation of individuals in a database by assuring them of incurring very little risk by such a participation. To capture the notion of an individual's absence or presence in a database, the "sameness" condition is defined to hold with respect to a neighbor relation; intuitively, two inputs are neighbors if they differ only in the presence or absence of a single individual. For example, Dwork et al. [33] define datasets to be neighbors if they differ in a single row. Formally,

Definition 2.2.1 (Differential Privacy [33])**.** *A randomized algorithm $\mathcal{A}$ provides ε-differential privacy if for all neighboring input data sets $\mathbf{x}$, $\mathbf{x}'$, and for all $S \subseteq Range(\mathcal{A})$,*

$$\Pr[\mathcal{A}(\mathbf{x}) \in S] \leq \exp(\varepsilon) \times \Pr[\mathcal{A}(\mathbf{x}') \in S].$$

An important thing to note is that differential privacy is a property, not of the data but of the (randomized) algorithm $\mathcal{A}$. Given such a randomized algorithm $\mathcal{A}$, an input $\mathbf{x}$ induces a probability distribution on its range ($\mathcal{R}(\mathcal{A})$). Differential privacy is a boundedness condition on the ratio of the two probability density functions— corresponding to the two distributions induced on $\mathcal{R}(\mathcal{A})$ by $\mathbf{x}$ and any possible neighbor $\mathbf{x}'$— in terms of the parameter ε. The closer ε, the privacy parameter, is to zero, the closer the two distributions are, and the higher the level of privacy.[2]

[2] There is however, a fundamental limitation on how small ε can be, ε has to be $\Omega(\frac{1}{n})$ [33] for a useful notion of privacy.

McGregor et al. [84] define differential privacy, equivalently, in terms of a family of probability distributions.

Definition 2.2.2. *[84] Let* $\mathbf{x}$ *be a database of length* n*, drawing each of its elements from an alphabet* $\mathcal{X}$*, then an* ε-differentially private mechanism *on* $\mathcal{X}^n$ *is a family of probability distributions* $\{\pi(\boldsymbol{o}|\mathbf{x}) : \mathbf{x} \in \mathcal{X}^n\}$ *on a range* $\mathcal{R}$*, such that for every neighboring* $\mathbf{x}$ *and* $\mathbf{x}'$*, and for every measurable subset* $\boldsymbol{o} \subset \mathcal{R}$*,*

$$\pi(\boldsymbol{o}|\mathbf{x}) \leq \pi(\boldsymbol{o}|\mathbf{x}') \exp(\varepsilon).$$

Notice that the distribution (or equivalently the mechanism) is parametrized by the input database $\mathbf{x}$ or $\mathbf{x}'$, whichever is relevant.

This definition constrains the distribution on the range $\mathcal{R}$ of the algorithm $\mathcal{A}$ to be exactly the same; otherwise the ratio of the two probabilities will be unbounded over some region in $\mathcal{R}$. To allow such occurrences, a relaxed notion of differential privacy called (ε, δ)-differential privacy comes in handy. It lets the ratio be unbounded for a (negligibly) small fraction, δ, of events in the range.

Definition 2.2.3 ((ε, δ)-differential privacy [33])**.** *A randomized algorithm* $\mathcal{A}$ *provides* (ε, δ)*-differential privacy if for all neighboring input data sets* $\mathbf{x}$*,* $\mathbf{x}'$*, and for all* $S \subseteq Range(f)$*,*

$$\Pr[\mathcal{A}(\mathbf{x}) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{A}(\mathbf{x}') \in S] + \delta.$$

When δ=0, this is identical to ε-differential privacy in Definition 2.2.1. (ε, δ)-differential privacy allows us to violate the restriction of the the probabilities ratio for some events, allowing them to be outside the interval $[\exp(\varepsilon), \exp(-\varepsilon)]$, as long as these events themselves are low probability occurrences (specified by δ). We generally strive for very small values of δ, often smaller than any inverse polynomial in n, the size of the database. However, ε itself cannot be smaller than $O(\frac{1}{n})$ as noted by Dwork et al. [33]. We will explain this in more detail when we delve into the property of "group privacy" below.

2.2.1 Advantages

Some of the advantages of the notion of differential privacy compared to earlier notions in the literature are:

- **Guarantees at the level of an individual, extensible to the level of groups:** Ultimately, any privacy notion seeks to provide guarantees at the level of individuals. Through the notion of neighbors, that captures the presence or absence of an individual from a database, differential privacy reasons about relative risks of an individual (and as we will shortly see, groups of individuals). Another way of thinking about differential privacy is that it assumes a worst-case adversary. Consider the case where an adversary knows everything in a database $\mathbf{x}$ except for one individual Tom's data, that is, a neighboring database $\mathbf{x}'$ is completely known to the adversary. Then, the probability of learning something about Tom from composing the output of a differentially private algorithm $\mathcal{A}(\mathbf{x})$ with this (and any other) background information will not change much compared to the case when Tom is not in the database (that is, if we compose the output of the algorithm $\mathcal{A}(\mathbf{x}')$, where $\mathbf{x}'$ is the same as $\mathbf{x}$ except for Tom's data, with any background information).

  Further, differential privacy naturally lends itself to the case of "group privacy". Consider the notion of ℓ-neighbors where two databases $\mathbf{x}$ and $\mathbf{x}'$ are said to be ℓ-neighbors if they differ in the presence or absence of exactly ℓ individuals. If algorithm $\mathcal{A}$ provides ε-differential privacy, then we can easily see that via a sequence of ℓ deletions or additions, it provides $\ell\varepsilon$-differential privacy with respect to ℓ-neighbors. This can be seen as a group privacy guarantee which (as it should) degrades with ℓ, the size of the group. This observation will also help us in reasoning about the fundamental limitation of $\Omega(\frac{1}{n})$ on the order of ε as discussed by Dwork et al. [33] If epsilon is $o(\frac{1}{n})$, then after deleting every

element of a database $\mathbf{x}$ of size n used as input to an algorithm $\mathcal{A}$, we end up with an empty database which is an n-neighbor of $\mathbf{x}$. If $\mathcal{A}$ is now run on the empty database as an input, the output is $n\varepsilon$-differentially private with respect to any of its n-neighbors. Notice that $n\varepsilon = n \times o(\frac{1}{n})$ which is a constant. This implies that the distribution on the output range, when the input is empty, is within a constant factor of the distribution when the input is $\mathbf{x}$ (and presumably informative). The output distribution would then be the same for *every* database (even an empty and uninformative one), and hence would not give any useful information on $\mathbf{x}$.

- **Independence from auxiliary information:** Differential privacy bypasses the issue of background information by making no assumptions about the information an attacker may have. Since the guarantees of differential privacy are relative, it bypasses the issue of having to explicitly model the background information. Instead, it provides a strong, robust, and natural notion of privacy, one that holds in the face of any auxiliary information. It captures the dilemma of information revelation (utility) and information hiding (privacy) in the form of this relative guarantee.

- **No assumptions about the computational power of an attacker:** Another advantage of differential privacy is that it makes no computational assumptions about the power of an adversary. The attacker is assumed to be a so-called *information-theoretic* attacker, and the guarantees hold in the face of unbounded computational power.

- **Composability:** Typically, one wants to compute more than one statistic on a database. In such situations, each time we might learn something more about

the participating individuals. Ideally, we would like a privacy definition to reflect this possible degradation in privacy guarantees. Differential privacy has appealing composition properties, where if a sequence of queries is made on the same database $\mathbf{x}$ (or on databases that have a non-zero intersection with $\mathbf{x}$), then the privacy guarantees degrade with the number of queries in the following manner:

Theorem 2.2.4 (Serial Composition [33]). *For $i \in [k]$, let $\mathcal{A}_i(\mathbf{x})$ be an ε_i-differentially private mechanism executed on database $\mathbf{x}$. Then, any mechanism $\mathcal{A}$ that is a composition of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$, is $\sum_i \varepsilon_i$-differentially private.*

On the other hand, if we know that a sequence of queries is being made on non-intersecting sets, then we can invoke a "parallel composition" theorem:

Theorem 2.2.5 (Parallel Composition [33]). *For $i \in [k]$, let $\mathcal{A}_i(\mathbf{x})$ be an ε_i-differentially private mechanism executed on partition $\mathbf{x}_i$ of the database $\mathbf{x}$, such that $\forall i, j$, $\left|\mathbf{x}_i \cap \mathbf{x}_j\right| = 0$, and each user appears in exactly one of the $\mathbf{x}_i$'s. Then, any mechanism $\mathcal{A}$ that is a composition of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ is $\max_i \varepsilon_i$-differentially private.*

Both these theorems help us reason about the overall privacy degradation over several measurements of a database.

- **Post-processing:** Yet another desirable characteristic of differential privacy is that any "post-processing" of an output of a differentially private algorithm is also differentially private. As long as the post-processing does not need to "dip" back into the original private data, the privacy guarantees hold.

  Theorem 2.2.6 (Post-Processing [106]). *Let $\mathcal{A} : \mathcal{X} \to \mathcal{R}$ be an (ε, δ)-differentially private mechanism, and let $f : \mathcal{R} \to \mathcal{Y}$ be any function of $\mathcal{R}$, that takes as input only the output of the mechanism and is independent of the data, then $f(\mathcal{A}) : \mathcal{X} \to \mathcal{Y}$ also preserves (ε, δ)-differential privacy.*

- **Amplification by sampling:**

  Differential privacy also captures the intuition that if we are uncertain about which members of a population appear in a database, this should help boost the level of privacy the algorithm is able to achieve. More specifically, given an underlying population from which we sample a set of individuals on whom a (private) statistic is computed, we can improve privacy guarantees of our algorithm, as noted by Cormode et. al [21] and Smith [112].

  Theorem 2.2.7 (Amplification by sampling [21])**.** *Given a database* $\mathbf{x}$ *of size* n*, and an algorithm* $\mathcal{A}$ *and that provides* ε*-differential privacy, for any* $0 < p < 1$*, including each user in the input* $\mathbf{x}$*, into a sample* S *with probability* p *and outputting* $\mathcal{A}(S)$ *is* $2p\varepsilon$*-differentially private.*

  Smith offers the following "privacy for free" interpretation of this result in his blog

  > "If you are doing a survey and you can reasonably expect that the sample itself will remain secret, then you get ε-differential privacy for very small ε essentially for free. This formalizes a common intuition among, say, statisticians at the census bureau, that the very uncertainty about which members of the US population were surveyed (for long form data) provides a large degree of protection."

In the next section we will see general methodologies to achieve differential privacy.

2.3 How to achieve differential privacy?

One mechanism that Dwork et al. [33] use to provide differential privacy is the *Laplacian noise method* which depends on the *global sensitivity* of a function:

Definition 2.3.1 (Global sensitivity [33])**.** *For* $f : \mathcal{X}^n \to \mathbb{R}^d$*, the global sensitivity of* f *is*

$$\mathrm{GS}_f = \max_{\mathbf{x} \sim \mathbf{x}'} \left|\left| f(\mathbf{x}) - f(\mathbf{x}') \right|\right|_1$$

.

Definition 2.3.2 (Laplace distribution)**.** *The Laplace Distribution with mean 0 and scale b is the distribution with the following probability density function:*

$$p(x) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$$

We will denote an ℓ-length vector, each element of which is drawn from a Laplace distribution with mean 0 and scale b as $\langle \mathrm{Lap}(b) \rangle^{\ell}$. Then the *Laplacian mechanism* is given by the following theorem of Dwork et al. [33].

Theorem 2.3.3 (Laplacian mechanism[33])**.** *For any $f : D \to \mathbb{R}^{\ell}$, and $\varepsilon > 0$, the following mechanism $\mathcal{A}$, called the* Laplace mechanism*, is ε-differentially private:*

$$\mathcal{A}_f(D) = f(D) + \langle \mathrm{Lap}(\mathrm{GS}_f / \varepsilon) \rangle^{\ell}.$$

Proof. Let $\mathbf{x}$ and $\mathbf{x}'$ be any pair of neighboring databases. Let $p_{\mathbf{x}}$ denote the probability density function on $\mathcal{R}$ induced by $\mathcal{A}(\mathbf{x})$. Similarly, let $p_{\mathbf{x}'}$ denote the probability density function on $\mathcal{R}$ induced by $\mathcal{A}(\mathbf{x}')$. Consider an $o \in \mathcal{R} = \mathbb{R}^{\ell}$. We have

$$\begin{aligned}
\frac{p_{\mathbf{x}}(\boldsymbol{o})}{p_{\mathbf{x}'}(\boldsymbol{o})} &= \prod_{i=1}^{\ell} \frac{\exp\left(-\dfrac{\varepsilon\,|f(\mathbf{x})_i - \boldsymbol{o}_i|}{\mathrm{GS}_f}\right)}{\exp\left(-\dfrac{\varepsilon\,|f(\mathbf{x}')_i - \boldsymbol{o}_i|}{\mathrm{GS}_f}\right)} \\
&= \prod_{i=1}^{\ell} \exp\left(\frac{\varepsilon(|f(\mathbf{x}')_i - \boldsymbol{o}_i| - |f(\mathbf{x})_i - \boldsymbol{o}_i|)}{\mathrm{GS}_f}\right) \\
&\leq \prod_{i=1}^{\ell} \exp\left(\frac{\varepsilon(|f(\mathbf{x}')_i - f(\mathbf{x})_i|}{\mathrm{GS}_f}\right) \\
&= \exp\left(\frac{\varepsilon(|f(\mathbf{x}')_i - f(\mathbf{x})_i|}{\mathrm{GS}_f}\right) \\
&\leq \exp\left(-\varepsilon\right).
\end{aligned}$$

The first inequality follows from the triangle inequality. The last follows from the definition of sensitivity (Definition 2.3.1). □

We make extensive use of Laplace mechanism throughout this dissertation.

Another, more general (though, not always computationally efficient) method of providing differential privacy is the *exponential mechanism* proposed by McSherry and Talwar [87]. This is especially helpful in situations when the output is non-numeric.

This mechanism is parametrized by a "quality function" $q(\mathbf{x}, \boldsymbol{o})$ that maps a pair of an input data set $\mathbf{x}$ (a vector over some arbitrary real-valued domain) and candidate output $\boldsymbol{o}$ (again over an arbitrary range $\mathcal{R}$) to a positive real-valued "quality score." Higher quality scores imply good input-output correspondences. It assumes a base measure π on the range $\mathcal{R}$. For a given input $\mathbf{x}$, the mechanism selects an output $\boldsymbol{o}$ with exponential bias in favor of " high quality" outputs by sampling from the following *exponential distribution* [87]:

$$\pi^{\varepsilon}(\boldsymbol{o}|\mathbf{x}) \propto \exp(\varepsilon q(\mathbf{x}, \boldsymbol{o})) \cdot \pi(\boldsymbol{o}). \tag{2.1}$$

The superscript ε in π^{ε} denotes the dependence of the distribution $\pi^{\varepsilon}(\boldsymbol{o}|\mathbf{x})$, on the privacy parameter ε.

Theorem 2.3.4 (Exponential mechanism [87])**.** *The exponential mechanism, when used to select an output $\boldsymbol{o} \in \mathcal{R}$, gives $2\varepsilon\,\mathrm{GS}_q$-differential privacy, where GS_q is the global sensitivity of the quality function, that is:*

$$\mathrm{GS}_q = \max_{\mathbf{x} \sim \mathbf{x}',\, \boldsymbol{o}} \left| q(\mathbf{x}, \boldsymbol{o}) - q(\mathbf{x}', \boldsymbol{o}) \right|.$$

Proof. Let $\boldsymbol{o} \in \mathcal{R}$, then from Equation 2.1, for any two neighboring $\mathbf{x}$ and $\mathbf{x}'$, we have:

$$\begin{aligned}
\frac{\pi^{\varepsilon}(\boldsymbol{o}|\mathbf{x})}{\pi^{\varepsilon}(\boldsymbol{o}|\mathbf{x}')} &= \frac{\dfrac{\exp(\varepsilon q(\mathbf{x}, \boldsymbol{o})) \cdot \pi(\boldsymbol{o})}{\int_{\boldsymbol{o}} \exp(\varepsilon q(\mathbf{x}, \boldsymbol{o})) \cdot \pi(\boldsymbol{o})}}{\dfrac{\exp(\varepsilon q(\mathbf{x}', \boldsymbol{o})) \cdot \pi(\boldsymbol{o})}{\int_{\boldsymbol{o}} \exp(\varepsilon q(\mathbf{x}', \boldsymbol{o})) \cdot \pi(\boldsymbol{o})}} \\
&= \frac{\exp(\varepsilon q(\mathbf{x}, \boldsymbol{o})) \cdot \pi(\boldsymbol{o})}{\exp(\varepsilon q(\mathbf{x}', \boldsymbol{o})) \cdot \pi(\boldsymbol{o})} \cdot \frac{\int_{\boldsymbol{o}} \exp(\varepsilon q(\mathbf{x}', \boldsymbol{o})) \cdot \pi(\boldsymbol{o})}{\int_{\boldsymbol{o}} \exp(\varepsilon q(\mathbf{x}, \boldsymbol{o})) \cdot \pi(\boldsymbol{o})} \\
&\leq \exp\left(\varepsilon\,\mathrm{GS}_q\right) \cdot \frac{1}{\int_{\boldsymbol{o}} \exp\left(-\varepsilon\,\mathrm{GS}_q\right) \pi(\boldsymbol{o})} \\
&\leq \exp\left(2\varepsilon\,\mathrm{GS}_q\right).
\end{aligned}$$

The proof for a discrete $\mathcal{R}$ follows a similar argument with the integrals replaced by sums. □

The exponential mechanism is a useful abstraction when trying to understand differential privacy because it generalizes all specific mechanisms, such as the Laplacian mechanism introduced above. The exponential mechanism because of the generality of the input space $\mathcal{X}$, the output range $\mathcal{R}$, and the quality function q, can be shown to capture all differentially private mechanisms:

Theorem 2.3.5. *The exponential mechanism captures all differentially private mechanisms.*

Proof Sketch. Let $\mathcal{A}$ be any ε-differentially private mechanism. Let, q the quality function be defined as the logarithm of the pdf of the distribution $\mathcal{A}$ induces on the range $\mathcal{R}$, that is,

$$q(\mathbf{x}, \boldsymbol{o}) = \log\left(p_{\mathcal{A}}(\boldsymbol{o}|\mathbf{x})\right).$$

Then the exponential mechanism of Equation 2.1 on applying Theorem 2.3.4 is $2\varepsilon\,\mathrm{GS}_q$-differentially private. As long as we can bound the sensitivity of the log-likelihood function (the scoring function) we have a differentially private mechanism with the appropriate parameters. □

Specifically, the Laplace mechanism is captured by the exponential mechanism when

$$q(\mathbf{x}, \boldsymbol{o}) = \log\left(p_{\mathcal{A}}(\boldsymbol{o}|\mathbf{x})\right) \propto \log(\exp\left(-\left|f(\mathbf{x}) - o\right|\right).$$

In the next section we review the notion of pan-privacy, an extension of differential privacy.

2.4 Pan-privacy

Pan privacy guarantees a participant that his/her risk of being identified by participating in a data set is very little even if there is an external intrusion on the *internal state* of the analyzing algorithm.

Consider a universe $\mathcal{U}$ of IDs, where $|\mathcal{U}| = m$. An *update* is defined as an ordered pair $(i, d) \in \mathcal{U} \times \mathbb{Z}$ where i represents the ID to be updated d represents the update itself. Consider two such online sequences of updates $S = \big((i_1, d_1), \ldots, (i_t, d_t)\big)$ and $S' = \big((i'_1, d'_1), \ldots, (i'_{t'}, d'_{t'})\big)$. We define the following notion of neighborhood of sequences.

Definition 1 (User-level neigbors [92])**.** *S and S' are said to be (user-level)* neighbors *if there exists a (multi)set of updates in S indexed by $K \subseteq [t]$ that update the same ID, $i \in \mathcal{U}$, and there exists a (multi)set of updates in S' indexed by $K' \subseteq [t']$ that updates some $j(\neq i) \in \mathcal{U}$ such that $\sum_{k \in K} d_k = \sum_{k \in K'} d'_k$ and for all other updates in S and S' indexed by $Q = [t] - K$ and $Q' = [t'] - K'$ respectively,*

$$\forall i \in \mathcal{U} \sum_{k \in Q, s.t.\ i_k = i} d_k = \sum_{k \in Q', s.t.\ i'_k = i} d'_k.$$

Notice that in the definition above t and t' do not have to be equal because we allow the d_i's to be integers. The definition ensures that two inputs are neighbors if some of the occurrences of an ID in S is replaced by some other ID in S' and everything else stays the same except (a) the order may be arbitrarily different and (b) the updates can be arbitrarily broken up since they are not constrained to be 1's. The neighbor relation preserves the first frequency moment of the sequence of updates, considered to be public information. Also, the graph induced by the neighbor relation on any set of sequences with the same first frequency moment is connected.

Our notion of neighborhood is slightly different from the definition of Dwork et al. [34] definition, where any two data streams S and S' are neighbors if they differ

only in the presence or absence of any number of occurrences of any element $i \in \mathcal{U}$. Our definition ensures that two neighboring sequences of updates are of the same "length," in the sense that the sum of the updates over all items is the same for both S and S', that is, $\sum_{i=1}^{t} d_k = \sum_{i=1}^{t'} d'_k$. For this purpose, we constrain the sum of the updates of the occurrences of item i in S to be conserved when they are replaced by item j in S'. In our definition, the total weight of updates is public, but, still, an adversary cannot distinguish between appearances of ID i or ID j, even if the adversary knows all other appearances of all other IDs.

Definition 2 (User-level pan privacy [34])**.** *Let* **Alg** *be an algorithm. Let I denote the set of internal states of the algorithm, and let σ the set of possible output sequences. Then algorithm* **Alg** *mapping input prefixes to the range $I \times \sigma$, is* pan-private (against a single intrusion) *if for all sets $I' \subseteq I$ and $\sigma' \subseteq \sigma$, and for all pairs of user-level neighboring data stream prefixes S and S'*

$$\Pr[\mathbf{Alg}(S) \in (I', \sigma')] \leq e^{\varepsilon} \Pr[\mathbf{Alg}(S') \in (I', \sigma')]$$

where the probability spaces are over the coin flips of the algorithm **Alg**.

Pan-privacy protects users appearing (possibly several times) in an online data sequence, even if the state of the algorithm is revealed once either by an insider (who may have turned rogue) or by a subpoena or fiat. While this is an extremely strong (some would say unduly restrictive) notion of privacy, it is interesting to note that some statistics can still be computed accurately in this model.

2.5 Related work in differential privacy and pan-privacy

The development of differential privacy can be traced to Dinur and Nissim's work on identifying *blatant non-privacy* [27]. They modeled a statistical database by an n-bit string $x_1, \ldots x_n$, with queries being sums of random subsets of these bits; the data

curator adds a certain magnitude of noise to such queries and releases these noisy answers. They proposed a polynomial reconstruction algorithm that using such noisy subset sums can recover a significant fraction of the the original database whenever the magnitude of noise added is less than $O(n)$.

Subsequently, differential privacy was developed as a notion of privacy over a series of papers [11, 31, 33]. Following this, a growing line of work has emerged identifying differentially private mechanisms for both interactive and non-interactive settings [12, 87, 95].

There is now a growing body of work on differential privacy, and this section is far from exhaustive. We refer the reader to a set of tutorials and surveys [29, 36]. Here, we summarize some work relevant to the problems we examine in this thesis. A more detailed treatment and a comparison with our results can be found in the relevant chapters.

Hay et al. [51] first examined differential privacy in graphs releasing a differentially private approximation of the degree distribution of a graph under the notion of *edge differential privacy*. Karwa et al. [64] apply the notion of *smooth sensitivity* formulated by Nissim et al. [95] to compute differentially private approximations to graph statistics such as the number of k-triangles and k-stars. Sala et al. [109] generate differentially private synthetic graphs that are similar to the original graph by extracting the original graph's detailed structure into degree correlation statistics, and then computing differentially private approximations of these statistics. Proserpio et al. compute several differentially private graph statistics using a version of the PINQ [85] programming language [100]. Task and Clifton [116] ask broader questions about what graph differential privacy should actually protect and introduce the notion of *outlink privacy*. This enables a participant (a node) to plausibly deny its out-links to another node. More recently, Kasiviswanathan et al. [65] study the problem of releasing graph

statistics under the (harder to achieve) notion of *node differential privacy.*

There has been some work on publishing summaries of spatiotemporal data (such as mobility data) in a differentially private manner. Machanavajjhala et al. [81] generate synthetic commuting patterns for the United States population using a notion of *probabilistic differential privacy*. Chen et al. [19] study the problem of publishing a differentially private version of the trajectory data of commuters in Montreal. Ho and Ruan [54] propose a differentially private pattern mining algorithm for geographic location discovery using a region quadtree for spatial decomposition. Qardaji et al. [101] propose methods of releasing differentially private summaries of two-dimensional datasets. Cormode et al. [21] also study the problem of releasing differentially private summaries of two-dimensional data by using *spatial decompositions*. Andrés et al. [6] introduce the notion of *geo-indistinguishability* in location-based systems, which protects the exact location of a user while allowing release of information needed to gain access to a service.

Chaudhuri et al. [18] propose a differentially private algorithms for logistic regression using the so-called *objective perturbation* method—of adding appropriate noise into the objective function instead of the output. Rubinstein et al. [107] propose privacy-preserving classification methods using support vector kernels with *output perturbation* methods. Dwork and Lei [32] also examine regression in a relaxed version of differential privacy, using the *Propose-Test-Release* framework. Zhang et al. [124] introduce a novel differentially private *functional mechanism* that expresses the optimization function as a decomposition of "well-behaved" functions (such as polynomials) and then uses objective perturbation to obtain a solution that optimizes this decomposition. All of these techniques work in the low-dimensional regime, leading to a prohibitive amount of noise in the high-dimensional setting. Kifer et al. [69] propose differentially private algorithms for sparse regression problems in high-dimensional

settings, where there is an assumption of the existence of a "good" sparse solution.

Pan-privacy was introduced by Dwork et al. [34]; they compute statistics such as density estimators, cropped means and heavy hitters in this model with only insertions allowed in the online stream. Dwork also studies *continual differential privacy* in the online streaming setting [35]. Chan et al. study continual online differential privacy in centralized [15] and distributed [16] settings.

Alvim et al. [4], [2] first studied the relationship between information theoretic notions of leakage and differential privacy. They use the information-theoretic notion of *min-entropy* for the information leakage of the private channel, and show that differential privacy implies a bound on the min-entropy of such a channel. Barthe and Kopf [9] also develop upper bounds for the information leakage of every ε-differentially private mechanism.

2.5.1 Beyond differential privacy

In this section we briefly summarize other notions of privacy that are emerging from examining the semantics and application of differential privacy to various problem domains. While some of these are motivated by identifying the implicit assumptions that differential privacy makes about the data, others are motivated by providing a formal framework that is a meaningful relaxation of differential privacy yet allows better utility. Some other frameworks seek to relate the guarantees that differential privacy provides at the level of an individual to that of a notion of an individual's identifiability.

While differential privacy is an extremely strong notion of privacy, it does make some implicit assumptions on the data. Further, in practice, sometimes it is too strong a notion to be of use. Kifer and Machanavajjhala [67] examine these aspects, and argue that without making assumptions about how the data are generated, it is not

possible to provide both privacy and utility guarantees. Further, they show that for a meaningful application of differential privacy in context to its hiding "evidence of participation" of an individual in a database, there is an implicit assumption about the independence of each tuple (corresponding to the data of a single individual). This may not always be realistic.

In subsequent work Kifer and Machanavajjhala [68] introduce a general privacy framework called *Pufferfish* with the goal of allowing experts in an application domain, who often are not privacy experts, to formulate relevant and rigorous privacy definitions. Pufferfish also generalizes existing privacy definitions including differential privacy.

Gehrke et al. [40] introduce the notion of *crowd-blending privacy* that is weaker than the notion of differential privacy. Informally, k-crowd blending privacy requires that each individual i in the database "blends" with k other individuals, in the sense that the private output is "indistinguishable" if individual i's data is replaced by any of the k other individuals. They demonstrate that crowd-blending private algorithms for tasks such as histogram releases, achieve better utility than differentially private algorithms. Further, when combined with a sampling step where individuals in the database are randomly drawn from an underlying population, crowd-blending privacy is shown to satisfy differential privacy as well as the stronger notion of zero-knowledge privacy [41].

Lee and Clifton [75] introduce the notion of *differential identifiability*. They remark that in ε-differential privacy, ε limits how much one individual can affect the resulting distribution without quantifying how much information is revealed about an individual. In fact, it can be argued that differential privacy was formulated to bypass the complex task of trying to quantify such quantities. However, a notion of "information revelation" or "risk of identifiability" is often desirable under legal formulations

such as the U.S. HIPAA safe harbor rule [1] which require protection of "individually identifiable data". In addition, interested stakeholders and the general public are more interested in such notions. Lee and Clifton propose ρ-differential identifiability that explicitly models the background information an adversary may possess. The adversary is assumed to have complete knowledge of the database except for one individual, and using this he constructs the *possible worlds* of input databases. Modeling the background information in this manner enables one to make a quantifiable statement about the identifiability of an individual database. Using the output of a ρ-differentially identifiable algorithm $\mathcal{A}$ and a prior distribution on all possible worlds, the adversary creates a new posterior distribution on the possible worlds of databases. This also enables one to construct a new posterior on a specific individual's data, associating the individual with an "identifiability risk". Informally, the mechanism $\mathcal{A}$ ensures that the identifiability risk of any individual in the universe is less than or equal to ρ.

3

Differentially Private Estimation of Random Graph Models

"[...] even from a single anonymized copy of a social network, it is possible for an adversary to learn whether edges exist or not between specific targeted pairs of nodes."

– Lars Backstrom, Cynthia Dwork and Jon Kleinberg. [8]

3.1 Introduction

As graph databases such as social networks become ubiquitous, researchers have an unprecedented opportunity to understand and analyze complex social phenomena. For example, access to a social network may help researchers track the spread of an epidemic or a sexually-transmitted disease in a community. While society would like to encourage such scientific endeavors, if individuals run the risk of being identified, they may be apprehensive of participating in, or making their social network data available for, such studies. To ensure that public policy promotes such scientific projects, we are faced with the problem of providing researchers with a fairly accurate picture of the quantities or trends they are looking for without disclosing sensitive information about participating individuals.

There are numerous examples of data that have associations between entities, such as social networks, routing networks, citation graphs, biological networks, etc. Such associations between entities may be modeled as a graph, where individuals are represented by the nodes, and relationships between individuals as edges. Each node may be associated with various attributes. The risk of being identified by participating in

such a database is two-fold: individuals may be identified by virtue of their attributes or they may be identified from their associations with other individuals and some background information, that they usually cannot predict or control, or they might be identified using a combination of the two. In this chapter, we will only be concerned about preventing identification of the nodes using associations between individuals and some possible background information, an approach that Korolova et al. [73] call *link privacy*. Using the work of Hay et al. [51], this can be extended to include a weak form of *node privacy*. Our proposed mechanism for synthetic graph generation, which aims to approximate certain statistics of the original graph, satisfies the rigorous definition of ε-differential privacy. Private estimation of the Stochastic Kronecker Graph (SKG) model parameter is an interesting problem, especially given the surge in the popularity of SKGs for graph modeling. Subsequent work by Gleich and Owen [46], which estimates the SKG model parameters by using a "moment matching" method, makes it possible for us to apply the work of Hay et al. [51] and Nissim et al. [95] to efficiently compute private approximations of the "matching statistics" and, hence, in obtaining private estimates of the model parameter.

To generate representative synthetic graphs, we use tools from statistical inference. Assuming that observed data is generated from an underlying, but unknown, probability distribution, we use the data to infer the distribution. A graph $G(V, E)$ is represented as a vector of random variables $\{E_1, E_2, \ldots, E_N\}$, where each of the E_i's are 0-1 random variables representing the presence or absence of an edge (assuming a specific known ordering of all potential edges between $|V|$ vertices). We assume that data is generated from a parameterized family of probability distributions. Given a graph, that is treated as a sequence of observations in such a model, our goal is to infer the parameter of the distribution and hence the distribution itself. If the estimator preserves differential privacy and is a good estimator, we can publish it and anyone interested

in studying statistical properties of the original graph G can sample the distribution to yield a synthetic graph G_S which mimics the statistical properties of G. We could also sample several graphs from the distribution and compute an average of the desired statistic over several such graphs. To use such an approach, we need to impose a relevant model on the kinds of graphs we are interested in. The choice of a model is typically guided by empirical and theoretical considerations of how well the model captures key properties of real-world graphs. For our purpose, we use Leskovec et al.'s Kronecker graph model [77, 78] that effectively models salient features of real-world graphs. We compute an estimator, based on Gleich and Owen's non-private estimator [46], that is provably differentially private and that still compares favorably with the estimators proposed by Leskovec et al. [77] and Gleich and Owen [46] in terms of matching several statistics of the original graph.

Section 3.2 summarizes related work in privacy and anonymization. In Section 3.3, we provide the required background about the stochastic Kronecker graph (SKG) model and parameter estimation in this model. In Section 3.4, we discuss our main results: we show how we can compute an estimator of a given graph in the SKG model in a differentially private manner and also experimentally demonstrate how well the private estimator does on mimicking statistical properties of the original graph when compared to non-private methods such as those of Gleich and Owen [46] and Leskovec et al. [77]. We observe that our private estimator performs almost similalrly to Gleich and Owen's non-private estimators, for meaningful values of the privacy parameter ε.

3.2 Related work in privacy and anonymization

The problem of anonymizing databases has been receiving considerable attention over the last decade. However, researchers have only recently started looking at the problem of privacy preservation in graphs. Backstrom, Dwork, and Kleinberg [8] describe

a family of attacks where access to a naively anonymized graph with the identifiers of the nodes stripped can enable an adversary to learn whether edges exist or not between specified pairs of nodes. Many solutions assuming various models of attacks have been proposed: see [23, 52, 73, 127, 128] for examples. Most of that work, provides guarantees only against a specific set of adversaries who are assumed to have specific background knowledge. In reality, however, individuals and even organizations managing the database have little or no control over auxiliary information available to the adversary. Hence, we use differential privacy as our notion of privacy.

We base our estimation algorithm on work by Gleich et al. [46] that estimates model parameters using a moment matching method rather than an approximation of the Maximum Likelihood Estimator as in Leskovec et al. [78]. The algorithm matches four statistics of the observed graph to the expected values of these statistics over the probability distribution on graphs defined by the parameters. This enables us to use the work of Hay et al. [51] and the results of Nissim et al. [95] to compute differentially private approximations to these features F of the observed graph that we seek to match. Hay et al. [51] compute a differentially private approximation to the degree distribution of a graph using post-processing techniques. Nissim et al. [95] compute a differentially private approximation to the number of triangles of a graph.

Karwa et al. [64] apply the notion of *smooth sensitivity* formulated by Nissim et al. [95] to compute differentially private approximations to other graph statistics such as the number of k-triangles and k-stars. Sala et al. [109] also generate synthetic graphs that are similar to the original graph by extracting the original graph's detailed structure into degree correlation statistics, and then computing differentially private approximations of these statistics to generate a private synthetic graph. This is the closest in spirit to our work.

3.3 Parametric models and estimation

This section provides background on parametric model estimation, the Stochastic Kronecker Graph Model [77, 78] and the moment based estimation method of Gleich [46] in which our work is grounded.

A parametric statistical model, say $\mathcal{F}$, is a set of probability distributions that can be parametrized by a finite set of parameters. Parametric estimation in such a model assumes that data observed is generated from a parametrized family of probability distributions $\mathcal{F} = \{f(x;\theta) : \theta \in \Theta)\}$, where θ is an unknown parameter (or vector of parameters) that can take values in the parameter space Θ. Let $X = (X_1, X_2, \ldots X_N)$ denote N random variables representing observations $X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N$, and let the joint probability density function of $(X_1, \ldots, X_N)$, given by $f(x_1, x_2, \ldots, x_N; \theta)$, depend on θ, the parameter of the distribution.

After observing this data, an estimate $\hat{\theta}$ of the unknown true parameter θ is formed. $\hat{\theta}$ is a function of the observations and hence, it is also a random variable. The problem of parameter estimation is to pick a $\hat{\theta}$ from the parameter space that best estimates the true parameter in some optimum sense. Parameter estimation is a well studied branch of statistics; see [122] for a review.

As mentioned before, the choice of a generative parametric model for graphs is typically based on empirical or theoretical considerations of how well a model captures significant descriptive properties of graphs, such as degree distribution, specific patterns observed, etc. Once such a model is defined, the task consists of estimating the parameter of the model that generated a particular instance G. G can be looked at as a sequence of observations $E_1, \ldots, E_N$ where the E_i's are 0-1 random variables representing the absence or presence of an edge i (according to a specific ordering). The estimated parameter defines a probability distribution on all graphs, one from which we assumed G was generated. One can then sample this probability distribution to

generate a synthetic graph G_S and run queries on it to get an approximation to the answers that one would have obtained from the original graph G. In this section, we introduce the Stochastic Kronecker Graph (SKG) model, the specific generative model we use. In Section 3.4, we show how to estimate the parameter in a differentially private manner that demonstrates experimental utility with respect to certain statistics.

3.3.1 Kronecker graph model

Modeling graphs, in general, and networks in particular, is an important problem. Most work in graph modeling consists of studying patterns and properties found in real-world graphs and then finding models that help understand the emergence of these properties. Some of the key properties studied are degree distribution, diameter, hop-plot, scree plot, and node triangle participation [77, 78]. The Kronecker graph model effectively captures some of the salient patterns of real-world graphs, such as heavy tailed in-degree and out-degree distributions, heavy tails for eigenvalues and eigenvectors, small diameters, and "densification power law" observed in the Internet, the Web, citation graphs, and online social networks. Many models in the literature focus on modeling one static property of the network model while neglecting others. Moreover, the properties of many such network models have not been formally analyzed. Leskovec et al.'s Kronecker graph model has been empirically shown to match multiple properties of real networks. It also facilitates formal analysis of these properties and establishes, empirically and analytically, that Kronecker graphs mimic some important properties of real-world graphs such as those described above. The Kronecker graph results of Leskovec et al. [77, 78] have three important contributions:

1. Their graph generation model provably produces networks with many properties often found in real-world graphs, such as a power-law degree distribution and small diameter.

2. Their approximate MLE algorithm is fast and scalable, being able to handle very large networks with millions of nodes.

3. The estimated parameter generates realistic looking graphs that empirically match the statistical properties of the target real graphs.

Kronecker graphs are based on a recursive construction, with an aim of creating self-similar graphs recursively. The process starts with an *initiator graph* G_1 with N_1 nodes. By a recursive procedure, larger graphs $G_2, \ldots, G_n$ are generated in succession such that the kth graph, G_k, has $N_k = N_1^k$ nodes. This procedure is formalized by introducing the concept of Kronecker product of the adjacency matrices of two graphs [78].

Definition 3.3.1 ([78]). *Given two matrices $\mathcal{A}$ and $\mathbf{B}$ of sizes $n \times m$ and $n' \times m'$ respectively, their* Kronecker product *is a matrix $\mathbf{C}$ of dimensions $(n \cdot n') \times (m \cdot m')$ defined as:*

$$C = \mathcal{A} \otimes \mathbf{B} = \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \ldots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \ldots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \ldots & a_{n,m}\mathbf{B} \end{pmatrix}$$

The Kronecker product of two graphs is the Kronecker product of their adjacency matrices, defined as:

Definition 3.3.2 ([78]). *Let G and H be graphs with adjacency matrices $A(G)$ and $A(H)$ respectively. The Kronecker product $G \otimes H$ of the two graphs is the graph whose adjacency matrix is the Kronecker product $A(G) \otimes A(H)$.*

Informally, the Kronecker product of two graphs G and H is the "expanded" graph obtained by replacing each node in G by a copy of H. G_2 is obtained by taking the Kronecker product of G_1 with itself, G_3 by taking the Kronecker product of G_2 with G_1, and so on, such that the kth Kronecker power of G_1 gives G_k. Formally:

Definition 3.3.3 (Kronecker power [78]). *Given a*

Kronecker initiator adjacency matrix Θ_1*, the kth* power of Θ_1 *defined by*

$$\Theta_1^{[k]} = \underbrace{\Theta_1 \otimes \Theta_1 \otimes \ldots \otimes \Theta_1}_{\textbf{\textit{k times}}} = \Theta_1^{[k-1]} \otimes \Theta_1$$

The graph G_k *defined by* $\Theta_1^{[k]}$ *is a* Kronecker graph of order *k with respect to* Θ_1.

3.3.2 Stochastic Kronecker graph model

In this section we review the SKG model and in Sections 3.3.3 and 3.3.4, we review parameter estimation in this model.

Leskovec et al. [78] introduce stochasticity in the Kronecker graphs model by letting each entry of the $N_1 \times N_1$ initiator matrix Θ_1 take values in the range $[0, 1]$ instead of binary values, representing the probability of that edge being present. If the Kronecker power of Θ_1 is computed in the manner explained above, larger and larger stochastic adjacency matrices are obtained where each entry represents the probability of that particular edge appearing in the graph. $\Theta_1^{[k]}$, therefore, defining a probability distribution on all graphs with N_1^k nodes. To obtain a stochastic Kronecker graph (SKG), an edge is independently chosen with a probability specified by the corresponding entry in the matrix.

Definition 3.3.4 (SKG). *If* Θ *is an* $N_1 \times N_1$ *probability matrix such that* $\theta_{ij} \in \Theta$ *denotes the probability that edge* (i, j) *is present,* $\theta_{ij} \in [0, 1]$*. Then the kth Kronecker power* $P = \Theta^{[k]}$*, is a stochastic matrix where each entry* $P_{uv} \in P$ *encodes the probability of edge* (u, v) *appearing. This stochastic matrix encodes a* stochastic Kronecker graph*. To obtain a graph* G^**, an instance or realization of the distribution, denoted as* $R(P)$*, an edge* (u, v) *is included in* $G^* = R(P)$ *with probability* P_{uv}*.*

Given a stochastic matrix P, and a graph G^* realized from P in the manner specified above, each edge (i, j) in G^* is picked independently by tossing a coin with a bias

specified by P_{ij}.

Notice that, G^* as defined is a directed graph, but in this work, like Gleich et al. [46] we examine modeling of undirected graphs only. If A^* is the adjacency matrix of G^*, then it may contain loops and may not necessarily be symmetric. These loops and the assymetry are removed by defining the random graph G with an adjacency matrix A such that, $A_{ij} = 0, \forall i = j$ and symmetrizing A^* by letting $A_{ij} = A^*_{i,j}$ if $i > j$ and having $A_{ji} = A^*_{ji}$ if $i < j$.

3.3.3 Parameter estimation in the SKG Model

For every graph G, $P(G)$ is the probability that a given stochastic graph model, with a given set of parameters, generates graph G. In the stochastic Kronecker graph model, probability distributions over graphs are parametrized by the initiator matrix Θ of size $N_1 \times N_1$. An appropriate size for N_1 is decided upon using standard techniques of model selection. Analysis in [78] shows that for many real-world graphs, having $N_1 > 2$ does not accrue a significant advantage as far as matching of some statistics is concerned. In this paper, we set $N_1 = 2$, to compare our results to those obtained by Gleich et al. [46].

Given a graph G that is assumed to be generated by an SKG model, we want to estimate the true parameter—the initiator matrix Θ—that generated G by an appropriate $\hat{\Theta}$. Leskovec et al. provide an algorithm that is linear in the number of edges to estimate the parameter $\hat{\Theta}$. Let G have N nodes and assume $N = N_1^k$, where the size of the initiator matrix is $N_1 \times N_1$. Using $\Theta^{[k]} = P$, P defines a SKG on N nodes: P_{uv} is the probability that there is an edge between nodes u and v. Hence, the probability $p(G|\Theta) = p(G = R(P))$ that G is a realization of P can be computed easily. The Maximum Likelihood Estimator $\hat{\Theta}$ maximizes the likelihood of realizing G. Formally, the

MLE solves:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, p(G|\Theta)$$

3.3.4 Moment based estimation of SKGs

Gleich and Owen [46] propose an alternative method to estimate SKG model parameters. They do so for reasons of computational cost of estimating the MLE of the SKG model. Leskovec et al. [77] try to approximate the MLE itself. Gleich and Owen use the so-called moment-based estimation of the model parameter, where the observed values of certain statistics of the graphs are equated with those of the expected value of these statistics over graphs that a parameter would define. They remark that "while moment methods can be statistically inefficient compared to maximum likelihood, statistical efficiency is of reduced importance for enormous samples and in settings where the dominant error is lack of fit."

Four statistics for matching (as explained above) are considered:

- number of edges (E),
- number of triangles (Δ),
- number of hairpins (2-stars or wedges) (H), and
- the number of tripins (3-stars) (T).

They consider graphs with a 2×2 initiator matrix of the form

$$\Theta = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

with $a, b, c, \in [0, 1]$ and $a \geq c$. The Kronecker structure of P makes it possible to compute closed formulae for these statistics from Θ. Given Θ of the form above,

the expected count for these statistics can be calculated explicitly. Specifically, given $P = [\Theta]^k$, closed formulae can be derived for H, Δ and T in terms of a, b, c, as follows:

$$\begin{aligned}
\mathbb{E}(E) =& \frac{1}{2}\left((a+2b+c)^k - (a+c)^k\right) \\
\mathbb{E}(H) =& \frac{1}{2}\Big(\left((a+b)^2+(b+c)^2\right)^k - 2\left(a(a+b)+c(c+b)\right)^k \\
& - (a^2+2b^2+c^2)^k + 2(a^2+c^2)^k\Big) \\
\mathbb{E}(\Delta) =& \frac{1}{6}\Big(((a^3+3b^2(a+c)+c^3)^k - 3a(a^2+b^2)+ \\
& c(b^2+c^2))^k + 2(a^3+c^3)^k\Big) \\
\mathbb{E}(T) =& \frac{1}{6}\Big(\left((a+b)^3+(b+c)^3\right)^k - 3\left(a(a+b)^2+c(b+c)^2\right)^k \\
& -3(a^3+c^3+b(a^2+c^2)+b^2(a+c)+2b^3)^k + 2(a^3+ \\
& 2b^3+c^3)^k + 5(a^3+c^3+b^2(a+c))^k \\
& +4(a^3+c^3+b(a^2+c^2))^k - 6(a^3+c^3)^k)
\end{aligned} \tag{3.1}$$

The problem then is to find an initiator matrix whose expected counts match the counts of the features $F(G)$ of the observed graph as closely as possible.

Given G, one way to choose $\hat{\Theta}$ (or equivalently, $\hat{a}$, $\hat{b}$, and $\hat{c}$) is to solve

$$\min_{a,b,c} \sum_F \frac{(F - \mathbb{E}_{a,b,c}(F))^2}{\mathbb{E}_{a,b,c}(F)}$$

where the sum is over three of four of the features $F \in \{E, \Delta, H, T\}$ and the minimization is taken over $0 \leq c \leq a \leq 1$ and $0 \leq b \leq 1$. A more general minimization method solves:

$$\min_{a,b,c} \sum_F \frac{\text{Dist}(F, \mathbb{E}_{a,b,c}(F))}{\text{Norm}(F, \mathbb{E}_{a,b,c}(F)),} \tag{3.2}$$

where Dist is either of the two distance functions:

$$\text{Dist}_{\text{Sq}}(x,y) = (x-y)^2 \text{ or } \text{Dist}_{\text{abs}}(x,y) = |x-y|$$

and Norm is one of the normalizations:

$$\text{Norm}_F = (F, \mathbb{E}) = F; \text{Norm}_{F^2}(F, \mathbb{E}) = F^2; \text{Norm}_{\mathbb{E}}(F, \mathbb{E}) = \mathbb{E}; \text{Norm}_{\mathbb{E}^2}(F; \mathbb{E}) = \mathbb{E}^2.$$

Gleich and Owen [46] find that robust results arise from the combination of Dist_{Sq} and Norm_{F^2}. The next section uses these results.

3.4 A differentially private graph estimator

We present our main result in this section. We use the results of Gleich and Owen [46] to provide a differentially private estimator of a given graph. Based on experimental results, in Section 3.4.2 we argue that a modification that makes the estimator differentially private does not destroy the desirable properties of the graph model estimator for both some real-world and synthetic networks.

3.4.1 Differential privacy for graphs

After a private estimator is computed, we may publish it and sample graphs from this distribution to compute an approximation of relevant statistics. Under the assumption that the model captures the essential properties of the graph, our estimator will define a probability distribution from which we can sample graphs that are "similar" to the original graph G. We emphasize here that we rely upon the results of [77] to justify using the SKG model to maintain "similarity" of synthetic graphs to the original graphs. Our private estimator suffers from the same limitation that the SKG does in capturing properties of a real-world network but also demonstrates almost the same accuracy. In this section, we present our main result showing how to compute an estimator for the SKG model that is also differentially private. We first formalize the idea that the output of the estimator should not change significantly if a link between two individuals is included or excluded from the observations.

Definition 3.4.1 (Edge neighborhood [51, 93])**.** *Given a graph $G(V, E)$, the (edge)* neighborhood of a graph *is the set*

$$\Gamma(G) = \{G'(V, E') \; s.t \; |E \oplus E'| = 1\}$$

Applying the standard definition of differential privacy to graphs instead of databases and using the above definition of neighborhood yields the following:

Definition 3.4.2 (Edge differential privacy [95])**.** *A parameter estimation algorithm that takes as input a graph G, and outputs $\tilde{\Theta}(G)$, preserves* (ε, δ)-differential edge privacy *if for all closed subsets S of the output parameter space, and all pairs of neighboring graphs G and G', and for all $\delta \in [0, 1]$,*

$$\Pr[\tilde{\Theta}(G) \in S] \leq \exp(\varepsilon) \cdot \Pr[\tilde{\Theta}(G') \in S] + \delta$$

The original notion of ε-differential privacy is a special case of the (ε, δ)-differential privacy in which $\delta = 0$.

Hay et al. [51] also define *node differential privacy*, by analogously defining the notion of *node neighborhood* of a graph. Two graphs are node neighbors if they differ by at most one node and all the incident edges. This notion of privacy is highly restrictive when trying to compute accurate approximations of graph statistics because of potentially high degree nodes and the loss of information that would accompany their deletion. To provide some degree of privacy to nodes, Hay et al. [51] introduce the notion of *k-edge differential privacy*. In k-edge differential privacy, graphs G and G' are *k-edge neighbors* if $|V \oplus V'| + |E \oplus E'| \leq k$. They also make the observation that any algorithm that provides ε-edge privacy with respect to 1-edge neighbors, will provide $k\varepsilon$-edge privacy with respect to k-edge neighbors using a well-known composition theorem (restated here as Theorem 3.4.8). In this work, we only examine 1-edge differential privacy.

According to Definition 3.4.2, for a graph estimator that preserves differential privacy, outputs of the estimating algorithm do not become significantly more or less likely if an edge is included or excluded from the database. If the inclusion or exclusion of a single link between individuals cannot change the output distribution appreciably, even an adversary who may have additional background information will not, by interacting with the algorithm, learn significantly more about an individual than could be learned about this individual otherwise.

Dwork et al. [33] and Nissim et al. [95] define the notions of *local sensitivity* and *global sensitivity*:

Definition 3.4.3 (Local Sensitivity [95])**.** *The* local sensitivity *of* $f : D \to \mathbb{R}$*, that maps a Domain* D *to reals, at* $G \in D$ *is*

$$\mathrm{LS}_f(G) := \max_{G' \text{ s.t. } G' \in \Gamma(G)} \|f(G) - f(G')\|_1$$

As an example, when computing the local sensitivity of the number of triangles in a graph G having N nodes, the domain D is the space of all graphs on N nodes. Here we express the notion of global sensitivity introduced in Definition 2.3.1 in terms of the local sensitivity.

Definition 3.4.4 (Global Sensitivity [33])**.** *The*global sensitivity *of a function of a graph* G*,* $f : D \to \mathbb{R}^\ell$ *is*

$$\mathrm{GS}_f := \max_{G \in D} \mathrm{LS}_f(G)$$

Using these notions we compute a differentially private estimator based on matching the expected count to the observed counts of the statistics—we supply differentially private approximations of the statistics E, H, Δ and T to Equation 3.2. We do this by computing differentially private approximation to the degree sequence vector of G and the number of triangles in G.

Let d be the vector of degrees of G, such that d_i is the degree of node i of graph G. Let d be sorted to yield d_S such that $d_S(i)$ is the i-th smallest degree. Hay et al. [51] propose a method of computing a differentially private approximation $\tilde{d}$ of the sorted degree vector d_S by adding a vector of appropriate Laplacian noise to d_S and then using post-processing techniques that they experimetally show to be a highly accurate approximation of d_S. Let $\langle \text{Lap}(\sigma) \rangle^N$ denote a N length vector of independent random samples from a Laplace distribution with mean zero and scale σ. We know that the global sensitivity of d_S, GS_d is equal to 2.

Hay et al. [51] use the Laplacian mechanism (Theorem 2.3.3) to compute a "noisy" degree sequence $\hat{d}$ as an approximation of d_S:

$$\hat{d} = d_S + \langle \text{Lap}(2/\varepsilon) \rangle^N.$$

Therefore, $\hat{d}$ is then an $(\varepsilon, 0)$-differentially private approximation of d_S. Hay et al. [51] use post-processing techniques that seek to "remove some of the extra noise" in $\hat{d}$, to compute a $\tilde{d}$ that is experimentally and theoretically shown to provide higher accuracy. Using $\tilde{d}$, we compute $(\varepsilon, 0)$-differentially private approximations of E, H, and T in the following manner:

$\tilde{E} = \frac{1}{2}\sum_i \tilde{d}_i$; $\tilde{H} = \frac{1}{2}\sum_i \tilde{d}_i(\tilde{d}_i - 1)$ and $\tilde{T} = \frac{1}{6}\sum_i \tilde{d}_i(\tilde{d}_i - 1)(\tilde{d}_i - 2)$. Hence, we have:

Fact 3.4.5. *Computing $\tilde{E}, \tilde{H}$ and $\tilde{T}$ using $\tilde{d}$ is $(\varepsilon, 0)$-differentially private.*

This is straightforward, as computing $\tilde{d}$ is $(\varepsilon, 0)$-differentially private. Since the number Δ of triangles is not a simple function of the degree distribution, we instead use the tehcniques of Nissim et al. [95] to compute an (ε, δ)-differentially private approximation of Δ. To reduce the amount of noise that needs to be added to compute an approximation to Δ, Nissim et al. [95] use an upper bound on the local sensitivity of $\Delta(G)$ by computing the *β-smooth sensitivity* of $\Delta(G)$.

Let $\text{dist}(G, G')$ be the symmetric difference between the edge sets of graphs G and G'. Hence, if G and G' are neighbors by Definition 4.3.1, $\text{dist}(G, G') = 1$.

Definition 3.4.6 (β-smooth sensitivity [95])**.**
For $\beta > 0$, the β-smooth sensitivity of f at G, is

$$\text{SS}_{\beta,f}(G) = \max_{G'} \left(\text{LS}_f(G) \cdot e^{-\beta \, \text{dist}(G;G)} \right)$$

The smooth sensitivity can be used to compute a differentially private approximation to a function f:

Theorem 3.4.7 ([95])**.** *Let $f : D^n \to R$ be any real-valued query function from an input $x \in D^n$ for some domain D,and let $\text{SS}_{\beta,f} : D^n \to R$ be the β-smooth sensitivity of f for some $\beta > 0$. Then, if $\beta < \frac{\varepsilon}{2\ln(2/\delta)}$ and $\delta \in (0, 1)$, the algorithm that outputs $\tilde{f} = f(D) + 2\frac{\text{SS}_{\beta,f}(D)}{\varepsilon}.\eta$, where $\eta \sim \text{Lap}(1)$, is (ε, δ)-differentially private.*

Algorithm 1 illustrates the process we adopt. Our results use the above theorem and the serial composition theorem 2.2.4 restated here:

Theorem 3.4.8 (Composition theorem [32])**.**
Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\ell$, be ℓ number of (ε, δ)-differentially private mechanisms computed using graph G. Then any mechanism $\mathcal{M}$ that is a composition of $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\ell$, is $(\ell\varepsilon, \ell\delta)$-differentially private.

Using these results we compute an (ε, δ)-differentially private approximation of Δ by outputting:

$$\tilde{\Delta} = \Delta + 2\frac{\text{SS}_{\beta,\Delta}\, 2}{\varepsilon}. \, \text{Lap}(1),$$

as an (ε, δ)-differentially private approximation to the number of triangles in G. Using Theorems 3.4.8, 3.4.7, and Fact 3.4.5, we have

Theorem 3.4.9.
The computation of $\tilde{F} = \{\tilde{E}, \tilde{H}, \tilde{T}, \tilde{\Delta}\}$ is $(2\varepsilon, \delta)$ differentially private.

Using these private statistics $\tilde{F}$, in the moment-matching algorithm of Gleich and Owen (Equation 3.2), we obtains a differentially private estimator. Algorithm 1 illustrates the process. Hence, we have:

Corollary 3.4.10. *$\tilde{\Theta}$ computed by Algorithm 1 is (ε, δ)-differentially private.*

Algorithm 1 Differentially private estimation of $\tilde{\Theta}$

Input: Graph G, privacy parameters (ε, δ)

1. Compute the degree vector d of G.
2. Using Hay et al. [51] compute a $\varepsilon/2$-differentially private approximation of d, $\tilde{d}$
3. Compute $\tilde{E}, \tilde{H}, \tilde{T}$ from $\tilde{d}$.
4. Compute the smooth sensitivity $\mathrm{SS}_{\beta,\Delta}$ of Δ
5. Use $\mathrm{SS}(G)$ to compute an $(\varepsilon/2, \delta)$ private approximation of Δ, $\tilde{\Delta}$.
6. Use the Kronecker Moment Estimation of [46] with $\{\tilde{E}, \tilde{H}, \tilde{T}, \tilde{\Delta}\}$ as inputs to Equation 3.2 to compute $\tilde{\Theta}$.

Output: $\tilde{\Theta}$

3.4.2 Experimental results

In this section, we discuss application of Algorithm 1 to three real-world networks and two synthetic Kronecker graphs. CA-GrQC and CA-HepTh are co-authorship networks from arXiv [77]. The nodes of the network represent authors, and there is an edge between two nodes when the authors jointly wrote a paper. AS20 is a real-world technological infrastructure network [77]. Each node represents a router on the internet and edges represent a physical or virtual connection between the routers. All these graphs are naturally undirected and all edges are unweighted. We downloaded these networks from Snap [76] and used the provided library for our experiments.

Network	**KronFit** (a, b, c)	**KronMom** (a, b, c)	**Private** (a, b, c)
CA-GrQC	0.999 0.245 0.691	1.000 0.4674 0.2790	1.000 0.4618 0.2930
CA-HepTh	0.999 0.271 0.587	1.000 0.4012 0.3789	1.000 0.4048 0.3720
AS20	0.987 0.571 0.049	1.000 0.6300 0.000	1.000 0.6286 0.000
Synthetic $\Theta = [.99.45; .45.25]$	0.9523 0.4743 0.2493	0.9894 0.5396 0.2388	0.9924 0.5343 0.2466

Table 3.1: Comparison of parameter estimation for $\varepsilon = 0.2, \delta = 0.01$

We also used the code provided by Gleich [45] to compute both the private and non-private moment-based estimators of the networks. Table 3.1 compares the results of Algorithm 1 (column titled "Private") to those of Gleich et al. [46] ("KronMom") and Leskovec et al. [77] ("KronFit"). Our results are based on Gleich et al.'s results, so it is not surprising that our results are close to theirs—we observe that the private parameters we compute are very similar. To provide a reasonable comparison, for each of the graphs, we use the same Dist and Norm functions in the parameter estimation of Equation 3.2 as in Gleich and Owen.

For the synthetic Kronecker graph we start with an initiator matrix

$$\Theta = \begin{pmatrix} 0.99 & 0.45 \\ 0.45 & 0.25 \end{pmatrix}$$

and $k = 14$ to obtain a synthetic graph on 2^{14} nodes. Then we try to recover the parameters of this synthetic graph by running all three algorithms on it. From Table 3.1, we see that all three algorithms do a satisfactory job in recovering the parameter when the modeling assumption is true, that is when the graph indeed is a stochastic Kronecker graph.

To further understand how well the private estimator captures various properties of the graph, we carry out further experiments. All experiments are conducted for $(0.2, 0.1)$-differential privacy. Using the parameter estimates of a graph, we generate 100 synthetic graphs from the estimated parameters for all three methods, and compute various expected statistics over these 100 graphs. These statistics have been computed in [77] for these graphs, so we compare the performance of our private estimator on these statistics to Leskovec et al.'s results. We summarize these statistics briefly:

1. The *degree distribution* plots the distribution of the degrees of the nodes.

2. The *Hop-plot* plots the number of reachable pairs of noded within h hops, as a function of the number of hops h.

3. The *Scree plot* plots the eigenvalues (or singular values) of the graph adjacency matrix, versus their rank, using the logarithmic scale.

4. The *Network values* plots the distribution of eigenvector components (indicators of "network value") associated with the largest eigenvalue of the graph adjacency matrix.

5. The *average clustering coefficient* plotted as a function of the node degree. The clustering coefficient is a measure of the extent to which nodes in a graph tend to cluster together.

For each of these graphs we plot these statistics. "*Original*" refers to the original graph, "*KronFit*" refers to a single synthetic Kronecker graph generated from the parameter $\hat{\Theta}$ which is computed using the KronFit algorithm of Leskovec et al. [78]. "*KronMom*" refers to a single synthetic Kronecker graph generated from the parameter $\hat{\Theta}$ that is computed using the "*KronMom*", moment-matching algorithm of Gleich and Owen [46]. "*Private*" refers to a single Kronecker graph generated from $\tilde{\Theta}$ computed

in Algorithm 1. The prefix "*Expected*" refers to the expected value of the statistics being computed over 100 synthetic realizations of the appropriate Kronecker graphs.

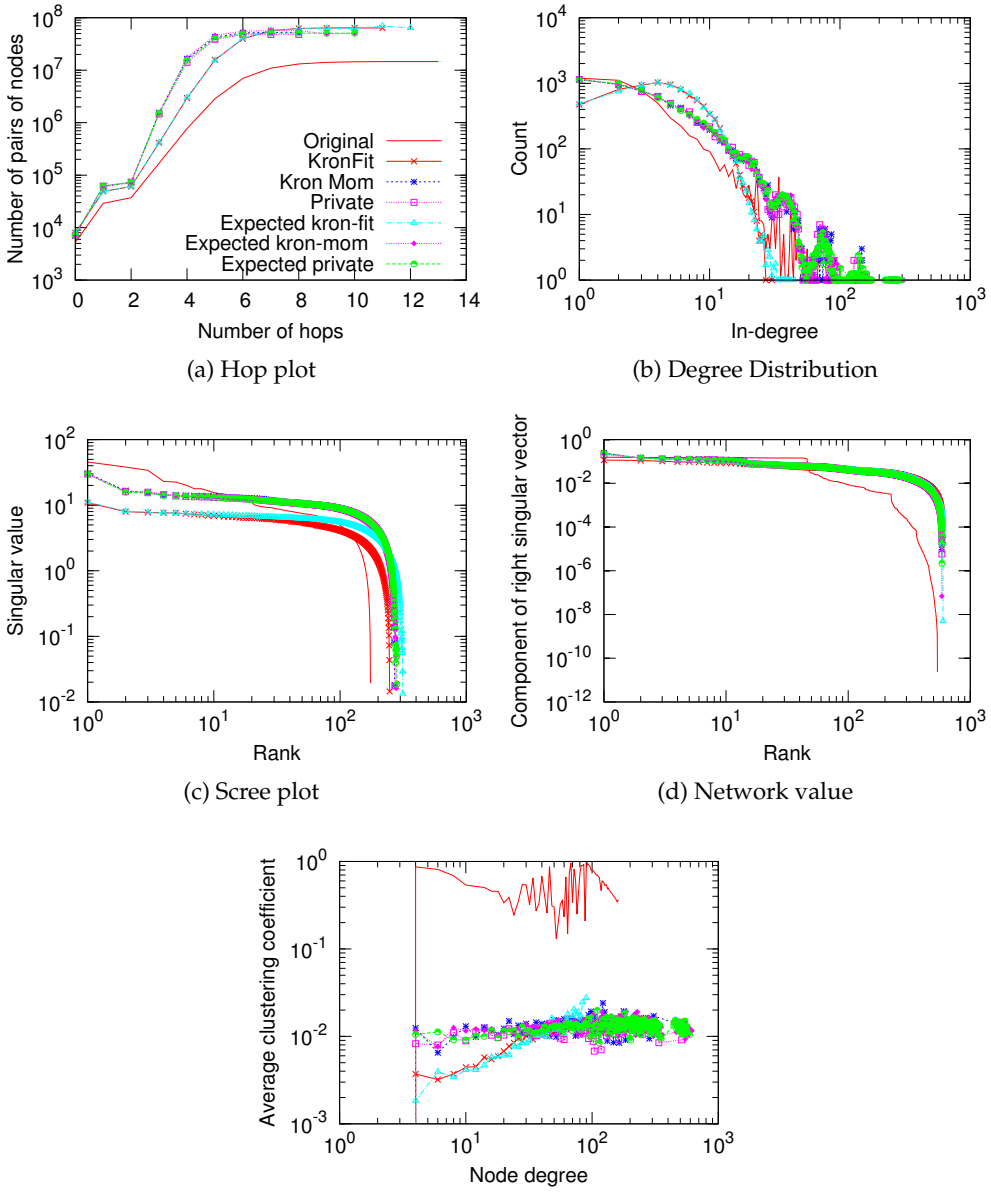
From Figure 3.1, we notice that the observed statistics for a single realization are very close to the expected values, hence one realization appears to give us a representative sample, at least for these four graphs.

To reduce clutter, for CA-HepTh (Figure 3.3), AS20 (Figure 3.2), and the synthetic Kronecker graph (Figure 3.4), we only show single realizations. We observe that in all four cases, the statistics are well-approximated and very close to the "predictions" made by both the "KronFit" and the "KronMom" estimators. In the case of the synthetic Kronecker graph we also observe a good matching of the average clustering coefficient which is usually not the case for real-world networks. This has to do with modeling assumptions. We see that the SKG models the clustering coefficient well for AS20 but not for CA-GrQC and CA-HepTh. The private estimators are also observed to perform comparably.

3.5 Conclusions and future work

We applied the rigorous differential privacy framework to problems of generating synthetic graphs that can be made publicly available for research purposes while providing privacy to the individual participants. We built upon the work of Leskovec et al. [77, 78] and Gleich and Owen [46] in the generative Kronecker graph model to demonstrate that synthetic graphs that are statistically similar to the original sensitive graphs can be generated in a manner that is differentially private. While we used a specific model and a specific estimator, our work can be broadly placed in the framework of private parametric estimation for graph models.

There are several future directions for future work. A comparison of our results to those of Sala et al. [109] seems most relevant. An empirical study of the smooth



(a) Hop plot

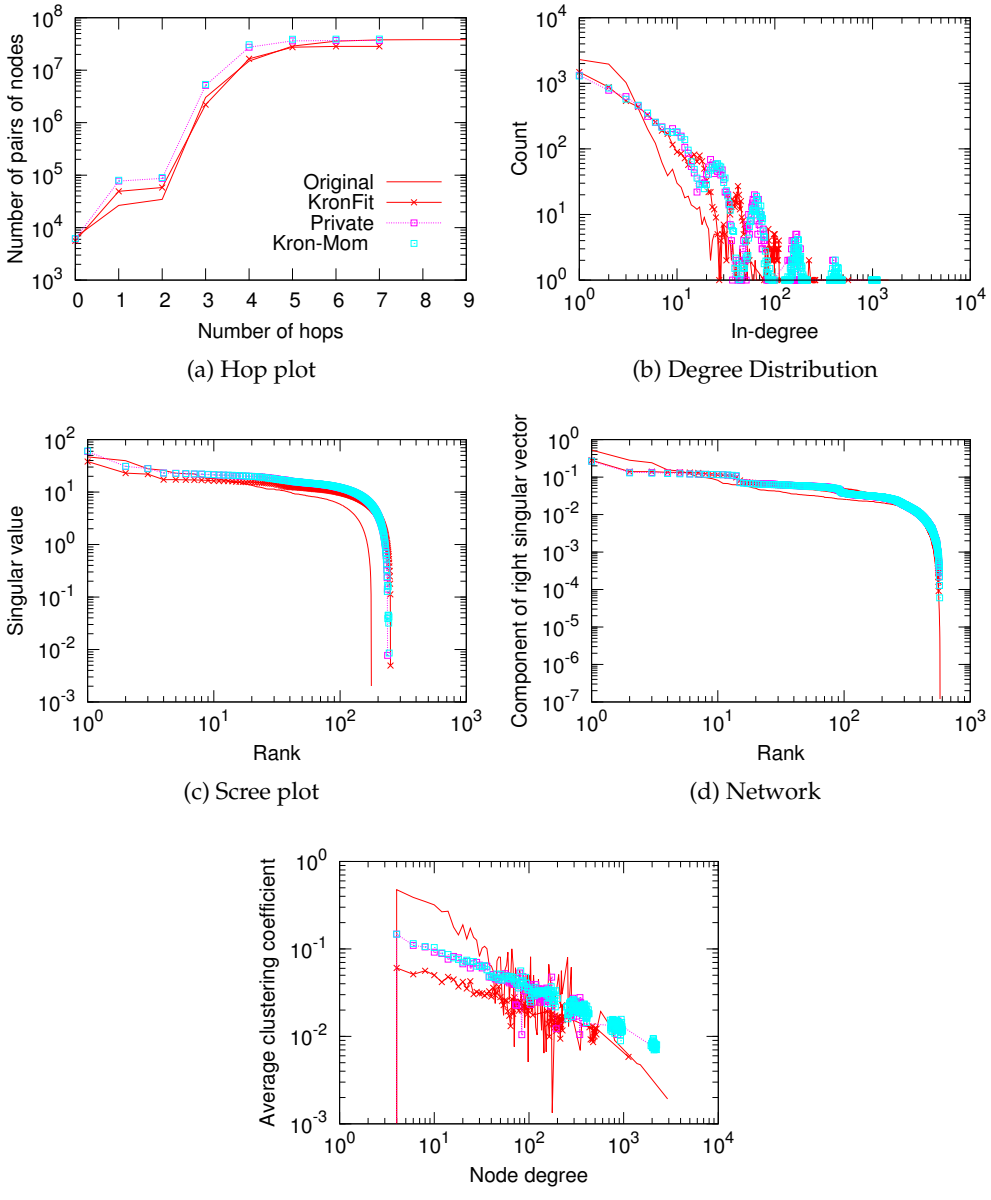
(b) Degree Distribution

(c) Scree plot

(d) Network value

(e) Clustering

Figure 3.1: Overlayed patterns of real network for CA-GrQC ($N = 5,242, E = 28,980$) and the estimated synthetic Kronecker graph using the three different estimators.



(a) Hop plot

(b) Degree Distribution

(c) Scree plot

(d) Network

(e) Clustering

Figure 3.2: Overlayed patterns of real network for AS20 ($N = 6,474,\ E = 26,467$) and the estimated synthetic Kronecker graphs using the three different estimators.

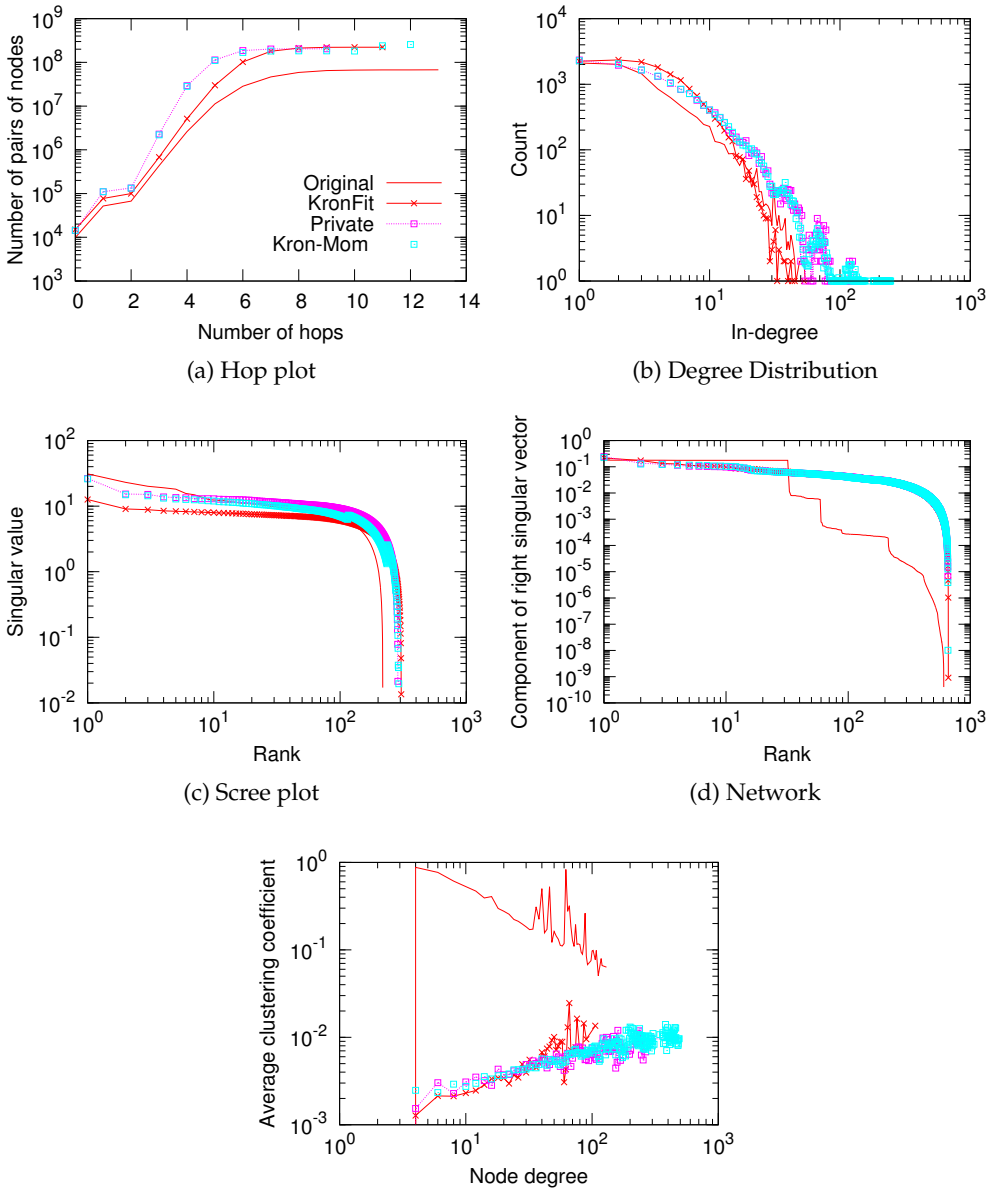
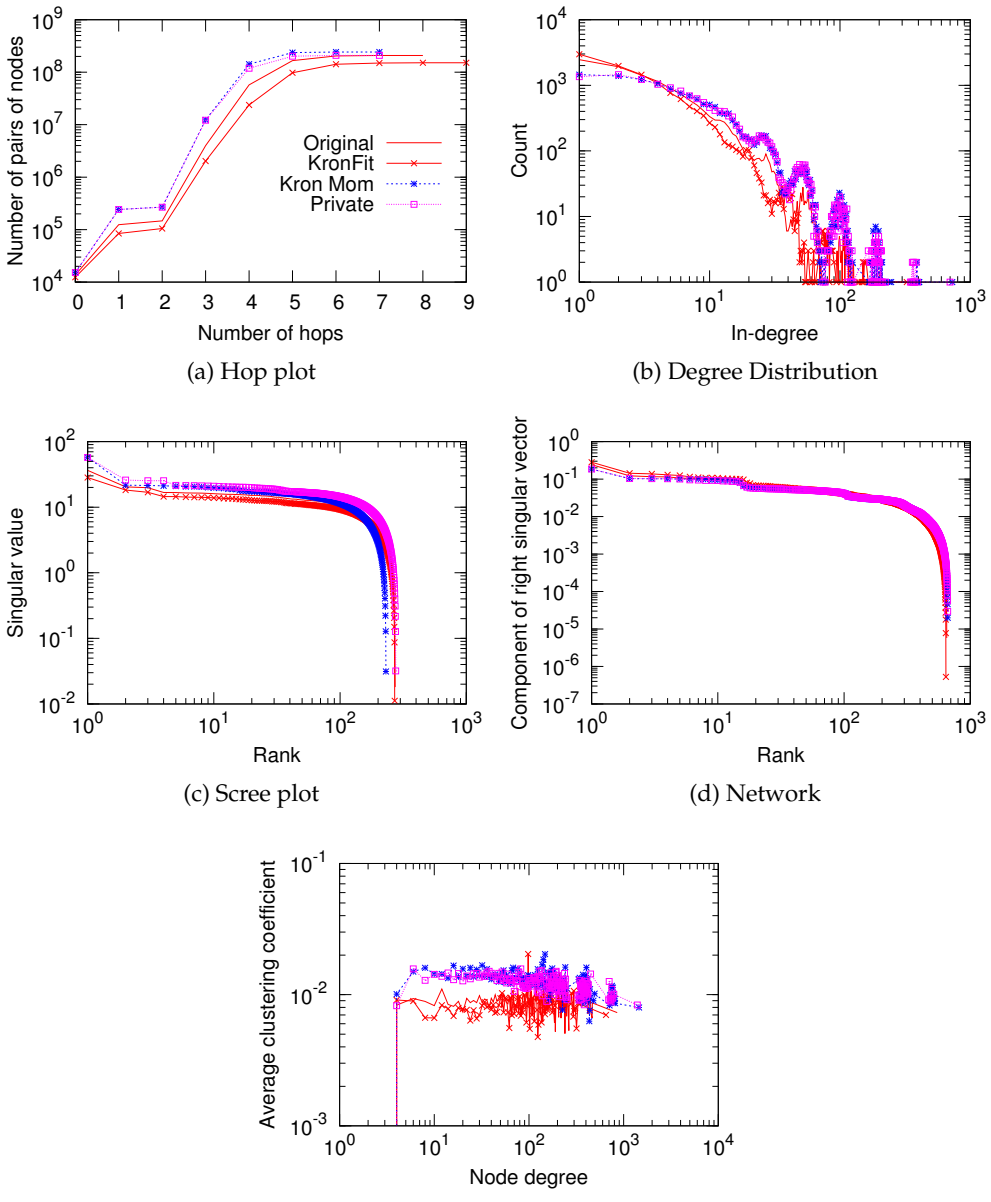


Figure 3.3: Overlayed patterns of real network for CA-HepTh ($N = 9,877$, $E = 51,971$) and the estimated synthetic Kronecker graph using the three different estimators.



(a) Hop plot

(b) Degree Distribution

(c) Scree plot

(d) Network

(e) Clustering

Figure 3.4: Overlayed patterns of a synthetic source Kronecker network and the estimated sythetic Kronecker graph using the three different estimators.

sensitivity of the number of triangles in the SKG is another direction we are currently pursuing. Nissim et al. [95] propose an upper bound on the smooth sensitivity of the number of triangles in the $G(n, p)$ Erdos-Renyi model. It would be interesting to examine the smooth sensitivity of Δ as a function of the size of the graph G. Preliminary experiments indicate that in the SKG model, SS_Δ might grow slowly. Yet another direction that presents itself is to examine private estimation in other graph models such as the *Exponential Random Graph Model* (ERGM) [105], especially since the results of Karwa et al. [64] provide accurate differentially private approximations to statistics used in ERGM estimation.

4

Differentially Private Modeling of Human Mobility at Metropolitan Scales

"In a dataset where the location of an individual is specified hourly, and with a spatial resolution equal to that given by the carrier's antennas, four spatio-temporal points are enough to uniquely identify 95% of the individuals."

– Yves-Alexandre de Montjoye et al. [26]

4.1 Introduction

Models of human mobility have wide applicability to infrastructure and resource planning, analysis of infectious disease dynamics, ecology, and more. The abundance of spatiotemporal data from cellular telephone networks affords new opportunities to construct such models. Furthermore, such data can be gathered with greater detail at larger scale and lower cost than traditional methods, for example a census survey.

Prior work introduced the WHERE (Work and Home Extracted REgions) approach to mobility modeling [62]. In WHERE, aggregated collections of cellphone Call Detail Records (CDRs) form the basis of a mobility model that can be used to characterize a city's commute patterns and enable the exploration of what-if scenarios regarding changes in residential density, telecommuting popularity, etc. Starting with CDRs from a cellular telephone network that have gone through a straightforward anonymization procedure, WHERE produces synthetic CDRs for a synthetic population. WHERE has been experimentally validated against billions of location samples

for hundreds of thousands of cell phones in the New York and Los Angeles metropolitan areas.

While human mobility models have the potential for great societal benefits, privacy concerns regarding their use of individuals' location data have inhibited their release and wider use. Despite the fact that WHERE intuitively provides some privacy because it rests on aggregated distributions of sampled and anonymized data, a more rigorous assurance of privacy can further advance safe and widespread use of such techniques.

In this chapter, we present and evaluate DP-WHERE, a *differentially private* version of WHERE. DP-WHERE satisfies the rigorous requirements of differential privacy while retaining WHERE's usefulness for predicting movement of human populations in metropolitan areas. Overall, our work demonstrates that modest revisions to a mobility model drawn from real-world and large-scale location data allow for rigorous demonstrations of its privacy without overly compromising its utility. Specific contributions of our work include the following:

- We produce and evaluate a differentially private approach for modeling human mobility based on large sets of cellular network data.

- Our experiments show that differential privacy can be achieved for a modest reduction in accuracy. In particular, across a wide array of experiments involving 10,000 synthetic users moving across more than 14,000 square miles, the distance between synthetic and real population density distributions for DP-WHERE differed by only 0.17–2.2 miles from those of the original WHERE approach.

- More broadly, this work shows that there is reason for optimism regarding the judicious use of Big Data repositories of potentially sensitive information. We show the value of a multi-pronged approach to privacy: Our model starts with

attributes (such as sampling and aggregation) that make it intrinsically well suited to offering some intuitive degree of privacy. We subsequently modify the steps of the modeling algorithm to rigorously implement differential privacy.

Figure 4.1 shows an overview of DP-WHERE and its changes to WHERE. We discuss related work in Section 4.2. We provide the necessary background on WHERE in Section 4.3.1. In Section 4.4 we describe the DP-WHERE algorithm in detail. In Section 4.5 we present our evaluation of the utility of DP-WHERE.

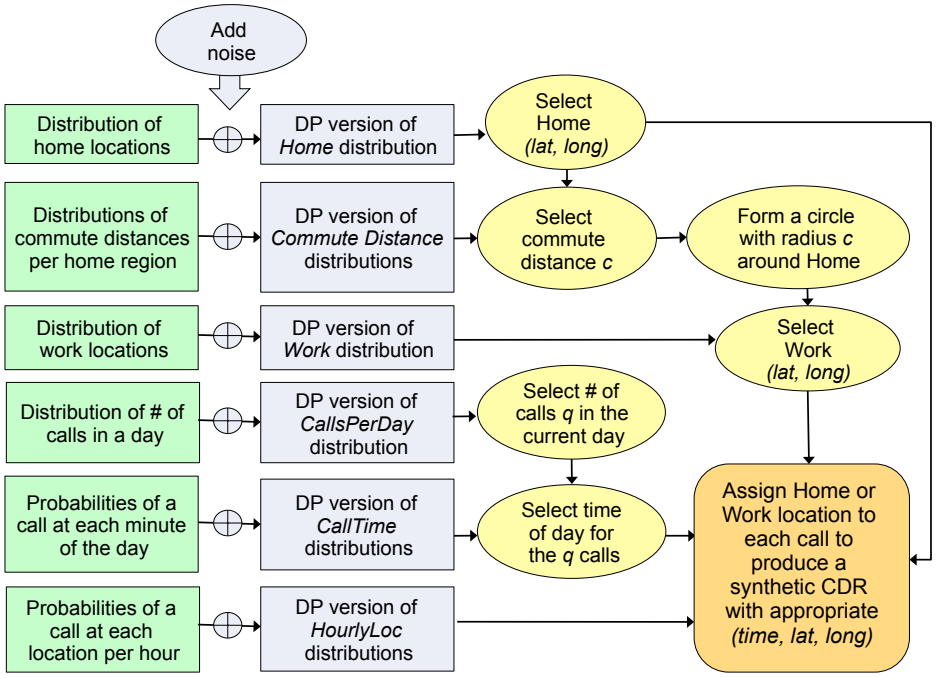


Figure 4.1: Overview of DP-WHERE, which modifies WHERE by adding noise to achieve differentially private versions of the input probability distributions. The rest of WHERE remains unchanged.

4.2 Related work

Mobility Modeling: Characterizing human mobility based on cellular network or

other position data has received considerable attention. Previous work developed algorithms for inferring important locations (for example, home and work) from anonymized cellular network data, and used such information to characterize metropolitan behaviors like commute patterns, to quantify carbon footprints, and to create the WHERE mobility model [59, 60, 61, 62].

Early mobility modeling work used either handheld GPSs or WiFi association behavior to model human mobility at much smaller scales, and with little privacy [55, 70, 104]. Previous uses of cellular data has also included some mobility modeling [42, 43, 44, 48, 113], but with little attention to formal privacy assurances. Such studies use at most anonymization and aggregation, and in some cases, actually point to data characteristics that increase the difficulty of creating privacy-preserving mobility models.

Privacy: To the best of our knowledge, the problem of creating differentially private human mobility models based on real-world cellular network data has not been studied previously. Differential privacy has been examined in other contexts of spatio-temporal data. Chen et al. [19] study the problem of publishing a differentially private version of the trajectory data of commuters in Montreal. They then evaluate the utility of published private data in terms of count queries and frequent sequential pattern mining. Similarly, [26] recently characterized sequences of movements from individual users and found them extremely resistant to privacy techniques. In contrast, WHERE does not directly model the sequentiality of the spatio-temporal data at the level of an individual. Some work [54, 101] considers aspects of differential privacy on spatial data, but without DP-WHERE's end-to-end treatment. Other characterization work also exists. Several recent papers have characterized the privacy risks of releasing location data, in each case demonstrating the ability to re-identify individual information from geospatial data sets [47, 74, 123]. These papers motivated us to look

beyond a simple anonymization of location traces. In addition, Andrés et al. [6] introduce the notion of *geo-indistinguishability* in location-based systems, which protects the exact location of a user while allowing release of information needed to gain access to a service.

4.3 Background

In this section, we provide background for WHERE and the application of differential privacy to WHERE.

4.3.1 WHERE

DP-WHERE is based on WHERE, which produces models of how large populations move within different real or hypothetical metropolitan areas [62]. WHERE generates sequences of times and associated locations that aim to capture how people move between important places in their lives, such as home and work. Previous work has shown that people spend most of their time at a few such places [48, 59, 113]. WHERE aggregates the movements of many synthetic individuals to reproduce human densities over time at the geographic scale of metropolitan areas.

WHERE draws information from either CDR traces or public sources (for example, the US Census Bureau). It then creates a set of probability distributions that it uses to "drive" the generation of synthetic CDRs for the region being modeled. This paper uses as its starting point the version of WHERE that uses CDR traces as its data source. As shown in Figure 4.6, this source yields substantially better experimental results than using current publicly available data sources.

The WHERE modeling algorithm takes as input a database of simplified CDRs. (Complete CDRs contain details not relevant to mobility, for example, call-termination codes.) Each row of this database corresponds to a single voice call or text message,

both of which we refer to interchangeably as calls. WHERE thus uses a database D of m entries corresponding to calls made by n distinct users. Each user is indexed by a unique anonymized user ID in the set $[n] = \{1, 2, \ldots n\}$. The calls were made in a given metropolitan area divided into smaller geographic areas by imposing a square grid of $d \times d$ cells.

WHERE leverages earlier work that estimates important places in people's lives (for example, home and work) by applying clustering and regression methods to the CDRs in D [59]. In order to work with a single database in DP-WHERE, we append to each CDR entry these inferred home and work locations for the corresponding user. Thus, for the purposes of DP-WHERE, each row of D contains the following fields:

id	date	time	lat	long	home	work

At its core, WHERE uses D to construct cumulative distribution functions (CDFs) for the following probability distributions (see also Figure 4.1):

Home **and** *Work*

For each grid cell, all users with inferred `home` locations in that grid cell are counted (and normalized) to produce a probability distribution *Home* over the grid cells. Similarly, a *Work* distribution is constructed from the inferred `work` locations of users in the database.

CommuteDistance

WHERE allows for a coarser grid to be used for commute distances than for `home` and `work` locations by merging adjoining cells in the underlying $d \times d$ grid to yield a $d_c \times d_c$ grid. We refer to this coarser grid as the *commute grid*. For each cell in the commute grid, WHERE creates an empirical distribution of commute distances (that is, distance

between `home` and `work`) for people whose `home` locations are in that grid cell, leading to a total of d_c^2 of these *CommuteDistance* distributions.

CallsPerDay

WHERE computes an empirical distribution *CallsPerDay* over the set $\mathbb{C} = \{\mu_{\min}, \ldots \mu_{\max}\} \times \{\sigma_{\min}, \ldots \sigma_{\max}\}$ of possible rounded values of means and standard deviations of numbers of calls per day made by users.

ClassProb **and** *CallTime*

For each user in D, WHERE computes the distribution of when calls are made throughout the day. These per-user distributions are then combined using X-Means clustering into two classes [62]. Each user belongs to one of two user-classes with a probability specified by *ClassProb*. Subsequently, using the CDR database, per-minute call probability distributions *CallTime* are computed separately for each user class.

HourlyLocs

For each hour of the day, WHERE computes a distribution of calls made over the grid cells. Each of those 24 distributions reflects the probability of users being at a given location during that hour. The *HourlyLocs* distributions are not tied to a specific user, but represent the calling activity across the entire metropolitan area during each hour.

As shown in Figure 4.1, subsequent stages of WHERE use the above distributions to produce synthetic CDRs for any number of synthetic users and a time period of any duration. WHERE generates a synthetic user as follows. It first selects a home location by sampling from *Home*. It then selects a commute distance c by sampling *CommuteDistance* c for the region the home lies in. Finally, it selects a work location by sampling from *Work* while restricted to locations at distance c from the home location.

WHERE then generates synthetic call times and locations for a synthetic user i as follows. First, it samples from *CallsPerDay* to obtain a (μ_i, σ_i) tuple that represents i's calling frequency and can itself be viewed as a distribution. Second, it samples from the normal distribution with a mean μ_i and standard deviation σ_i to determine the number of calls q that i makes in the current simulated day. Third, it samples from *ClassProb* to assign i one of two classes of calling time patterns. Fourth, it samples *CallTime* to select the times of day for the q calls that day. Finally, it samples *HourlyLocs* to determine the locations of these calls while restricted to the user's `home` and `work` locations.

The synthetic CDR traces that comprise the output of WHERE have been shown to agree closely across a variety of metrics with real-world CDR traces for hundreds of thousands of users moving over metropolitan regions of thousands of square miles [62].

4.3.2 Differential privacy for Call Detail Records databases

Differential privacy relies on the notion of neighboring databases [33]—in our context, two neighboring CDR databases. Intuitively, two databases are neighbors if they differ only in one individual's data.

Definition 4.3.1 (Neighbors)**.** *Two CDR databases D and D' are* neighbors *if $D \subset D'$ and there is some $k \in [n]$ such that for every record $r \in D' \oplus D$, $\mathtt{id}(r) = k$ (where $\mathtt{id}(r)$ denotes the user id in r).*

That is, neighboring CDR databases D and D' differ in the records of exactly one user (who may have made many calls).

Differential privacy for multi-step algorithms can be provided by breaking the algorithm down into multiple interactions with the database, each of which is itself differentially private. We use the parallel and serial composition theorems from Chapter 2 (Theorems 2.2.4 and 2.2.5) for such algorithms in this work.

4.4 Differentially private WHERE

Our new approach, DP-WHERE, modifies WHERE to provide differential privacy, while still offering the high accuracy of the original approach. As described in Section 4.3.1, WHERE creates and samples from several spatio-temporal distributions. Our approach is to render each of these empirical distributions ε_i-differentially private, using different values of ε_i, and then to apply Theorems 2.2.4 and 2.2.5 to arrive at an ε-differentially private modeling algorithm, where $\varepsilon = \sum_i \varepsilon_i$. For each distribution, we specify a privacy "budget" ε_i that will not be exceeded. The remainder of this section describes our methodology in detail.

4.4.1 Pre-processing

Before the algorithm executes, we perform a pre-processing step that removes all users who make more than a maximum threshold MaxCallsHr of calls per hour. This limits the impact of any one user on the dataset. Our experimental evaluation sets MaxCallsHr to 120 which makes it likely that any filtered caller is an auto-dialer; Section 4.5 shows it yields good results.

4.4.2 Distributions

Home and Work

We compute differentially private empirical CDFs for *Home* and *Work*. Let $\varepsilon_{\texttt{home}}$ and $\varepsilon_{\texttt{work}}$ be the privacy budgets allocated to computing *Home* and *Work*, respectively. CountHomeNum(i) is defined as the function that returns the number of distinct users in the database D with homes in the ith grid cell (in the chosen canonical ordering). Note that the global sensitivity of the vector (Definition 2.3.1) of $\langle$CountHomeNum(1), $\ldots$, CountHomeNum$(d^2)\rangle$ is 2, since each user can change his home location from grid

cell i to another grid cell j, reducing the count in grid cell i by 1 and increasing j's count by 1. Applying the Laplace mechanism described in Theorem 2.3.3, Algorithm 2 provides an $\varepsilon_{\texttt{home}}$-differentially private approximation of *Home*. Similarly, the Laplace mechanism achieves an $\varepsilon_{\texttt{work}}$-differentially private empirical CDF for *Work*.

Algorithm 2 Algorithm to compute an $\varepsilon_{\texttt{home}}$-differentially private CDF of the *Home* distribution.

DPhomeCDF(D, $\varepsilon_{\texttt{home}}$)
Count $\leftarrow 0$
for $i \leftarrow 1$ to d^2 **do**
  Count $\leftarrow$ Count + CountHomeNum(i) + Lap $\left(0, \frac{2}{\varepsilon_{\texttt{home}}}\right)$
  CDF$[i] \leftarrow$ Count
end for
CDF $\leftarrow$ PostProc(CDF)

return CDF

The noisy CDF does not correspond to a legitimate probability distribution, as the noisy counts are not necessarily non-decreasing. We use Hay et al.'s post-processing techniques [53] to "clean up" this noise and create a legitimate (non-decreasing) CDF, denoted by PostProc in Algorithm 2. The postprocessing method does not need to access the original private data, so Theorems 2.3.3 and 2.2.5 imply:

Lemma 4.4.1. *Algorithm 2 is $\varepsilon_{\texttt{home}}$-differentially private. The equivalent algorithm for Work is $\varepsilon_{\texttt{work}}$-differentially private.*

Figure 4.2 shows the CDFs of the *Home* distribution for different values of $\varepsilon_{\texttt{home}}$ and the original empirical CDF. (The dataset and parameters used for the figures are described in detail in Section 4.5.) The private version of *Home* is very close to its non-private counterparts even for very low values of $\varepsilon_{\texttt{home}}$. Only for extreme values of ε such as 0.000001 are the differences even noticeable at this graph scale.

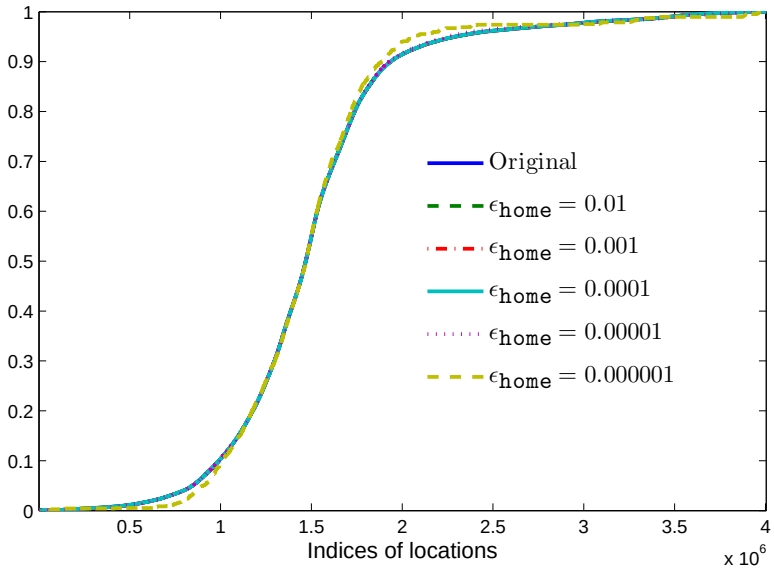


Figure 4.2: CDF of *Home* distribution for different values of $\varepsilon_{\texttt{home}}$.

Commute distance

As in WHERE, we impose a $d_c \times d_c$ grid on the geographical area. We first create a data structure D_i that contains the counts of commute distances of users (in CDR database D) with `home` locations in grid cell i. In order to avoid having empty grid cells (in a data-oblivious manner so that we do not incur a privacy budget expenditure), we add two commute distances (of 0 and 0.1 miles) to every grid cell.

In WHERE, the CDF of the commute distribution is constructed by using the actual commute distances as histogram bins. In DP-WHERE, for privacy reasons, we cannot use the actual commute distances of people living in a grid cell as histogram bins. Instead, as shown in Algorithm 3, we create a per-commute-grid-cell data-dependent histogram of commute distances in a differentially private way, and then sample from this (normalized) histogram. The data-dependent histogram bins also need to be created in a differentially private manner. Let $\varepsilon_{\texttt{commute}}$ be the privacy budget for the commute distribution. We allocate half of this to determine the histogram bin ranges

Algorithm 3 Algorithm to compute $\varepsilon_{\texttt{commute}}$-differentially private CDFs of the Commute distributions.

commuteCDFs$(D_i, i, \varepsilon_{\texttt{commute}})$

CREATE BINS:
dpmedian $\leftarrow$ ExpoMedian$(D_i, \frac{\varepsilon_{\texttt{commute}}}{2})$
synthdata $\leftarrow$ GenExpoSynthData(dpmedian)
bins $\leftarrow$ FindPercentiles(synthdata)

CREATE NOISY HISTOGRAM:
for $j \leftarrow 1$ to numbins **do**
  $\text{CDF}[i,j] \leftarrow \text{CountCommute}(\text{bins}_j, i) + \text{Lap}\left(0, \frac{2 \cdot 2}{\varepsilon_{\texttt{commute}}}\right)$
end for

$\text{CDF}[i] \leftarrow \text{PostProc}(\text{CDF}[i])$
return $\text{CDF}[i]$

(because they are data dependent) and the other half to compute the counts themselves. To determine the bins, we assume that the commute distances in each grid cell are modeled by an exponential distribution—a popular model for positively skewed distributions such as commute distances—e.g., see [5]. Let $\eta(x)$ be the (normalized) frequency of the distance x in the dataset D_i. Then, if it follows an exponential distribution with rate parameter λ, we have

$$\eta(x) = \lambda e^{-\lambda x}$$

.

The rate parameter can be estimated using the median of the empirical data, by

$$\hat{\lambda} = \text{median} \,/\, \log(2).$$

The differentially private approximation to the median of the commute distances in grid cell i is called dpmedian and is computed using a computationally efficient version of the *exponential mechanism* [87], as in [22]. In Algorithm 3, ExpoMedian$(D_i, \frac{\varepsilon_{\texttt{commute}}}{2})$

implements this algorithm to compute dpmedian, an $\frac{\varepsilon_{\texttt{commute}}}{2}$-differentially private approximation of the median of the commute distances.

Next, we determine the histogram bins by creating a large synthetic set of commute distances that are sampled from an exponential distribution whose parameter is given by $\lambda = \frac{\text{dpmedian}}{\log(2)}$. In Algorithm 3, GenExpoSynthData(dpmedian) generates a set of synthetic commute distances, synthdata, from such a distribution. We determine the $10, 20, 30, \ldots, 90, 95$ percentiles of this set of distances using FindPercentiles. The distances corresponding to these percentiles form the edges of the histogram bins.

CountCommute(bins_j, i) counts the number of distances in the data structure D_i that fall in bins_j. $\langle \text{CountCommute}(\text{bins}_1, j), \ldots, \text{CountCommute}(\text{bins}_{10}, i) \rangle$ has a global sensitivity of 2. Applying the Laplace mechanism yields an $\frac{\varepsilon_{\texttt{commute}}}{2}$-differentially private computation of the approximate histogram counts. Since each user appears in only one of the $d_c \times d_c$ grid cells, by Theorems 2.2.4 and 2.2.5 and the privacy of the ExpoMedian [22]:

Lemma 4.4.2. *Using Algorithm 3 to compute* commuteCDF$(D_i, i, \varepsilon_{\texttt{commute}}), \forall i \in \{1, \ldots, d_c^2\}$ *is* $\varepsilon_{\texttt{commute}}$*-differentially private.*

Calls per day per user

To create the CDF of *CallsPerDay* in a differentially private manner, we begin, as in WHERE, by assuming that the average number of calls per day for any user is from the set $\mathbb{M} = \{\mu_{\min}, \ldots, \mu_{\max}\}$. Similarly, the standard deviation of the number of calls per day is from the set $\Sigma = \{\sigma_{\min}, \ldots, \sigma_{\max}\}$. Just as for WHERE, each μ_i and σ_i corresponding to a user i is rounded to the nearest value in the sets $\mathbb{M}$ and Σ, respectively.

Let CountAvgStd(μ, σ) be a function that counts the number of users whose calls made per day have a (rounded) mean and standard deviation of μ and σ respectively. Consider the matrix M, of size $|\mathbb{M}| \times |\Sigma|$, each element of this matrix corresponds to

Algorithm 4 Algorithm to compute an $\varepsilon_{\texttt{cpday}}$-differentially private CDF of the *CallsPerDay* distribution

CallsPerDayCDF$(D, \varepsilon_{\texttt{cpday}})$

COUNT:
for $\mu \leftarrow \mu_{\min}$ to μmax **do**
  for $\sigma \leftarrow \sigma_{\min}$ to σmax **do**
    $\widehat{\mathrm{M}}(\mu, \sigma) \leftarrow \mathrm{CountAvgStd}(\mu, \sigma)$
  end for
end for

NOISE ADDITION:
for $\mu \leftarrow \mu_{\min}$ to μmax **do**
  for $\sigma \leftarrow \sigma_{\min}$ to σmax **do**
    $\widehat{\mathrm{M}}(\mu, \sigma) \leftarrow \widehat{\mathrm{M}}(\mu, \sigma) + \mathrm{Lap}\left(0, \frac{2}{\varepsilon_{\texttt{cpday}}}\right)$
  end for
end for

CONVERT TO CDF:
CDF $\leftarrow \mathrm{PostProc}(\widehat{\mathrm{M}})$
return CDF

$\mathrm{CountAvgStd}(\mu, \sigma)$, for $\mu \in \mathbb{M}$ and $\sigma \in \Sigma$. Any addition or deletion of calls by a single user can change the mean / standard deviation pair from (μ, σ) to another pair (μ', σ'), decreasing the count for at most one element of the matrix M by at most 1 and increasing the count for another element by 1. Therefore, the global sensitivity of the vector $\langle M(\mu_{\min}, \sigma_{\min}), \ldots M(\mu_{\max}, \mu_{\min}) \rangle$ is 2.

Algorithm 4 first counts each user's (μ, σ). At the end of the **COUNT** process in Algorithm 4, element $\widehat{\mathrm{M}}(\mu, \sigma)$ contains $\mathrm{CountAvgStd}(\mu, \sigma)$, $\forall \mu \in \mathbb{M}$ $\sigma \in \Sigma$. Using Theorems 2.3.3 and 2.2.5, the computation of $\widehat{\mathrm{M}}$ after it goes through **NOISE ADDITION** is differentially private. Next, the noisy matrix $\widehat{\mathrm{M}}$ is converted to a CDF by applying post-processing techniques [53] to further reduce the noise. Figure 4.3 shows the differentially private approximation of the CDF of the *CallsPerDay* distribution for different values of $\varepsilon_{\texttt{cpday}}$.

Lemma 4.4.3. *Algorithm 4's computation of* $\widehat{\mathrm{M}}$ *and the CDF of the CallsPerDay distribution*

is ε_{cpday}-differentially private.

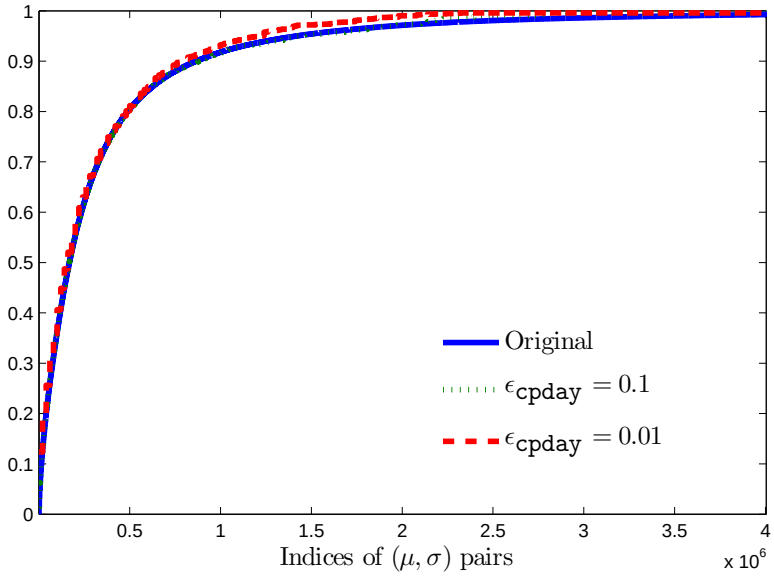


Figure 4.3: CDF of *CallsPerDay*

Call times per user class

In DP-WHERE, we cluster users into one of the two classes using differentially private k-means clustering [86] (rather than X-means as used in WHERE). From the CDR database D, just as in WHERE, we compute the number of calls each user makes during each hour of the day. From this, a 24-dimensional probability vector (one dimension for each hour) is constructed so that each element represents the probability that a user makes calls during that hour. We classify users based on this 24-dimensional probability vector. An intermediate data structure P that is input for the clustering algorithm (Algorithm 5) is the set of probability vectors p_i for all users i. Each row of P consists of the id of the user and his calling probability vector p_i. The input to Algorithm 5 consists of P, the target number k (2 in our work) of cluster centers, the privacy budget for the clustering algorithm $\varepsilon_{\texttt{bdg}}$, the amount of the privacy budget $\varepsilon_{\texttt{it}}$ that is spent for each iteration within the clustering algorithm, and the error tolerance

$\texttt{tol}$. Algorithm 5 will iterate until either the error is within the range of tolerance or the privacy budget is used up, whichever comes first.

Algorithm 5 Differentially private k-means algorithm

DP-Kmeans$(P, \varepsilon_{\texttt{bdg}}, \varepsilon_{\texttt{it}}, \texttt{tol})$

INITIALIZE:
$\text{ClustCtr}_1 \leftarrow \langle \text{Rand} \rangle^{24}$
$\text{ClustCtr}_2 \leftarrow \langle \text{Rand} \rangle^{24}$
$\varepsilon_{\texttt{calltime}} \leftarrow 0$

ITERATE:
while $\varepsilon_{\texttt{calltime}} \leq \varepsilon_{\texttt{bdg}}$ **or** $\texttt{err} < \texttt{tol}$ **do**
  $\text{OldCtr} \leftarrow \text{ClustCtr}_1$
  $\text{OldCtr} \leftarrow \text{ClustCtr}_2$
  $\text{ClustSize}_1 \leftarrow \text{ClustSize}_1 + \text{Lap}(0, \frac{1}{\varepsilon})$
  $\text{ClustSize}_2 \leftarrow \text{ClustSize}_2 + \text{Lap}(0, \frac{1}{\varepsilon})$
  $\varepsilon_{\texttt{calltime}} \leftarrow \varepsilon_{\texttt{calltime}} + \varepsilon_{\texttt{it}}$
  $\text{Sum}_1 \leftarrow \text{Sum}(\text{Cluster}_1) + \langle \text{Lap}(0, \frac{2}{\varepsilon}) \rangle^{24}$
  $\text{Sum}_2 \leftarrow \text{Sum}(\text{Cluster}_2) + \langle \text{Lap}(0, \frac{2}{\varepsilon}) \rangle^{24}$
  $\varepsilon_{\texttt{calltime}} \leftarrow \varepsilon_{\texttt{calltime}} + \varepsilon_{\texttt{it}}$
  $\text{ClustCtr}_1 \leftarrow \text{Sum}_1 \,/\, \text{ClustSize}_1$
  $\text{ClustCtr}_2 \leftarrow \text{Sum}_2 \,/\, \text{ClustSize}_2$
  $\text{ClustCtr}_1 \leftarrow \text{PostProc}(\text{ClustCtr}_1)$
  $\text{ClustCtr}_2 \leftarrow \text{PostProc}(\text{ClustCtr}_2)$
  $\texttt{err} = \text{dist}(\text{OldCtr}_1, \text{ClustCtr}_1) + \text{dist}(\text{OldCtr}_2, \text{ClustCtr}_2)$
end while

return $\text{ClustSize}_1, \text{ClustSize}_2$
return $\text{ClustCtr}_1, \text{ClustCtr}_2, \varepsilon_{\texttt{calltime}}$

As shown in Algorithm 5, we initialize the cluster centers by picking two random 24-dimensional probability vectors $\langle \text{Rand} \rangle^{24}$. Vectors in P are assigned to a cluster depending on which of the two current cluster centers they are closer to. Over each iteration, the noisy sum of the vectors in ClustCtr_i for each current cluster i is computed. The global sensitivity of the sum of vectors in $\text{ClustSum}_i = \sum_{j \in \text{Cluster}_i} p_j$ is 2, because $\forall j \in [n], \|p_j\|_1 = 1$ (since each of the vectors is a probability vector). Any change in one person's data can change the ClustSum_i to another vector $\text{ClustSum}_i + \delta$, where

$|\delta| \leq 2$. The size of a cluster has global sensitivity 1. A differentially private computation of the cluster size and the cluster sum enables a differentially private approximation of the mean vector of the cluster. Additionally, for each iteration, the computation of each ClustSize_i is $\varepsilon_{\texttt{it}}$-differentially private and the computation of Sum_i is also $\varepsilon_{\texttt{it}}$-differentially private. Using Theorem 2.2.4, this leads to a $2\varepsilon_{\texttt{it}}$-differentially private computation of ClustCtr_i. The overall privacy level over an iteration, on applying Theorem 2.2.5, is also $2\varepsilon_{\texttt{it}}$, as the clusters are non-intersecting subsets of the dataset P. A user and, consequently, his probability vector appears in exactly one cluster.

At this point, ClustCtr_i, the noisy mean of the vectors in Cluster i, will not necessarily correspond to a probability vector, as some of its elements may be negative and their sum may not add to 1. To correct for this, we apply post-processing noise correction techniques on each of these cluster centers before returning to the next iteration. After an iteration where either the privacy budget is exhausted or the error falls below the given threshold `tol`, the algorithm returns the differentially private cluster centers, the total privacy budget spent ($\varepsilon_{\texttt{calltime}}$), and a differentially private computation of the cluster sizes (the vector ClustSize). All of this incurs a privacy expenditure of $\varepsilon_{\texttt{calltime}}$.

We use the cluster centers as calling time probability distributions: each element of the cluster center vectors represents the probability that a user in that cluster makes a call during that hour. We compute one probability distribution *CallTime* for each minute of the day and for each user class by interpolating the probability distribution over all minutes between the hours (elements of the cluster centers). We use ClustSize to determine *ClassProb*, the probability of a user belonging to one of the two classes. Using Theorems 2.2.4 and 2.2.5:

Lemma 4.4.4. *Algorithm 5 gives an $\varepsilon_{\texttt{calltime}}$-differentially private clustering of the user calling probability vectors.*

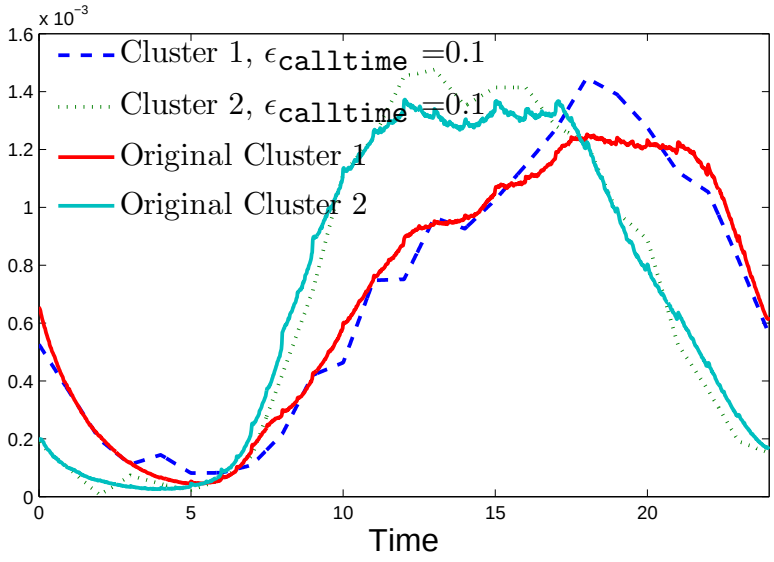


Figure 4.4: Comparison of distribution of call times for two classes of users as determined by Algorithm 5 to the non-private clustering.

Figure 4.4 shows that DP-WHERE preserves typical diurnal patterns for both classes, even for low values of $\varepsilon_{\texttt{calltime}}$.

Hourly calls per location

For every hour of the day, DP-WHERE differentially privately computes an empirical distribution of calls made over every grid cell. To do this, CountCallsNum(i, j) is defined as the function that returns the number of calls users in D make in the ith grid cell between the hour $j-1$ and j. We wish to determine a matrix H of size $d^2 \times 24$; each row of H corresponds to a grid cell $i \in [d^2]$ and each column to an hour $j \in [24]$. Element $H(i, j)$ of the matrix has value CountCallsNum(i, j). Let NumDays be the number of days that the database D corresponds to. The (column) vector corresponding to calls made over the geographical area during hour j is written as $\langle H(\ \times *, j)\rangle = \langle H(1, j) \ldots H(d^2, j)\rangle$. Since any change in exactly one user's data can cause a change of at most MaxCallsHr for every hour of each of these days, the global sensitivity of this vector is MaxCallsHr $\cdot$ NumDays.

Algorithm 6 Algorithm to compute $\varepsilon_{\texttt{hrlocs}}$-differentially private CDFs of the *HourlyLocs* distributions.

HourlyCDFs$(D, \varepsilon_{\texttt{hrlocs}})$
gnums $\leftarrow \lfloor \frac{d^2}{\text{gsize}} \rfloor$
for $j \leftarrow 1$ to 24 **do**
  for $\ell \leftarrow 1$ to gnums **do**

    GROUP:
    $g_\ell \leftarrow 0$
    for $i \leftarrow 1$ to gsize **do**
      $g_\ell \leftarrow g_\ell + \text{CountCallsNum}(i, j)$
    end for
  end for

  NOISE ADDITION:
  $\langle g \rangle \leftarrow \langle g \rangle + \left\langle \text{Lap}(0, \frac{\text{MaxCallsHr} \cdot \text{NumDays}}{\varepsilon_{\texttt{hrlocs}}/24}) \right\rangle^{\text{gnums}}$

  RECONSTRUCT:
  $\langle H(*, j) \rangle \leftarrow \text{Reconstruct}(\langle g \rangle)$ CDF$[j] \leftarrow \text{PostProc}(\langle H(*, j) \rangle)$
end for

return CDF

Direct use of the Laplace mechanism with this level of global sensitivity would add a lot of noise relative to the individual counts. To reduce the overall magnitude of noise added, we make use of *grouping* [66], which groups similar counts together and allows the magnitude of the noise added to each group count to be lower as compared to the total group count. Specifically, we set the group size gsize to be equal to $24 \cdot \text{NumDays}$, comparable to the magnitude of noise we will add to the resulting grouped-counts vector. Grouping gsize contiguous elements together yields a vector $\langle g \rangle$ of size gnums $= \lfloor \frac{d^2}{\text{gsize}} \rfloor$. Each element g_ℓ of $\langle g \rangle$ counts the total number of calls made in locations that appear in group ℓ. Note that the global sensitivity of $\langle g \rangle$ is still $\text{MaxCallsHr} \cdot \text{NumDays}$ because any one user can make upto a maximum of MaxCallsHr calls during a particular hour of each of these days. We then apply the

Laplace mechanism to add noise to each group count. Finally, we replace every individual count $H(i,j)$ by the average of the noisy group count it belongs to (as denoted by Reconstruct in Algorithm 6).

Algorithm 6 applies a similar grouping scheme for each hour $(1,...,24)$. By Theorem 2.3.3, each of these computations is $\frac{\epsilon_{\texttt{hrlocs}}}{24}$-differentially private. Thus by Theorem 2.2.4:

Lemma 4.4.5. *Algorithm 6 is $\epsilon_{\texttt{hrlocs}}$-differentially private.*

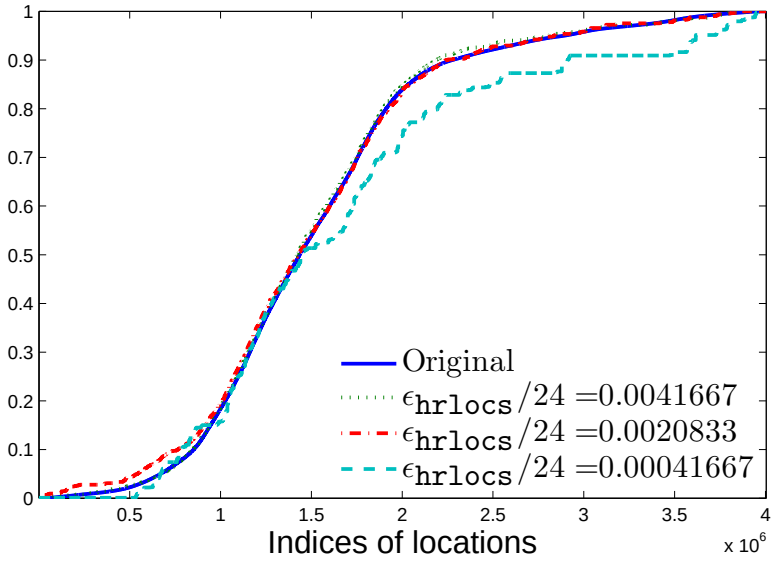


Figure 4.5: *HourlyLocs* Distribution for 5:00pm to 6:00pm

While this kind of grouping may not always yield highly accurate results, in our case each of the hourly distributions is defined on a geographical area, so we can expect call counts within a group (corresponding to call counts in contiguous geographical areas) to be similar to each other for many groups. As demonstrated by Figure 4.5, showing *HourlyLocs* for different values of $\epsilon_{\texttt{hrlocs}}/24$, corresponding to an overall $\epsilon_{\texttt{hrlocs}}$ (over all the *HourlyLocs* distributions) of 0.1, 0.05, and 0.01, respectively, this method works well in our experiments.

4.4.3 DP-WHERE: putting it all together

All of the approximations to the empirical distributions computed above are ε_i-differentially private for different values of ε_i. DP-WHERE composes these individual differentially private mechanisms to yield the overall algorithm. To generate synthetic CDRs from these distributions, DP-WHERE performs the same steps as WHERE to sample from each of these private distributions to generate synthetic CDRs without "dipping" back into the original data. Applying Theorem 2.2.4 to Lemmas 4.4.1–4.4.5 yields:

Theorem 4.4.6. *DP-WHERE is ε-differentially private, where*

$$\varepsilon = \varepsilon_{home} + \varepsilon_{work} + \varepsilon_{commute} + \varepsilon_{cpday} + \varepsilon_{calltimes} + \varepsilon_{hrlocs}.$$

It is important to note that, because none of the sampling steps in DP-WHERE require further access to the original data, it is possible for the data holder to release the noisy distributions while retaining differential privacy. This would allow others to produce their own synthetic CDR traces for any desired number of users, time duration, or other parameters.

4.5 Experimental evaluation

We have shown that DP-WHERE achieves differential privacy. Because it achieves this by injecting noise, we must also assess the impact on utility. In this section, we explore this impact by comparing the utility of the models produced by DP-WHERE and by WHERE, both WHERE using real CDRs as input and WHERE using public data (for example, the US Census) as input. To evaluate the utility of our models, we are interested in how closely our synthetic users mimic the behavior of real cellular network subscribers. Specifically, for multiple kinds of uses, we demonstrate that DP-WHERE achieves similar accuracy to WHERE using CDRs, and far better accuracy

than WHERE using only public data.

4.5.1 Datasets and methodology

The input data for our DP-WHERE and WHERE experiments come from a large set of CDRs generated by actual cellphone use over 91 consecutive days from April 1 to June 30, 2011. This dataset contains over 1 billion records for both voice calls and text messages involving over 250,000 unique phones chosen at random from phones billed to ZIP codes within 50 miles of the center of New York City.

In addition to the differential privacy provided by DP-WHERE, we took several steps to preserve the privacy of individuals represented in our input datasets throughout our handling of those datasets. First, we used only anonymized CDRs containing no Personally Identifying Information (PII). Second, we did not focus our analysis on any individual phone. Third, we present only aggregate results.

In each of our DP-WHERE and WHERE experiments, we generate 10,000 synthetic users that travel for 30 consecutive days in an area of more than 14,000 mi^2 around New York City, more specifically bounded by latitudes 40°N & 42°N and longitudes 73°W & 75°W. This area is further broken down into squares 0.001° on a side to construct the $d \times d$ grid discussed in Section 4.3.1, with d = 2,000.

4.5.2 Earth Mover's Distance

An important goal of our modeling approach is that a synthetic CDR trace should produce population density distributions that closely match those produced by a real CDR trace at every time of day. We therefore need a quantitative measure for comparing two spatial probability distributions at a given time. Our chosen metric is Earth Mover's Distance (EMD) [108], which we compute efficiently using the Fast EMD code from [97].

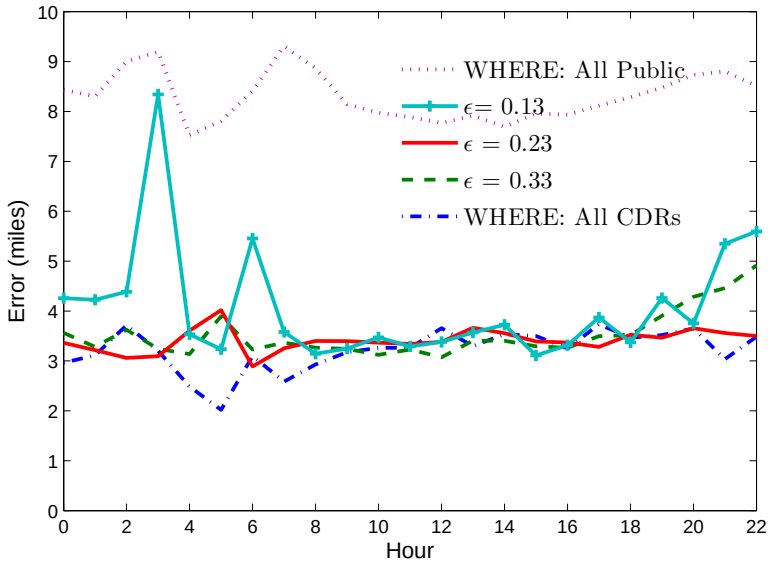


Figure 4.6: EMD error for DP-WHERE using different values of ε and a fixed commute-grid cell size of $0.01^\circ \times 0.01^\circ$, as compared to WHERE using CDRs and WHERE using public data.

EMD finds the minimum amount of energy required to transform one probability distribution into another. If one visualizes the problem as reshaping one mound of earth to match another, this energy is given by the "amount" of probability to be moved and the "distance" to move it. Thus, a lower EMD value indicates a stronger similarity between two distributions. Since different distance weightings lead to different EMD values, we follow the method in [62] and convert a raw EMD value to miles of error by using a normalizing factor. We obtain this factor by calculating the EMD between two spatial probability distributions with their entire populations concentrated in one of two places one mile apart.

The differential privacy parameter ε gives us a "knob" by which to trade privacy for accuracy. Figure 4.6 compares DP-WHERE using different values of ε to WHERE using CDRs and WHERE using public data. The size of the commute-grid cells is held constant at $0.01^\circ \times 0.01^\circ$. As shown, WHERE using CDRs has the lowest overall EMD,

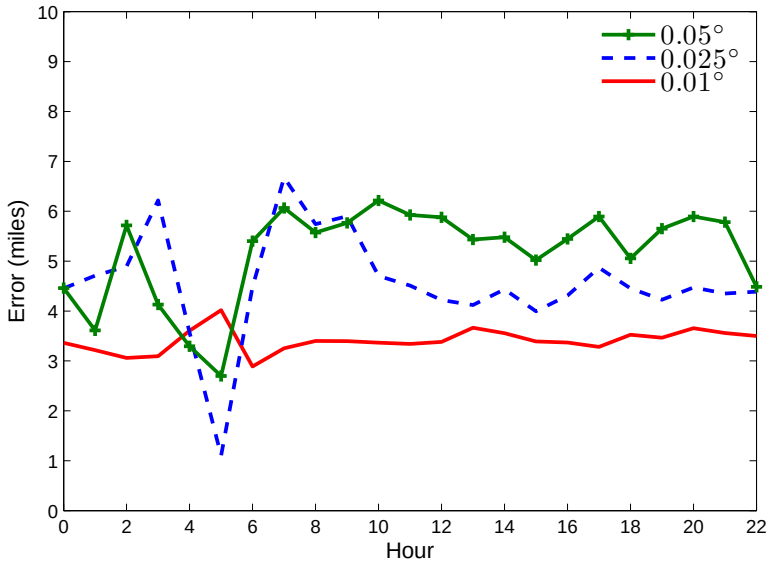


Figure 4.7: EMD error for DP-WHERE using different sizes of commute-grid cell side and a fixed ε of 0.23.

but DP-WHERE performs favorably across a range of ε values, always performing better than WHERE using public data. Just as in [62], we compare our results to WHERE using public data; the dotted curve at the top in Figure 4.6 represents the EMD error incurred by WHERE using publicly available data (such as the US census data) and models of some of the probability distributions. This comparison is especially relevant in our case, because if the EMD error incurred by DP-WHERE (for a reasonable value of ε) were to exceed the "all-public" WHERE error, one could arguably make a case for using public data sources instead. This, however, is not the case as the figure shows; even for an overall ε of 0.23, the average EMD error for DP-WHERE is well below the "all-public" error. As ε is made smaller to achieve better privacy, more noise is added and the EMD creeps upward. Accuracy is better in some times of day than in others. In particular, hours before 8 or after 22 suffer from a smaller sample of locations in the input CDRs because fewer people make calls then, so adding noise has more of an impact during those hours. We note that the exact choice of an appropriate ε is a (largely open) policy question.

The accuracy of DP-WHERE also depends on the granularities of the grids used to divide the geographic region of interest. Figure 4.7 compares DP-WHERE using different commute-grid sizes for the same ε of 0.23. At this ε value, coarser commute grids provide less accurate EMD results.

	commute-grid cell size		
	0.01°	**0.025°**	**0.05°**
WHERE	3.2150	3.3396	3.0871
ε **= 0.33**	3.5316	3.1655	4.5687
ε **= 0.23**	3.4066	4.5577	5.1691
ε **= 0.13**	5.3391	5.3194	5.2754

Table 4.1: Average EMD error for WHERE using CDRs and DP-WHERE using various ε, as the commute-grid cell size changes.

We ran a wide range of experiments to explore the effects of ε and commute-grid cell size. Table 4.1 summarizes the EMD error averaged over the hours of the day for each choice of ε and cell size. Across all our experiments, the EMD error for DP-WHERE differed by only 0.17–2.2 miles from those of WHERE using CDRs. Our results confirm that differential privacy can be achieved for a modest reduction in accuracy.

4.5.3 Daily range

Daily range, or the maximum distance between any two points a person visits in a day, has proven useful for characterizing human mobility patterns [61, 70]. We can therefore demonstrate the value of our modeling techniques by showing that the daily range computed from synthetic CDRs closely match those computed from real CDRs.

Figure 4.8 demonstrates the utility of DP-WHERE for daily range experiments. We compare daily ranges produced by DP-WHERE, WHERE from CDRs, and the original CDRs. We use boxplots to summarize the resulting empirical distributions, where the "box" represents the 25^{th}, 50^{th}, and 75^{th} percentiles, while the "whiskers" indicate the

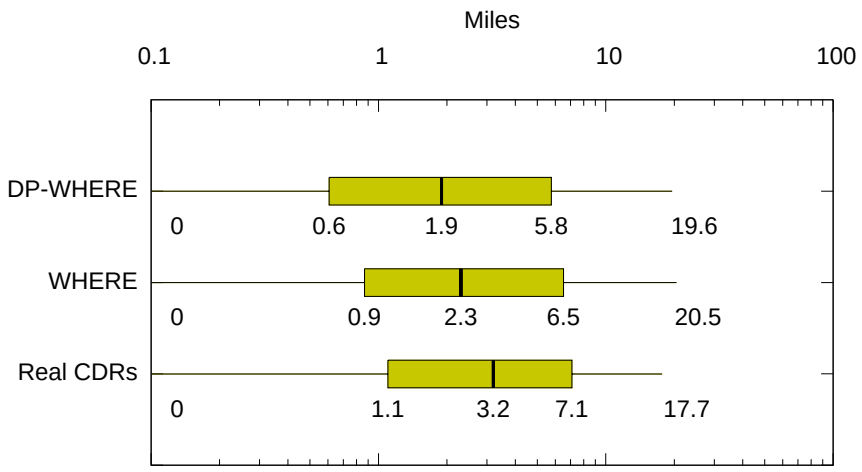


Figure 4.8: Daily range for DP-WHERE ($\varepsilon = 0.23$, commute-grid size $= 0.01^\circ \times 0.01^\circ$), WHERE from CDRs, and the real CDR dataset.

2^{nd} and 98^{th} percentiles. The horizontal axis shows miles on a logarithmic scale. Like WHERE, DP-WHERE exhibits daily ranges that are qualitatively similar to those from real CDRs, with differences of 0.5–1.3 miles across the middle two quartiles.

EMD and daily range serve as important and complementary metrics for validating our synthetic models. EMD measures the aggregate behavior of synthetic users, while daily range yields results at a per-user granularity. In summary, our EMD and daily range results confirm that DP-WHERE produces synthetic CDRs that closely mimic the behavior of large populations of real cellphone users.

4.6 Conclusions and future work

We have introduced DP-WHERE, which provides differential privacy while maintaining the utility of WHERE for modeling human mobility from real-world cellular network data. Our work demonstrates that it is possible to balance privacy and utility concerns in practical Big Data applications.

We note that the input data set for this study comes from a much larger dataset of CDRs of millions of users per metropolitan area. The CDRs used in this study have

been randomly sampled from this larger database. If we undertake the sampling in a systematic way, we can improve overall the privacy guarantees of our algorithm, as noted by Cormode et. al [21]. In our case, if we start with a larger database D of CDRs and include each user and his associated calls with a probability of p in the final dataset S, then our privacy parameters for DP-WHERE show a further considerable improvement. For example, sampling 5% of users from a database of millions of users and running our DP-WHERE algorithm over the sampled database would yield an order of magnitude improvement in the privacy parameter ε. Conversely, this could be used to achieve a given ε with less noise addition. However, to explore how sampling can help us reduce the magnitude of noise that needs to be added to each distribution, one would need to formally quantify the relative error of each of the private CDFs.

In our experimental evaluations, described in Section 4.5, we only report the overall privacy value ε for each instance of the DP-WHERE modeling process. The choice of individual privacy budgets (such as $\varepsilon_{\texttt{home}}$) spent on each distribution (or family of distributions), obviously, affects the overall ε. However, all distributions do not exhibit a similar utility-privacy tradeoff. For example, as we can see from Figures 4.2 and 4.5, values of $\varepsilon_{\texttt{home}}$ and $\varepsilon_{\texttt{hrlocs}}$ that are of the same order (0.0001 and 0.0004 respectively) lead to varying utility (informally, seen as the extent to which private approximations are "close" to the original distributions). It would be interesting to conduct a systematic study of how to divide the overall budget among the constituent distributions. Here, again a formal quantification of the approximation error of each of the private CDFs may be helpful.

In DP-WHERE none of these sampling steps require further access to the original CDRs, making it possible for the data holder to release the noisy distributions while retaining differential privacy. Among possible uses, these distributions would allow others to produce their own synthetic CDR traces for any desired population size,

time duration, or other parameters. This, among other things, would entail a study—experimental or theoretical—of the approximation error of each of the private CDFs, perhaps, measured by the relative error in computing the CDFs.

It is our hope that our work on DP-WHERE constitutes a significant step towards enabling cellular telephony providers to unlock the value of their data for applications with broad societal benefits, such as urban planning and epidemiology, without compromising privacy.

5

Differentially Private Gaussian Regression

Several machine learning tasks use data that is increasingly coming from individuals. This leads to concerns about preserving the privacy of participating individuals while enabling as little loss in utility of the machine learning algorithm as possible. In this chapter, we consider the problem of privacy-preserving *linear regression*. Regression is a staple methodology in machine learning and has been considered before in various privacy preserving paradigms (see [118] for example). We will adhere to the notion of differential privacy and consider the problem of regression in this model. In the context of machine learning, differential privacy informally requires that if, using the training data, we learn a probability distribution on the predictor space, then this distribution should remain (almost) unchanged whether or not an individual is included in the training data.

We examine the problem of differentially private regression, a supervised learning task concerned with the prediction of continuous quantities rather than discrete labels as in classification. The training set for this task consists of individuals' sensitive data. For example, consider a database that consists of individuals' smoking frequencies and their associated risk of lung cancer. Using this data— the *training data*— we would like to determine a function that predicts an individual's risk of lung cancer given their smoking frequency. This function is learned by using individuals' private data, but clearly has great social benefit.

Our work makes a connection between Gaussian regression (see [102] for example)

and differentially private regression. We establish this via the so-called *exponential mechanism* [87]. This connection to Gaussian regression, in particular, and Bayesian regression in general (using an appropriate *conjugate prior*) helps us propose a general computationally efficient exponential mechanism for regression problems. This novel exponential mechanism is based on the concept of *probabilistic sensitivity* that may be of independent interest. Gaussian regression also has well known connections to ridge-regression; we show how the regularization/penalization constant in the case of ridge regression is related to the privacy level of the solution. This connection also helps us in high-dimensional regression, if the L_2-norm of the true predictor is low. We show that the regularization constant, in the case of ridge regression, (or equivalently the Gaussian prior in the case of Gaussian regression) is related to the privacy parameter ε. Additionally, we also show how this enables us to provide a "dimension-free" bound for a differentially private predictor, using the concept of *effective dimension* [125] or *effective degrees of freedom* [50]. This, to the best of our knowledge is the first such bound for differentially private supervised learning. We also experimentally validate our results by applying our method to the Boston housing data set from the UCI repository [39].

5.1 Related Work

Chaudhuri et al. [18] propose differentially private prediction by adding appropriate noise into the objective function instead of the predictor, a technique called *objective perturbation*. Though they propose a method for classification, their technique is applicable to regression problems in which the objective function is strongly convex. This is the case for L_2-regularized least squares regression which we consider here. As we show in Section 5.5.1 our results, unlike theirs, are not always dependent on the underlying dimension of the predictor space.

Dwork and Lei [32] also examine private regression adding noise proportional to the dimension of the input space. Rubinstein et al. [107] propose privacy-preserving classification methods using support vector kernels. Chaudhuri and Hsu [17] propose sample-complexity bounds for differentially private learning. Our generalization bounds are in terms of the expected risk of the private predictor, where the expectation is over the private distribution, rather than in terms of the risk of a single predictor. Our experimental results show that the risk of a (private) predictor sampled from such a distribution is fairly concentrated around the expected risk. In the next section we introduce the framework of differential privacy and in Section 5.3 we introduce the problem of private regression.

5.1.1 Differential Privacy

We introduce a novel exponential mechanism in this section via the notion of *probabilistic sensitivity* which may be of independent interest.

5.2 A relaxed exponential mechanism

First, we specify the input and output of the scoring function q of the exponential mechanism. Recall that the exponential mechanism maps a pair of an input data set $\mathbf{X}$ (a vector or matrix over some arbitrary real-valued domain) and candidate output $\boldsymbol{\omega}$ (again over an arbitrary range Ω) to a real-valued "quality score." Higher valued scores imply good input-output correspondences. It assumes a base measure π on the range Ω. The global sensitivity of the scoring function q (of the exponential mechanism) as outlined in Chapter 2 is given by

$$\mathrm{GS}_q = \max_{\mathbf{X}\,\mathbf{X}',\boldsymbol{\omega}} \left| q(\mathbf{X}, \boldsymbol{\omega}) - q(\mathbf{X}', vw) \right|$$

where $\mathbf{X}$ and $\mathbf{X}'$ are any two neighboring inputs datasets. Global sensitivity is a worst case notion that takes into account the "worst" neighboring pair of datasets $\mathbf{X}$ and $\mathbf{X}'$ over all outputs $\boldsymbol{\omega}$. Sometimes, it is hard to bound the effect of worst-case neighbors over every $\boldsymbol{\omega}$ in Ω, the output space. However, if we show that this effect is bounded with a high probability, over the choice of $\boldsymbol{\omega}$ sampled from the exponential mechanism, then we can prove the existence of a (ε, δ)-differentially private "relaxed" exponential mechanism.

We introduce the notion of a probabilistic upper bound on the global sensitivity of the scoring function. Recall, that the exponential mechanism is given by the following family of distributions:

$$\pi_{\mathbf{X},\varepsilon}(\boldsymbol{\omega}) \propto \exp(\varepsilon q(\mathbf{X}, \boldsymbol{\omega})) \cdot \pi(\boldsymbol{\omega}) \tag{5.1}$$

Definition 5.2.1 (Probabilistic global sensitivity)**.** *Let $\pi_{\boldsymbol{X},\varepsilon}$ denote the exponential mechanism as in Equation 5.1. $(\mathcal{U}, \delta)$ is said be a probabilistic upper bound on the global sensitivity of a scoring function $q(\boldsymbol{X}, \boldsymbol{\omega})$, with respect to π_ε, if with probability at least $1 - \delta$ over the choice of $\boldsymbol{\omega}$ (sampled from π_ε), for any neighboring $\boldsymbol{X}$ and $\boldsymbol{X}'$,*

$$\max_{\boldsymbol{X} \sim \boldsymbol{X}'} \left| q(\boldsymbol{X}, \boldsymbol{\omega}) - q(\boldsymbol{X}', \boldsymbol{\omega}) \right| \leq \mathcal{U}.$$

For convenience we refer to this quantity as probabilistic global sensitivity, but in fact, it is only a probabilistic upper bound. We also prove that sampling from an exponential mechanism whose score function has a probabilistic global sensitivity of $(\mathcal{U}, \delta)$ is $(2\varepsilon\mathcal{U}, \delta)$-differentially private.

Theorem 5.2.2 (Relaxed exponential mechanism)**.** *If, with probability $1 - \delta$, over the choice of $\boldsymbol{\omega}$ with respect to the measure π_ε*

$$\max_{X_1 \sim X_2} |q(\boldsymbol{X}_1, \boldsymbol{\omega}) - q(\boldsymbol{X}_2, \boldsymbol{\omega})| \leq \mathcal{U},$$

then sampling from the exponential distribution is $(2\varepsilon\mathcal{U}, \delta)$-differentially private

Proof. Let **GOOD** be the set of ω's such that for any $\mathbf{x}$,

$$\left|q(\mathbf{X},\omega) - q(\mathbf{X}',\omega)\right| \leq \mathcal{U}.$$

If the probabilistic sensitivity of the q with respect to the mechanism π_ε is given by $(\mathcal{U},\delta)$, then we have $\pi_\varepsilon(\mathbf{GOOD}) \geq 1-\delta$. Let ω^{private} be the result of sampling π_ε once.

Now we have

$$\text{pdf}(\omega^{\text{private}} = \omega|\mathbf{X}_1) =$$

$$\pi_\varepsilon(\mathbf{GOOD}).\,\text{pdf}(\omega^{\text{private}} = \omega|\omega \in \mathbf{GOOD}, \mathbf{X}_1)$$

$$+\pi_\varepsilon(\overline{\mathbf{GOOD}}) \cdot \text{pdf}(\omega^{\text{private}} = \omega|\omega \in \overline{\mathbf{GOOD}}, \mathbf{X}_1)$$

$$\leq \pi_\varepsilon(\mathbf{GOOD}) \cdot \text{pdf}(\omega^{\text{private}} = \omega|\omega \in \mathbf{GOOD}, \mathbf{X}_1) + \delta$$

$$\leq \pi_\varepsilon(\mathbf{GOOD}) \cdot e^{2\varepsilon\mathcal{U}}\,\text{pdf}(\omega^{\text{private}} = \omega|\omega \in \mathbf{GOOD}, \mathbf{X}_2) + \delta$$

$$\leq e^{2\varepsilon\mathcal{U}}\,\text{pdf}(\omega^{\text{private}} = \omega|\mathbf{X}_2) + \delta$$

□

The idea of probabilistic global sensitivity, to the best of our knowledge has not been used before. It would be interesting to compare this notion to that of *smooth sensitivity*. Another probabilistic notion of sensitivity is used by Dwork and Lei in the *Propose-Test-Release* mechanism [32], where they propose a (asymptotic) bound on the local sensitivity of a function, and subsequently test (in differentially private manner) whether the local sensitivity is sufficient for the current problem instance. The proposed bound fails with a small probability δ. Subsequently a noisy statistic (with the added noise being proportional to this bound on the local sensitivity) is released.

We will make use of Theorem 5.2.2 to propose a computationally efficient mechanism for differentially private regression. In the next section we introduce the general framework of linear and Gaussian regression.

5.3 Linear Regression

First, we introduce the general framework of statistical prediction, in which there is an input space $\mathcal{X}$, an output space $\mathcal{Y}$, and a space of predictors $\mathcal{F}$. For any $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, and any predictor $f \in \mathcal{F}$, a loss quantified by a loss function

$$\ell_f(\mathbf{x}, y) = \ell_f(\mathbf{z})$$

is incurred, where $\mathbf{z} = (\mathbf{x}, y), \in \mathcal{Z} = (\mathcal{X} \times \mathcal{Y})$. We assume a fixed but unknown, probability measure $\mathbb{P}$ on $\mathcal{Z}$.

The *true risk* of a predictor f is the expected loss of the predictor given by:

$$R(f) = \mathbb{E}_{\mathbf{z}} \ell_f(\mathbf{z}).$$

Given a set of n random independent samples $\hat{\mathbf{Z}} = \{(\mathbf{x}_i, y_i)_{i=1}^n\} \in \mathcal{Z}^n$, each drawn from $\mathbb{P}$, and a predictor f, the *empirical risk* of f on $\hat{\mathbf{Z}}$, is given by:

$$\hat{R}_{\hat{\mathbf{Z}}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell_f(\mathbf{x}_i, y_i)$$

Empirical Risk Minimization (see [7] for example) seeks to formalize a relationship between the true risk and the empirical risk of the best (in the empirical sense) predictor in the form of generalization bounds. Specifically, one is interested in how much difference there will be between the empirical risk and the true risk of a predictor (that belongs to a class $\mathcal{F}$) given n random, independent samples. This relationship is mainly studied using probabilistic bounds.

Another quantity that interests us is the *estimation error*: that is, how different is the true risk of a predictor that minimizes empirical risk compared to the true risk of the "best" predictor in the predictor space. This also takes the form of *sample complexity bounds* where one can ask how large does the sample size have for this error to be within a given $\mathcal{E}$.

5.3.1 Gaussian Regression

In the case of parametric linear regression (see [102] for example), the input space, $\mathcal{X} \subset \mathbb{R}^d$ and the output space, $\mathcal{Y} \subset \mathbb{R}$.

Let Ω be a (often bounded convex) subset of $\mathbb{R}^d$. Then, each function $f \in \mathcal{F}$ is parametrized in the following manner:

$$f_{\omega}(\mathbf{x}) = \mathbf{x}^T \cdot \boldsymbol{\omega}$$

for any $\mathbf{x} \in \mathcal{X}$.

Let $\mathbf{X}$ be a matrix, each column of which corresponds to an input $\mathbf{x}_i$ in the sample set $\hat{\boldsymbol{Z}}$, and let $\mathbf{y}$ be a vector, each element of which is a y_i from $\hat{\boldsymbol{Z}}$.

Next, we review the Bayesian linear model with Gaussian noise. Let $\mathbf{x}$ be an input vector, f_{ω^*} be the "true" function, that is,

$$f_{\omega^*}(\mathbf{x}) = (\mathbf{x})^T \cdot \boldsymbol{\omega}^*.$$

Each observed value y_i differs from the function value $f_{\omega^*}(\mathbf{x}_i)$ by a additive noise, which is assumed to be generated from a Gaussian distribution, $\mathcal{N}(0, \sigma_n^2)$ with mean 0 and variance σ_n^2. We have

$$y = f_{\omega^*}(\mathbf{x}) + \eta, \;\; \eta \sim \mathcal{N}(0, \sigma_n^2).$$

This noise assumption implies that the likelihood of the observations given the model also follows a normal distribution, that is:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\omega}) &= \frac{1}{(\sqrt{2\pi}\sigma_n)^n} \exp\left(\frac{-1}{2{\sigma_n}^2} \left|\left|\mathbf{X}^T\boldsymbol{\omega} - \mathbf{y}\right|\right|^2\right) \\ &= \mathcal{N}(\mathbf{X}^T\boldsymbol{\omega}, \sigma_n^2 I). \end{aligned} \tag{5.2}$$

In Bayesian settings, a *prior* on the parameter space $\boldsymbol{\omega}$ is specified, which captures the "beliefs" about the parameters before looking at the data. The belief is then updated to the so-called *posterior* probability, which signifies the fact that the probability

distribution has been updated after observing the data. Like Rasmussen et al. [102], we assume a Gaussian prior with zero mean and covariance matrix Σ_p on the weights:

$$\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \Sigma_p).$$

Using Bayes' rule, the posterior is given by:

$$p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \boldsymbol{\omega})p(\boldsymbol{\omega})}{p(\mathbf{y}|\mathbf{X})}$$

After alegbraic manipulation it can be simplified to

$$p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\bar{\boldsymbol{\omega}} = \frac{1}{\sigma_n^2} A^{-1}\mathbf{X}\mathbf{y}, A^{-1})$$

where

$$A = \sigma_n^{-2}\mathbf{X}\mathbf{X}^T + \Sigma_p^{-1}.$$

Since the posterior is also a normal distribution, its mean is the same as its mode, also called the *maximum a posteriori (MAP)* estimate of $\boldsymbol{\omega}$.

In Bayesian prediction given a test input $\mathbf{x}_{\text{test}}$, the predictive distribution is given by

$$\begin{aligned} p(f_{\text{test}}|\mathbf{x}_{\text{test}}, \mathbf{X}, \mathbf{y}) &= \int p(f_{\text{test}}|\mathbf{x}_{\text{test}}, \boldsymbol{\omega})p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{y})d\boldsymbol{\omega}. \\ &= \mathcal{N}(\frac{1}{\sigma_n^2}\mathbf{x}_{\text{test}}A^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}_{\text{test}}A^{-1}\mathbf{x}_{\text{test}}) \\ &= \mathcal{N}(\mathbf{x}_{\text{test}}^T \cdot \overline{\boldsymbol{\omega}}, \mathbf{x}_{\text{test}}A^{-1}\mathbf{x}_{\text{test}}) \end{aligned} \tag{5.3}$$

Given a $\mathbf{x}_{test}$, the mean (or mode) , $\overline{\boldsymbol{\omega}}$, of this normal distribution is used to make the prediction, that is:

$$f_{\overline{\boldsymbol{\omega}}}(\mathbf{x}_{\text{test}}) = \mathbf{x}_{\text{test}}^T \cdot \overline{\boldsymbol{\omega}}.$$

In the next section, we adapt this mechanism to propose a differentially private prediction function. We do this by establishing a connection to the exponential mechanism and proving a probabilistic upper bound on the global sensitivity of its score function.

5.4 Differentially Private Gaussian Regression

A prior $p(\boldsymbol{\omega})$ over the predictor space Ω captures the beliefs we have without having seen the training data $\hat{\mathbf{Z}}$. Consider the score function of the exponential mechanism to be the log-likelihood of the data, given the model. That is, let

$$q((\mathbf{X},\mathbf{y}),\boldsymbol{\omega}) = \log\left(\frac{1}{(2\pi\sigma_n^2)^{n/2}}\exp\left(\frac{-1}{2{\sigma_n}^2}\left|\left|\mathbf{X}^T\boldsymbol{\omega}-\mathbf{y}\right|\right|^2\right)\right). \tag{5.4}$$

The exponential mechanism corresponds to the following density

$$\pi_{\mathbf{X},\alpha}(\boldsymbol{\omega}|\mathbf{X},\mathbf{y}) = \frac{1}{\beta(\mathbf{X},\mathbf{y})}\exp\left(\alpha q\big((\mathbf{X},\mathbf{y}),\boldsymbol{\omega}\big)\right).p(\boldsymbol{\omega}) \tag{5.5}$$

where α is a privacy parameter and $\beta(\mathbf{X},\mathbf{y})$ is a normalizing factor. This is similar to the analysis above yielding

$$\pi_{\mathbf{X},\alpha}(\boldsymbol{\omega}|\mathbf{X},\mathbf{y}) = \mathcal{N}(\frac{\alpha}{\sigma_n^2}A^{-1}\mathbf{X}\mathbf{y},\Sigma_\alpha^{-1})$$
$$\boldsymbol{\omega}_\alpha = \frac{\alpha}{\sigma_n^2}A^{-1}\mathbf{X}\mathbf{y},\ \Sigma_\alpha = (\frac{\alpha}{\sigma_n^2})^{-1}\mathbf{X}\mathbf{X}^T+\Sigma_p^{-1}. \tag{5.6}$$

In the case of Bayesian regression, as we saw above, the prediction is made by using the mean of the normal distribution $\pi_{\mathbf{X},\alpha}$. But, we will instead sample $\boldsymbol{\omega}^{\text{private}}$ from $\pi_{\mathbf{X},\alpha}$ and prove the privacy properties of such a sampling.

To prove the privacy properties of sampling from $\pi_{\mathbf{X},\alpha}$ we prove a probabilistic upper bound on the global sensitivity of the score function as given in Equation 5.4.

Before that, we briefly remark on the well-known connection between Gaussian and ridge regression, as we use some of the results from ridge regression in our work.

Connections to ridge regression

There is well-known equivalence between ridge regression and Gaussian regression with a Gaussian prior [50]. Ridge regression shrinks the regression coefficients $\boldsymbol{\omega}$ by

imposing a penalty on their size. It minimizes the sum of least squares error between the prediction and the actual y-values, while penalizing large regression coefficients in the following manner.

$$\boldsymbol{\omega}^{\text{ridge}} = \text{argmin}_{\boldsymbol{\omega}} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\omega})^2 + \lambda \, ||\boldsymbol{\omega}||_2^2 \right\}. \tag{5.7}$$

Here, $\lambda \geq 0$ is the regularization or penalty constant that determines the shrinkage of $\boldsymbol{\omega}$. The larger λ, the greater the shrinkage, leading the coefficients to be shrunk towards zero. The solution to this is given by

$$\boldsymbol{\omega}^{\text{ridge}} = [\mathbf{X}^T\mathbf{X} + \lambda I]^{-1}\mathbf{X}^T\mathbf{y}, \tag{5.8}$$

where I is a $d \times d$-identity matrix.

Consider a Gaussian noise model as in Equation 5.3.1 and assume a Gaussian prior $\Sigma_p = \tau^2 I$ on the parameter space Ω. Then, assuming that, τ^2 and σ_n^2 are known, the negative log-posterior of $\boldsymbol{\omega}$, is given by the expression in the RHS of Equation 5.7 with

$$\lambda = \sigma_n^2/(\alpha\tau^2).$$

$\boldsymbol{\omega}^{\text{ridge}}$ is the same as $\boldsymbol{\omega}_\alpha$ in equations 5.6 and we will use the two interchangeably, as convenient.

Ridge regression is used in situations to control the norm of the predictor. The larger the value of λ, the smaller the L_2-norm of the predictor will be. Gaussian regression has an equivalent way of achieving this by imposing a prior on the parameter space given by $\Sigma_p = \tau^2$, where τ is small. We see that the privacy parameter is inversely related to the regularization parameter, if τ and σ_n are considered to be constants.

The predicted values will then be given by

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\omega}^{\text{ridge}} = \mathbf{X}[\mathbf{X}^T\mathbf{X} + \lambda I]^{-1}\mathbf{X}^T\mathbf{y}.$$

5.4.1 A probabilistic upper bound on the global sensitivity of the score function in Gaussian regression

Theorem 5.4.1. *A probabilistic upper bound on the global sensitivity of the scoring funtion associated with π_α is $(\frac{S}{2\sigma_n^2}, \delta)$, where:*

$$\begin{aligned} S \leq & \left(\frac{M^4 n^2 \, {\mathrm{Y}_{\max}}^2}{\lambda^2} + \frac{2M^2 \, \mathrm{Y}_{\max} \sqrt{2\log(1/\delta)}}{\lambda^{\frac{3}{2}}} + \frac{2M^2 \log(1/\delta)}{\lambda} \right. \\ & \left. + \frac{2M^2 \, {\mathrm{Y}_{\max}}^2 \, n}{\lambda} + \frac{2M \, \mathrm{Y}_{\max} \sqrt{2\log(1/\delta)}}{\sqrt{\lambda}} + {\mathrm{Y}_{\max}}^2 \right) \end{aligned}$$

Before we can prove this theorem, we will need to prove a lemma that bounds the norm of $\boldsymbol{\omega}^{\text{ridge}}$. We also state two lemmas here (one on Gaussian concentration and the other on linearity of Gaussian distributions) that we will use in our proof below.

Lemma 5.4.2. *[122][Concentration of a Gaussian distribution] Let $X \sim \mathcal{N}(0,1)$, then for all $t > 1$,*

$$\Pr\left[|X| > t\right] \leq \exp\left(\frac{-t^2}{2}\right).$$

Lemma 5.4.3. *[122] If $\boldsymbol{\omega} \in \mathbb{R}^d$, such that it is drawn from a multivariate normal distribution, that is $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\omega}', \Sigma')$, and $\mathbf{x}$ is any fixed vector in $\mathbb{R}^d$, then*

$$\langle \mathbf{x}, \boldsymbol{\omega} \rangle \sim \mathcal{N}\left(\langle \mathbf{x}', \boldsymbol{\omega} \rangle, \mathbf{x}^T \Sigma' \mathbf{x}\right).$$

Lemma 5.4.4 (Bound on the norm of the ridge solution)**.**

$$\left|\left|\boldsymbol{\omega}^{\text{ridge}}\right|\right| \leq \frac{M \, \mathrm{Y}_{\max} \, n}{\lambda}.$$

Proof. We have

$$\begin{aligned}
\left|\left|\boldsymbol{\omega}^{\text{ridge}}\right|\right| &= \left|\left|[\mathbf{X}^T\mathbf{X}+\lambda I]^{-1}\mathbf{X}^T\mathbf{y}\right|\right| \\
&\leq \left|\left|[\mathbf{X}^T\mathbf{X}+\lambda I]^{-1}\right|\right| \left|\left|\mathbf{X}^T\right|\right| ||\mathbf{y}|| \\
&\leq \frac{1}{\lambda}\cdot\left|\left|\mathbf{X}^T\right|\right|\sqrt{n}\,\mathrm{Y}_{\max} \\
&\leq \frac{1}{\lambda}\sqrt{\mathbf{X}^T\mathbf{X}}\sqrt{n}\,\mathrm{Y}_{\max} \\
&\leq \frac{1}{\lambda}\sqrt{M^2 n}\sqrt{n}\,\mathrm{Y}_{\max} \\
&\leq \frac{M\,\mathrm{Y}_{\max}\,n}{\lambda}.
\end{aligned}$$

□

Proof of Theorem 5.4.1. Consider the score function

$$q((\mathbf{X},\mathbf{y}),\boldsymbol{\omega}) = \log(p(\mathbf{y}|\mathbf{X},\boldsymbol{\omega})).$$

Now,

$$\begin{aligned}
&\max_{(\mathbf{x}_1,\mathbf{y}_1)\sim(\mathbf{X}_2,\mathbf{y}_2)} |\log(p(\mathbf{y}_2|\mathbf{X}_2,\boldsymbol{\omega})) - \log(p(\mathbf{y}_1|\mathbf{X}_2,\boldsymbol{\omega}))| \\
&= \max_{(\mathbf{x}_1,\mathbf{y}_1),\,(\mathbf{x}_2,\mathbf{y}_2)} \frac{\left|(\langle\mathbf{x}_1,\boldsymbol{\omega}\rangle - y_1)^2\right|}{2{\sigma_n}^2}
\end{aligned}$$

This equation follows from the definition of neighbors; without loss of generalization we assume that neighboring $(\mathbf{X}_1,\mathbf{y}_1)$ and $(\mathbf{X}_2,\mathbf{y}_2)$ are exactly the same except for the first tuple $(\mathbf{x}_1, y_1)$. Then,

$$\left|(\langle\mathbf{x}_1,\boldsymbol{\omega}\rangle - y_1)^2 - (\langle\mathbf{x}_2,\boldsymbol{\omega}\rangle - y_2)^2\right| \leq (\langle\mathbf{x}_1,\boldsymbol{\omega}\rangle - y_1)^2.$$

$$\begin{aligned}
Pr\left\{\left|\langle\mathbf{x}_1,\boldsymbol{\omega}\rangle - \langle\mathbf{x}_1,\boldsymbol{\omega}^{\text{ridge}}\rangle\right| \geq t\right\} &\leq \exp\left(\frac{-t^2}{2(\mathbf{x}_1^T\Sigma_\lambda\mathbf{x}_1)}\right) \\
&\leq \exp\left(\frac{-t^2}{2\lambda_{\max}(\Sigma_\lambda)\,||\mathbf{x}_2||^2}\right)
\end{aligned}$$

where the first inequality follows from Lemmas 5.4.2 and 5.4.3. In the second inequality $\lambda_{\max}$ is the maximum eigen value of the covariance matrix Σ_λ.

Since the minimum eigen-value of $\Sigma_\lambda = [\mathbf{X}\mathbf{X}^T + \lambda I]^{-1}$ is greater than or equal to λ, we have

$$Pr\left\{ \left| \langle \mathbf{x}_1, \boldsymbol{\omega} \rangle - \langle \mathbf{x}_1, \boldsymbol{\omega}^{\text{ridge}} \rangle \right| \geq t \right\} \leq \exp\left(\frac{-\lambda t^2}{2M^2} \right)$$

This implies that with probability at least $1 - \exp\left(\frac{-\lambda t^2}{2M^2}\right)$,

$$(\langle \mathbf{x}_1, \boldsymbol{\omega} \rangle^2 - y_1)^2 \leq (\langle \mathbf{x}_1, \boldsymbol{\omega}^{\text{ridge}} \rangle - y_1)^2 + t^2 - 2t(\langle \mathbf{x}_1, \boldsymbol{\omega}^{\text{ridge}} \rangle - y_1)$$

$$\leq \langle \mathbf{x}_1, \boldsymbol{\omega}^{\text{ridge}} \rangle^2 + y_1^2 + 2y_1 \left| \langle \mathbf{x}_1, \boldsymbol{\omega}^{\text{ridge}} \rangle \right| + t^2 + 2t \left| \langle \mathbf{x}_1, \boldsymbol{\omega}^{\text{ridge}} \rangle \right| + 2t \left| y_1 \right|$$

Let, $\delta = \exp\left(\frac{-\lambda t^2}{2M^2}, \right)$ such that $t \leq \dfrac{\sqrt{2\log(1/\delta)}M}{\sqrt{\lambda}}$.

Hence, with probability $1 - \delta$, we have

$$\begin{aligned}(\langle \mathbf{x}_1, \boldsymbol{\omega} \rangle^2 - y_1)^2 \quad \leq \quad & ||\mathbf{x}||^2 \left|\left| \boldsymbol{\omega}^{\text{ridge}} \right|\right|^2 + {\mathrm{Y}_{\max}}^2 + 2\,\mathrm{Y}_{\max} ||\mathbf{x}|| \left|\left| \boldsymbol{\omega}^{\text{ridge}} \right|\right| + \\ & \frac{2\log(1/\delta)M^2}{\lambda} + 2\frac{\sqrt{2\log(1/\delta)}M}{\sqrt{\lambda}} \left(||\mathbf{x}|| \left|\left| \boldsymbol{\omega}^{\text{ridge}} \right|\right| + \mathrm{Y}_{\max} \right).\end{aligned}$$

Combining this with Lemma 5.4.4, we get the result. □

From Theorems 5.4.1 and 5.2.2 we have:

Theorem 5.4.5. *Sampling $\boldsymbol{\omega}^{\text{private}}$ from $\mathcal{N}(\bar{\boldsymbol{\omega}}, \bar{\Sigma})$ is (ε, δ)-differentially private, where*

$$\varepsilon = \frac{\alpha S}{\sigma_n^2}.$$

After sampling an $\boldsymbol{\omega}^{\text{private}}$ from $\pi_{\mathbf{X},\alpha}$, given an $\mathbf{x}^+$, the output is predicted using $f_{\boldsymbol{\omega}^{\text{private}}}(\mathbf{x}^+) = (\boldsymbol{\omega}^{\text{private}})^T \mathbf{x}^+$.

Since any $\boldsymbol{\omega}^{\text{private}}$ is sampled from the multivariate normal distribution π_α, any linear combination of $\boldsymbol{\omega}^{\text{private}}$ is drawn from the univariate normal distribution, that is

$$\boldsymbol{\omega}^{\text{private}} \sim \mathcal{N}({\mathbf{x}^+}^T \bar{\boldsymbol{\omega}}, {\mathbf{x}^+}^T \bar{\Sigma} \mathbf{x})$$

where $\bar{\omega}$ is the mean and $\bar{\Sigma}$ is the co-variance matrix of π_α.

5.4.2 Computationally efficient exponential mechanism for Bayesian regression

We briefly remark on a general relaxed exponential mechanism for Bayesian regression.

Using Bayes' rule, a differentially private posterior given a prior and a likelihood is:

$$p_\alpha(\boldsymbol{\omega}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \boldsymbol{\omega})^\alpha p(\boldsymbol{\omega})}{\int p(\mathbf{y}|\mathbf{X}, \boldsymbol{\omega})^\alpha p(\boldsymbol{\omega}) d\boldsymbol{\omega}}.$$

We saw an instance of this in Gaussian regression. Using this observation, we have the following general result.

Theorem 5.4.6. *If (U, δ) is a probabilistic upper bound on the global sensitivity of the log-likelihood $\log p(\mathbf{y}|\boldsymbol{X}, \boldsymbol{\omega}))$ (the scoring function) in Bayesian regression, then sampling from the posterior $p_\alpha(\boldsymbol{\omega}|\boldsymbol{X}, \mathbf{y})$ is $(2\alpha U, \delta)$-differentially private.*

We notice here that the exponential mechanism in case of Gaussian regression has a closed form, since it is also a Gaussian. In the case of Gaussian regression, since the likelihood is a Gaussian, corresponding to a (scaled) least square loss function, and the prior is also a Gaussian, the resulting differentially private posterior is Gaussian. In general, this may not be the case. For example, if the prior on Ω is Laplacian and the likelihood is Gaussian, the resulting posterior distribution does not have a closed form and it is difficult to sample from it. In Bayesian analysis, if the posterior and the prior distributions are of the same family, then they are called conjugate distributions and the prior is called the conjugate distribution for the likelihood (See [122] for example). The Gaussian distribution is a conjugate distribution for the Gaussian likelihood. There are others, such as the Gamma and Wishart distributions. If the likelihood has another form, such as an exponential distribution, the gamma distribution

is a conjugate prior for this.

5.5 Risk Bounds

Zhang [125] discusses the role of the regularization constant λ in inducing the so-called *effective dimension* on the predictor space. Assume that the true underlying function is f. To illustrate the effect of λ, we assume that the design matrix $\mathbf{X}$ is fixed, the so-called *fixed-design* setting. We see that the expected square error of the predictor $\hat{\mathbf{y}} = \hat{f}(\mathbf{X})$ is given by $\mathbb{E}_{\mathbf{y}} ||\hat{\mathbf{y}} - f(\mathbf{X})||^2$. Since this is the fixed-design setting, the randomization comes only from the noise, leading to randomization in $\mathbf{y}$.

The ridge regression predictor from Equation 5.4 is a linear estimator of the form $f = P\mathbf{y}$, where

$$P = \mathbf{X}[\mathbf{X}^T\mathbf{X} + \lambda I]^{-1}\mathbf{X}^T.$$

The expected mean-squared-error of $\hat{f}$ is given by

$$\mathbb{E}\left|\left|\hat{f} - f\right|\right|^2 = ||Pf - f||^2 + \frac{\mathrm{Tr}(P^TP)}{n}\sigma_n^2 \tag{5.9}$$

since $y_i = f(\mathbf{x}_i) + \eta_i$, where $\eta_i \sim \mathcal{N}(0, \sigma_n^2)$. The first term represents the bias and the second the variance. If P is close to the identity matrix, the bias will be small, but on the other hand, the quantity $\mathrm{Tr}(P^TP)$ will be close to d the dimensionality of the data, and this may not be desirable in high-dimensional settings. This is why Zhang uses the term effective dimensionality for this quantity and discusses how the regularization term λ influences this quantiy. For the ridge regression the term is

$$\begin{aligned}
&\mathrm{Tr}(P^TP) \\
&\quad = \mathrm{Tr}[\mathbf{X}[\mathbf{X}^T\mathbf{X} + \lambda I]^{-1}\mathbf{X}^T]^T\mathbf{X}[\mathbf{X}^T\mathbf{X} + \lambda I]^{-1}\mathbf{X}^T \\
&\quad \leq \mathrm{Tr}[\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T]
\end{aligned}$$

This quantity $\mathrm{Tr}[\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T]$ is called the effective dimension of the predictor

at scale λ [125] or effective degrees of freedom [50]. We have

$$D_\lambda = \text{Tr}[\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T] \tag{5.10}$$

Let $\lambda_j, \sim 1 \leq j \leq d$ represent the eigen-values of the the matrix $\mathbf{X}^T\mathbf{X}$, then we have

$$D_\lambda = \sum_{j=1}^{d} \frac{\lambda_j}{\lambda_j + \lambda}.$$

When $\lambda \to 0$, then the effective dimension at scale λ is close to d the underlying dimension of the predictor space. When $\lambda \to \infty$, the effective dimension tends to zero.

We have seen before that the regularization constant of ridge regression is related to the noise and prior parameters of Gaussian regression, via the following relationship, $\lambda = \sigma_n^2/(\alpha\tau^2)$. Therefore, lower values of ε and consequently α (higher privacy) correspond to higher values of λ, which results in lower effective dimensionality of the predictor. This is the reason why we can apply this methodology to high-dimensional differentially-private regression. In the next section, we derive bounds on the expected error of sampling from π_α in terms of the effective dimension of the predictor space and the norm of the "best" predictor in Ω rather than the actual dimension d of the input space.

5.5.1 Risk bounds for differentially private regression

Let,

$$f_{\omega*} \in \inf_{\omega} R(f_\omega)$$

Generalization bounds state the risk of the predictor in terms of the risk of the best predictor $f_{\omega*}$ in the predictor space Ω. Using generalization bounds from Zhang [125], we derive bounds on the expected risk of the private predictor $\boldsymbol{\omega}^{\text{private}}$.

We reproduce the following theorem of Zhang [125] that provides generalization bounds for ridge regression.

Theorem 5.5.1. *[125] Assume that the derivative of the loss function is bounded, that is,* $|f_{\omega}(\mathbf{x}) - y| \leq b_{\Omega}$ *and that* $||\mathbf{x}|| \leq M$*, almost surely. Then* $\forall \lambda > 0$*, with probability at least* $1 - 4\exp(-t)$*, we have:*

$$R(f_{\boldsymbol{\omega}^{\text{ridge}}}) \leq R(f_{\boldsymbol{\omega}*}) + \frac{\lambda}{n}\left|\left|\boldsymbol{\omega}^{\text{ridge}} - \boldsymbol{\omega}*\right|\right|^2 +$$
$$\left[\frac{4tD_{\frac{\lambda}{n}}}{n} + \frac{8t^2M^2}{\lambda n}\right]\left(b_{\Omega} + M\left|\left|\boldsymbol{\omega}^{\text{ridge}} - \boldsymbol{\omega}*\right|\right|^2\right)$$

This bound holds for using the ridge estimator $f_{\boldsymbol{\omega}^{\text{ridge}}}$ to make a prediction. However, in our case, we need to sample a $\boldsymbol{\omega}^{\text{private}}$ from π_{α} which will then be used to make a prediction. We bound the expected risk $\mathbb{E}_{\boldsymbol{\omega}\sim\pi_{\alpha}}R(f_{\boldsymbol{\omega}})$ of a private predictor $\boldsymbol{\omega}$ sampled from π_{α}, to the risk of the ridge predictor, $R(f_{\boldsymbol{\omega}^{\text{ridge}}})$. Let $\lambda_{\Sigma_{\alpha}}$ be the maximum eigen value of the co-variance matrix Σ_{α} of π_{α}, then assuming that $||\mathbf{x}|| \leq M$, $\forall \mathbf{x}$, we have

Lemma 5.5.2.

$$\mathbb{E}_{\boldsymbol{\omega}\sim\pi_{\alpha}}R(f_{\boldsymbol{\omega}}) \leq R(f_{\boldsymbol{\omega}^{\text{ridge}}}) + \lambda_{\Sigma_{\alpha}}M^2$$

Proof. We have

$$\mathbb{E}_{\boldsymbol{\omega}}(\mathbf{x}^T \cdot \boldsymbol{\omega} - y)^2 = \mathbb{E}_{\boldsymbol{\omega}}(\mathbf{x}^T \cdot \boldsymbol{\omega})^2 + \mathbb{E}_{\boldsymbol{\omega}}y^2 - \mathbb{E}_{\boldsymbol{\omega}}2y(\mathbf{x} \cdot \boldsymbol{\omega})$$

$$= Var(\boldsymbol{\omega}.\mathbf{x}) + (\mathbb{E}_{\boldsymbol{\omega}}\mathbf{x}^T.\boldsymbol{\omega})^2 + y^2 + 2y\mathbb{E}_{\boldsymbol{\omega}}(\mathbf{x}^T \cdot \boldsymbol{\omega})$$

$$= \mathbf{x}^T A^{-1}\mathbf{x} + (\mathbf{x}^T \cdot \boldsymbol{\omega}^{\text{ridge}} - y)^2$$

$$\leq (\mathbf{x}^T \cdot \boldsymbol{\omega}^{\text{ridge}} - y)^2 + \lambda_{\Sigma_{\alpha}}||\mathbf{x}||^2$$

Taking expectation over $\mathbf{x}, y$, we have the result. □

Using this lemma with Theorem 5.5.1 and the conditions therein, we have

Theorem 5.5.3. *With a probability of at least* $1 - 4\exp(-t)$*,* $\forall \lambda > 0$*, we have:*

$$\mathbb{E}_{\boldsymbol{\omega}\sim\pi_{\alpha}}R(f_{\boldsymbol{\omega}}) \leq R(f_{\boldsymbol{\omega}*}) + \frac{\lambda}{n}\left|\left|\boldsymbol{\omega}^{\text{ridge}} - \boldsymbol{\omega}*\right|\right|^2 +$$

$$\left[\frac{4tD_{\frac{\lambda}{n}}}{n} + \frac{8t^2M^2}{\lambda n}\right]\left(b_\Omega + M\left|\left|\boldsymbol{\omega}^{\text{ridge}} - \boldsymbol{\omega}*\right|\right|^2\right) + \frac{M^2}{\lambda}$$

Compare this to earlier bounds proved by Chaudhuri et al. [18] of the form, where with probability at least $1 - 4\exp(-t)$,

$$R(f_{\boldsymbol{\omega}^{\text{private}}}) \leq R(f_{\boldsymbol{\omega}*}) + O\left(\frac{d^2\log^2(d)t^2}{\lambda n} + \frac{t}{\lambda} + \frac{\lambda}{n}\left|\left|\boldsymbol{\omega}*\right|\right|^2 + \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{d}}\right)\sqrt{t}\right)$$

which necessarily depend on the dimensionality d of the predictor (or input) space.

Notice that the risk bounds for differentially private Gaussian regression have a linear dependence on the effective dimension $D_{\frac{\lambda}{n}}$ which (as noted earlier in this section) in the worst case is d (but, generally smaller because of the regularization factor λ). The risk bounds of Chaudhuri et al. [18], on the other hand, have a quadratic dependence on d. However, when d the dimension of the predictor space is a small constant, it may not be clear a priori which method is better to use because of the constants involved that depend on assumptions on the data.

The next section presents experimental results that demonstrate the accuracy of $\boldsymbol{\omega}^{\text{private}}$.

5.6 Experiments

We carry out experiments on real-world data using the Boston housing data set [39], and compare the ridge regression predictor $\boldsymbol{\omega}_\lambda^{\text{ridge}}$ to the private predictor $\boldsymbol{\omega}_\lambda^{\text{private}}$. The Boston Housing data set has thirteen predictor variables and one response variable (median value of the owner-occupied homes) and consists of 506 samples. We use 5-fold cross validation and repeat it 100 times for each value of ε and report the average mean-squared-error over the resulting 500 iterations in Figure 5.1. We normalize the dataset so that each value in a column lies in the range $[-1, 1]$ and such that for every row vector $\mathbf{x}$, $||\mathbf{x}|| \leq 1$. The x-axis shows different values of ε corresponding to a value of $\delta = 0.001$ in the (ε, δ) from Theorem 5.4.1. The "mean of GR" curve refers

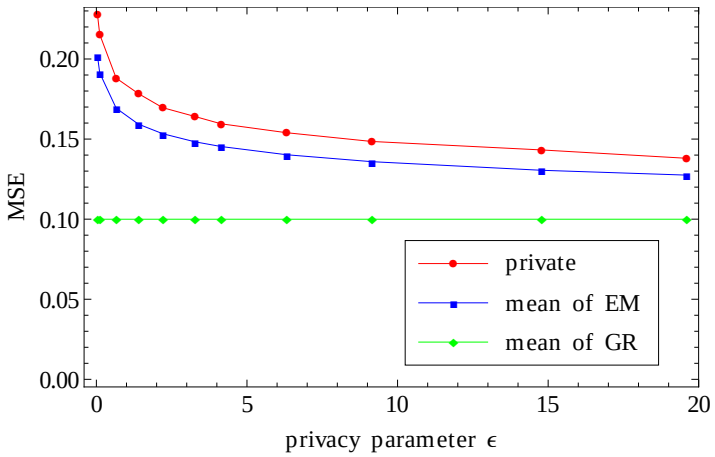


Figure 5.1: Mean Squared Error (MSE) of the predictor for the Boston Housing data set, averaged over 100 instances of 5-fold cross validation

to Gaussian Regression without privacy with an "optimal" σ_n^2 and τ^2 (corresponding to an "optimal" λ in the regularized ridge regression formulation). This will be our baseline to compare against. The "mean of EM" curve refers to prediction done by the mean of the exponential mechanism for the corresponding ε. The "private" curve refers to prediction done by sampling from the Exponential Mechanism. In order to provide a smoother estimate, we sample from the Exponential Mechanism 10 times and use the average predictor (while accounting for the overall privacy budget using Composition Theorem 2.2.4).

As we can see, the (average) sample over 10 draws from the Exponential Mechanism is quite close to its actual mean, which in turn for increasing values of privacy ε, converges to the "optimal" predictor. Even for values of $\varepsilon = 0.1$, for example, the Mean Squared Error using the private predictor is close to that achieved by the optimal one. In Figure 5.2 we look into the region of smaller (and, therefore, generally more meaningful) values of ε.

From Figure 5.1, we see that the private estimator performs quite favorably as compared to the ridge estimator, even as we do expect the performance to decrease with increasing privacy.

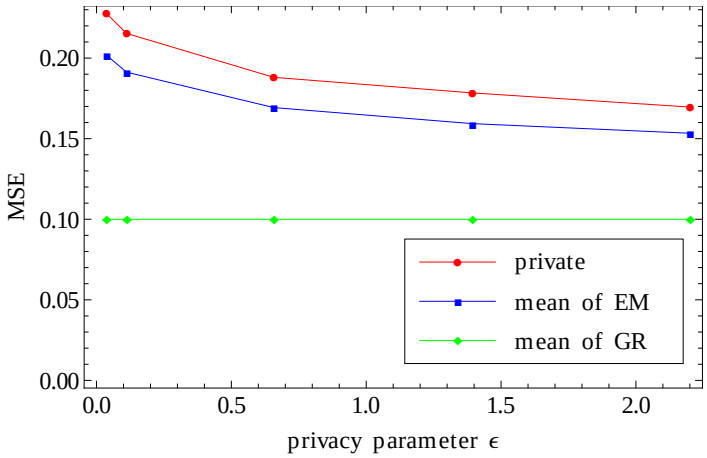


Figure 5.2: Mean Squared Error (MSE) of the predictor for the Boston Housing data set, for lower values of ε

5.7 Conclusions and future work

We propose a method for differentially private Gaussian regression. Using experimental results, we show that this method performs favorably over real-world data. We achieve generalization error bounds that are not not much worse than those for ridge regression. Further, utilizing the convenience of conjugate priors for Gaussian regression, we can compute the differentially private predictor without necessarily adding noise proportional to the dimension of the predictor space, which the Laplacian mechanism necessitates. This connection to Gaussian regression in particular and Bayesian regression, in general, may help in application of a host of techniques from Bayesian analysis to differentially private machine learning.

Several future directions present themselves. First, we can examine the conjugate priors of several other classes of likelihood to propose computationally efficient exponential mechanisms for other regression and classification problems. Second, in cases where the exponential mechanism does not have a closed form, we can examine approximation techniques prevalent in the literature such as Markov Chain Monte-Carlo methods. However, this will need to be done in a privacy-preserving manner and

involves more than a straightforward application. Third, we could apply Gaussian regression methods to online learning to propose methods for differentially private online learning.

6

Pan-private Algorithms via Statistics on Sketches

6.1 Introduction

Consider the following simple, motivating example. Say we keep track of visitors that enter or leave a large facility (offline sites like a corporate or government office or online like websites). When queried, we wish to determine how many different visitors are on-site. This is a *distinct count* query. Unlike a data publication scenario where data is static after it is published, here the data is dynamic, varying over time, and the distinct count query may be posed any time, or even multiple times.

Our focus is first on privacy. Known methods for instance would be able to maintain the list of all IDs currently on site and, when queried, compute the precise answer D but return $D + \alpha$ for some suitable α that balances utility of the approximate distinct count against compromising the privacy of any particular ID. This intuitive approach has been formalized and a rich theory of differential privacy now exists for limitations and successes of answering this and many other queries privately.

Now, we go beyond privacy, and consider security. In particular, suppose the program—that tracks the data and answers the query—is compromised. Of course, this may happen because a malicious intruder hacks the system. But more subtly, this may happen because an insider with access, such as a systems administrator, may turn curious or crooked; data analysis may be outsourced to far away countries where people and laws are less stringent; or the contents of the registers may be subpoenaed by

law or security officials. How can distinct count query processing be done securely, as well as with privacy? Maintaining a list of IDs on-site will not work, since it compromises all such IDs when a breach occurs. A natural idea is to hash (or encrypt) IDs into a new space that hides the identity. On a closer look, this too will not work since a breach will reveal the hash function or the encrypting key, and the intruder can exhaustively enumerate potential visitors to a site and determine the identity of all visitors currently on-site; this is known as a *dictionary attack*. (Notice that we are not limiting the intruder to have any computational constraints; however, even for computationally bounded adversaries, no cryptographic guarantees are known when the adversary has full access to the private key.)

Maintaining a random sample of the IDs too will not work since it compromises the sampled IDs, and further, sample-based solutions are not known for estimating D with dynamic data when visitors arrive and depart, only for *partly dynamic* case when departure of visitors is not recorded. One can be principled and use well-known *sketches* since they only keep aggregate information (like counts, projections), rather than explicit IDs, and therefore afford natural obfuscation. While such solutions approximate distinct count well with dynamic data, they also do not work because they rely on hash functions to aggregate IDs: during the breach, the intruder obtains access to the hash functions, and can carry out a dictionary attack, compromising some of the IDs.

This example illustrates the issues involved when one seeks privacy and security simultaneously: even if we rely on cryptography and use exponential space or time to process the dynamic data, there are no known methods for even simple queries like distinct count. Of course, in reality, the dynamic data may have more attributes and many queries are of interest from estimating statistics like averages, to data mining tasks like finding heavy hitters, anomalies and others.

In this chapter we develop algorithmic techniques to compute the distinct count

statistic for fully dynamic data. In order to do that, we need to formalize security and privacy.

We consider the *fully dynamic* setting in which for each user, represented by an ID i, (drawn from a universe $\mathcal{U}$), we maintain a *state* a_i, which consists of cumulative updates to i until time t. At each time step, the state of a single user is modified by incrementing or decrementing updates (in arbitrary integral values). In *partly dynamic* data, only increments are allowed. In addition, we call this *fully* or *partly streaming*, respectively, if the algorithms use sublinear space (typically, space polylogarithmic in various parameters).

We adopt the notion of differential privacy. In the context of online sequences, a randomized function f is *differentially private* with respect to the IDs if the probability distribution on the range of f is not sensitive to changing the state of any single user ID. To add security to privacy, Dwork et al. [30, 34] formalized the notion of *pan-privacy*. Informally, both the distribution of the internal states of the algorithm and the distribution of outputs should be insensitive to changing the state of a single user. This addresses privacy even in the case when there is one unannounced memory breach by the adversary. In this chapter, we study this model (see [34] for variants of the model). Without some "secret state" (such as a secret set of hash or cryptographic keys), it might seem impossible to estimate statistics privately, but, surprisingly, Dwork et al. [34] showed that several interesting statistics on streams can be estimated accurately on *partly dynamic data*. Their algorithms are based on the technique of randomized response [121] and sampling.

6.1.1 Our contributions

We design the first known pan-private algorithms for *distinct count* in the fully dynamic model. We refer to our full paper [92] for algorithms on *cropped first moment*

and *heavy hitters count* for fully dynamic data and lower bound techniques. Our algorithms rely on *sketches* widely useful in streaming: in some cases, we add suitable noise using a novel approach of calibrating noise to the underlying problem structure and the projection matrix of the sketch; in other cases, we maintain certain statistics on sketches, and in yet others, we define novel sketches. In what follows, m is the size of the universe of IDs. These statistics, in one form or the other, have a long history, and are considered basic in data analysis tasks over dynamic data in the past few decades and different streaming solutions are known for these problems:

Given a sequence of updates, the *Distinct Count* statistic D is the number of user IDs with nonzero state:

$$D = |\{i \in \mathcal{U} : a_i \neq 0\}|.$$

We present an algorithm that is ε-pan private and outputs an estimate $(1+\alpha)D \pm$ **polylog** with probability at least $1-\delta$, where **polylog** is a polylogarithmic function of various input parameters and m is the size of the universe. It directly uses a sketch known before based on stable distributions for estimating distinct count [20, 58], but maintains noisy versions based on a new method of adding noise tailored to the sketch and the underlying problem.

In [92] this result is complemented by showing lower bounds. Let $\mathcal{A}$ be an online (not necessarily streaming) algorithm that outputs $D \pm o(\sqrt{m})$ with small constant probability. Then we show that $\mathcal{A}$ is not ε-pan private for any constant ε. The lower bound holds irrespective of the memory used by $\mathcal{A}$—even if the memory is $\Omega(m)$. Further, in [92], we show a lower bound of $(1+\alpha)D \pm \textbf{polylog}(1/\delta)$ for algorithms that succeed with probability $1-\delta$, essentially tight up to additive polylog terms with the pan-private algorithm presented in this thesis.

We emphasize that our algorithm works on fully dynamic data which has not been considered in pan-privacy before. Dwork et al. [34] provide pan-private algorithms

for the distinct count statistic for partly dynamic data. Our definitions of the problem we consider differs slightly from those in [34]: we consider distinct count instead of density but our definition specifies a problem that is at least as hard to approximate as those in [34].

The algorithms presented in [34] are based on sampling and randomized response and do not work with fully dynamic data. This is why we had to develop alternative techniques based on maintaining statistics over sketches. Surprisingly, our distinct counts algorithm provides estimates for fully dynamic data that matches the best bounds from [34] for partly dynamic data (up to additive polylog factors for distinct counts, and multiplicative factor 2 for cropped sum). The hashing technique used in [34] to obtain a constant multiplicative approximation for distinct count and cropped sum has an implicit additive factor of $O(\log m)$ because of adding Laplacian noise linear in $\log m$, giving an approximation of $(1 \pm \alpha)D \pm O(\log m)$. In fact a pure multiplicative approximation of $1 \pm \alpha$, for any constant α, is prohibited by our lower bounds on distinct counts [92].

Finally, we make an intriguing observation. Pan-privacy does not require algorithms to have any computational or storage constraints; it only requires differential privacy and security against intrusion. In fact, the lower bounds proved in [92] hold against algorithms that can use unbounded storage and perform arbitrary computations per update. On the other hand, the pan-private algorithm for distinct count we present here are actually streaming algorithms that use only polylogarithmic time per update and polylogarithmic space. This may be an artifact of the techniques we use. We leave it open to find problems for which pan-private algorithms exist that necessarily use large (say polynomial) space.

We start in Section 6.2 by introducing relevant definitions and notation. In Section 6.3, we present our pan-private algorithm by keeping statistics on sketches.

6.2 Background

In this section we introduce notation and definitions and recapitulate earlier work on pan-privacy and sketches that we build on.

6.2.1 Model and notation

We are given a universe $\mathcal{U}$, where $|\mathcal{U}| = m$. An *update* is defined as an ordered pair $(i, d) \in \mathcal{U} \times \mathbb{Z}$. Consider a semi-infinite sequence of updates $(i_1, d_1), (i_2, d_2), \ldots$; the *input* for all our algorithms consists of the first t updates, denoted $S_t = (i_1, d_1), \ldots, (i_t, d_t)$. The *state vector* after t updates is an m-dimensional vector $\mathbf{a}^{(t)}$, indexed by the elements in $\mathcal{U}$. (We omit the superscript when it is clear from the context.) The elements of the vector state vector $\mathbf{a} = \mathbf{a}^{(t)}$, store the cumulative updates to i: $a_i = \sum_{j:i_j=i} d_j$. Each a_i is referred to as the state of ID i. In the *partly dynamic* model, all updates are positive, i.e. $\forall j : d_j \geq 0$; in the *fully dynamic* model, updates can be both positive (*inserts*), i.e. $d_j \geq 0$, and negative (*deletes*), i.e. $d_j < 0$, but at any time, $a_i \geq 0$ (since deletes cannot exceed inserts). We assume an upper bound Z on the maximum absolute value of the state of any $i \in \mathcal{U}$, i.e. $a_i \leq Z$ at any time step.

6.2.2 Pan-privacy

We refer the reader to Chapter 2 to review the notion of pan-privacy and for formal definitions. Here we recapitulate a few of these definitions and theorems for convenience.

We will use composition theorem (Theorem 2.2.4) from Chapter 3 reproduced here:

Theorem 6.2.1. *[33] Given mechanisms $\mathcal{M}_i$, $i \in [r]$ each of which provide ε_i-differential privacy, then the overall mechanism $\mathcal{M}$ that executes these r mechanisms with independent randomness and outputs the vector of their outputs, provides $\left(\sum_{i\in[r]} \varepsilon_i\right)$-differential privacy.*

The second composition result we use concerns composition of the neighbor relation. First we define the notion of ℓ-neighborhood, which is a binary relation induced by the neighbor relation.

Definition 3. *Given a neighbor relation $\sim$, the ℓ-neighbor relation $\sim_\ell$ is defined as follows. Two input datasets D, D' are said to be* 1-neighbors*—i.e. $D \sim_1 D'$, if $D \sim D'$. For a natural number $\ell > 1$, D, D' are said to be ℓ-*neighbors*—i.e. $D \sim_\ell D'$ if $D \sim_{\ell-1} D'$ or there exists a dataset $D'' \sim D'$ such that $D'' \sim_{\ell-1} D$.*

Another way to think of ℓ-neighbors is as inputs that are linked by a path of length at most ℓ in the graph induced by the neighbor relation. Next we present a theorem of Dwork et al. formally showing that differential privacy is resilient to composition of the neighbor relation.

Theorem 6.2.2. *[33] If a function f provides ε-differential privacy with respect to $\sim$, then f provides $\ell\varepsilon$-differential privacy with respect to $\sim_\ell$.*

Pan privacy, as introduced in Chapter 2, guarantees a participant that his/her risk of being identified by participating in a data set is very little even if there is an external intrusion on the internal state of the analyzing algorithm. Consider two neighboring online sequences of updates $S = \big((i_1, d_1), \ldots, (i_t, d_t)\big)$ and $S' = \big((i'_1, d'_1), \ldots, (i'_{t'}, d'_{t'})\big)$ associated with state vectors $\mathbf{a}$ and $\mathbf{a}'$ respectively as introduced in Definition 4.3.1.

As mentioned in Chapter 2, our notion of neighborhood is slightly different from the Dwork et al. [34] definition, where any two data streams S and S' are neighbors if they differ only in the presence or absence of any number of occurrences of any element $i \in \mathcal{U}$ (i.e. $\mathbf{a}$ and $\mathbf{a}'$ have hamming distance at most 1). Our definition ensures that two neighboring sequences of updates are of the same "length," in the sense that the sum of the updates over all items is the same for both S and S', that is, $\sum_{i=1}^{t} d_k = \sum_{i=1}^{t'} d'_k$. For this purpose, we constrain the sum of the updates of the occurrences of

item i in S to be conserved when they are replaced by item j in S'. In our definition, the total weight of updates is public, but, still, an adversary cannot distinguish between appearances of ID i or ID j, even if the adversary knows all other appearances of all other IDs.

We comment on the composability of our definition of neighborhood. Applying Definition 3, we see that two sequences S and S' will be ℓ-neighbors if there exist (possibly multi) sets of ID's of cardinality ℓ: $\{i_1, i_2 \dots i_\ell\}$ and $\{j_1, j_2, \dots, j_\ell\}$ all from $\mathcal{U}$, such that some occurrences of each $i_k, 1 \leq k \leq \ell$ in S are replaced by some occurrences of $j_k \neq i_k, 1 \leq k \leq \ell$ in S'. There is no other restriction on the $j_k's$; they may be all equal, different or any subset of these may be equal. Hence Theorem 6.2.2 is applicable to our definition of ℓ-neighbors. We also reproduce the definition of user-level pan-privacy here:

Definition 4 (User-level pan privacy [34])**.** *Let* **Alg** *be an algorithm. Let I denote the set of internal states of the algorithm, and let σ the set of possible output sequences. Then algorithm* **Alg** *mapping input prefixes to the range $I \times \sigma$, is* pan-private (against a single intrusion) *if for all sets $I' \subseteq I$ and $\sigma' \subseteq \sigma$, and for all pairs of user-level neighboring data stream prefixes S and S'*

$$\Pr[\mathbf{Alg}(S) \in (I', \sigma')] \leq e^{\varepsilon} \Pr[\mathbf{Alg}(S') \in (I', \sigma')]$$

where the probability spaces are over the coin flips of the algorithm **Alg***.*

6.2.3 Sketches and stable distributions

In this section we discuss previous work in sketch-based streaming.

Definition 5. *[96] A distribution $\mathcal{S}(\mathrm{p})$ over $\mathbb{R}$ is said to be p-stable if there exists $p \geq 0$ such that for any n real numbers $b_1, \dots, b_m$ and i.i.d. variables $Y_1, \dots, Y_m$ with distribution $\mathcal{S}(\mathrm{p})$, the random variable $\sum_i b_i Y_i$ has the same distribution as the random variable $(\sum_i |b_i|^p)^{1/p} Y$, where Y is a random variable with distribution $\mathcal{S}(\mathrm{p})$.*

Examples of p-stable distributions are the Gaussian distribution, which is 2-stable, and the Cauchy distribution, which is 1-stable. Stable distributions have been used to compute the L_p norms of vectors ($L_p = (\sum_i a_i^p)^{1/p}$) in the streaming model [20, 58].

Let X be a matrix of random values of dimension $m \times r$, where each entry of the matrix $X_{i,j}$, $1 \leq i \leq m$, and $1 \leq j \leq r$, is drawn independently from $\mathcal{S}(\mathrm{p})$, with p as small as possible. The *sketch vector* $\mathrm{sk}(\mathbf{a})$ is defined as the dot product of matrix X^T with $\mathbf{a}$, so $\mathrm{sk}(\mathbf{a}) = X^T \cdot \mathbf{a}$. From the property of stable distributions we know that each entry of $\mathrm{sk}(\mathbf{a})$ is distributed as $(\sum_i |a_i|^p)^{1/p} X_0$, where X_0 is a random variable chosen from a p-stable distribution. The sketch is used to compute $\sum_i |a_i|^p$ for $0 < p < \alpha / \log Z$, from which we can approximate $D^{(t)}$ up to a $(1+\alpha)$ factor(See [20] for details). By construction, any $\mathrm{sk}(\mathbf{a})_j$ can be used to estimate L_p^p. Cormode et al. [20] and Indyk [58] obtain a low-space good estimator for $(\sum_i |a_i|^p)$ by taking the median of all entries $\left|\mathrm{sk}(\mathbf{a})_j\right|^p$ over j:

Theorem 6.2.3. *If the continuous stable distribution is approximated by discretizing it to a grid of size* $(\frac{mZ}{\alpha\delta})^{O(1)}$*, the support of the distribution* $\mathcal{S}(\mathrm{p})$ *from which the values* $X_{i,j}$ *are drawn is truncated beyond* $(mZ)^{O(1)}$*, and* $r = O(1/\alpha^2 \cdot \log(1/\delta))$*, then with probability* $1 - \delta$*,*

$$(1-\alpha)^p \,\mathrm{median}_j \left|\mathrm{sk}(\mathbf{a})_j\right|^p \leq \mathrm{median}\,|X_0|^p \,(\sum_i |a_i|^p)$$
$$\leq (1+\alpha)^p \,\mathrm{median}_j \left|\mathrm{sk}(\mathbf{a})_j\right|^p$$

where $\mathrm{median}\,|X_0|^p$ *is the median of absolute values (raised to the power p) from a (truncated, discretized) p-stable distribution.*

We will use these details in Algorithm 7 in Section 6.3.1 to propose a pan-private algorithm for distinct counts.

6.3 Pan-private algorithms for fully dynamic data

In this section we present our pan-private algorithms that work for fully dynamic data. Our algorithms follow the outline:

1. initialize a sketch to a noisy vector chosen from an appropriate distribution;
2. update the sketch linearly (linearity may be over the real field, or modulo a real number); and
3. compute a global statistic of the sketch.

The fact that, for all the algorithms, the state of the algorithm is a linear function of its input and the noisy initialization allows us to characterize the distribution of the state of the algorithm at any time step; this property is essential to both the privacy and utility analyses of our algorithms. While particular entries in the sketches may not be accurate approximations of the states of the user IDs, the global statistic computed at the end can be shown to be an accurate estimate of the desired value.

6.3.1 Distinct count

We use sketching based on stable distributions outlined in Section 6.2.3 to design an algorithm for pan-private estimation of the distinct count statistic $D^{(t)}$. We exploit the linearity property of the sketches by maintaining a noisy version of the sketch in order to achieve pan-privacy. Because the sketch is a linear function of the state vector, it is enough to add an initial noise vector drawn from the appropriate distribution. To do so without adding too much noise, we develop a new technique of adding noise calibrated to the underlying random projection matrix and the nature of the statistic we are computing, using the exponential mechanism of McSherry and Talwar [87]. As a consequence, while this mechanism, in general, is not computationally efficient, it

provides us with a new framework for adding noise that is not "function oblivious." The established Laplace mechanism [33], that adds noise calibrated to the *global sensitivity* of the function, beyond being aware of the global sensitivity of the function is oblivious of the underlying structure of the problem. This is important for our application as the sensitivity of the stable distribution sketch can be very high due to the heavy tails of p-stable distributions for small p.

Next we describe the mechanism we use to draw the noise vector.

An initializing noise vector: We use the exponential distribution to generate a random noise vector that initializes the sketch. The sketch vector has dimension r; let us denote the i-th row of X as X_{i*} and the j-th column of X as X_{*j}.

We use the exponential mechanism of McSherry and Talwar with the following quality function q. If the true sketch vector is $\mathrm{sk}(\mathbf{a})$, then

$$q\left(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}\right) = -d\left(\mathrm{sk}(\mathbf{a}) - \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}\right),$$

where d is defined as:

$$\begin{aligned} &d(\mathbf{z}) := \min \|\mathbf{c}\|_0 \ \text{ s.t.} \\ &\mathbf{z} = X^T\mathbf{c} \\ &\forall i \in [m] : c_i \in [-2Z, 2Z]. \end{aligned}$$

If the above program is infeasible, then $d(\mathbf{z}) = \infty$.

Given sketch vector $\mathrm{sk}(\mathbf{a})$, the mechanism picks a sketch $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$ from a distribution, μ_ε given by

$$\mu_\varepsilon(\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}) \propto \exp\left(\frac{\varepsilon}{4} q(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}})\right).$$

Intuitively, the distance function d roughly measures the minimum number of items in the state vector $\mathbf{a}$, whose entries need to be changed in order to get from $\mathrm{sk}(\mathbf{a})$ to $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$. This is used in the utility analysis.

Next, we need to compute the sensitivity Δ_q of q defined as

$$\mathrm{GS}_q = \max_{\mathbf{x} \sim \mathbf{z}, \mathbf{y}} |d(\mathbf{x}, \mathbf{y}) - d(\mathbf{z}, \mathbf{y})| .$$

Lemma 6.3.1. *For q as defined above, $\mathrm{GS}_q \leq 2$.*

Proof. If $\mathrm{sk}(\mathbf{a})$ and $\mathrm{sk}(\mathbf{a}')$ are the true sketch vectors for neighboring sequences of updates corresponding to state vectors $\mathbf{a}$ and $\mathbf{a}'$ respectively, then for some $i, j \in \mathcal{U}$, $i \neq j$, $\mathrm{sk}(\mathbf{a}') = \mathrm{sk}(\mathbf{a}) + c_i X_{i*} + c_j X_{j*}$, for some $c_i, c_j \in [-2Z, 2Z]$. Therefore,

$$\begin{aligned}
\mathrm{GS}_q &\leq \max_{\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a}'), \mathbf{y}} |d(\mathrm{sk}(\mathbf{a}) - \mathbf{y}) - d(\mathrm{sk}(\mathbf{a}') - \mathbf{y})| \\
&\leq \max_{\mathrm{sk}(\mathbf{a}), c_i, c_j, \mathbf{y}} |d(\mathrm{sk}(\mathbf{a}) - \mathbf{y}) - d(\mathrm{sk}(\mathbf{a}) - \mathbf{y} + c_i X_{i*} + c_j X_{j*})| \\
&\leq 2.
\end{aligned}$$

□

Let $B = \mathbf{poly}(m, Z)$ be large enough so that: (1) Theorem 6.2.3 holds, (2) for any $c \in [-2Z, 2Z]^m$, $X^T c \in [-B, B]^r$. We pick an initializing vector $\mathbf{y}$ using the exponential distribution with quality function q from the range $\mathcal{R} = [-B, B]^r \cap \langle X_{1*}, \ldots, X_{m*} \rangle$, discretized to within $\mathbf{poly}(m, Z, 1/\alpha, 1/\delta)$ precision, again so Theorem 6.2.3 holds. Notice that $\log \mathcal{R} = O(r \cdot \log(\mathbf{poly}(m, Z, 1/\alpha, 1/\delta)))$, which implies that $\log \mathcal{R} = \mathbf{poly}(\log m, \log Z, 1/\varepsilon, 1/\alpha, \log(1/\delta))$.

The Algorithm: After initializing, we update and decode the sketch as in the non-private algorithm. Before outputting the final answer, we draw another vector using the exponential mechanism with the same parameters. The algorithm is shown below as Algorithm 7.

Since updates are linear, and $q(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}})$ is a function of $\mathrm{sk}(\mathbf{a}) - \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$, initializing the sketch to a vector picked using the exponential mechanism with quality function $q(\mathbf{y}, 0) = -d(\mathbf{y})$ ensures that any state is $2\frac{\varepsilon}{4}\,\mathrm{GS}_q$-differentially private. More formally, from Theorems 2.3.4 and 6.2.1, and Definition 4:

Lemma 6.3.2. *At any step in Algorithm 7, the state of the algorithm is a sketch and the distribution over states is given by the exponential mechanism with quality function*

$$q(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}) = -d(\mathrm{sk}(\mathbf{a}) - \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}).$$

Hence the algorithm is ε-pan private.

Also, by simple application of Theorem 2.3.4:

Lemma 6.3.3. *The initializing vector* $\mathbf{y}$ *has*

$$d(\mathbf{y}) \leq 4\frac{\log|\mathcal{R}|}{\varepsilon} + \frac{4}{\varepsilon}\log 1/\delta \leq \mathbf{polylog}(m, Z, 1/\varepsilon, \log(1/\delta), 1/\alpha)$$

with probability $1-\delta$*. The same holds for* $\mathbf{y}'$

Theorem 6.3.4. *With probability* $1-\delta$*, Algorithm 7 outputs an estimate in* $(1\pm\alpha)D^{(t)} \pm \mathbf{poly}(\log m, \log Z, \frac{1}{\varepsilon}, \frac{1}{\alpha}, \log\frac{1}{\delta})$.

Proof. Follows by the previous lemma, the definition of d, the fact that

$$\|\mathbf{a}\|_0 - \|\mathbf{c}\|_0 \leq \|\mathbf{a}+\mathbf{c}\|_0 \leq \|\mathbf{a}\|_0 + \|\mathbf{c}\|_0$$

, Theorem 6.2.3 and the linearity property of sketches:

$$\mathrm{sk}(\mathbf{a}) \pm \mathrm{sk}(\mathbf{b}) = \mathrm{sk}(\mathbf{a} \pm \mathbf{b}).$$

□

Algorithm 7 is a streaming algorithm since it uses space polylogarithmic in m and takes time polylogarithmic in m per new update.

Since we use the exponential mechanism, our techniques are not efficient in general. We need to sample from a space of 2^S different possible sketches, where S is the maximum bit size of a sketch. When S is polylogarithmic, we need to sample from

Algorithm 7 Pan-private approximation of $D^{(t)}$

INPUT: privacy parameter ε, $0 < p < \alpha/Z < 1$, matrix X computed off-line [a], $\mathrm{sf}(p) = \mathrm{median}\,|X_0|^p$ also computed off-line numerically.

Initialize the r-dimensional sketch vector $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$ to $\mathbf{y}$, by picking $\mathbf{y}$ from μ_ε
for all tuples (i, d_t) **do**
  for all $j = 1$ to r **do**
    $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}{}_j \leftarrow \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}{}_j + d_t * X_{ij}$
  end for
end for

OUTPUT: Draw r-dimensional vector $\mathbf{y}'$ from μ_ε, assign $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}} \leftarrow \mathrm{sk}(\mathbf{a})^{\mathrm{priv}} + \mathbf{y}'$.
return $\tilde{\mathcal{D}} = \mathrm{median}_j \left(\left| \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}{}_j \right|^p \right) \cdot \mathrm{sf}(p)$

[a] See [20] for converting this to the on-line setting using seeded pseudorandom constructions.

a quasipolynomial set of objects. Note that a noise vector is only drawn during the preprocessing and postprocessing phases of the algorithm. While these phases take time 2^s, the time per update is only polylogarithmic.

6.3.2 A general noise-calibrating technique for sketches.

The construction above gives a more general "recipe." Assume that a function f from state vectors to the reals ($f : [-Z, Z]^{\mathcal{U}} \to \mathbb{R}$) with $f(0) = 0$ can be approximated by a sketch. More precisely, the sketch is given by a linear map L and there exists a procedure that given the sketch outputs $\tilde{f}(\mathbf{a}) \in [\gamma_1 f(\mathbf{a}), \gamma_2 f(\mathbf{a})]$. Then we can use the technique above with $d(\mathbf{z}) = \min\{f(\mathbf{c}) : L\mathbf{c} = \mathbf{z}\}$, where the minimum is over valid differences of state vectors, i.e. $\mathbf{c} \in [-2Z, 2Z]^{\mathcal{U}}$. By identical proofs to the ones above, the algorithm is $\varepsilon/2\Delta_q$-pan private and computes an approximation of f in $[\gamma_1 f(\mathbf{a}) - O(S), \gamma_2 f(\mathbf{a}) + O(S)]$, where S is a bound on the bitsize of a sketch, provided that $f(\mathbf{a} + \mathbf{y}) \in f(\mathbf{a}) \pm f(\mathbf{y})$. Note also that $\mathrm{GS}_q = \max_{\mathbf{y}:\|\mathbf{y}\|_0=1} |f(\mathbf{y})|$, where $\mathbf{y}$ has one nonzero component, and that component is bounded in $[-2Z, 2Z]$.

In particular, a variant of Theorem 6.3.4 can be easily achieved for pan-private computation of L_1 and L_2. A key fact that helps us adapt such a technique is the linearity of the sketches, which renders the state of the algorithm to be a linear function of the sketch and the initializing noise vector. The noise vector, itself is picked using a technique that relies on the linearity of these sketches, as outlined above. However, for both L_1 and L_2, this results in an additive factor that is linear in Z, the upper bound on each $|a_i|$. This is because for L_1 or L_2, the sensitivity of the quality function is $\mathrm{GS}_q = 2Z$ (where d minimizes $||c||_1$ and $||c||_2$, respectively, instead of $||c||_0$) and we need to sample the noise vector from $\mu_{\varepsilon'}$, where $\varepsilon' = \varepsilon/2Z$. In turn, this results in linear dependence on Z in the bound on $d(\mathbf{y})$. The linear dependence is inherent in trying to estimate L_1 and L_2, due to their high sensitivity.

This is because the proof of Theorem 6.3.4 will involve a triangle inequality of $||\mathbf{a}+\mathbf{c}||_p \leq ||\mathbf{a}||_p + ||\mathbf{c}||_p$ (for $p = 1$ and 2) and $||\mathbf{c}||_p \leq Z \cdot ||\mathbf{c}||_0$, giving an additive error term, $Z\,\mathbf{poly}(\log m, \log Z, \frac{1}{\varepsilon}, \frac{1}{\alpha}, \log\frac{1}{\delta})$ An additive error term linear in Z seems to be inherent to such a statistic as it is highly sensitive (sensitivity $2Z$), unlike the distinct count statistic.

6.4 Discussion

We focus not only on privacy of data analysis, but also security, formulated as pan-privacy in [34]. We present the first pan-private algorithms on fully dynamic data for distinct count that almost matches the lower bound for this problem. Privacy with security is an important issue, and pan-privacy [34] is an effective and interesting formulation of this problem. A number of extensions are of interest.

Other Security Models. In this work we focus on security against a single unannounced intrusion. A natural extension is to protect against multiple intrusions. If the occurrence of an intrusion is announced before or immediately after the intrusion,

such as in applications where they are legally mandated or are detected by the system, then our results will still hold, with the simple fix to randomize anew after each intrusion. If the intrusions are unannounced, there are extreme cases when differential privacy cannot be ensured even with partly dynamic data [34]. We leave it open to formulate a realistic model of multiple unannounced intrusions and investigate tradeoffs between privacy and accuracy guarantees.

In a dynamic data scenario, it is often desirable to *continuously monitor* some set of statistics in order to detect trends in the data in a timely manner [35]. Our results can also be used to provide *continual event-level pan-privacy* [35]—that is, to provide the ability to monitor the statistics we have considered while ensuring privacy and security. *Event-level pan-privacy* can be defined analogously as in Definition 4 by considering event-level neighbors instead. Two sequences S and S' are said to be event-level neighbors if some "event" (i_k, d_k) in S is replaced by some other event (j, d_k), where $j \neq i_k$ in S'. While the notion of user-level privacy offers protection to a user, event-level privacy seeks to protect an "event,"—i.e., a particular update. Continual event-level pan privacy addresses the problem of providing continual outputs over dynamic data (over time $1 \leq t \leq T$), that are event-level pan-private with respect to one intrusion. As further evidence of the utility of linear sketches (and linear measurements of data, in general), we notice that along with L_p sketches, our noise adding technique of Section 6.3.1 can easily be extended to provide a continual event-level pan-private data structure for computing the number of distinct elements in a dynamic stream by a simple extension of the results in [35]. They propose a counter that within a bounded time period of T provides an accurate estimate of the number of ones in a binary stream, with an additive error term scaled by $O(\log(T)^{2.5})$. A key ingredient in their construction is the linearity of the binary count operation; since operations on sketches are also linear, essentially the same construction (replaced by linear sketch updates) works for

our case.

Other Data Models. We studied the fully dynamic data where items may be inserted or deleted. In such applications, at all times, for all i, $a_i \geq 0$ since one does not delete an item or copy that was not inserted. Still, there are applications with for example, distributed data, which may be modeled by dynamic data where some a_i's may be negative. Our algorithm for distinct count from Section 6.3.1 still works and provides the same guarantees, but we need new algorithms for estimating cropped sum and heavy hitter count in such a data model.

Other Queries. We studied a basic statistical query in this chapter. Many richer queries are of interest, including estimating the entropy of dynamic data, join size estimation for dynamic relations, graph quantities on dynamic graphs, rank and compressibility of dynamic matrices and so on.

We believe that there is a rich theory of pan-private algorithms that needs to be developed, inspired by recent work on differential privacy and streaming algorithms, but already quite distinct as we know from [34] and this work.

7

Information Theoretic Foundations of Differential Privacy

7.1 Introduction

The application of differential privacy to several problems of private data analysis has made it clear that the utility of the data for a specific measurement degrades with the level of privacy. For any analysis to provide a useful notion of utility, it must leak information about the underlying data. The greater this information leakage, the more useful the data is, and vice versa. This chapter attempts to understand the precise information-theoretic conditions that necessitate such a trade-off and examines its relationship to differential privacy. We observe that differentially private mechanisms arise out of minimizing the information leakage (measured using *mutual information*) while trying to maximize utility. The notion of utility is captured by the use of an abstract distortion function dist that measures the distortion between the input and the output of the mechanism. This is a general mechanism, and can be instantiated appropriately depending on the problem domain. The main contribution of this chapter is that differentially private mechanisms can be characterized as probability distributions that achieve this constrained minimization—of minimizing mutual information between the input and output while trying to minimize the distortion (or maximizing the utility). Conversely, through such a characterization, we show that for a fixed value of distortion, the higher the level of differential privacy (ε) a mechanism achieves, the more information it will leak about the input.

We also show how differentially private mechanisms arise out of the application of the *principle of maximum entropy*, first formulated by Jaynes [63]. We see that among all probability distributions that constrain the expected distortion to stay within a given value, the differentially private mechanism corresponds to the distribution that maximizes the conditional entropy of output given the input. This, to our knowledge, is the first attempt at providing an information theoretic foundation for differential privacy. In Section 7.2 we review the appropriate definitions and notions from differential privacy. In Section 7.2.1 we discuss related work. In Sections 7.3 and 7.4 we present our main results.

7.2 Definitions and background

Notation:. Symbols in upper case represent random variables and in lower case, the values the random variables take. Vectors are represented using bold symbols. The measurable space in which random variables take various values is referred to as the alphabet and is represented using script symbols. Both the probability mass function (PMF) and the probability distribution function (PDF) are represented by p. Both $p(\mathbf{x})$ and $p_{\mathbf{X}}(\mathbf{x})$ represent the value of the function p at $\mathbf{x}$, and are used interchangeably throughout the chapter. The only other symbol used for PMF's or PDF's is π to distinguish the case of distribution on the output of an algorithm.

Assume a probability distribution $\mathbb{P}$ on an alphabet $\mathcal{X}$, $\mathcal{X}$ may either be a scalar or vector space. Let $\mathbf{X}_i \in \mathcal{X}$ be a random variable representing the i-th row of a database. Then the random variable representing a database of size n, (whose elements are drawn from $\mathcal{X}$) is $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2 \ldots, \mathbf{X}_n)$. $\mathbf{x}$ represents the value that the random variable $\mathbf{X}$ takes, that is the observed database $\mathbf{x}$. Note that the $\mathbf{X}$'s themselves may be multi-dimensional representing the k attributes of the database.

We reproduce McGregor et al.'s [84] definition of differential privacy in terms of

probability distributions for convenience. This formulation is more useful for us.

Definition 7.2.1. *[84] Let* $\mathbf{x}$ *be a database of length* n*, drawing each of its elements from an alphabet* $\mathcal{X}$*, then an* ε-differentially private mechanism *on* $\mathcal{X}^n$ *is a family of probability distributions* $\{\pi(\boldsymbol{o}|\mathbf{x}) : \mathbf{x} \in \mathcal{X}^n\}$ *on a range* $\mathcal{O}$*, such that for every neighboring* $\mathbf{x}$ *and* $\mathbf{x}'$*, and for every measurable subset* $\boldsymbol{o} \subset \mathcal{O}$*,*

$$\pi(\boldsymbol{o}|\mathbf{x}) \leq \pi(\boldsymbol{o}|\mathbf{x}') \exp(\varepsilon).$$

Notice that the distribution (or equivalently) mechanism is parametrized by the input database $\mathbf{x}$ or $\mathbf{x}'$, whichever is relevant.

We now review the exponential mechanism [87] and restate the related privacy theorem as introduced in Chapter 2, using the negative of a distortion function dist. This mechanism can be said to be parametrized by a "distortion function" $\text{dist}(\mathbf{x}, \boldsymbol{o})$ that maps a pair of an input data set $\mathbf{x}$ (a vector over some arbitrary real-valued domain) and candidate output $\boldsymbol{o}$ (again over an arbitrary range $\mathcal{O}$) to a real valued "distortion score." Lower valued distortions imply good input-output correspondences. It assumes a base measure π on the range $\mathcal{O}$. For a given input $\mathbf{x}$, the mechanism selects an output $\boldsymbol{o}$ with exponential bias in favor of low distorting outputs by sampling from the following *exponential distribution* [87]:

$$\pi^{\varepsilon}(\boldsymbol{o}) \propto \exp(-\varepsilon \, \text{dist}(\mathbf{x}, \boldsymbol{o})) \cdot \pi(\boldsymbol{o}). \tag{7.1}$$

Here the π^{ε} denotes the dependence of the posterior on $\pi(\boldsymbol{o}|\mathbf{x})$, on the parameter ε.

Theorem 7.2.2. *[87] The exponential mechanism, when used to select an output* $\boldsymbol{o} \in \mathcal{O}$*, gives* $2\varepsilon \, \text{GS}_{\text{dist}}$*-differential privacy, where* GS_{dist} *is the global sensitivity of the distortion function* dist.

7.2.1 Related work

Some information-theoretic notions and metrics of data privacy exist in the literature. See [119], [10], [99], for example. Sankar et. al [110] consider the problem of quantifying the privacy risk and utility of a data transformation in an information-theoretic framework. Rebello-Monedero [103] consider the problem in a similar framework and define an information-theoretic privacy measure similar to an earlier defined measure of *t-closeness* [79]. A connection between information theory and differential privacy through *Quantitative flow* has been made by Alvim et al. [4], [2]. Alvim et al. [2] use the information-theoretic notion of min-entropy for the information leakage of the private channel, and show that differential privacy implies a bound on the min-entropy of such a channel. They also show how differential privacy imposes a bound on the utility of a randomized mechanism and under certain conditions propose an optimal randomization mechanism that achieves a certain level of differential privacy. Barthe and Kopf [9] also develop upper bounds for the leakage of every ε-differentially private mechanism. Our work is different from (but related to) theirs in the sense that we do not aim at finding bounds for the information leakage (or risk) of the differentially-private mechanisms. Our aim is to understand the information-theoretic foundations of the framework of differential privacy. Our work is in the spirit of Sankar et al. [110] and Rebello-Monedero et al. [103] but examines how a risk-distortion tradeoff gives rise to differentially-private mechanisms. In previous work [88] we examine the information theoretic connections of differentially-private learning. This was done in the specific context of learning, and the general implications were not clear.

7.3 Differentially private mechanisms in a rate-distortion framework

Assume an input space $\mathcal{X}^n$, and a range $\mathcal{O}$. For any $\mathbf{x} \in \mathcal{X}^n$, and any output $\boldsymbol{o} \in \mathcal{O}$, a distortion function dist is specified. Consider a probability measure $\mathbb{P}$ on $\mathcal{X}$ and a

prior probability π on $\mathcal{O}$.

Given a database $\mathbf{x}$, which is a set of n random independent samples $\hat{\boldsymbol{Z}} = \{\mathbf{X}_1, \ldots \mathbf{X}_n\} \in \mathcal{X}^n$, where each $\mathcal{X}_i$ is drawn i.i.d from $\mathbb{P}$, and an output $\boldsymbol{o}$, the "utility" of $\boldsymbol{o}$ for $\mathbf{x}$, is given by (the negative of) a function dist : $\mathcal{X}^n \times \mathcal{O} \to \mathbb{R}$.

The expected distortion of a mechanism $\pi_{\boldsymbol{O}|\mathbf{X}}(\boldsymbol{o}|\mathbf{x})$ is given by

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{P}^n}\mathbb{E}_{\boldsymbol{o}\sim\pi(\boldsymbol{o}|\mathbf{x})}\,\mathrm{dist}(\mathbf{x}, \boldsymbol{o})$$

Rebollo-Monedero et. al [103] define a privacy risk function to be the mutual information between the revealed and the hidden random variables. Similarly, we define a privacy risk function $\mathcal{R}$ to be the mutual information between the input (the underlying database) and the output of the differentially private mechanism, that is

$$\mathcal{R} = I(\mathbf{X}; \boldsymbol{O}).$$

We know that the mutual information

$$I(\mathbf{X}; \boldsymbol{O}) = H(\boldsymbol{O}) - H(\boldsymbol{O}|\mathbf{X}) = H(\mathbf{X}) - H(\mathbf{X}|\boldsymbol{O}), \tag{7.2}$$

where $H(\mathbf{X})$ represents the entropy of the random variable of $\mathbf{X}$ and $H(\boldsymbol{O}|\mathbf{X})$ the conditional entropy of $\boldsymbol{O}$ given $\mathbf{X}$. So, the mutual information is the reduction in the uncertainty about $\mathbf{X}$ by knowledge of output $\boldsymbol{O}$ or vice versa (See [24] for example). Also we have that

$$\mathcal{R} = I(\mathbf{X}; \boldsymbol{O}) = \mathbb{E}\log\frac{\pi(\boldsymbol{O}|\mathbf{X})p(\mathbf{X})}{\pi(\boldsymbol{O})p(\mathbf{X})} = \mathbb{E}\log\frac{\pi(\boldsymbol{O}|\mathbf{X})}{\pi(\boldsymbol{O})}. \tag{7.3}$$

This is equal to the conditional Kullback-Leibler divergence between the posterior and prior distributions denoted by $D_{KL}\left(\pi(\boldsymbol{O}|\mathbf{X})\|\pi(\boldsymbol{O})\right)$. If the prior and posterior distributions are the same, then the privacy risk is zero, but that also means that the distortion may be arbitrarily high. However, we are interested in minimizing the distortion function associated with the posterior distribution, while minimizing the

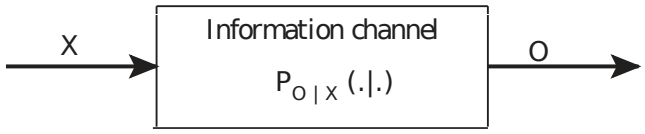


Figure 7.1: Information theoretic model of differentially private channel

privacy risk $\mathcal{R}$. As a result, we are interested in quantifying this risk-distortion trade-off. Notice that until this point, our risk-distortion framework is formulated only in information-theoretic terms. We will see how a differentially private mechanism arises out of this framework.

As in Rebollo-Mondero et al. [103], we are interested in a randomized output, minimizing the privacy risk given a distortion constraint (or vice versa). Unlike their treatment, however, the potential outputs are more general than perturbations of the input database elements to capture differentially private mechanims (both interactive and noninteractive). The privacy risk-distortion function is defined analogously (as in Rebollo-Mondero [103]):

$$\mathcal{R}(D) = \inf_{\pi_{\boldsymbol{O}|\mathbf{X}} : \mathbb{E}_{\mathbf{x},\boldsymbol{o}} \operatorname{dist}(\mathbf{x},\boldsymbol{o}) \leq D} I(\mathbf{X};\boldsymbol{O}) \tag{7.4}$$

7.3.1 An information channel

In view of this we present an information-theoretic view of differentially private information channels. Given a random sample $\mathbf{X}$ of cardinality n from a probability distribution $\mathbb{P}$, the mechanism outputs a $\boldsymbol{o}$ from $\mathcal{O}$. This process sets up an information channel, whose input is a $\mathbf{X}$ and output is $\boldsymbol{O}$. Figure 7.1 shows the channel. $p_{\boldsymbol{O}|\mathbf{X}}(\boldsymbol{o}|\mathbf{x})$ represents the probability that the channel will output $\boldsymbol{o}$ when the secret is $\mathbf{x}$, and from above we know this is specified by the posterior, π^{ε}. Hence, the problem of differential privacy can be looked at as designing an information channel that minimizes the (regularized) mutual information between $\boldsymbol{O}$ and $\mathbf{X}$, subject to constraints of minimizing the distortion.

7.3.2 Connection to the rate-distortion framework

Rebollo-Mondero et. al relate the risk-distortion function formulated in Equation 7.4 [103] to the well-known *rate-distortion* problem in information theory first formulated by Shannon. (See [24], for example). Shannon's rate-distortion theory is applied in the context of lossy compression of data. The objective is to construct a compact representation (a code) of the underlying signal (or data), such that the average distortion of the signal reconstructed from this compact representation is low. Rate-distortion theory determines the level of expected distortion D, given the desired information rate $\mathcal{R}$ of the code or vice-versa using the rate-distortion function $\mathcal{R}(D)$ similar to that in Equation 7.4 where $\mathcal{R}$ is the information rate of the code, when applied to the compression problem. So, the rate-distortion function is defined as the infinimum of the rates of codes whose distortion is bounded by D.

Using this connection, Rebello-Mondero et al. prove that:

Theorem 7.3.1. *[103] The privacy risk-distortion function is a convex and non-increasing function of D.*

The problem is to minimize the privacy risk, defined thus, under the expected distortion constraint. As a function of the probability density, $\pi_{O|\mathbf{X}}(\boldsymbol{o}|\mathbf{x})$, the problem is also convex. We can also use Lagrangian multipliers to write Equation 7.4 in an equivalent unconstrained form. We have the functional:

$$\mathcal{F}[\pi(\boldsymbol{o}|\mathbf{x})] = \frac{1}{\varepsilon} I(\mathbf{X};\boldsymbol{O}) + \mathbb{E}\,\mathrm{dist}(\mathbf{X},\boldsymbol{O}) \tag{7.5}$$

for a positive ε. Functional $\mathcal{F}$ needs to be minimized among all normalized $\pi(\boldsymbol{o}|\mathbf{x})$. We can find the distribution that minimizes this function, by using standard optimization techniques. Standard arithmetic manipulation, leads Tishby et al. [117] to prove the following theorem:

Theorem 7.3.2. *[117] The solution of the variational problem,*

$$\frac{\partial \mathcal{F}}{\partial \pi(\boldsymbol{o}|\mathbf{x})} = 0,$$

for normalized distributions $\pi(\boldsymbol{o}|\mathbf{x})$, is given by the exponential form

$$\pi^{\varepsilon}(\boldsymbol{o}|\mathbf{x}) = \frac{\exp(-\varepsilon \operatorname{dist}(\mathbf{x}, \boldsymbol{o}))}{Z(\mathbf{x})}\pi(\boldsymbol{o}). \tag{7.6}$$

where $Z(\mathbf{x}, \varepsilon)$ is a normalization (partition) function. Moreover, the Lagrange multiplier ε is determined by the value of the expected distortion, D, is positive and satisfies

$$\frac{\partial \mathcal{R}}{\partial D} = -\varepsilon$$

Among all the conditional distributions, the one that optimizes the functional in Equation 7.5 is π^{ε} (in Equation 8.2 above). This is our main result, that the distribution that minimizes the privacy risk, given a distortion constraint is a differentially private distribution. From examining Equation 7.1 and Theorem 7.2.2 we have

Theorem 7.3.3. *The distribution that minimizes Equation 7.4 defines a $2\varepsilon\,\mathrm{GS}_{\mathrm{dist}}$-differentially private mechanism.*

Figure 7.3.2 illustrates the tradeoff. It plots the unconstrained Lagrangian function $L(D, \mathcal{R}) = D + \frac{1}{\varepsilon}\mathcal{R}$, which because of the convexity of the risk-distortion function is also convex. For a given privacy parameter ε, we consider lines of slope -ε. We see that these lines intersect the curve at various points, these points represent the risk-distortion tradeoffs for those values. As we should expect, a high privacy-risk implies a low distortion and vice versa. We see that for a given value of slope -ε, the line that is tangent to the curve represents the optimal tradeoff point between the risk and the distortion. The value of the function $L(D, \mathcal{R})$ on these lines is a constant, which implies that in some way the level of privacy imposes a value on the function L, since such a line can only intersect the curve in at most two places.

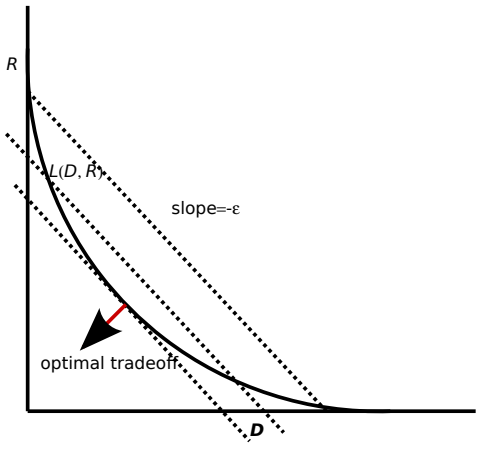


Figure 7.2: Risk-distortion curve

As ε the privacy parameter decreases, for the same level of distortion, the slope of the tangent to the curve decreases, (since it is -ε), leading to lower mutual information $\mathcal{R}$ (the y-axis intercept).

It is important to note that the distributions that minimize Equation 7.4 are differentially private, since they can be expressed as exponential mechanisms. Exponential distributions that correspond to points on the $R - D$ curve correspond to different values of ε. However, the converse is not, in general true. That is, all differentially private mechanisms do not lie on the $R - D$ curve. For mechanisms to lie on the $R - D$ curve, the prior distribution has to satisfy certain conditions as specified by Tishby et al. [117]; it would be interesting to examine what this means in the context of differential privacy.

7.4 Differential privacy arising out of the Maximum Entropy principle or Minimum Discrimination Information principle

The *principle of maximum entropy* was proposed by Jaynes [63]. Suppose, a random variable $\mathbf{X}$ takes a discrete set of values $\mathbf{x}_i$ with probabilities specified by $p_{\mathbf{X}}(\mathbf{x}_i)$, and we know of constraints on the distribution $p_{\mathbf{X}}$, in the form of expectations of some

functions of these random variables. Then the principle of maximum entropy states that of all distributions $p_{\mathbf{X}}$ that satisfy the constraints, one should choose the one with the largest entropy $H(\mathbf{X}) = -\sum_i p(\mathbf{x}_i) \log(p(\mathbf{x}_i))$.

In the case of a continuous random variable, we apply the principle of minimum discrimination information [63]. It states that given a prior p on $\mathbf{X}$, a new distribution q should be chosen so that it as hard as possible to distinguish it from the prior distribution p, that is the new data should produce as small a gain in information as possible given by $D_{KL}(q\|p)$.

We show that the application of the principle of Maximum Entropy to the distribution $\pi(\boldsymbol{o}|\mathbf{x})$ gives rise to a differentially private mechanism.

When trying to find a distribution $\pi_{O|\mathbf{X}}(\boldsymbol{o}|\mathbf{x})$, we utilize the Maximum Entropy Principle. Among all distributions $p(\boldsymbol{o}|\mathbf{x})$, we choose the one that maximizes the entropy $H(\boldsymbol{O}|\mathbf{X})$ subject to satisfying the constraint that the expected distortion function $\text{dist}(\boldsymbol{o}, \mathbf{x})$ is bounded by a quantity D. So we have,

$$\text{maximize } H(\boldsymbol{O}|\mathbf{X})$$

$$\text{subject to } \sum \text{dist}(\mathbf{x}, \boldsymbol{o}) p(\boldsymbol{o}|\mathbf{x}) p(\mathbf{x}) \leq D.$$

From Equation 7.2 we observe that minimizing the mutual information as in Equation 7.4 is equivalent to maximizing the entropy $H(\boldsymbol{O}|\mathbf{X})$ for a fixed prior and hence $H(\boldsymbol{O})$ is fixed.

Shannon introduced the concept of *equivocation* as the conditional entropy of a private message given the observable [111]. Sankar et al. [110] use equivocation as a measure of privacy of their data transformation. Their aim is also to maximize the average equivocation of the underlying secret sample given the observables.

Since $I(\mathbf{X};\boldsymbol{O}) = H(\mathbf{X}|\boldsymbol{O}) - H(\mathbf{X})$, minimizing $I(\mathbf{X};\boldsymbol{O})$ is also equivalent to maximizing the conditional entropy $H(\mathbf{X}|\boldsymbol{O})$, subject to constraints on the expected distortion. Therefore, the exponential distribution $\pi^{\varepsilon}(\boldsymbol{o}|\mathbf{x})$ as defined in Equation 8.2 maximizes the conditional uncertainty about the underlying sample given a constraint on the distortion function.

Now consider the worst case which differential privacy protects against, that is given knowledge of the entire database except for one row i, represented as $\mathbf{X}_{-i}$, if we look at the problem of maximizing the uncertainty of the random variable $\mathbf{X}_i$, we have

$$\text{maximize } H(\mathbf{X}_i|\boldsymbol{O},\mathbf{X}_{-i})$$

$$\text{subject to } \sum \text{dist}(x_i,\mathbf{x}_{-i},\boldsymbol{o})p(x_i|\mathbf{x}_{-i},\boldsymbol{o})p(\mathbf{x}_{-i},\boldsymbol{o}) \leq D$$

Again this is equivalent to minimizing the mutual information $I(\mathbf{X},\boldsymbol{O})$ when $\mathbf{X}_{-i}$ and $\boldsymbol{O}$ are given.

A note on incorporating auxiliary information:.

Usually, differential privacy provides guarantees on the inference, irrespective of any side or auxilliary information. This can be easily incorporated in our framework like Sankar et. al [110] by making all the distributions above conditional on the side information.

7.5 Conclusion and future work

We presented an information-theoretic foundation for differential privacy, which to our knowledge is the first such attempt. We formulated differential privacy within the broader frameworks of rate-distortion and the maximum entropy principle in information theory. There are several directions for future work.

One, we can try to apply the risk-distortion framework to examine the generation of private synthetic data when the underlying data generating distribution $p_{\mathbf{X}}(\mathbf{x})$ is

known. Additionally, one could try to derive bounds on the mutual information in such cases. Second, we can examine the deployment of this framework to problems where the distortion function dist is specified. Another direction is to examine the notion of compressive privacy [80] in this rate-distortion framework and derive bounds for the rate or mutual information of such a mechanism.

8

Differentially Private Learning and PAC-Bayesian Bounds

8.1 Introduction

In this chapter we examine the most general problem of differentially private learning, and establish a connection to PAC-Bayesian bounds [14, 83, 126]. To our knowledge, this is the first such connection. We discover that the so-called Gibbs estimator, that arises when minimizing PAC-Bayesian bounds, corresponds to the exponential mechanism [87], which is the most general formulation of a differentially-private mechanism. This PAC-Bayesian connection to differentially private learning also helps us place the problem in an information theoretic framework similar to that in Chapter 7 with the distortion function specified by the *empiricial risk of* a predictor. A connection between information theory and differential privacy through *Quantitative flow* has been made by Alvim et al. [3]. They propose upper and lower bounds on the mutual information between the input and the differentially private output and the connections this has to the utility of the algorithm. Our connection to information theory, on the other hand demonstrates that differentially private learning is really a problem of minimizing (regularized) mutual information between the data (the sample) and the predictor, under the constraints of minimizing expected risk of the algorithm, and through such a treatment relate this problem to PAC-Bayesian learning.

In Section 8.1.1 we review the general problem of differentially private learning.

In Section 8.2 we introduce the relevant PAC-Bayesian bounds and the Gibbs estimator and establish its connection to differentially private learning. We also use PAC-Bayesian bounds to interpret how differentially private predictors arise out of balancing the requirements of minimizing the mutual information between the predictor and the underlying sample and minimizing the expected risk, with the balance "tilt" being determined by the privacy level.

8.1.1 Differentially private learning

We use the general framework of statistical prediction/learning, in which there is an input space $\mathcal{X}$, an (optional) output space $\mathcal{Y}$, and a space of predictors Θ. For any $X \in \mathcal{X}$, $Y \in \mathcal{Y}$, and any predictor $\theta \in \Theta$, a loss quantified by a loss function $\ell_\theta(X, Y) = \ell_\theta(Z)$ is incurred, where $Z = (X, Y)$, $\in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Consider a probability measure $\mathbb{P}$ on $\mathcal{Z}$.

The true risk of a predictor θ is given by:

$$R = \mathbb{E}_Z \ell_\theta(Z)$$

Given a set of n random independent samples $\hat{\boldsymbol{Z}} = \{(X_i, Y_i), \ldots (X_n, Y_n)\} \in \mathcal{Z}^n$, each one i.i.d, drawn from $\mathbb{P}$, and a predictor θ, the *e*mpirical risk of θ on $\hat{\boldsymbol{Z}}$, is given by:

$$\hat{R}_{\hat{\boldsymbol{Z}}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell_\theta(X_i, Y_i)$$

Given a set of random samples $\hat{\boldsymbol{Z}} = \{\hat{Z}_1 \ldots \hat{Z}_n\}$ from $\mathbb{P}$, our goal is to find a parameter, $\hat{\theta}(\hat{\boldsymbol{Z}})$, such that the true expected risk $L(\hat{\theta}) = \mathbb{E}_Z \ell_{\hat{\theta}(\hat{\boldsymbol{Z}})}(Z)$ is small, where $\mathbb{E}_Z$ is the expectation with respect to $\mathbb{P}$ and Z is independent of $\hat{\boldsymbol{Z}}$. The predictor may be deterministic or randomized, which is equivalent to specifying *a sample-dependent posterior* probability distribution on Θ. The use of a posterior signifies the fact that the probability distribution on Θ was arrived at after processing the sample $\hat{\boldsymbol{Z}}$.

The goal of differentially private learning is to learn a predictor $\hat{\theta}(\hat{\boldsymbol{Z}})$ from the data $\hat{\boldsymbol{Z}}$, that respects the definition of differential privacy. For this purpose any two sample sets, $\hat{\boldsymbol{Z}}$ and $\hat{\boldsymbol{Z}}'$ are neighbors if they differ in exactly one of the samples, that is for some $i \in [n]$, $(X_i, Y_i) \neq (X_i', Y_i')$, and for every other $j \in [n], j \neq i$, $(X_j, Y_j) = (X_j', Y_j')$. As introduced in Chapter 2, a mechanism M on $\hat{\boldsymbol{Z}}$ is a family of probability distributions $\hat{\pi}_{\lambda,\hat{\boldsymbol{Z}}} : \hat{\boldsymbol{Z}} \in \mathcal{Z}n$ on Θ. The mechanism is λ-differentially private if for every neighboring $\hat{\boldsymbol{Z}}$ and $\hat{\boldsymbol{Z}}'$ and for every measurable subset $S \subset \Theta$, we have

$$\hat{\pi}_{\lambda,\hat{\boldsymbol{Z}}}(S) \leq \exp(\lambda)\hat{\pi}_{\lambda,\hat{\boldsymbol{Z}}'}(S)$$

8.2 PAC-Bayesian bounds and differentially private learning

Since the true risk is defined with respect to the unknown distribution $\mathbb{P}$, one needs to specify which function of the sample (or training) set, $\hat{\boldsymbol{Z}}$, needs to be optimized to find a suitable predictor. The so-called *generalization bounds* provide an upper bound on the true risk of a predictor θ in terms of the empirical risk of θ on the training data $\hat{\boldsymbol{Z}}$ and some function of a measure of the complexity of the predictors, that may be output by the learning algorithm, and a confidence term $\delta \in [0, 1]$. Given such a (hopefully tight) upper bound which can be computed from the performance of a predictor on the training set, one can compute the predictor that minimizes it. For example, Chaudhuri et al. [18] use this methodology to compute a differentially private predictor in the case of machine learning tasks such as logistic regression, support vector machines etc.

In bounds such as the VC-Dimension bounds, (see for example [7]) the data-dependencies only come from the empirical risk of the predictor on the training set. This data-independence constrains the predictor to come from some restricted class of finite complexity. This restriction is data-independent, it does not look at the training set $\hat{\boldsymbol{Z}}$ and by virtue of this restriction allows the difference between the empirical risk and the true risk to be bounded uniformly for all predictors in this class. As a result such

bounds are often loose. For data-dependent bounds, on the other hand, the difference between the true risk and the empirical risk depends on the training set $\hat{\boldsymbol{Z}}$. In data-dependent bounds such as PAC-Bayesian bounds possible, prior knowledge about the unknown data distribution is incorporated into a model that places a prior distribution on the space of possible predictors, which is updated to a posterior distribution after observing the data.

We can already see the parallels between PAC-Bayesian bounds and differntially-private learning. Given a $\hat{\boldsymbol{Z}}$, and a *prior* distribution π on Θ, the goal of differentially private statistical prediction is to find a randomized estimator specified by a posterior probability measure $d\hat{\pi}_{\hat{\boldsymbol{Z}}}(\theta)$ on Θ, that fulfills the privacy property. As in PAC-Bayesian bounds, the posterior on Θ is learned after processing the training set $\hat{\boldsymbol{Z}}$, even though the goals are different. PAC-Bayesian learning starts out with a prior on Θ which after getting information from $\hat{\boldsymbol{Z}}$ is updated to the posterior measure $d\hat{\pi}_{\hat{\boldsymbol{Z}}}(\theta)$, the goal being to choose a "good" randomized predictor. The goal of differential privacy is to arrive at a "good" randomized predictor that also satisfies the property specified in Definition 2.2.1.

Catoni [14] quantifies these bounds in the following manner: Let $D_{KL}\left(\pi \| \hat{\pi}\right)$ represent the Kullback-Leibler divergence between two distributions.

Theorem 8.2.1. *[14] For any posterior $\hat{\pi}$ on Θ, any prior π on Θ, any sample set $\hat{\boldsymbol{Z}}$, and for any positive λ, with probability at least $1-\delta$ over the choice of $\hat{\boldsymbol{Z}}$, we have:*

$$\begin{aligned}
\mathbb{E}_{\theta\sim\hat{\pi}} R((\theta) \leq & \quad \frac{1-\exp\left\{-\frac{\lambda \mathbb{E}_{\theta\sim\hat{\pi}}\hat{R}_{\hat{\boldsymbol{Z}}}(f)}{n} - \frac{D_{KL}\left(\hat{\pi}\|\pi\right)-\log\delta}{n}\right\}}{1-\exp\left(\frac{-\lambda}{n}\right)} \\
\leq & \quad \frac{\lambda}{n\left[1-\exp(\frac{-\lambda}{n})\right]}\left[\mathbb{E}_{\theta\sim\hat{\pi}}\hat{R}_{\hat{\boldsymbol{Z}}}(f) + \frac{D_{KL}(\hat{\pi}\|\pi)-\log(\delta)}{\lambda}\right]
\end{aligned}$$

In expectation we have:

$$\mathbb{E}_{\hat{\boldsymbol{Z}}}\mathbb{E}_{\theta\sim\hat{\pi}}R(()\theta) \leq \frac{1-\exp\left(-n^{-1}\mathbb{E}_{\hat{\boldsymbol{Z}}}\left[\lambda\cdot\mathbb{E}_{\theta\sim\hat{\pi}}\hat{R}_{\hat{\boldsymbol{Z}}}(f)+D_{KL}(\hat{\pi}\|\pi)\right]\right)}{1-\exp(\frac{-\lambda}{n})}$$
$$\leq \frac{\lambda}{n\left[1-\exp(\frac{-\lambda}{n})\right]}\mathbb{E}_{\hat{\boldsymbol{Z}}}\left[\mathbb{E}_{\theta\sim\hat{\pi}}\hat{R}_{\hat{\boldsymbol{Z}}}(f)+\frac{D_{KL}(\hat{\pi}\|\pi)}{\lambda}\right]$$
$$= \frac{\lambda}{n\left[1-\exp(\frac{-\lambda}{n})\right]}\left\{\mathbb{E}_{\hat{\boldsymbol{Z}}}\left[\mathbb{E}_{\theta\sim\hat{\pi}}\hat{R}_{\hat{\boldsymbol{Z}}}(f)\right]+\frac{\mathbb{E}_{\hat{\boldsymbol{Z}}}[D_{KL}(\hat{\pi}\|\pi)]}{\lambda}\right\} \tag{8.1}$$

Notice that, the bounds hold for any π and $\hat{\pi}$. Usually, these bounds are optmized to yield an "optimal" posterior. Also, as noticed by Catoni, $1 \leq \frac{\lambda}{n\left(1-\exp(\frac{-\lambda}{n})\right)} \leq \left[1-\frac{\lambda}{2n}\right]^{-1}$ and hence this factor is close to 1 when λ is much smaller than n (which will always the case for us).

If the prior π and λ are considered to be fixed, then the goal is to come up with a posterior $\hat{\pi}$ that minimizes this bound. Similar bounds were proved by Zhang [126].

We have the following lemma from Catoni [14] and Zhang [126]:

Lemma 8.2.2. *[14, 126] Given a $\lambda > 0$ and a prior distribution π on Θ, the posterior $\hat{\pi}$ that minimizes the unbiased empirical upper bound given by Theorem 8.2.1 is the Gibbs posterior, denoted as $\hat{\pi}_\lambda$:*

$$d\hat{\pi}_\lambda = \frac{\exp(-\lambda\hat{R}_{\hat{\boldsymbol{Z}}}(f))}{\mathbb{E}_{\theta\sim\pi}\exp(-\lambda\hat{R}_{\hat{\boldsymbol{Z}}}(f))}d\pi \tag{8.2}$$

We observe that the Gibbs estimator of Lemma 8.2.2 is differentially private, provided the empirical risk function has a bounded global sensitivity. Applying McSherry and Talwar's [87] Theorem 2.3.4, we have the following:

Theorem 8.2.3. *Given a sample $\hat{\boldsymbol{Z}}$, the mechanism given by the posterior $\hat{\pi}$ is $2\lambda\,\mathrm{GS}_{\hat{R}_{\hat{\boldsymbol{Z}}}(f)}$, differentially private, where $\mathrm{GS}_{\hat{R}_{\hat{\boldsymbol{Z}}}(f)}$ is the global sensitivity of the empirical risk.*

The fact that the Gibbs estimator is differentially private, establishes a connection between information theory and differential privacy. Catoni [14] remarks that in Equation 8.1, the quantity $\mathbb{E}_{\hat{\boldsymbol{Z}}}\left[D_{KL}\left(\hat{\pi}\|\pi\right)\right]$ is equal to

$$\mathbb{E}_{\hat{\boldsymbol{Z}}}\{D_{KL}\left(\hat{\pi}\|\mathbb{E}_{\hat{\boldsymbol{Z}}}\hat{\pi}\right)\} + D_{KL}\left(\mathbb{E}_{\hat{\boldsymbol{Z}}}\hat{\pi}\|\pi\right).$$

The quantity $\mathbb{E}_{\hat{\boldsymbol{Z}}}\{D_{KL}\left(\hat{\pi}\|\mathbb{E}_{\hat{\boldsymbol{Z}}}\hat{\pi}\right)\}$ is actually the *mutual information* $I(\hat{\boldsymbol{Z}},\theta)$ between the sample $\hat{\boldsymbol{Z}}$ drawn from $\mathbb{P}$ and the parameter θ drawn from $\hat{\pi}$ under the joint probability distribution $\mathbb{P}\hat{\pi}$. The mutual information between $\hat{\boldsymbol{Z}}$ and θ can be interpreted as the average amount of information contained in the predictor θ about the sample $\hat{\boldsymbol{Z}}$. Intuitively, we know that the problem of privacy is a tradeoff between minimizing this mutual information and learning a (possibly) randomized predictor from the data in order to make meaningful predictions.

As noticed by Catoni [14], from this equation we see that the expected KL-divergence between $\hat{\pi}$ and π, for any $\hat{\pi}$, is equal to the mutual information between the sample and the parameter when the prior $\pi = \mathbb{E}_{\hat{\boldsymbol{Z}}}\hat{\pi}$. Hence for a given posterior $\hat{\pi}$, the optimal choice for π, is $\pi_{OPT} = \mathbb{E}_{\hat{\boldsymbol{Z}}}\hat{\pi}$. However, since finding the bound-optimal $\mathbb{E}_{\hat{\boldsymbol{Z}}}\hat{\pi}$ is not better known than $\mathbb{P}$, there is an additional additive factor of $D_{KL}\left(\mathbb{E}_{\hat{\boldsymbol{Z}}}\hat{\pi}\|\pi\right)$. To illustrate the relationship of differential privacy with mutual information, we assume that we can find the "optimal prior" in this sense. Conceptually, the argument holds even if an "'optimal" prior is not assumed, but we make the assumption for clarity of exposition. Then the Gibbs estimator minimizes the expected empirical risk and the regularized mutual information between the sample and the predictor:

$$\hat{\pi}_\lambda = \arg\inf_{\hat{\pi}} \left[\mathbb{E}_{\hat{\boldsymbol{Z}}} \left[\mathbb{E}_{\theta\sim\hat{\pi}} \hat{R}_{\hat{\boldsymbol{Z}}}(f) \right] + \frac{1}{\lambda} I(\hat{\boldsymbol{Z}},\theta) \right].$$

This relationship quantifies the tradeoff that was intuitively understood before. The privacy parameter λ weighs the effect of the mutual information on this tradeoff. For a small λ, which corresponds to higher privacy, the mutual information penalizes the bound more than for a larger λ, biasing it towards solutions that have a smaller mutual information between the parameter and the sample. This tendency towards picking distributions that induce smaller $I(\hat{\boldsymbol{Z}},\theta)$, needs to be traded with picking a $\hat{\pi}$ that also minimizes the expected emprical risk. For a larger λ, the Gibbs estimator is not considerably biased towards solutions having smaller mutual information. We have:

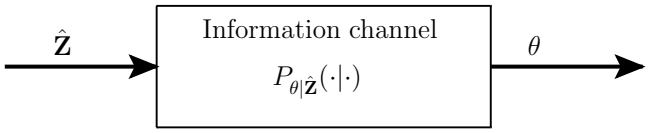


Figure 8.1: Information theoretic model of differentially private learning

Theorem 8.2.4. *The minimization of regularized mutual information (or entropy), regularized by the privacy parameter, under constraints of minimizing expected empirical risk gives rise to a differentially private predictor (the Gibbs estimator).*

In view of this we present an information-theoretic view of differentially private learning. Given a random sample $\hat{\boldsymbol{Z}}$ of cardinality n from a probability distribution $\mathbb{P}$, we come up with a predictor θ from Θ. This process sets up an information channel, whose input is a $\hat{\boldsymbol{Z}}$ and output is θ. The sample $\hat{\boldsymbol{Z}}$ is the secret and the predictor θ the ouput of the channel, which should be differentially private. Figure 8.1 shows the channel. $p_{\theta|\hat{\boldsymbol{Z}}}(\theta|\hat{\boldsymbol{Z}})$ represents the probability that the channel will output θ when the secret is $\hat{\boldsymbol{Z}}$, and from above we know this is specified by the Gibbs posterior, $\hat{\pi}_\lambda$. Hence, the problem of differentially-private learning can be looked at as designing an information channel that minimizes the (regularized) mutual information between $\hat{\boldsymbol{Z}}$ and θ, subject to constraints of minimizing the expected empirical risk.

8.3 Conclusion and future work

We have established a connection between PAC-Bayesian bounds and differentially-private learning that helps us interpret differentially-private learning in an information theoretic framework. This will hopefully help us both apply PAC-Bayes bounds to investigate more problems in differentially-private learning as well help us understand the connections between differentially private learning and information theory

in a deeper manner. This connection could also help us in coming up with generalization bounds for various machine learning algorithms. As an example the risk bounds in Chapter 4 for differentially private Gaussian regression can also be obtained by the use of PAC-Bayesian bounds. We could also examine the use of upper and lower bounds on the mutual information between the sample and the predictor and their implication on the utility of differentially-private learning algorithms similar to Alvim et al. [3], and compare these bounds.

9

Conclusion

In this dissertation we carried out a study of the privacy-utility tradeoff in various contexts of sensitive data analysis. We chose differential privacy as our metric of privacy, and studied a range of sensitive data analysis tasks in this model. We proposed differentially private algorithms for problems over a range of settings—both interactive and noninteractive .

These tasks, namely, synthetic graph generation, human mobility modeling, statistical prediction, and computation of online statistics are deployed in a range of real-world problems. They have important real-world applications and there is a dire need for providing provably private solutions to these problems. Proposing efficient algorithms for these tasks within a strong model of privacy, is the first step to bridging the gap between the practice and theory of privacy. We see our work as a step in this direction.

We also studied the privacy-utility tradeoff in the context of information theory and showed how a higher level of differential privacy constrains outputs of private mechanisms to reveal less information about the inputs and vice-versa. We also studied the equivalence between PAC-Bayesian bounds and differentially private learning.

Hence, we contributed to both an algorithmic and conceptual study of differential privacy.

Bibliography

[1] Standard for privacy of individually identifiable health information. Federal Register, August 2002. URL `http://www.hhs.gov/ocr/privacy/hipaa`. 29

[2] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: On the trade-off between utility and information leakage. In *Formal Aspects in Security and Trust '11*, pages 39–54. 2011. 27, 124

[3] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative information flow and applications to differential privacy. In *International School on Foundations of Security Analysis and Design '10*, pages 211–230. 2010. 133, 140

[4] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. On the relation between differential privacy and quantitative information flow. In *International Colloquium on Automata, Languages and Programming (2) '11*, pages 60–76. 2011. 27, 124

[5] Jillian Anable, Christian Brand, Martino Tran, and Nick Eyre. Modelling transport energy demand: A socio-technical approach. *Energy Policy*, 41:125 – 138, 2012. ISSN 0301-4215. Modeling Transport (Energy) Demand and Policies. 66

[6] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. CoRR '12 abs/1212.1984, 2012. 26, 59

[7] Martin Anthony and Peter L. Bartlett. *Neural Network Learning - Theoretical Foundations*. Cambridge University Press, 2002. ISBN 978-0-521-57353-5. 7, 88, 135

[8] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *World Wide Web '07*, pages 181–190. 2007. 30, 32

[9] Gilles Barthe and Boris Kopf. Information-theoretic bounds for differentially private mechanisms. In *Computer Security Foundations Symposium '11*, pages 191–204. 2011. 27, 124

[10] Michele Bezzi. An information theoretic approach for privacy metrics. *Transactions on Data Privacy*, 3(3):199–215, December 2010. ISSN 1888-5063. 124

[11] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *Principle Of Database Systems '05*, pages 128–138. 2005. 25

[12] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *Symposium on Theory of Computing '08*, pages 609–618. 2008. 25

[13] Elit Van Buskirk. How the netflix prize was won. `http://www.wired.com/business/2009/09/how-the-netflix-prize-was-won/`, 2009. 2

[14] Olivier Catoni. Pac-bayesian supervised classification: The thermodynamics of statistical learning. *Monograph series of the Institute of Mathematical Statistics*, 2007. 133, 136, 137, 138

[15] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 14(3):26:1–26:24, November 2011. 27

[16] T.H.Hubert Chan, Mingfei Li, Elaine Shi, and Wenchang Xu. Differentially private continual monitoring of heavy hitters from distributed streams. In *Privacy Enhancing Technologies '12*, pages 140–159. 2012. 27

[17] Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. *Journal of Machine Learning Research - Proceedings Track*, 19:155–186, 2011. 85

[18] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011. 26, 84, 100, 135

[19] Rui Chen, Benjamin C. M. Fung, Bipin C. Desai, and Nériah M. Sossou. Differentially private transit data publication: a case study on the Montreal transportation system. In *Knowledge Discovery and Datamining '12*, pages 213–221. 2012. 26, 58

[20] Graham Cormode, Mayur Datar, P. Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *ACM Transactions on Knowledge and Data Engineering*, 15(3):529–540, 2003. 107, 112, 117

[21] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. Differentially private spatial decompositions. In *International Conference on Data Engineering '12*, pages 20–31. 2012. 19, 26, 81

[22] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Thanh T. L. Tran. Differentially private summaries for sparse data. In *International Conference on Database Theory '12*, pages 299–311. 2012. 66, 67

[23] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. Anonymizing bipartite graph data using safe groupings. In *Very Large Data Bases '08*, pages 833–844. 2008. 33

[24] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN 0471241954. 125, 127

[25] Tore Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15(429-444):2–1, 1977. 10

[26] Yves-Alexandre de Montjoye, Cesar A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, Mar 2013. 55, 58

[27] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Principles Of Database Systems '03*, pages 202–210. 2003. 24

[28] Pat Doyle, Julia I. Lane, Jules J.M. Theeuwes, and Laura M. (eds.) Zayatz. *Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies*. Elsevier, 2001. ISBN 9780444507617. 10

[29] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages and Programming (2) '06*, pages 1–12. 2006. 25

[30] Cynthia Dwork. Differential privacy in new settings. In *Symposium On Discrete Algorithms '10*. 2010. 106

[31] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. In *Eurocrypt '06*, pages 486–503. 2006. 25

[32] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Symposium on Theory of Computing '09*, pages 371–380. 2009. 26, 45, 85, 87

[33] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference '06*, pages 265–284. 2006. 3, 14, 15, 16, 18, 19, 20, 25, 43, 62, 109, 110, 114

[34] Cynthia Dwork, Moni Naor, Toni Pitassi, Guy Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Innovations in Computer Science '10*, pages 66–80. 2010. 6, 23, 24, 27, 106, 107, 108, 110, 111, 118, 119, 120

[35] Cynthia Dwork, Toni Pitassi, Moni Naor, and Guy Rothblum. Differential privacy under continual observation. In *Symposium on Theory Of Computing '10*, pages 715–724. 2010. 27, 119

[36] Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. In NCHS/CDC Data Confidentiality Workshop, 2008. 25

[37] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *Advances in Cryptology '08*, pages 469–480. 2008. 13

[38] Mark Elliot, Anco Hundepool, Eric Schulte Nordholt, Jean-Louis Tambay, and Thomas Wende. Glossary on statistical disclosure control. `http://neon.vb.cbs.nl/casc/glossary.htm`, May 2009. 11

[39] A. Frank and A. Asuncion. UCI machine learning repository. 2010. URL `http://archive.ics.uci.edu/ml`. 84, 100

[40] Johannes Gehrke, Michael Hay, Edward Lui, and Rafael Pass. Crowd-blending privacy. In *International Conference on Cryptology '12*, pages 479–496. 2012. 28

[41] Johannes Gehrke, Edward Lui, and Rafael Pass. Towards privacy for social networks: A zero-knowledge based definition of privacy. In *Theory of Cryptography Conference '11*, volume 6597, pages 432–449. 2011. 28

[42] Fabien Girardin, Francesco Calabrese, Fillipo Dal Fiorre, Assaf Biderman, Carlo Ratti, and Josep Blat. Uncovering the presence and movements of tourists from user-generated content. In *International Forum on Tourism Statistics*. 2008. 58

[43] Fabien Girardin, F. Dal Fiore, Josep Blat, and Carlo Ratti. Understanding of tourist dynamics from explicitly disclosed location information. *Symposium on LBS and Telecartography*, 2007. 58

[44] Fabien Girardin, A. Vaccari, A. Gerber, A. Biderman, and C. Ratti. Towards estimating the presence of visitors from the aggregate mobile phone network activity they generate. In *Computers in Urban Planning and Urban Management*, pages 52–61. 2009. 58

[45] David Gleich. Kronecker moment based estimation code. https://dgleich.com/gitweb/?p=kgmoments;a=summary, 2011. 47

[46] David F. Gleich and Art B. Owen. Moment based estimation of stochastic Kronecker graph parameters. *Internet Mathematics*, 8(3):232–256, 2012. 31, 32, 33, 34, 38, 39, 41, 46, 47, 48, 49

[47] Philippe Golle and Kurt Partridge. On the anonymity of home/work location pairs. In *Pervasive '09*, pages 390–397. 2009. 58

[48] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008. 58, 59

[49] Dov Greenbaum, Andrea Sboner, Xinmeng Jasmine Mu, and Mark Gerstein. Genomics and privacy: Implications of the new reality of closed data for the field. *PLoS Computational Biology*, 7(12), 2011. 1

[50] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001. 84, 91, 98

[51] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *International Conference on Data Mining '09*, pages 169–178. 2009. 25, 31, 33, 42, 44, 46

[52] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. In *Very Large Databases '08*, pages 102–114. 2008. 33

[53] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the Very Large DataBases Endowment*, 3(1):1021–1032, 2010. 64, 68

[54] Shen-Shyang Ho and Shuhua Ruan. Differential privacy for location pattern mining. In *SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS '11*, pages 17–24. 2011. 26, 58

[55] Wei-Jen Hsu, Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Ahmed Helmy. Modeling time-variant user mobility in wireless mobile networks. In *International Conference on Computer Communications '07*, pages 758–766. 2007. 58

[56] Anco Hundepool, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter-Paul de Wolf. *Statistical Disclosure Control*. Wiley Series in Survey Methodology. Wiley, 2012. ISBN 9781118348222. 10

[57] Neil Hunt. Netflix prize update. `http://blog.netflix.com/2010/03/this-is-neil-hunt-chief-product-officer.html`, 2010. 2

[58] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the Association of Computing Machinery*, 53(3):307–323, May 2006. 107, 112

[59] Sibren Isaacman, Richard Becker, Ramón Cáceres, Stephen Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. Identifying important places in people's lives from cellular network data. In *International Conference on Pervasive Computing '11*, pages 133–151. 2011. 58, 59, 60

[60] Sibren Isaacman, Richard Becker, Ramón Cáceres, Stephen Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. Ranges of human mobility in Los Angeles and New York. In *International Conference on Pervasive Computing and Communications, Workshops '11*, pages 88–93. 2011. 58

[61] Sibren Isaacman, Richard Becker, Ramón Cáceres, Stephen Kobourov, James Rowland, and Alexander Varshavsky. A tale of two cities. In *Workshop on Mobile Computing Systems and Applications '10*, pages 19–24. 2010. 58, 79

[62] Sibren Isaacman, Richard A. Becker, Ramón Cáceres, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. Human mobility modeling at metropolitan scales. In *The International Conference on Mobile Systems, Applications, and Services '12*, pages 239–252. 2012. 5, 55, 58, 59, 61, 62, 77, 78

[63] E. T. Jaynes. Information theory and statistical mechanics. ii. *Physical Review*, 108:171–190, Oct 1957. 122, 129, 130

[64] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *Proceedings of the Very Large DataBases Endowment*, 4(11):1146–1157, 2011. 25, 33, 54

[65] ShivaPrasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography*, volume 7785, pages 457–476. 2013. 25

[66] Georgios Kellaris and Stavros Papadopoulos. Practical differential privacy via grouping and smoothing. In *Very Large DataBases '13*, pages 301–312. 2013. 73

[67] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Principles of Database Systems '11*. 2011. 27

[68] Daniel Kifer and Ashwin Machanavajjhala. A rigorous and customizable framework for privacy. In *Principles of Database Systems '12*, pages 77–88. 2012. 28

[69] Daniel Kifer, Adam D. Smith, and Abhradeep Thakurta. Private convex optimization for empirical risk minimization with applications to high-dimensional regression. *Journal of Machine Learning Research - Proceedings Track*, 23:25.1–25.40, 2012. 26

[70] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *International Conference on Computer Communications '06*, pages 1–13. 2006. 58, 79

[71] Gary King. Ensuring the data-rich future of the social sciences. *Science*, 331(6018):719–721, 2011. 1

[72] Aleksandra Korolova. Protecting privacy while mining and sharing user data. Ph.D. Thesis, 2011. 2

[73] Aleksandra Korolova, Rajeev Motwani, Shubha U. Nabar, and Ying Xu. Link privacy in social networks. In *Conference on Information and Knowledge Management '08*, pages 289–298. 2008. 31, 33

[74] John Krumm. Inference attacks on location tracks. In *Pervasive '07*, pages 127–143. 2007. 58

[75] Jaewoo Lee and Chris Clifton. Differential identifiability. In *Knowledge Discovery and Data mining '12*, pages 1041–1049. 2012. 28

[76] Jure Leskovec. Snap: Stanford network analysis platform. http://snap.stanford.edu/snap/index.html, 2010. URL `http://snap.stanford.edu/snap/index.html`. 46

[77] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042, March 2010. 4, 32, 34, 35, 39, 41, 46, 47, 48, 49

[78] Jure Leskovec and Christos Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *International Conference on Machine Learning '07*, pages 497–504. 2007. 4, 32, 33, 34, 35, 36, 37, 38, 48, 49

[79] Ninghui Li and Tiancheng Li. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *International Conference on Data Engineering 07*, pages 106–115. 2007. 12, 124

[80] Yang D. Li, Zhenjie Zhang, Marianne Winslett, and Yin Yang. Compressive mechanism: Utilizing sparse representation in differential privacy. *CoRR*, abs/1107.3350, 2011. 132

[81] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. Privacy: Theory meets practice on the map. In *IEEE International Conference on Data Engineering '08*, pages 277–286. 2008. 26

[82] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), March 2007. 12

[83] David A. McAllester. Pac-bayesian model averaging. In *Conference on Computational Learning Theory '99*. 1999. 133

[84] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil P. Vadhan. The limits of two-party differential privacy. In *Foundations of Computer Science '10*, pages 81–90. 2010. 15, 122, 123

[85] Frank McSherry. Privacy integrated queries (pinq). `http://research.microsoft.com/en-us/projects/pinq`. 25

[86] Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Communication of the Association of Computing Machinery*, 53(9):89–97, 2010. 69

[87] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science '07*, pages 94–103. 2007. 21, 25, 66, 84, 113, 123, 133, 137

[88] Darakhshan Mir. Differentially-private learning and information theory. In *Joint EDBT/ICDT Workshops '12*, pages 206–210. 2012. iv, 124

[89] Darakhshan Mir and Rebecca N. Wright. A differentially private estimator for the stochastic kronecker graph model. In *Joint EDBT/ICDT Workshops*, pages 167–176. 2012. iv

[90] Darakhshan J. Mir. Information-theoretic foundations of differential privacy. In *Foundations and Practice of Security '12*, pages 374–381. 2013. iv

[91] Darakhshan J. Mir, Ramón Cáceres, Sibren Isaacman, Margaret Martonosi, and Rebecca N. Wright. Differentially private modeling of human mobility at metropolitan scales. In *IEEE International Conference on BigData '13, To Appear*. 2013. iv, 5

[92] Darakhshan J. Mir, S. Muthukrishnan, Aleksandar Nikolov, and Rebecca N. Wright. Pan-private algorithms via statistics on sketches. In *Principle Of Database Systems*, pages 37–48. 2011. iv, 6, 23, 106, 107, 108

[93] Darakhshan J. Mir and Rebecca N. Wright. A differentially private graph estimator. In *ICDM Workshops '09*, pages 122–129. 2009. iv, 4, 42

[94] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy '08*, pages 111–125. 2008. 2

[95] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Symposium on Theory Of Computing '07*, pages 75–84. 2007. 25, 31, 33, 42, 43, 44, 45, 54

[96] J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, 2010. In progress, Chapter 1 online at academic2.american.edu/∼jpnolan. 111

[97] Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *International Conferenece on Computer Vision '09*, pages 460–467. 2009. 76

[98] Marc Perry. As libraries go digital, sharing of data is at odds with tradition of privacy. `http://chronicle.com/article/As-Libraries-Go-Digital/135514/`, 2012. 1

[99] Silvia Polettini. Maximum entropy simulation for microdata protection. *Statistics and Computing*, 13(4):307–320, Oct 2003. 124

[100] Davide Proserpio, Sharon Goldberg, and Frank McSherry. A workflow for differentially-private graph synthesis. In *Workshop on Online Social Networks '12*, pages 13–18. 2012. 25

[101] Wahbeh H. Qardaji, Weining Yang, and Ninghui Li. Differentially private grids for geospatial data. In *International Conference on Data Engineering, '13*, pages 757–768. 2013. 26, 58

[102] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN ISBN 0-262-18253-X. 83, 89, 90

[103] David Rebollo-Monedero, Jordi Forne, and Josep Domingo-Ferrer. From t-closeness-like privacy to postrandomization via information theory. *IEEE Transactions on Knowledge and Data Engineering*, 22:1623–1636, 2010. 124, 125, 126, 127

[104] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking*, 19(3):630–643, June 2011. 58

[105] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p) models for social networks. *Social Networks*, 29(2):173–191, 2007. 54

[106] Aaron Roth. New algorithms for preserving differential privacy. Ph.D. Thesis. CMU-CS-10-135, 2010. 18

[107] Benjamin I. P. Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *CoRR*, abs/0911.5708, 2009. 26, 85

[108] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40, 2000. 76

[109] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y. Zhao. Sharing graphs using differentially private graph models. In *Internet Measurement Conference '11*, pages 81–98. 2011. 25, 33, 49

[110] Lalitha Sankar, S.R. Rajagopalan, and H.V. Poor. A theory of utility and privacy of data sources. In *International Symposium on Information Theory '10*, pages 2642 –2646. 2010. 124, 130, 131

[111] Claude .E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record, Part 4*, pages 142–163, 1959. 130

[112] Adam Smith. Differential privacy and the secrecy of the sample. Blog: Oddly Shaped Pegs, 2009. URL `http://adamdsmith.wordpress.com/2009/09/02/sample-secrecy`. 19

[113] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, February 2010. 58, 59

[114] Latanya Sweeney. Uniqueness of simple demographics in the u.s. population. In Carnegie Mellon University, School of Computer Science, Data Privacy Lab White Paper Series LIDAP-WP4, 2000. 11

[115] Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty Fuzziness and Knowledge-Based Systems*, 10(5):557–570, October 2002. 12

[116] Christine Task and Christopher Clifton. A guide to differential privacy theory in social network analysis. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining '12*, pages 411–417. 2012. 25

[117] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Annual Allerton Conference on Communication, Control and Computing '99*, pages 368–377. 1999. 127, 128, 129

[118] Jaideep Vaidya, Yu Michael Zhu, and Christopher W. Clifton. *Privacy Preserving Data Mining (Advances in Information Security)*. Springer-Verlag New York, Inc., 2005. ISBN 0387258868. 83

[119] Poorvi L. Vora. An information-theoretic approach to inference attacks on random data perturbation and a related privacy measure. *IEEE Transactions on Information Theory*, 53(8):2971–2977, August 2007. 124

[120] James Waldo, Herbert Lin, and Lynette I. Millett. *Engaging privacy and information technology in a digital age*. National Academies Press, 2007. ISBN 9780309103923. 1

[121] Stanley L. Warner. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *Journal of the American Statistical Association*, 60(309):63+, March 1965. 10, 106

[122] Larry Wasserman. *All of Statistics : A Concise Course in Statistical Inference (Springer Texts in Statistics)*. Springer, September 2004. ISBN 0387402721. 34, 93, 96

[123] Hui Zang and Jean Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Mobile Computing and Networking '11*, pages 145–156. 2011. 58

[124] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *Proceedings of the Very Large DataBases Endowment*, 5(11):1364–1375, 2012. 26

[125] Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computing*, 17(9):2077–2098, September 2005. 84, 97, 98, 99

[126] Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006. 133, 137

[127] Elena Zheleva and Lise Getoor. Preserving the privacy of sensitive relationships in graph data. In *Proceedings of the First International Workshop on Privacy, Security, and Trust in KDD '07*, pages 153–171. 2007. 33

[128] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *International Conference on Data Engineering '08*, pages 506–515. 2008. 33