Differential privacy for survival analysis and user data collection

Nguyen, Thong T.

2019

Nguyen, T. T. (2019). Differential privacy for survival analysis and user data collection.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/85347>

<https://doi.org/10.32657/10220/48212>

DIFFERENTIAL PRIVACY FOR SURVIVAL ANALYSIS AND USER DATA COLLECTION

THÔNG T. NGUYỄN

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University in
fulfilment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

Thông T. NGUYÊN

29 March 2019

Date

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

Dr. Siu Cheung Hui

Date

Authorship Attribution Statement

This thesis contains material from 4 papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as *Thông T Nguyễn and Siu Cheung Hui. "Privacy-Preserving Mechanisms for Parametric Survival Analysis with Weibull Distribution". In: The 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom 2017). 2017, pp. 456–463.*

The contributions of the co-authors are as follows:

- A/Prof Hui and I discussed the initial project direction. A/Prof Hui edited and revised the manuscript drafts.
- I proposed the approaches, conducted the experiments and prepared the manuscript drafts.

Chapter 4 is published as *Thông T Nguyễn and Siu Cheung Hui. "Bayesian Inference for Survival Analysis in Private Setting". In: The 2018 IEEE International Conference on Big Knowledge (ICBK 2018). 2018, pp. 314–321.*

The contributions of the co-authors are as follows:

- A/Prof Hui and I discussed the initial project direction. A/Prof Hui edited and revised the manuscript drafts.
- I proposed the approaches, conducted the experiments and prepared the manuscript drafts.

Chapter 5 is published as *Thông T Nguyễn and Siu Cheung Hui. "Differentially Private Regression for Discrete-Time Survival Analysis". In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017). 2017, pp. 1199–1208.*

The contributions of the co-authors are as follows:

- A/Prof Hui and I discussed the initial project direction. A/Prof Hui edited and revised the manuscript drafts.
- I proposed the approaches, conducted the experiments and prepared the manuscript drafts.

Chapter 6 is published as *Thông T Nguyễn and Siu Cheung Hui. "Privacy Protection for Flexible Parametric Survival Models". In: Proceedings of the 2018 ACM on Conference on Information and Knowledge Management (CIKM 2018). 2018, pp. 1273–1282.*

The contributions of the co-authors are as follows:

- A/Prof Hui and I discussed the initial project direction. A/Prof Hui edited and revised the manuscript drafts.
- I proposed the approaches, conducted the experiments and prepared the manuscript drafts.

Thông T. NGUYỄN

29 March 2019

Date

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Hui Siu Cheung for the continuous support of my Ph.D. study. His guidance has helped me in conducting my research and writing this thesis. My sincere thanks also go to Dr. Xiao Xiaokui, who has provided me with an opportunity to work with him on data privacy problems.

I would like to thank NTU for providing me the scholarship to pursue this research, as well as supporting the finance for me to present papers in conferences overseas.

Last but not the least, I would like to thank my parents, and my brother and sister for supporting me throughout the writing of this thesis and my life in general.

Contents

Acknowledgements

Abstract

1	Introduction	1
1.1	Motivation	1
1.1.1	Data Privacy in Survival Analysis	4
1.1.2	Data Privacy in User Data Collection	6
1.2	Objectives	7
1.3	Contributions	8
1.4	Organization of the Thesis	11
2	Background and Related Work	13
2.1	Early Privacy-Preservation Techniques	13
2.1.1	Absolute Disclosure Prevention	13
2.1.2	Randomized Response	14
2.1.3	k-anonymity	15
2.1.4	Other Techniques	16
2.2	Differential Privacy	16
2.2.1	Basic Differentially Private Techniques	19
2.3	Private Survival Analysis	25
2.4	Local Differential Privacy	27
2.5	Summary	28

3	Parametric and Nonparametric Survival Analysis	31
3.1	Survival Estimations	31
3.1.1	Nonparametric Estimations	32
3.1.2	Parametric Estimations	33
3.2	Problem Definitions	33
3.3	Private Nonparametric Estimations	36
3.4	Private Parametric Estimations for Exponential Distribution	37
3.5	Private Parametric Estimation for Weibull Distribution	38
3.5.1	Local Sensitivity Mechanism for Estimating p (LSP)	38
3.5.2	Two Laplace Mechanism for Estimating λ (TLL)	44
3.5.3	Proof of Privacy Protection	44
3.6	Experimental Evaluation	47
3.6.1	Datasets	47
3.6.2	Nonparametric Estimations	48
3.6.3	Parametric Estimations for Exponential Distribution	49
3.6.4	Parametric Estimations for Weibull Distribution	51
3.7	Summary	53
4	Bayesian Parametric Survival Analysis	55
4.1	Bayesian Inference Framework	55
4.2	Problem Definitions	57
4.3	The Sampling-A-Posterior Framework	57
4.3.1	Approximating Log-likelihood Function	58
4.3.2	Publishing Coordinate Vector of Log-likelihood Function .	59
4.3.3	Drawing Random Samples	60
4.4	Applying SAP Framework to Parametric Models	61
4.4.1	Bayesian Modeling for Parametric Models	62
4.4.2	Differentially Private Protection	63
4.5	Applying SAP Framework to Flexible Parametric Models	66

4.5.1	Bayesian Modeling for Flexible Parametric Models	67
4.5.2	Differentially Private Protection	69
4.6	Experimental Evaluation	70
4.6.1	Datasets	70
4.6.2	Publishing the Posterior Distribution	70
4.6.3	Point Estimations of Parametric Models and Flexible Parametric Models	72
4.6.4	Estimating the Underlying Distribution of T	75
4.7	Summary	76
5	Discrete-time Survival Regression	77
5.1	Discrete-time Survival Analysis	78
5.1.1	Discrete-time Survival Model	78
5.1.2	Discrete-time Survival Regression	79
5.2	Problem Definitions	81
5.3	Extended Output Perturbation Approach	82
5.3.1	Proposed Algorithm	83
5.3.2	Privacy Guarantee	83
5.4	Extended Objective Perturbation Approach	87
5.4.1	Proposed Algorithm	87
5.4.2	Privacy Guarantee	88
5.5	Proposed Sampling Approach	91
5.6	Experimental Evaluation	94
5.6.1	Datasets	94
5.6.2	Trade-off between Privacy and Accuracy	95
5.6.3	Convergence of the Proposed Sampling Approach	97
5.7	Summary	98
6	Continuous-time Survival Regression	99
6.1	Continuous-time Survival Regression	100

6.1.1	Proportional Hazard Models	100
6.1.2	Proportional Odds Models	102
6.2	Problem Definitions	103
6.3	Proposed Posterior Sampling Approach	103
6.4	Proposed Posterior Perturbing Approach	108
6.4.1	Random Neural Networks	112
6.4.2	Model-dependent Noisy Prior	113
6.5	Experimental Evaluation	114
6.5.1	Datasets	114
6.5.2	Proposed Posterior Sampling Approach	114
6.5.3	Proposed Posterior Perturbing Approach	116
6.6	Summary	118
7	Local Differential Privacy for User Data Collection	121
7.1	Privacy in User Data Collection	121
7.2	Problem Definitions	123
7.3	Estimating Means and Frequencies	124
7.3.1	Estimating Mean Values for Numeric Attributes	124
7.3.2	Estimating Frequencies for Categorical Attributes	130
7.4	Empirical Risk Minimization	136
7.4.1	Basic Methods	136
7.4.2	Proposed Improvements	138
7.5	Experimental Evaluation	140
7.5.1	Datasets	141
7.5.2	Estimating Means and Frequencies	141
7.5.3	Empirical Risk Minimization	143
7.6	Summary	146
8	Conclusion and Future Work	149
8.1	Conclusion	149

8.2	Future Work	151
8.2.1	Differentially Private Cox Regression	151
8.2.2	Differential Privacy for Bayesian Inference	152
8.2.3	Local Differential Privacy	152
8.2.4	Differential Privacy for Deep Learning Models	153
Appendix A Spline Basis Functions		155
Bibliography		157

List of Figures

1.1	The generic data querying setting for data privacy.	2
1.2	Contributions and organization of chapters in the thesis.	10
2.1	Two neighboring datasets $\mathcal{D}, \mathcal{D}'$ which are different by only one row.	17
2.2	The output distributions for two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ . .	18
3.1	An example of the utility function $\mathcal{U}(p, \mathcal{D})$. The utility function has 6 rungs. The root of Equation (3.7) on dataset $\mathcal{D}$ is $p(\mathcal{D})$, which is located at rung 0.	39
3.2	The estimated value of p on dataset $\mathcal{D}^{(k)}$ is determined by the intersection C of two curves $f(p, \mathcal{D}^{(k)})$ and $g(p, \mathcal{D}^{(k)})$	42
3.3	Experimental results (in MdAE) for nonparametric estimators. . .	49
3.4	Experimental results (in MdAE) for parametric estimations of ex- ponential distribution.	50
3.5	Performance results (in MdAE) on estimating the value of the shape parameter p	52
3.6	Performance results (in MdAE) on estimating the value of the scale parameter λ	53
4.1	Convex hull K for a parametric model with exponential distribution.	60
4.2	An illustration of t_i^p and C-spline basis functions which are used to approximate t_i^p	64

4.3	Performance results (in KLD) of our proposed SAP framework for parametric models and flexible parametric models.	72
4.4	Performance results (in MdAE) for parametric models with Weibull distribution.	73
4.5	Performance results (in MdAE) for flexible parametric models. . .	74
4.6	Estimated distribution of survival functions using our proposed SAP framework (in red) in comparison with that of Kaplan-Meier estimator and its 95% confidence interval (in black).	75
5.1	The illustration of link functions: logit (green), cloglog (blue) and probit (red).	80
5.2	The performance results (in AUC-ROC) of our proposed approaches with privacy budget ϵ from 0.1 to 1.6.	96
5.3	The convergence (in MRE) of the pSGLD sampling algorithm. . . .	98
6.1	An illustration of the sanitizer functions with a maximum value of 0 and minimum value of -3 . $C_1(\cdot)$ is our proposed sanitizer function, while $C_2(\cdot)$ was previously proposed in Chapter 5.	104
6.2	Log-likelihood function before and after adding a noisy prior. . . .	110
6.3	A feed-forward neural network whose network weights are randomly initialized.	112
6.4	Performance results (in AUC-ROC) of the posterior sampling approach (PSA) with different privacy budgets ϵ	116
6.5	Performance results of the posterior perturbing approach (PPA) with different numbers of fake records.	117
6.6	Estimated credible intervals of the model's parameters.	118
7.1	Experimental results on estimating means and frequencies.	143
7.2	Experimental results for linear regression.	145
7.3	Experimental results for logistic regression and SVM.	146

A.1	I-spline basis functions of order 3 with 5 knots.	156

List of Tables

2.1	A 2-anonymized dataset.	15
2.2	Probability distributions of $\mathcal{M}(x)$	18
3.1	Datasets used in the experiments.	48
5.1	Statistics of the datasets.	94
7.1	Statistics of the datasets.	141

List of Notations

$\Delta, \mathcal{D}$	A dataset of user data records
$\Delta', \mathcal{D}'$	A neighboring dataset
ϵ	Privacy budget
(x_i, t_i, d_i)	A tuple of covariate vector x_i , survival time t_i and death indicator d_i
$h(t_i)$	Hazard function at t_i
$S(t_i)$	Survival function at t_i
$h(t_i \mid x_i)$	Hazard function at t_i given x_i
$S(t_i \mid x_i)$	Survival function at t_i given x_i
$h(t; \theta)$	Hazard function with parameter θ
$S(t; \theta)$	Survival function with parameter θ
$H(t)$	Cumulative hazard function
$D_{KL}(P Q)$	The Kullback-Leibler divergence between two probabilities P and Q
$\text{Lap}(s)$	A Laplacian random variable with scale s
MdAE	Mean absolute error
MLE	Maximum Likelihood Estimation
MRE	Mean Relative Error
pdf	Probability density function
cdf	Cumulative distribution function
$\ell(\cdot)$	Loss function
$\text{Var}[X]$	Variance of random variable X
$\mathbb{E}[X]$	Expectation of random variable X
$\text{Pr}[X], \text{Pr}(X)$	Probability of event X

NANYANG TECHNOLOGICAL UNIVERSITY

Abstract

School of Computer Science and Engineering

Doctor of Philosophy

Differential Privacy for Survival Analysis and User Data Collection

by Thông T. NGUYỄN

Supervisor:

Dr. Siu Cheung Hui

Most of the personal information nowadays exist in the form of digital data which includes sensitive information such as medical records, credit card information, private instant messages, etc. In this research, we aim to investigate the data privacy problem in collecting and mining user sensitive information. We focus our research on: (i) data privacy in survival analysis which uses medical records to learn useful survival models in medical research; and (ii) data privacy in collecting user data which is the current practice of many corporations and governments. We use differential privacy, which is the golden standard in privacy protection, to address the data privacy problem in survival analysis and user data collection. To this end, we aim to achieve the following:

- Guaranteeing privacy for survival analysis models which include (i) parametric and nonparametric survival models; and (ii) continuous-time and discrete-time survival regression models.
- Guaranteeing privacy for users whose data is collected by corporations and governments.

The main contributions of this thesis are given as follows:

- For nonparametric survival models, we have proposed a private mechanism for two popular nonparametric estimators, namely Kaplan-Meier estimator and Nelson-Aalen estimator. For parametric survival models, we have proposed a simple private mechanism for accurately estimating the parameter of the exponential distribution. In addition, we have also proposed a private mechanism based on the local sensitivity concept for estimating the parameters of the Weibull distribution.
- For estimating uncertainty in parametric survival models, we have proposed a private framework which allows learning the posterior function. Moreover, we have applied the proposed framework to parametric models with Weibull distribution and flexible parametric models.
- We have proposed three private approaches for estimating the discrete-time survival regression model, namely extended output perturbation approach, extended objective perturbation approach, and posterior sampling approach.
- We have proposed a posterior sampling approach for continuous-time survival regression model. In addition, we have also proposed a posterior perturbing approach which supports a relaxation of differential privacy for scenarios in which differential privacy is impractical.
- For user data collection, we have proposed mechanisms which allow each user to publish a randomized vector of categorical data and numerical data. The proposed mechanisms are asymptotically optimal in both accuracy and run-time. Moreover, we have applied the proposed mechanisms to supervised learning problems under the empirical risk minimization framework.

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

Alan Turing

For my family.

Chapter 1

Introduction

1.1 Motivation

Most of the personal information nowadays exist in the form of digital data such as web browsing data, medical data, employee data, social networking data, etc. Currently, many companies and governments gather data from their users/citizens on a regular basis. Many of these collected data contain sensitive information such as medical records, credit card information, private instant messages, and information which can be used to identify a person uniquely. For example, de Montjoye *et al.* [80] showed that it can uniquely identify 95% of individuals by using only four spatiotemporal data points collected from them. The personal information collected may lead to a serious data privacy problem. For example, in the year 2015, the customer data of Ashley Madison (a popular online dating service) had been stolen¹. The stolen data include personal information such as name, email address, home address, sexual fantasies and credit card information. One approach to solving the data privacy problem is to simply let people keep their own data without sharing it with others. However, this approach denies many benefits from sharing information. For example, Google uses users' suggestions on spam emails for training its machine learning engine which decides whether an email is a spam email. In another example, medical

---

¹https://en.wikipedia.org/wiki/Ashley_Madison

Figure 1.1: The generic data querying setting for data privacy.

records of patients can be used to identify the causes and discover the cures for many diseases.

It is important to clarify the notion of privacy that is considered in this research. We consider a privacy setting in which the result from a data analysis procedure is published for the benefit of many people. It is different from the notion of privacy in cryptography in which information is only distributed between restricted parties. In cryptography, privacy is an *all-or-nothing* property. While in the data privacy setting, there will always be a *trade-off* between the privacy of individuals and the benefit of the society. In particular, we consider a generic data privacy setting which is shown in Figure 1.1. There are different actors in this setting as follows:

- **Data Analyst** - An actor who wants to extract useful information from a dataset. However, the actor cannot access the dataset directly and has to query information through a *privacy protection mechanism* $\mathcal{M}$, which acts like a proxy layer and thus controls the output of queries.
- **Adversary** - An actor who is not allowed to access the private information kept in the dataset. However, the actor can know the private information by observing the queries and answers between the data analyst and mechanism $\mathcal{M}$.

- Curator - An actor who is allowed to access the dataset directly and also allowed to control the mechanism $\mathcal{M}$.

In this setting, we distinguish between *public domain* and *private domain*. In the public domain, information is, in principle, shared between actors. Whereas in the private domain, information is kept private from actors in the public domain. Only trusted parties (e.g., the curator) are allowed to access the data. In general, there are two modes for querying information from a dataset:

- Offline Mode - The mechanism $\mathcal{M}$ only publishes the result to the public domain once. Any further information must be inferred from this one-time published result.
- Online Mode - The data analyst can query the dataset many times. Moreover, it is also possible that subsequent queries may depend on the earlier answers.

Many years of research have shown that protecting privacy in the above setting is not straightforward. Intuitive solutions are usually wrong. For example, the presumption that when the identification attributes (e.g., social security number, full name, home address, email address, phone number, etc.) are removed from a dataset, it will become anonymous and can be published, turned out to be *completely wrong*. In 1996, Sweeney [96] was able to identify the medical record of William Weld, a governor of Massachusetts at that time, from a dataset which was believed to be anonymous. Sweeney recovered a unique tuple in the medical dataset which had the values of ZIPCODE, Birthdate, and Gender matched with the public information of the governor William Weld, which was extracted from a public voter list. This kind of attack is called *linkage attack*. It uses the links between attributes in an anonymous dataset and attributes in a public dataset to identify the identity of individuals in the anonymous dataset. This attack is also used in another famous case of the Netflix Prize competition where the identities of Netflix's users are discovered by linking them with the public IMDB

dataset [81]. In another case, Homer *et al.* [57] demonstrated that it is possible to determine whether a person is in an anonymous DNA dataset. This causes the National Institutes of Health and the Wellcome Trust Foundation to cease public access to the DNA dataset.

Many techniques have been proposed to protect privacy in the areas of statistics [96, 75, 72] and machine learning [46, 5, 104]. However, these techniques have suffered from the same weakness of lacking a strong and rigorous framework of privacy protection. In this research, we aim to investigate privacy protection mechanisms based on the foundation of differential privacy [41] which offers a strong and robust definition of privacy. From that, we are able to define and guarantee the trade-off between privacy and the utility of information rigorously. The core idea behind differential privacy is that: to protect privacy, the presence or absence of any individual in a dataset should not affect much on the answers for queries from the data analyst. Hence, it guarantees that the adversary will not be able to learn much about individuals' information from the answers.

In recent years, a lot of work has been done on the theoretical and fundamental aspects of differential privacy such as the work on the definition of differential privacy and its composability [38, 36], Laplace mechanism [38], exponential mechanism [76], sparse vector technique [39], etc. However, there has not been much work done on the applications of differential privacy for data analysis and machine learning problems. In this research, we will investigate how to apply differential privacy to deal with two important data analysis problems, namely survival analysis and user data collection.

1.1.1 Data Privacy in Survival Analysis

We focus on the data privacy problem in survival analysis because of its important implications for society. Survival analysis is a topic in statistics which studies the statistical properties of the lifetime of objects. Its applications include

studying the lifetime of a population, the lifetime of patients, the duration of a customer using a service, etc. In survival analysis, the survival function and hazard function are most commonly estimated by data analyst. The survival function $S(t)$ is the probability that a patient survives during the time interval $[0, t]$. The hazard function $h(t)$ is the risk of dying of a patient at time t . In principle, we could estimate these functions by (i) nonparametric models such as the Kaplan-Meier estimator [65] which can be used to estimate survival function and the Nelson-Aalen estimator [82] which can be used to estimate hazard function, or (ii) parametric models which assume an underlying probability distribution for the time of death. Survival analysis also requires solving regression problems which estimate the effects of explanatory variables (also known as covariates) to the survival chance of patients. For example, attributes such as gender, age, weight, and other medical conditions can greatly affect the probability of success and recovery in a medical operation.

However, survival analysis can leak private information to the public. For example, the results from a study on the lifetime of HIV patients can leak the identity of the individuals participated in the study. In general, it is widely accepted that personal health records are confidential. Therefore, it is an ethical and legal problem to guarantee privacy for people whose data are being used.

To protect confidential information in survival analysis, the state-of-the-art method [86] controls the output of queries by a central server. The central server does not give the data analyst the exact answer but uses confidentialising techniques (such as choosing 95% random samples or adding noise) to create a randomized answer and give that to the data analyst. However, this approach still lacks a robust framework to guarantee privacy. In this research, we aim to propose privacy mechanisms for survival analysis based on the differential privacy framework, a strong mathematical privacy framework that has been studied in recent years.

1.1.2 Data Privacy in User Data Collection

In this research, we also investigate the data privacy problem in the practice of collecting massive data of user information by corporations or governments. Smart devices connected to the Internet, including mobile phones, wearables, home appliances, sensors, etc. have become part of daily life. The data collected by these devices could be an invaluable asset to hardware designers and application developers. For instance, a smartphone maker with its own customized user interface (UI) could learn about the usage patterns of the various UI features such as Multi-Window and One-Handed Mode, and focus on improving them. However, user data privacy remains a major hurdle in collecting users' data.

Local Differential Privacy (LDP) [33], which has been used in Google's Chrome browser [45], addresses the above dilemma. The idea is to compute aggregates of users' data without collecting individuals' precise personal information. Unlike other models of differential privacy [37, 40] which publish randomized aggregates but still collect the exact sensitive data, LDP avoids collecting exact personal information in the first place, thus providing a stronger assurance to the users and to the curator. Meanwhile, LDP satisfies the strong and rigorous privacy guarantees of differential privacy, i.e., the adversary cannot infer sensitive information about an individual with high confidence, regardless of the adversary's background knowledge.

Google's LDP solution in its Chrome browser, namely Rappor [45], has rather limited functionalities. The core of Rappor is a randomized response mechanism, and this randomized approach provides plausible deniability to the individuals. However, three major limitations remain. Firstly, Rappor cannot calculate aggregates on numeric attributes, e.g., the average running time of an app. Secondly, the accuracy of Rappor deteriorates quickly with increasing number of attributes; for example, to compute aggregates on d independent attributes, the

error of Rappor grows linearly with d , which is sub-optimal [33, 32]. Thirdly, it is unclear whether or how Rappor can handle complex and yet commonly used machine learning tasks, such as logistic regression and SVM classification.

LDP has drawn considerable interest from the research community. However, the current approaches focus mainly in analyzing the asymptotical performance of basic building blocks of LDP, rather than practical systems and performance. Meanwhile, we are not aware of any systematic approach that can perform common machine learning tasks under LDP on a large data domain containing both numeric and categorical attributes. In this research, we aim to resolve the problems of collecting both categorical and numerical data for LDP and its application for training machine learning models.

1.2 Objectives

In this section, we state the objectives of this research as follows:

- **Data Privacy in Survival Analysis** - We aim to apply differential privacy, which is a robust privacy protection framework, to address the data privacy problems in survival analysis. The problem set includes: (i) parametric and nonparametric estimation of survival function and hazard function; and (ii) survival regression for explanatory variables. This addresses the data privacy problem in mining useful information from personal health records for survival analysis.
- **Data Privacy in User Data Collection** - We aim to apply local differential privacy to the problem of collecting user data. The problem set includes: (i) collecting both categorical and numerical data from users; and (ii) using the collected data to train complex machine learning models. This addresses the data privacy problem in collecting massive data of user information by corporations or governments.

1.3 Contributions

In this research, we have achieved the following:

- Nonparametric Survival Analysis (Chapter 3) - We have proposed a private mechanism which releases a private histogram of the survival dataset. The proposed mechanism supports both the Kaplan-Meier estimator and the Nelson-Aalen estimator. It allows data analyst to use these two estimators to estimate the survival function and hazard function without the need of any modifications.
- Parametric Survival Analysis (Chapters 3 and 4) - We have proposed private mechanisms for the exponential distribution and Weibull distribution, which are the two most common distributions in survival analysis. The proposed mechanisms support estimating the parameters of the exponential distribution and Weibull distribution. In addition, we have also addressed the problem of estimating uncertainty in parametric survival analysis. Based on the Bayesian framework, we have proposed the Sampling-A-Posterior (SAP) framework which allows data analysis to learn the posterior distributions of parametric models. The proposed mechanisms support parametric models with exponential distribution and Weibull distribution. Moreover, the SAP framework also supports *flexible* parametric models, in which we use spline functions to approximate the cumulative hazard function. As such, we have extended the expressive power of parametric models. The proposed framework uses the K -norm mechanism to guarantee differential privacy for the posterior function in the functional space.
- Discrete-time Survival Regression (Chapter 5) - We have extended the output perturbation and objective perturbation mechanisms [24], which guarantee differential privacy for the empirical risk minimization framework,

to the discrete-time survival regression. The proposed extended output perturbation approach guarantees differential privacy by adding noise to the regression output. The proposed extended objective perturbation approach adds noise to the empirical risk function, which is then minimized to obtain the regression output. In addition, we have also proposed a novel posterior sampling approach which guarantees differential privacy for a random sample drawn from a distribution which approximates the true posterior distribution.

- Continuous-time Survival Regression (Chapter 6) - We have applied the posterior sampling approach to continuous-time survival regression models. Moreover, we have also proposed the posterior perturbing approach which supports a relaxation of differential privacy for scenarios in which differential privacy is impractical.
- Local Differential Privacy (Chapter 7) - We have proposed a local private mechanism for gathering user information. It allows the estimation of the mean of numeric vectors which are collected from a population of users. Our proposed mechanism is asymptotically optimal in both accuracy and run-time. In addition, we have also proposed a private mechanism for estimating the frequency of categorical values which is simpler than other solutions and also achieves the asymptotically optimal accuracy. The above private mechanisms have been applied to the problem of training machine learning models under the empirical risk minimization framework.

Figure 1.2 illustrates the contributions and organization of chapters in the thesis.

Figure 1.2: Contributions and organization of chapters in the thesis.

1.4 Organization of the Thesis

This chapter discusses the motivation, objectives, and contributions in this thesis.

The rest of the thesis is organized as follows:

- In Chapter 2, we first present the early privacy preservation techniques. Then, we introduce the definition of differential privacy and some basic differentially private techniques. We also discuss the related work on survival analysis and local differential privacy.
- In Chapter 3, we consider the application of differential privacy in survival analysis. In particular, we propose mechanisms to tackle the problems of estimating the survival function and the hazard function for both parametric and nonparametric models.
- In Chapter 4, we propose a differentially private framework for Bayesian parametric survival analysis.
- In Chapter 5, we present our solutions for protecting privacy in discrete-time survival regression. We propose three private approaches for estimating the effects of the explanatory variable to survival probability in a discrete-time setting.
- In Chapter 6, we present our solutions for continuous-time survival regression. We propose a private approach for estimating the survival model. Moreover, we also propose an approach which supports a relaxation of differential privacy for certain application scenarios in which differential privacy is not practical.
- In Chapter 7, we propose mechanisms for the problem of collecting user data of categorical and numeric values. Then, we discuss how to apply the proposed mechanisms to train machine learning models under the empirical risk minimization framework.

- In Chapter 8, we give a summary of our work on survival analysis and local differential privacy. In addition, we also discuss the directions for future research work.

Chapter 2

Background and Related Work

In this chapter, we first review early works in privacy-preservation techniques. We then introduce differential privacy and its basic techniques. After that, we discuss the related work in private survival analysis and local differential privacy.

2.1 Early Privacy-Preservation Techniques

There are many early works in the research community to formalize data privacy. In this section, we will introduce these works and discuss their weaknesses, which motivate the development of differential privacy.

2.1.1 Absolute Disclosure Prevention

Dalenius [29] introduced the idea of absolute disclosure prevention (or ideal privacy) where the adversary cannot learn anything about individuals in the dataset from the mechanism $\mathcal{M}$'s answers. However, Dwork [34] showed that ideal privacy is impossible in practice. In theory, the mechanism $\mathcal{M}$ can publish completely random or nonsense answers so no one can learn anything from the answers. However, in practice, we would like to have something useful from the answers. The following example will illustrate that ideal privacy is

impossible when the answers are meaningful and the adversary has enough auxiliary information.

Example 2.1 (Smoking Causes Cancer). *A researcher studies the association between smoking and lung cancer based on the data from a group of participants in an experiment. The researcher comes to a conclusion that if someone is a smoker, he will have a higher risk of having lung cancer than those who are not. Thus, if Bob is one of the participants and the adversary (perhaps, an insurance company) has an auxiliary information that “Bob is a smoker,” the adversary can easily infer that Bob has a higher risk of having cancer than usual. Further, if the adversary has information about Alex, who is also a smoker, but did not take part in the experiment, then the adversary could also infer about Alex’s risk of having cancer.*

We can observe from this example that when the answer is *meaningful*, then the adversary could learn something about someone, who is not even in the dataset, if the adversary has enough auxiliary information. This example illustrates the fact that ideal privacy is impossible in practice.

2.1.2 Randomized Response

Randomized response [100, 19] is a method for collecting sensitive *yes* or *no* answers from participants in a survey, e.g., collecting information about whether a person uses drugs or not. For a participant in the survey, let $x, y \in \{0, 1\}$ be the true answer and the reported answer respectively. The participant tosses a coin with probability p for a head. If the coin is head, the participant will answer truthfully $y = x$. Otherwise, if the coin is tail, the participant will answer $y = 1 - x$. We have $\mathbb{E}[y] = px + (1 - p)(1 - x) = 1 - p + (2p - 1)x$. Thus, y is a biased estimation of x , and the unbiased estimation is $y' = \frac{y+p-1}{2p-1}$.

This method allows the participants to have the deniability on their answers. Hence, it protects participants’ privacy. However, it only allows the curator to

Table 2.1: A 2-anonymized dataset.

Name	Age	Gender	Disease
*	<20	*	Cancer
*	20-40	*	TB
*	<20	*	Cancer
*	20-40	*	TB
*	<20	*	Cancer

collect binary information. There is no obvious way to extend this method to other data types.

2.1.3 k-anonymity

Sweeney proposed the *k-anonymity* notion as a guarantee for privacy [96, 44]. To protect privacy, the curator releases a synthetic version of the dataset which has a property that any tuple (user) in the synthetic dataset is indistinguishable from at least $k - 1$ other tuples (users) in the same dataset. Table 2.1 gives an example of a 2-anonymized dataset. The attributes Name and Gender are replaced by “*” values; and the attribute Age is converted to generic categories.

Although the problem of optimally converting the original dataset to a k -anonymized dataset is NP-hard [77], the k-Optimize method [16] can be used to obtain good performance results in a reasonable time. However, k -anonymity can suffer from various kinds of attacks. Charu [4] showed that on a dataset with a high number of attributes, one can identify (with high probability) a person in the dataset by an inference attack over all possible combinations of attributes. Machanavajjhala *et al.* [75] showed that if there is little diversity in the values of sensitive attributes, the adversary can easily infer those sensitive values. In addition, k -anonymity is also vulnerable when the adversary has enough auxiliary information.

2.1.4 Other Techniques

There are many other randomization techniques [46, 5, 104] used in data mining for protecting privacy. In general, these techniques add noise and randomly distort the dataset to protect privacy. However, the fundamental problem of these techniques is that even though the results are randomized, there is still no rigorous definition on privacy guarantee.

2.2 Differential Privacy

The notion of differential privacy is based on the idea that the privacy of an individual is protected if the addition or removal of the data record of that person from the dataset will not significantly affect the output. In other words, the output of queries should not strongly depend on any particular individual in the dataset. Dwork *et al.* set the foundation for this notion of privacy in their seminal papers [40, 38, 76]. Before introducing the formal definition of differential privacy, we first give the definition of a dataset and neighboring datasets.

Definition 2.1 (Dataset). *Suppose each user is represented by a tuple t of d values, $t = (a_1, a_2, \dots, a_d)$, in a universe $\mathcal{X}$. A dataset $\mathcal{D}$ is a collection of tuples, $\mathcal{D} = \{t_1, t_2, \dots, t_n\}$. In addition, $\mathcal{D}$ can also be represented as a histogram of $\mathcal{X}$, $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$, where $\mathbb{N}^{|\mathcal{X}|}$ is the set of all possible datasets.*

The tuple t for each user is regarded as private information of that user, and the goal is to hide it from the public domain while providing meaningful answers to queries by data analyst.

Definition 2.2 (Neighboring Datasets). *Two datasets $\mathcal{D}, \mathcal{D}'$ are neighbors if and only if they agree in all tuples except one (as shown in Figure 2.1). Moreover, if we represent*

Figure 2.1: Two neighboring datasets $\mathcal{D}, \mathcal{D}'$ which are different by only one row.

$\mathcal{D}$ and $\mathcal{D}'$ by histograms, where $\mathcal{D}, \mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|}$, then

$$d(\mathcal{D}, \mathcal{D}') = \|\mathcal{D} - \mathcal{D}'\|_1 = 1$$

After defining the concept of neighboring datasets, we now can formalize the definition of differential privacy based on the idea that the difference between the probability distributions of the outputs from two neighboring datasets is limited.

Definition 2.3 ((ϵ, δ) –Differential Privacy [38, 36]). A mechanism (function) $\mathcal{M}$ is called (ϵ, δ) –differential privacy if for any pair of neighboring datasets $\mathcal{D}, \mathcal{D}'$ and any event $\mathcal{S} \subset \text{range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta$$

For convenience, we use ϵ –differential privacy as the alias for $(\epsilon, 0)$ –differential privacy. The parameter ϵ is called *privacy budget* which has the role in controlling the trade-off between privacy and accuracy of the output. Large ϵ will, in general, lead to better accuracy but leak more private information. On the other hand, small ϵ is better in protecting privacy but poorer in returning a useful answer.

Figure 2.2 illustrates the two probability distributions of a randomized mechanism on two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. An ϵ –differentially private mechanism will guarantee that the ratio between two probability densities at any

Figure 2.2: The output distributions for two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$.

Table 2.2: Probability distributions of $\mathcal{M}(x)$.

$\mathcal{M}(x)$	0	1
$x = 0$	p	$1 - p$
$x = 1$	$1 - p$	p

location is bounded by $\exp(\epsilon)$.

We now illustrate the notion of differential privacy by proving that the *randomized response* method is a differentially private mechanism.

Lemma 2.1. Suppose $0 < p < 1$ and $x \in \{0, 1\}$, a mechanism $\mathcal{M}$ such that

$$\mathcal{M}(x) = \begin{cases} x & \text{with probability } p \\ 1 - x & \text{with probability } 1 - p \end{cases}$$

then $\mathcal{M}$ is $\left| \log \left(\frac{p}{1-p} \right) \right|$ -differentially private.

Proof. This problem is perhaps the simplest problem we may have, the “dataset” has only one value x , and there are only two possibilities $x = 0$ or $x = 1$. We can easily create a table on the output’s distributions for both cases as shown in Table 2.2.

By following the definition of differential privacy, we consider the probability ratios:

$$\frac{\Pr[\mathcal{M}(x) = 0 \mid x = 0]}{\Pr[\mathcal{M}(x) = 0 \mid x = 1]} = \frac{p}{1 - p} \quad (2.1)$$

and

$$\frac{\Pr[\mathcal{M}(x) = 1 \mid x = 0]}{\Pr[\mathcal{M}(x) = 1 \mid x = 1]} = \frac{1 - p}{p} \quad (2.2)$$

By symmetry, we also get the same ratios after swapping the denominators and numerators of the left-hand sides of Equation (2.1) and Equation (2.2). We conclude that $\mathcal{M}$ is $\left\lceil \log \left(\frac{p}{1-p} \right) \right\rceil$ -differential privacy. $\square$

Compared to the earlier approaches, some of the advantages of differential privacy are given below:

- Differential privacy is a rigorous mathematical definition of privacy while the early methods guarantee privacy in an ad-hoc manner.
- Differential privacy does not try to achieve *absolute disclosure prevention* but instead, it proposes a trade-off between privacy and accuracy of the output.
- Differential privacy is a property of the mechanism only. It does not depend on any inference after the information is released, or make any assumption on the computational power and prior knowledge of the adversary.

2.2.1 Basic Differentially Private Techniques

In this section, we will review some basic methods that allow us to publish real values and categorical values. Besides, we will also discuss the guarantee in the accuracy of these methods. After that, we present the compositional theorems which will allow us to combine basic mechanisms together.

Laplace mechanism [38]: Suppose we want to compute the value of a function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}$ on a dataset $\mathcal{D}$ (e.g., $f(\mathcal{D})$ is the number of patients who have cancer in the dataset $\mathcal{D}$). The goal is to design an ϵ -differentially private mechanism $\mathcal{M}$ which takes the dataset $\mathcal{D}$ as its input and publishes an approximated value of $f(\mathcal{D})$.

The idea of Laplace mechanism is that we first compute the value of $f(\mathcal{D})$ and then we add Laplace noise¹ to the result such that the magnitude of noise is proportional to the “sensitivity” of the function f . The “sensitivity” roughly means how much the value of $f(\mathcal{D})$ changes when we replace $\mathcal{D}$ by its neighbors. The formal definition is given below.

Definition 2.4 (Global Sensitivity of a Function). *Let $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}$, the global sensitivity of f is:*

$$\Delta_f = GS(f) = \max_{d(\mathcal{D}, \mathcal{D}')=1} |f(\mathcal{D}) - f(\mathcal{D}')|$$

The Laplace mechanism can be expressed by the following formula:

$$\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + b \tag{2.3}$$

where $b \sim \text{Lap}(\Delta_f/\epsilon)$.

The following theorem proves that the Laplace mechanism given in Equation (2.3) satisfies the definition of differential privacy.

Theorem 2.1. *Laplace mechanism is ϵ -differential privacy.*

Proof. For any $x \in \mathbb{R}$, consider the probability ratio:

$$\frac{\text{pdf}(\mathcal{M}(\mathcal{D}) = x)}{\text{pdf}(\mathcal{M}(\mathcal{D}') = x)} = \frac{\text{pdf}(b = x - f(\mathcal{D}))}{\text{pdf}(b = x - f(\mathcal{D}'))} \tag{2.4}$$

$$= \frac{\exp\left(-\frac{\epsilon|x-f(\mathcal{D})|}{\Delta_f}\right)}{\exp\left(-\frac{\epsilon|x-f(\mathcal{D}')|}{\Delta_f}\right)} \tag{2.5}$$

$$\leq \exp\left(\frac{\epsilon|f(\mathcal{D}) - f(\mathcal{D}')|}{\Delta_f}\right) \tag{2.6}$$

$$\leq \exp(\epsilon) \tag{2.7}$$

---

¹Recall that if $x \sim \text{Lap}(s)$, then $\text{pdf}(x; s) = \frac{1}{2s} \exp\left(-\frac{|x|}{s}\right)$ where $\text{pdf}(x; s)$ is the probability density function of x .

We have Equation (2.6) by using the triangle inequality and Equation (2.7) by using the definition of *global sensitivity*. By symmetry, the inequality still holds when we change the role of $\mathcal{D}$ and $\mathcal{D}'$. $\square$

In fact, we can extend the Laplace mechanism for functions whose return value is a vector instead of a real number, $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$, where k is the dimensionality of the output. The global sensitivity becomes the maximum l_1 -norm distance between the outputs on two neighboring datasets $\mathcal{D}, \mathcal{D}'$:

$$\Delta_f = \max_{d(\mathcal{D}, \mathcal{D}')=1} \|f(\mathcal{D}) - f(\mathcal{D}')\|_1$$

Moreover, the guarantee on the accuracy of the Laplace mechanism is given below.

Theorem 2.2 (Laplace Mechanism's Accuracy Guarantee [41]). *Let $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$. $\forall \delta \in (0, 1]$,*

$$\Pr \left[\|f(\mathcal{D}) - \mathcal{M}(\mathcal{D})\|_\infty \geq \log \left(\frac{k}{\delta} \right) \cdot \left(\frac{\Delta_f}{\epsilon} \right) \right] \leq \delta$$

Let $d = \|f(\mathcal{D}) - \mathcal{M}(\mathcal{D})\|_\infty$ denote the distance between the true value and the returning value from Laplace mechanism. Theorem 2.2 assures that the probability will decay exponentially when d is increasing.

Exponential mechanism [76]: After discussing the mechanism to publish real values, the next natural question is how to publish categorical values. Suppose we want to compute the function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{C}$ where $\mathcal{C} = \{0, 1, \dots, k-1\}$. The idea behind Laplace mechanism is that the values, which are closer to the “truth” value of $f(\mathcal{D})$, should have a higher probability to be published. Unfortunately, for categorical values, we lose the notion of “distance”. The exponential mechanism solves this problem by generalizing the notion of distance by using *utility function*. Instead of constructing the probability distribution based on the distance to the “truth” value, it constructs the probability distribution based on

the *usefulness* of these values. “Usefulness” is a conceptual property depending on the characteristics of the problem. For example, if the published result is the price in an auction, then the revenue will be the usefulness of that price.

Definition 2.5 (Utility Function). *Let $u : \mathcal{C} \times \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}$ be a utility function, then $u(x, \mathcal{D})$ is defined as the “usefulness” when the private mechanism publishes x as the output of a query on the dataset $\mathcal{D}$.*

Similar to the definition of the global sensitivity of a function in the Laplace mechanism, we also have the definition of *sensitivity* for a utility function.

Definition 2.6 (Sensitivity of a Utility Function). *The sensitivity of a utility function u is*

$$\Delta_u = \max_{x, d(\mathcal{D}, \mathcal{D}')=1} |u(x, \mathcal{D}) - u(x, \mathcal{D}')|$$

Definition 2.7 (Exponential Mechanism). *A randomized mechanism (function) $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{C}$ is an exponential mechanism with a utility function $u : \mathcal{C} \times \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}$ if*

$$\Pr[\mathcal{M}(\mathcal{D}) = x] \propto \exp\left(\epsilon \cdot \frac{u(x, \mathcal{D})}{2\Delta_u}\right)$$

Theorem 2.3 proves that the exponential mechanism satisfies the definition of differential privacy.

Theorem 2.3. *Exponential mechanism is ϵ -differential privacy.*

Proof. For any $x \in \mathcal{C}$, consider the probability ratio:

$$\frac{\Pr[\mathcal{M}(\mathcal{D}) = x]}{\Pr[\mathcal{M}(\mathcal{D}') = x]} = \frac{\frac{\exp\left(\epsilon \cdot \frac{u(x, \mathcal{D})}{2\Delta_u}\right)}{\sum_{k \in \mathcal{C}} \exp\left(\epsilon \cdot \frac{u(k, \mathcal{D})}{2\Delta_u}\right)}}{\frac{\exp\left(\epsilon \cdot \frac{u(x, \mathcal{D}')}{2\Delta_u}\right)}{\sum_{k \in \mathcal{C}} \exp\left(\epsilon \cdot \frac{u(k, \mathcal{D}')}{2\Delta_u}\right)}} \quad (2.8)$$

$$= \frac{\exp\left(\epsilon \cdot \frac{u(x, \mathcal{D})}{2\Delta_u}\right)}{\exp\left(\epsilon \cdot \frac{u(x, \mathcal{D}')}{2\Delta_u}\right)} \cdot \frac{\sum_{k \in \mathcal{C}} \exp\left(\epsilon \cdot \frac{u(k, \mathcal{D}')}{2\Delta_u}\right)}{\sum_{k \in \mathcal{C}} \exp\left(\epsilon \cdot \frac{u(k, \mathcal{D})}{2\Delta_u}\right)} \quad (2.9)$$

$$\leq \exp\left(\frac{\epsilon}{2}\right) \cdot \exp\left(\frac{\epsilon}{2}\right) \quad (2.10)$$

$$= \exp(\epsilon) \quad (2.11)$$

Equation (2.10) is derived using the definition of *sensitivity of a utility function*. By symmetry, the inequality still holds when we change the role of $\mathcal{D}$ and $\mathcal{D}'$. $\square$

The guarantee on the accuracy of the exponential mechanism is given below.

Theorem 2.4 (Exponential Mechanism's Accuracy Guarantee [41]). *Let $OPT_u(\mathcal{D}) = \max_{x \in \mathcal{C}} u(x, \mathcal{D})$ be the maximum usefulness from the output of $\mathcal{M}$ on the dataset $\mathcal{D}$, and let $\mathcal{T}_{OPT} = \{x \in \mathcal{C} : u(x, \mathcal{D}) = OPT_u(\mathcal{D})\}$. Then, $\forall t > 0$,*

$$\Pr \left[u(\mathcal{M}(\mathcal{D}), \mathcal{D}) \leq OPT_u(\mathcal{D}) - \frac{2\Delta_u}{\epsilon} \left(\log \left(\frac{|\mathcal{C}|}{|\mathcal{T}_{OPT}|} \right) + t \right) \right] \leq \exp(-t)$$

Let's consider a possibility, in which the utility of the result from the exponential mechanism is much less than the maximum utility, namely a bad event. Then, Theorem 2.4 assures that the probability of the bad event will decay exponentially with the difference between the maximum utility and the utility of the result.

Compositional theorems: We usually use the divide-and-conquer strategy to construct a differentially private mechanism. We keep dividing the problem into smaller units until we can use the basic mechanisms (e.g., Laplace mechanism or exponential mechanism) to construct these units. This approach allows the creation of more complicated mechanisms from simpler ones. However, combining the smaller units together is likely required to query the same set of data many times. This action may cause leakage of information. For example, if we publish many values of the same Laplace mechanism, by the law of large numbers, the average of these values will converge to the true value and cause leakage of information. Hence, we want to know how much privacy budget be degraded when we combine many differentially private mechanisms together. This problem is addressed in the following theorem.

Theorem 2.5 (Compositional Theorem). *Let $\mathcal{M}_1$ be a (ϵ_1, δ_1) -differentially private mechanism and $\mathcal{M}_2$ be a (ϵ_2, δ_2) -differentially private mechanism. Then, the mechanism $\mathcal{M}$ such that $\mathcal{M}(\mathcal{D}) = (\mathcal{M}_1(\mathcal{D}), \mathcal{M}_2(\mathcal{D}))$ is a $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -differentially private mechanism.*

Proof. Here we only give the proof for the case $\delta_1 = \delta_2 = 0$. The proof for the general case is given in Appendix B of [41].

For any pair of neighboring datasets $\mathcal{D}, \mathcal{D}'$:

$$\begin{aligned} \frac{\Pr[\mathcal{M}(\mathcal{D}) = (a, b)]}{\Pr[\mathcal{M}(\mathcal{D}') = (a, b)]} &= \frac{\Pr[\mathcal{M}_1(\mathcal{D}) = a] \cdot \Pr[\mathcal{M}_2(\mathcal{D}) = b]}{\Pr[\mathcal{M}_1(\mathcal{D}') = a] \cdot \Pr[\mathcal{M}_2(\mathcal{D}') = b]} \\ &\leq \exp(\epsilon_1) \cdot \exp(\epsilon_2) = \exp(\epsilon_1 + \epsilon_2) \end{aligned}$$

Therefore, $\mathcal{M}(\mathcal{D})$ is a $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -differentially private mechanism. $\square$

Furthermore, if we can tolerate a little loss in the parameter δ , we can get a much slower degradation rate for ϵ .

Theorem 2.6 (Advanced Composition [41]). *Let $\mathcal{M}_i$ ($i = 1, \dots, k$) be k adaptive (ϵ, δ) -differentially private mechanisms where the latter mechanisms may depend on the values of previous mechanisms. Then, the mechanism $\mathcal{M} = (\mathcal{M}_1(\mathcal{D}), \dots, \mathcal{M}_k(\mathcal{D}))$ is $(\epsilon', k\delta + \delta')$ -differential privacy for any $\delta' > 0$ and*

$$\epsilon' = \epsilon \sqrt{2k \log(1/\delta')} + k\epsilon(\exp(\epsilon) - 1)$$

Recently, Kairouz [64] investigated the problem of privacy degradation and improved the above result to $\epsilon' = O\left(k\epsilon^2 + \epsilon\sqrt{k \log(e + (\epsilon\sqrt{k}/\delta'))}\right)$.

Other techniques: Abadi et al. [2] proposed a technique which adds Gaussian noise to the gradient in the stochastic gradient descent (SGD) algorithm. Papernot et al. [88] proposed a technique which creates a non-private dataset from an ensemble of private classifiers and uses this dataset to train another model

while guaranteeing privacy for individuals in the original dataset. These two techniques guarantee (ϵ, δ) –differential privacy.

2.3 Private Survival Analysis

Survival analysis studies the statistical properties of special events, namely *death events*. Let T be a random variable representing the period when a death event occurs. Two characteristic functions of T are defined below.

Definition 2.8 (Survival Function [68]). *The survival function $S(t)$ is defined as follows:*

$$S(t) = \Pr [T > t]$$

$S(t)$ is the probability that an object survives in the time interval $[0, t]$.

Definition 2.9 (Hazard Function [68]). *The hazard function $h(t)$ is defined as follows:*

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{\Pr [t \leq T < t + \Delta t \mid T \geq t]}{\Delta t}$$

The hazard function $h(t)$ can be interpreted as the risk of death at time t .

The relation between the survival function and the hazard function is given as follows:

$$h(t) = \frac{f(t)}{S(t)}$$

where $f(t)$ is the probability density function of random variable T . We also have:

$$S(t) = \exp (-H(t))$$

where $H(t) = \int_0^t h(u) du$ which is the *accumulative hazard function* over time.

Censored dataset: A dataset $\mathcal{D}$ is the collection of n tuples (x_i, t_i, d_i) ($i = 1, \dots, n$); where $x_i \in \mathbb{R}^k$ is k –dimensional covariate vector, t_i is the survival time, and $d_i \in \{0, 1\}$ is the death indicator. We assume that x_i is inside the unit ball: $\forall i : \|x_i\| \leq 1$

and survival time is scaled to a fixed interval $t_i \in [e^{-\Omega}, 1]$, where Ω is a positive constant. We consider a typical data censoring situation that some objects are still alive when the experiment ends. Therefore, we do not know the actual time a death event occurs, in these cases $d_i = 0$. Otherwise, if we know the actual time of a death event, then $d_i = 1$.

In survival analysis, we usually need to work on sensitive information, for example, the medical data of patients. Moreover, it is often illegal for research institutions and hospitals to distribute patients' medical data without an appropriate privacy preservation mechanism. Therefore, providing privacy preservation mechanism is usually a mandatory requirement when conducting survival analysis.

O'Keefe *et al.* [86] discussed many privacy problems in survival analysis. The results of survival analysis can potentially reveal confidential personal information such as name, address, and health care number. In their work, O'Keefe *et al.* proposed that the privacy of individuals in a survival dataset is protected if the results of a survival analysis are unlikely to lead to the identification of a person. They proposed many techniques to achieve the goal of privacy protection. For Kaplan-Meier estimation, they proposed the method of smoothing the resulting curve and adding a small amount of noise to the output. For regression model, they created random samples from the original survival dataset.

Yu *et al.* [102] proposed a privacy-preserving Cox's model for survival analysis in which the data is distributed over many data holders. The proposed approach preserves privacy by projecting the original data space into a lower dimensional space using a linear projection. However, there is no guarantee that the projection can protect the privacy of individuals. Fung *et al.* [50] proposed privacy-preserving data mining technique for the predictive models for lung cancer survival analysis. The proposed technique uses a random projection to protect privacy, but unable to prove the privacy guarantee mathematically. Chen *et al.* [26] proposed a model for privately comparing survival curves using

the log-rank test. However, the proposed method is unable to prove the privacy guarantee.

We observe that the above proposed approaches lack a strong privacy framework to ensure privacy protection. In this research, we will propose differentially private mechanisms for survival analysis which can rigorously control the disclosure risk.

2.4 Local Differential Privacy

Earlier models of differential privacy assume a trusted data curator, who collects and manages the exact private information of individuals, and releases statistics derived from the data under differential privacy requirements. However, in practice, users may not want to share private information with anyone, including the central data curator. Recently, much attention has been shifted to the local differential privacy model (LDP) [33, 63, 99, 60] which eliminates the data curator and the collection of exact private information. The formal definition of LDP is given as follows.

Definition 2.10 (ϵ -Local Differential Privacy [38, 36]). *A mechanism (function) $\mathcal{M}$ satisfies ϵ -differential privacy if and only if for any pair of input tuples t, t' and any event $\mathcal{S} \subset \text{range}(\mathcal{M})$:*

$$\Pr[\mathcal{M}(t) \in \mathcal{S}] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(t') \in \mathcal{S}]$$

In the LDP model, each user keeps her own input tuple t and applies mechanism $\mathcal{M}$ locally, and only the result $\mathcal{M}(t)$ is sent to the central data curator. Duchi *et al.* [33] investigated the notion of local differential privacy systematically, proposed the minimax framework for LDP based on information theory, proved the upper and lower error bounds of LDP-compliant methods, and analyzed the trade-off between privacy and accuracy. Kairouz *et al.* [62] showed that

a version of the randomized response is an optimal mechanism for frequency estimation on a single binary attribute. Kairouz *et al.* [63] investigated the problem with a categorical attribute with an arbitrary number of possible values. Two mechanisms, namely binary and randomized response mechanisms, were proposed and proved their optimality when the privacy budget is low and high, respectively. Bassily and Smith [14] proposed an asymptotically optimal solution for building succinct histograms over a large categorical domain under LDP.

Erlingsson *et al.* [45] proposed the Rappor framework, which is based on the randomized response mechanism for publishing a vector of binary values under LDP. They use this mechanism with a Bloom filter, which intuitively adds another level of protection and increases the difficulty for the adversary to infer private information. As a result, it also becomes more difficult to derive statistics from the collected data, and they propose a sophisticated solution for this purpose. A follow-up paper [47] extended Rappor to more complex statistics such as joint distributions and association testing, as well as categorical attributes that contain a large number of potential values, such as a user's home page.

We observe that the current approaches do not support common machine learning tasks under LDP on a large dataset. In this research, we will propose an approach which can apply LDP for empirical risk minimization models.

2.5 Summary

In this chapter, we have reviewed the early privacy preservation methods. We also introduce the definition of differential privacy and its basic private mechanisms such as the Laplace mechanism and exponential mechanism. We emphasize that differential privacy is a practical privacy preservation framework which enables the trade-off between privacy and utility of the published results. Besides, we also discuss the important composition theorems which can combine

---

private mechanisms together. After that, we review the related work on techniques used in survival analysis to protect privacy. However, these techniques are unable to formally prove the guarantee on privacy protection. Finally, we review the related work on the local differential privacy framework which is a specific setting of differential privacy in which each user keeps his private data and only publishes randomized information. However, the current methods do not support the training of machine learning models.

Chapter 3

Parametric and Nonparametric Survival Analysis

In this chapter, we propose differentially private mechanisms for parametric and nonparametric survival models. We first introduce the parametric and nonparametric estimations in survival analysis and then discuss privacy problems that we aim to address in this research. We consider two kinds of models: (i) nonparametric models and (ii) parametric models. For nonparametric models, we propose a differentially private mechanism which can release a histogram of the dataset privately. The private histogram is then used in nonparametric estimations as the replacement for the dataset. For parametric models, we propose two private mechanisms, one for exponential distribution and another one for Weibull distribution. The experimental results show that our proposed mechanisms have outperformed other methods on real datasets.

3.1 Survival Estimations

The goal of survival analysis is to estimate the survival function and the hazard function. There are two different approaches to estimate these two functions. In a nonparametric approach, we do not use any assumption on the underlying probability distribution of survival time. We estimate the pointwise values of the survival function and the hazard function directly. In a parametric approach,

we use an assumption on the family of the underlying distribution (e.g. Weibull distribution) of survival time. Therefore, instead of estimating the pointwise values of the functions, we estimate the parameters of the underlying distribution. The choice of which approach to use depends on the application and the data that the model needs to learn from.

3.1.1 Nonparametric Estimations

There are two popular nonparametric estimations in survival analysis. The Kaplan-Meier estimator [65, 92] is used to estimate the values of survival function at discrete points of time, and the Nelson-Aalen estimator [82, 27] is used to estimate the values of cumulative hazard function at discrete points of time. The definitions of these estimators are given below.

Definition 3.1 (Kaplan-Meier Estimator [65]). *The Kaplan-Meier estimator is defined as follows:*

$$\hat{S}(t) = \prod_{t_i \leq t} \frac{n_i - p_i}{n_i}$$

where n_i is the number of survivors at time t_i and p_i is the number of death events at time t_i .

Definition 3.2 (Nelson-Aalen Estimator [1]). *The Nelson-Aalen estimator is defined as follows:*

$$\hat{H}(t) = \sum_{t_i \leq t} \frac{d_i}{n_i}$$

where n_i is the number of survivors at time t_i . Also, t_i is the survival time and d_i is the death indicator as defined in Chapter 2.

One important property of the Kaplan-Meier estimator and the Nelson-Aalen estimator is that both depend only on counting the number of death events and survivors. This property will be useful when we come to develop a solution for the private versions of these estimators.

3.1.2 Parametric Estimations

In the parametric approach, the family of the underlying distribution of T is known in advance. Let's assume that the underlying distribution is specified by the parametric vector θ . Let t_i be the survival time, d_i be the death indicator, $h(t_i | x_i)$ be the hazard function and $S(t_i | x_i)$ be the survival function of the i -th data record. The generic form of the likelihood function used to estimate θ is given as follows:

$$\mathcal{L}(\theta) = \prod_{i=1}^n h(t_i | x_i)^{d_i} S(t_i | x_i) \quad (3.1)$$

Take log on both sides of Equation (3.1), we have the following log-likelihood function:

$$\ell(\theta) = \sum_{i=1}^n d_i \cdot \log h(t_i | x_i) + \log S(t_i | x_i) \quad (3.2)$$

In the traditional approach, we will choose the parameter θ such that the value of the log-likelihood function $\ell(\theta)$ is maximized. However, this property cannot guarantee privacy protection for individuals in the dataset. This leads to the need of defining precisely the privacy guarantee when publishing θ .

3.2 Problem Definitions

In this research, we address two different kinds of problems in survival analysis: parametric problems and nonparametric problems. In the following problem definitions, Problem 3.1 is a nonparametric problem, while Problem 3.2 and Problem 3.3 are parametric problems.

Problem 3.1: Given a dataset $\mathcal{D}$ and privacy budget ϵ , release the output of the Kaplan-Meier estimator and Nelson-Aalen estimator such that the output is ϵ -differentially private.

The goal of Problem 3.1 is to guarantee that the output of the Kaplan-Meier estimator and the Nelson-Aalen estimator will not leak confidential information of individuals in the survival dataset. To achieve the goal, we require the output of these estimators to be differentially private.

For parametric problems, the goal is to estimate parameters of the underlying distribution while guaranteeing that private information of individuals in the dataset is protected. In this research, we focus on two popular probability distributions used in survival analysis: exponential distribution and Weibull distribution [7, 97].

Exponential distribution: The probability density function of exponential distribution is $f(t; \lambda) = \lambda e^{-\lambda t}$, where λ is the parameter we want to estimate. Therefore, $h(t; \lambda) = \lambda$ and $S(t; \lambda) = e^{-\lambda t}$. Substitute these two functions into Equation (3.2), we get the log-likelihood function:

$$\ell(\lambda) = \log \lambda \sum_{i=1}^n d_i - \lambda \sum_{i=1}^n t_i \quad (3.3)$$

Set the first derivative equal to zero, we get:

$$\frac{d}{d\lambda} \ell(\hat{\lambda}) = 0 \iff \hat{\lambda} = \frac{\sum_{i=1}^n d_i}{\sum_{i=1}^n t_i} \quad (3.4)$$

The traditional solution for estimating parameter λ is given in Equation (3.4). However, returning the value of $\hat{\lambda}$ directly can potentially breach the privacy of individuals in the survival dataset. We formalize the privacy problem for $\hat{\lambda}$ in the problem statement below.

Problem 3.2: Given a dataset $\mathcal{D}$ and privacy budget ϵ , release the parameter λ based on Equation (3.4) and guarantee that the result satisfies ϵ -differential privacy.

Weibull distribution: In this case, T follows the Weibull distribution with the following cumulative distribution function:

$$cdf(t; \lambda, p) = 1 - \exp \left(- (t/\lambda)^p \right) \quad (t \geq 0)$$

where $p > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution. The Weibull distribution is used as the underlying distribution of T due to its effectiveness in representing survival processes. In fact, it is currently the most common model used for parametric survival analysis. The survival function is $S(t; \lambda, p) = \exp \left(- (t/\lambda)^p \right)$. The hazard function is $h(t; \lambda, p) = pt^{p-1}/\lambda^p$. λ and p are the two parameters which we want to estimate from the survival dataset. Substitute $S(t)$ and $h(t)$ into Equation (3.2), we get the log-likelihood function:

$$\ell(\lambda, p) = \log \prod_{i=1}^n h(t_i; \lambda, p)^{d_i} \cdot S(t_i; \lambda, p) \quad (3.5)$$

$$= \sum_{i=1}^n \{ d_i \cdot (\log p + (p-1) \cdot \log t_i - p \log \lambda) - t_i^p / \lambda^p \} \quad (3.6)$$

We can maximize $\ell(\lambda, p)$ by setting the partial derivatives with respect to λ and p equal to zero. We derive the following system of equations:

$$\frac{\sum_i t_i^p \log t_i}{\sum_i t_i^p} = \frac{1}{p} + \frac{\sum_i d_i \log t_i}{\sum_i d_i}, \quad (3.7)$$

$$\lambda^p = \frac{\sum_i t_i^p}{\sum_i d_i} \quad (3.8)$$

However, we cannot derive a closed-form solution for parameters p and λ . In the traditional approach, the solution of Equation (3.7) can be calculated by numerical methods (e.g. Newton-Raphson method). After that, we can find λ from the value of p and Equation (3.8). We formalize the privacy guarantee for p and λ in the problem statement below.

Algorithm 3.1 $\mathcal{A}_{\text{Priv-Hist}}$: Privately Releasing the Histogram of a Survival Dataset**Input:** Dataset $\mathcal{D}$, privacy budget ϵ , m , n **Output:** Vector $y \in \mathbb{R}^{m+n}$

- 1: Let $\mathcal{S}_0 = \{t_i \mid d_i = 0, i = 1 \dots |\mathcal{D}|\}$ and $\mathcal{S}_1 = \{t_i \mid d_i = 1, i = 1 \dots |\mathcal{D}|\}$
- 2: Compute histogram of $\mathcal{S}_0$: $(a_1, a_2, \dots, a_m) = \mathbf{histogram}(\mathcal{S}_0)$
- 3: Compute histogram of $\mathcal{S}_1$: $(b_1, b_2, \dots, b_n) = \mathbf{histogram}(\mathcal{S}_1)$
- 4: Concatenate two histograms $x = \mathcal{S}_0 \parallel \mathcal{S}_1 = (a_1, \dots, a_m, b_1, \dots, b_n)$
- 5: Draw a random vector $r \in \mathbb{R}^{m+n}$ such that $r_i \sim \text{Lap}(2/\epsilon), i = 1 \dots m+n$
- 6: Output $y = x + r$

Problem 3.3: Given a dataset $\mathcal{D}$ and privacy budget ϵ , release p and λ from Equation (3.7) and Equation (3.8), and guarantee that the result is ϵ -differentially private.

3.3 Private Nonparametric Estimations

In this section, we discuss our proposed solution to Problem 3.1. Inspecting the computations of the Kaplan-Meier estimator (KME) and the Nelson-Aalen estimator (NAE), we observe that they only need histograms to work with. This property leads to the idea of privately publishing two disjointed histograms: one for uncensored tuples ($d_i = 1$) and the other for censored tuples ($d_i = 0$). These two histograms together form an approximation of the original dataset. We can then perform the estimations based on the histograms instead of the dataset. The proposed mechanism is given in Algorithm 3.1.

We observe that the two histograms can be merged into a single histogram based on the combination of time t_i and death indicator d_i attributes. Lines 1-4 in Algorithm 3.1 compute the merged histogram x . At Lines 5-6 in Algorithm 3.1, we add a Laplace noise to x for guaranteeing that the output is differential privacy. Theorem 3.1 gives the formal proof of differential privacy protection.

Theorem 3.1. *Algorithm 3.1 is ϵ -differentially private.*

Algorithm 3.2 $\mathcal{A}_{\text{Priv-Exp}}$: Proposed Private Mechanism for Exponential Distribution

---

Input: Dataset $\mathcal{D}$, privacy budget ϵ

Output: Parameter λ

- 1: Compute $q_d = (\sum_{i=1}^n d_i) + \text{Lap}(2/\epsilon)$
  - 2: Compute $q_t = (\sum_{i=1}^n t_i) + \text{Lap}(2/\epsilon)$
  - 3: Output $\lambda = \frac{q_d}{q_t}$
-

Proof. As the merged histogram is *actually* a histogram based on the combination of time and death indicator, its global ℓ_1 -norm sensitivity is 2. Therefore, by property of Laplace mechanism, we conclude that Algorithm 3.1 is ϵ -differentially private. $\square$

A private histogram from Algorithm 3.1's output would allow us to use the traditional Kaplan-Meier estimator and Nelson-Aalen estimator without modification. The results of these estimators on the private histogram will satisfy differential privacy automatically because the differential privacy property does not depend on any post-processing computation.

3.4 Private Parametric Estimations for Exponential Distribution

In this section, we propose a solution for Problem 3.2 on estimating the parameter λ of exponential distribution privately. The proposed solution is given in Algorithm 3.2.

We construct the solution from an observation that λ can be computed from two independent quantities: $q_d = \sum_{i=1}^n d_i$ and $q_t = \sum_{i=1}^n t_i$. Therefore, we can release these two quantities independently. Lines 1-2 in Algorithm 3.2 compute the two quantities and add independent Laplace noise to the result; thereby guaranteeing the results are differential privacy. Line 3 in Algorithm 3.2 returns the approximated value of λ based on Equation (3.4). We can prove that the

result is ϵ -differentially private in Theorem 3.2 by applying the compositional theorem.

Theorem 3.2. *Algorithm 3.2 is ϵ -differentially private.*

Proof. The global sensitivity of $\sum_{i=1}^n d_i$ is 1 and the global sensitivity of $\sum_{i=1}^n t_i$ is bounded by 1. Therefore, q_d and q_t are $\epsilon/2$ -differentially private. Thus, Algorithm 3.2 is ϵ -differentially private by the compositional theorem. $\square$

3.5 Private Parametric Estimation for Weibull Distribution

In this section, we propose a solution for Problem 3.3. Problem 3.3 is much more complicated than Problem 3.2 because we cannot derive a closed-form solution. In the traditional approach, we use the Newton-Raphson method to compute the solution of Equation (3.7). However, this approach cannot guarantee that the output is differentially private.

Our approach consists of two steps. In the first step, we propose Local Sensitivity Mechanism for p (LSP), a private mechanism which allows us to estimate the value of p as the root of Equation (3.7). In the second step, we propose Two Laplace Mechanism for λ (TLL), a private mechanism which uses the value of p from LSP to estimate λ from Equation (3.8). For each step, we will spend $\epsilon/2$ privacy budget. By the composability property of differential privacy, it is guaranteed that the pair (λ, p) to be published is ϵ -differentially private.

3.5.1 Local Sensitivity Mechanism for Estimating p (LSP)

The proposed mechanism for estimating the shape parameter p , LSP, is based on our experimental observation: for small-sized datasets, the estimated value of p from Equation (3.7) can fluctuate a lot when we switch from one dataset to its

Figure 3.1: An example of the utility function $\mathcal{U}(p, \mathcal{D})$. The utility function has 6 rungs. The root of Equation (3.7) on dataset $\mathcal{D}$ is $p(\mathcal{D})$, which is located at rung 0.

adjacent datasets. However, for large-sized and medium-sized datasets, the fluctuation is very little. This observation suggests that techniques which are based on additive noise calibrated with global sensitivity, i.e., Laplace mechanism, are not applicable. Instead, our proposed LSP mechanism calibrates the noise with local sensitivity of p . Local sensitivity can be understood as the fluctuation of the output when we switch from the *actual dataset* $\mathcal{D}$ to its adjacent datasets. It is different from the global sensitivity which considers any pair of adjacent datasets. Local sensitivity is first proposed by Kobbi *et al.* [84] and later used by Jun *et al.* [103] on privately counting sub-graphs.

The LSP mechanism is constructed based on the exponential mechanism. It defines a utility function $\mathcal{U}(p, \mathcal{D})$ with the shape of a ladder as illustrated in Figure 3.1. Each rung of the ladder is defined as the set of points where the root of Equation (3.7) is placed when we replace the actual dataset $\mathcal{D}$ by a hypothetical dataset $\mathcal{D}^{(k)}$ such that $d(\mathcal{D}, \mathcal{D}^{(k)}) \leq -k$, where k is the rung's level. For example, at rung level 0 we get the exact value of p from Equation (3.7) on dataset $\mathcal{D}$; while at rung level -1 , the root of Equation (3.7) on any adjacent dataset of the dataset $\mathcal{D}$ is placed in the segment bounded by two extremal points of p at which the values of the utility function $\mathcal{U}(p, \mathcal{D})$ are equal to -1 .

However, it is very hard to compute exactly the extremal points at each rung of the utility function. Instead, we propose approximated intervals such that the interval at each rung will contain extremal points. The mathematical definition of these intervals is given below.

Definition 3.3 (Local Sensitivity Interval (LSI) at distance k). *The interval $\mathcal{S}^{(k)}(\mathcal{D}) = [l^{(k)}(\mathcal{D}), u^{(k)}(\mathcal{D})]$ is the local sensitivity interval at distance k of parameter p if and only if for any dataset $\mathcal{D}^{(k)}$ such that $d(\mathcal{D}, \mathcal{D}^{(k)}) \leq k$, then $p(\mathcal{D}^{(k)}) \in \mathcal{S}^{(k)}(\mathcal{D})$, where $p(\mathcal{D}^{(k)})$ denotes the root of Equation (3.7) on dataset $\mathcal{D}^{(k)}$.*

By constructing the utility function $\mathcal{U}(p, \mathcal{D})$ as a ladder of rungs created by intervals $[l^{(k)}(\mathcal{D}), u^{(k)}(\mathcal{D})]$, we will prove in Section 3.5.3 that the sensitivity of $\mathcal{U}(p, \mathcal{D})$ is equal to 1. This will consequently prove that our mechanism is differentially private. However, we need a method to compute LSI first.

Computing the LSI at distance k : For computing LSI, we introduce the boundary functions which are the upper bounds (resp., lower bounds) of the left hand side, LHS, (resp., right hand side, RHS) of Equation (3.7). We pick these boundary functions to guarantee that the LHS and the RHS of Equation (3.7) are always bounded by the boundary functions when we apply Equation (3.7) to any hypothetical dataset $\mathcal{D}^{(k)}$ at distance k to the actual dataset $\mathcal{D}$. Consequently, this property will guarantee that the root of Equation (3.7) on $\mathcal{D}^{(k)}$ is also bounded by the intersections of these boundary functions. Let $f(p, \mathcal{D}^{(k)})$ and $g(p, \mathcal{D}^{(k)})$ be the LHS and the RHS of Equation (3.7) on dataset $\mathcal{D}^{(k)}$, e be the base of the natural logarithm, and t_i^* be the i -th smallest survival time in the dataset. We can bound

Algorithm 3.3 $\mathcal{A}_{\text{LSI}}$: Computing Local Sensitivity Intervals**Input:** Dataset $\mathcal{D}$, number of rungs K **Output:** Lower bounds l , upper bounds u 1: $l^{(0)} = u^{(0)} = \text{root of Equation (3.7)}$ 2: **for** $i = 1 \dots K$ **do**3: $l^{(i)} = \text{root of equation } f_U^k(p) = g_L^k(p)$ 4: $u^{(i)} = \text{root of equation } f_L^k(p) = g_U^k(p)$ 5: **end for**6: $l^{(K+1)} = 0, u^{(K+1)} = \gamma$ 7: Output l, u

$f(p, \mathcal{D}^{(k)})$ and $g(p, \mathcal{D}^{(k)})$ by four families of functions:

$$f_L^k(p, \mathcal{D}) = \frac{\sum_i t_i^p \log t_i - \frac{k}{ep}}{\sum_{i=1}^{n-k} (t_i^*)^p}$$

$$f_U^k(p, \mathcal{D}) = \frac{\sum_i t_i^p \log t_i + \frac{k}{ep}}{\sum_{i=1}^n t_i^p + k}$$

$$g_L^k(p, \mathcal{D}) = \frac{1}{p} + \frac{\sum_i d_i \log t_i - k\Omega}{\sum_i d_i - k}$$

$$g_U^k(p, \mathcal{D}) = \frac{1}{p} + \frac{\sum_i d_i \log t_i + k\Omega}{\sum_i d_i + k}$$

The function $f_L^k(p, \mathcal{D})$ (resp., $f_U^k(p, \mathcal{D})$) is chosen to be the lower bound (resp., upper bound) of $f(p, \mathcal{D}^{(k)})$. These functions are derived from the fact that the minimum value of the function $h(t) = t^p \log t$ for a fixed p and $t \in [\exp(-\Omega), 1]$ is $-\frac{1}{ep}$. The function $g_L^k(p, \mathcal{D})$ (resp., $g_U^k(p, \mathcal{D})$) is the lower bound (resp., upper bound) of $g(p, \mathcal{D}^{(k)})$. This is from the fact that the minimum value of $k(t) = \log t$ for $t \in [\exp(-\Omega), 1]$ is $-\Omega$. Our observation is that the estimation of p on $\mathcal{D}^{(k)}$, referred as p_C , is bounded by an interval $[p_A, p_B]$, where p_A is the root of equation $f_U^k(p, \mathcal{D}) = g_L^k(p, \mathcal{D})$ and p_B is the root of equation $f_L^k(p, \mathcal{D}) = g_U^k(p, \mathcal{D})$. This observation is illustrated in Figure 3.2. The p -coordinate of point C is bounded by the p -coordinate of point A and the p -coordinate of point B where point A is the intersection between $f_U^k(p, \mathcal{D})$ and $g_L^k(p, \mathcal{D})$; and point B is the intersection between $f_L^k(p, \mathcal{D})$ and $g_U^k(p, \mathcal{D})$.

Algorithm 3.3 computes the lower bounds and upper bounds of p . At Line 1,

Figure 3.2: The estimated value of p on dataset $\mathcal{D}^{(k)}$ is determined by the intersection C of two curves $f(p, \mathcal{D}^{(k)})$ and $g(p, \mathcal{D}^{(k)})$.

we compute the exact solution of Equation (3.7) on dataset $\mathcal{D}$ and use this value as the first lower bound and upper bound. The **for** loop at Lines 2-5 gives the K rungs defined by the pairs of lower bounds $l^{(i)}$ and upper bounds $u^{(i)}$, where K is a hyperparameter of the solution which will be decided based on the trade-off between accuracy and runtime. In detail, we use the Newton-Raphson method to compute the roots of equations at Lines 3-4. The *floor rung* is given at Line 6 to guarantee that all values of p are contained in the interval $[0, \gamma]$, where the hyperparameter γ will be decided based on the nature of the survival analysis problem. Line 7 returns all the lower bounds and upper bounds of LSIs in two lists l and u .

Publishing the value of p by exponential mechanism: LSP uses the exponential mechanism as the framework to estimate the value of p . We define the following utility function which is based on LSI $\mathcal{S}^{(k)}(\mathcal{D})$ computed from the previous step.

Algorithm 3.4 $\mathcal{A}_{\text{LSP}}$: Local Sensitivity Mechanism for p (LSP)

-
- 1: **Input:** Dataset $\mathcal{D}$, number of rungs K , privacy budget $\epsilon/2$, lower bounds l , upper bounds u
  - 2: **Output:** p
  - 3: **for** $i = 1 \dots K + 1$ **do**
  - 4: $level^{(i)} = [l^{(i)}, l^{(i-1)}) \cup (u^{(i-1)}, u^{(i)}]$
  - 5: $length^{(i)} = u^{(i)} - u^{(i-1)} + l^{(i-1)} - l^{(i)}$
  - 6: $weight^{(i)} = length^{(i)} \cdot \exp(-i\frac{\epsilon}{4})$
  - 7: **end for**
  - 8: Sampling a level from distribution: $\Pr[level^{(i)}] = \frac{weight^{(i)}}{\sum(weight)}$
  - 9: Output a value of p sampling from uniform distribution over the chosen level at Line 6
-

Definition 3.4 (Ladder-shaped utility function). *The ladder-shaped utility function $\mathcal{U}(p, \mathcal{D})$ is defined as:*

$$\mathcal{U}(p, \mathcal{D}) = -k \iff p \in [l^{(k)}(\mathcal{D}), l^{(k-1)}(\mathcal{D})) \cup (u^{(k-1)}(\mathcal{D}), u^{(k)}(\mathcal{D})] \quad (3.9)$$

where $k = 1 \dots (K + 1)$.

From the definition of the utility function, we propose the LSP mechanism (Algorithm 3.4) for estimating the value of p . Algorithm 3.4 draws a randomized value of p from a probability density function which is constructed from the ladder-shaped utility function. Line 2 in Algorithm 3.4 specifies $level^{(i)}$ as the union of two intervals in the i -th rung of the utility function. The values of p in $level^{(i)}$ share the same probability density $\exp(-i\epsilon/4)$. The probability at level i is accumulated in $weight^{(i)}$ at Line 4. Line 6 draws a random rung from its weights and Line 7 draws a randomly uniform value of p from the chosen rung at Line 6. In Section 3.5.3, we will prove that the LSP mechanism is $\epsilon/2$ -differentially private.

Algorithm 3.5 $\mathcal{A}_{\text{TLL}}$: Two Laplace Mechanism for λ (TLL)

-
- 1: **Input:** Dataset $\mathcal{D}$, privacy budget $\epsilon/2$, p
  - 2: **Output:** λ
  - 3: Compute $\delta = \sum_{i=1}^n d_i + \text{Laplace}(4/\epsilon)$
  - 4: Compute $\tau = \sum_{i=1}^n t_i^p + \text{Laplace}(4/\epsilon)$
  - 5: Output $\lambda = \sqrt[p]{\tau/\delta}$
-

3.5.2 Two Laplace Mechanism for Estimating λ (TLL)

In order to estimate the scale parameter λ , we propose Two Laplace Mechanism for λ (TLL), which is based on the Laplace mechanism. We observe that for small datasets, the value of $\lambda = \sqrt[p]{\frac{\sum_i t_i^p}{\sum_i d_i}}$ can vary a lot between adjacent datasets. Therefore, we cannot apply the Laplace mechanism to λ directly because its global sensitivity is too large. Since the global sensitivities of $\sum_i d_i$ and $\sum_i t_i^p$ are equal to 1, instead of adding noise to λ directly, our proposed TLL mechanism spends $\epsilon/4$ privacy budget to publish $\sum_i d_i$ and another $\epsilon/4$ privacy budget to publish $\sum_i t_i^p$. By the composability property of differential privacy, TLL is $\epsilon/2$ -differentially private. The formal proof is given in Section 3.5.3.

Algorithm 3.5 gives the proposed TLL mechanism. Lines 1-2 in Algorithm 3.5 compute the two quantities δ and τ by adding independent Laplacian noises to $\sum_i d_i$ and $\sum_i t_i^p$. Line 3 in Algorithm 3.5 returns the value of λ as derived from Equation (3.8).

3.5.3 Proof of Privacy Protection

In order to prove the differentially private protection of the LSP mechanism, we need to introduce a constraint on the LSI $\mathcal{S}^{(k)}(\mathcal{D})$. We will use this constraint later to prove that the sensitivity of the utility function is equal to 1.

Definition 3.5 (Ladder constraint). *Local sensitivity intervals $\mathcal{S}^{(k)}(\mathcal{D})$ are **ladder local sensitivity intervals** if and only if for any pair of adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$,*

the following constraint holds:

$$\mathcal{S}^{(k-1)}(\mathcal{D}) \subset \mathcal{S}^{(k)}(\mathcal{D}) \subset \mathcal{S}^{(k+1)}(\mathcal{D}')$$

We now prove that the output of the first step in our solution satisfies the ladder constraint.

Lemma 3.1. *For any pair of adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, we have:*

- (a) $g_L^{k+1}(p, \mathcal{D}') \leq g_L^k(p, \mathcal{D})$ and $g_U^k(p, \mathcal{D}) \leq g_U^{k+1}(p, \mathcal{D}')$
- (b) $f_L^k(p, \mathcal{D}) \geq f_L^{k+1}(p, \mathcal{D}')$ and $f_U^k(p, \mathcal{D}) \leq f_U^{k+1}(p, \mathcal{D}')$

Proof. (a) We have $g_L^k(p, \mathcal{D}) - g_L^{k+1}(p, \mathcal{D}') \geq 0$ and $g_U^k(p, \mathcal{D}) - g_U^{k+1}(p, \mathcal{D}') \leq 0$. Therefore, $g_L^{k+1}(p, \mathcal{D}') \leq g_L^k(p, \mathcal{D})$ and $g_U^k(p, \mathcal{D}) \leq g_U^{k+1}(p, \mathcal{D}')$.

(b) We have $f_L^k(p, \mathcal{D}) - f_L^{k+1}(p, \mathcal{D}') \geq 0$ and $f_U^k(p, \mathcal{D}) - f_U^{k+1}(p, \mathcal{D}') \leq 0$. Therefore, $f_L^k(p, \mathcal{D}) \geq f_L^{k+1}(p, \mathcal{D}')$ and $f_U^k(p, \mathcal{D}) \leq f_U^{k+1}(p, \mathcal{D}')$. $\square$

Theorem 3.3. *Let $\mathcal{S}^k(\mathcal{D}) = [l^{(k)}(\mathcal{D}), u^{(k)}(\mathcal{D})]$, then*

- (a) $\mathcal{S}^{(k-1)}(\mathcal{D}) \subset \mathcal{S}^{(k)}(\mathcal{D})$
- (b) $\mathcal{S}^{(k)}(\mathcal{D}) \subset \mathcal{S}^{(k+1)}(\mathcal{D}')$

Proof. (a) We have $f_U^{k-1}(p, \mathcal{D}) \leq f_U^k(p, \mathcal{D})$ and $g_L^{k-1}(p, \mathcal{D}) \geq g_L^k(p, \mathcal{D})$, and $g_L^{k-1}(p, \mathcal{D})$ and $g_L^k(p, \mathcal{D})$ are non-increasing functions. Thus, we have $l^{(k-1)}(\mathcal{D}) \geq l^{(k)}(\mathcal{D})$. We also have $f_L^{k-1}(p, \mathcal{D}) \leq f_L^k(p, \mathcal{D})$ and $g_U^{k-1}(p, \mathcal{D}) \geq g_U^k(p, \mathcal{D})$, and $g_U^{k-1}(p, \mathcal{D})$ and $g_U^k(p, \mathcal{D})$ are non-increasing functions. Thus, we have $u^{(k-1)}(\mathcal{D}) \leq u^{(k)}(\mathcal{D})$. Therefore, $\mathcal{S}^{(k-1)}(\mathcal{D}) \subset \mathcal{S}^{(k)}(\mathcal{D})$.

(b) By Lemma 3.1 and similar arguments as (a), we have $l^{(k)}(\mathcal{D}) \geq l^{(k+1)}(\mathcal{D}')$ and $u^{(k)}(\mathcal{D}) \leq u^{(k+1)}(\mathcal{D}')$. Therefore, $\mathcal{S}^k(\mathcal{D}) \subset \mathcal{S}^{(k+1)}(\mathcal{D}')$. $\square$

Lemma 3.2. *If $p \in \mathcal{S}^{(k)}(\mathcal{D})$, then $-\mathcal{U}(p, \mathcal{D}) \leq k$.*

Proof. By the definition of $\mathcal{U}(\cdot)$, if $-\mathcal{U}(p, \mathcal{D}) = h$, then $p \in \mathcal{S}^{(h)}(\mathcal{D})$ and $p \notin \mathcal{S}^{(h-1)}(\mathcal{D})$. By the ladder constraint, we have $\mathcal{S}^{(i)}(\mathcal{D}) \subset \mathcal{S}^{(h-1)}(\mathcal{D})$ for any $i < h - 1$. Therefore, if $p \notin \mathcal{S}^{(h-1)}(\mathcal{D})$, then $p \notin \mathcal{S}^{(i)}(\mathcal{D})$ for any $i \leq h - 1$. In other

words, if $-\mathcal{U}(p, \mathcal{D}) = h$, then $p \notin \mathcal{S}^{(i)}(\mathcal{D})$ for any $i \leq h - 1$. Now assuming $h > k$, from the above argument it implies that $p \notin \mathcal{S}^{(k)}$. This contradicts the assumption of the lemma. Therefore, the lemma is proven by contradiction. $\square$

Lemma 3.3. *The sensitivity of utility function $\mathcal{U}(p, \mathcal{D})$ is 1.*

$$\max_{p, \mathcal{D}, \mathcal{D}': d(\mathcal{D}, \mathcal{D}')=1} |\mathcal{U}(p, \mathcal{D}) - \mathcal{U}(p, \mathcal{D}')| \leq 1$$

Proof. For simplicity, let k be the short form for $-\mathcal{U}(p, \mathcal{D})$ and h be the short form for $-\mathcal{U}(p, \mathcal{D}')$. To prove the lemma, we need to prove that $k - 1 \leq h \leq k + 1$. Assuming that $h < k - 1$. We have $p \in \mathcal{S}^{(h)}(\mathcal{D}')$. By the ladder constraint, we have $\mathcal{S}^{(h)}(\mathcal{D}') \subset \mathcal{S}^{(h+1)}(\mathcal{D}) \subset \mathcal{S}^{(k-1)}(\mathcal{D})$. So $p \in \mathcal{S}^{(k-1)}(\mathcal{D})$. Lemma 3.2 implies $-\mathcal{U}(p, \mathcal{D}) \leq k - 1$ and this contradicts with the assumption that $-\mathcal{U}(p, \mathcal{D}) = k$. Hence, $h \geq k + 1$. By the ladder constraint, we also have $p \in \mathcal{S}^{(k)}(\mathcal{D}) \subset \mathcal{S}^{(k+1)}(\mathcal{D}')$. Finally, from Lemma 3.2, we have $h \leq k + 1$. Therefore, the lemma is proven. $\square$

Theorem 3.4. *LSP is $\epsilon/2$ -differentially private.*

Proof. Lemma 3.3 claims that the sensitivity of the utility function $\mathcal{U}(\cdot)$ is equal to 1. Hence, LSP is $\epsilon/2$ -differentially private by the property of the exponential mechanism. $\square$

Theorem 3.5. *TLL is $\epsilon/2$ -differentially private.*

Proof. The global sensitivity of $\sum_i d_i$ is equal to 1 because $d_i \in \{0, 1\}$. Therefore, the value δ in Algorithm 3.5 is $\epsilon/4$ -differentially private. We have $t_i \in [0, 1]$, hence $t_i^p \in [0, 1]$ for $p > 0$. Therefore, the global sensitivity of $\sum_i t_i^p$ is equal to 1. Consequently, the value τ in Algorithm 3.5 is $\epsilon/4$ -differentially private. By the composability property of differential privacy, $\lambda = \sqrt[p]{\tau/\delta}$ is $\epsilon/2$ -differentially private. $\square$

3.6 Experimental Evaluation

In this section, we present the experimental results on real datasets. We have designed our experiments for answering the following research questions:

- What is the performance of the proposed private nonparametric mechanism as compared to non-private estimations?
- What is the performance of the proposed private mechanisms in estimating parameters of exponential and Weibull distributions as compared to other methods?

3.6.1 Datasets

We use four real datasets in the experiments. Table 3.1 gives a summary of these datasets.

- **The FL chain dataset (FL)** - It is obtained from a study on the association of the serum free light chain with higher death rates [69, 30]. The survival time of a patient is measured in days from enrollment until death. The uncensored cases are patients who were dead at the last contact.
- **The time-to-second-birth (SB) and time-to-third-birth (TB) datasets** - They are obtained from The Medical Birth Registry of Norway [59]. The survival time is the time between the first and second births, and between the second and third births respectively. The uncensored cases are women who have the second birth, and the third birth respectively, at the time the data are collected.
- **The Wichert dataset (WT)** - It contains records on unemployment duration of people in Germany [101]. The survival time is the duration of unemployment until having a job again. The uncensored cases are the ones who have a new job at the time the data are collected.

Table 3.1: Datasets used in the experiments.

Dataset	Size	#uncensored
FL	7874	2169
TB	16116	1761
WT	21685	18615
SB	53558	16341

Moreover, we normalize all the survival times into the interval $[\exp(-6), 1]$.

3.6.2 Nonparametric Estimations

In this set of experiments, we consider the trade-off between privacy and accuracy in estimating survival probability and cumulative hazard rate using nonparametric estimators, i.e. Kaplan-Meier estimator and Nelson-Aalen estimator. We use Algorithm 3.1 to publish a private histogram of the dataset. The private histogram has 20 bins in total. Then, we use the private histogram as input for both Kaplan-Meier estimator and Nelson-Aalen estimator to estimate the survival probabilities and cumulative hazard rates.

Baseline: We compare our private estimations to non-private estimations from Nelson-Aalen estimator and Kaplan-Meier estimator.

Evaluation metric: We use *median absolute error* (MdAE) to measure the distance between our estimated survival probabilities (and cumulative hazard rates) to the non-private values obtained from applying the Kaplan-Meier estimator (and Nelson-Aalen estimator, respectively) on the dataset without privacy protection. MdAE is defined as:

$$\text{MdAE} = \text{median}(\{\|x_i - x^*\|_1, i = 1 \dots t\}), \quad (3.10)$$

where x_i is an real-valued vector estimated from the private histogram, x^* is the vector obtained from the non-private estimation and t is the number of tries. As

Figure 3.3: Experimental results (in MdAE) for nonparametric estimators.

MdAE is the distance between a private result and a non-private result, a non-private model has $\text{MdAE} = 0$. Therefore, the MdAE score can be interpreted as the performance gap between private models and non-private models.

Experimental results: As shown in Figure 3.3, all the MdAEs are consistently decreased when the privacy budget is increased. At high privacy budgets, the private estimations approximate the non-private estimations very well. In addition, we use the same histogram output from Algorithm 3.1 for both Nelson-Aalen estimator and Kaplan-Meier estimator without spending the privacy budget twice. In other words, Algorithm 3.1 saves half of the privacy budget.

3.6.3 Parametric Estimations for Exponential Distribution

In this set of experiments, we consider the privacy trade-off in estimating the parameter λ in parametric models with exponential distribution. Our proposed private mechanism uses Algorithm 3.2 to estimate λ from the dataset.

Baselines: We compare our proposed private mechanism (Algorithm 3.2) with the Laplace mechanism and the *subsample-and-aggregate* (SAA) mechanism [95, 43].

- The Laplace mechanism - It adds Laplacian noise to the estimated parameters independently.

Figure 3.4: Experimental results (in MdAE) for parametric estimations of exponential distribution.

- The SAA mechanism - It divides the dataset into partitions of random non-overlap subsets and uses a non-private approach to estimate the parameters on these subsets. The estimated parameters are then aggregated together to derive the estimation of the whole dataset. The aggregated results are published by applying the Laplace mechanism. Please refer to [95] for the details of the SAA mechanism.

Evaluation metric: We use the median absolute error (MdAE) for performance measurement.

Experimental results: Figure 3.4 shows the performance results of our proposed mechanism (labeled as Our mechanism) in comparison with the Laplace mechanism and subsample-and-aggregate mechanism (SAA). It shows that our mechanism clearly outperforms the other two mechanisms. The estimations from our proposed mechanism are very close to the non-private estimations even at low

privacy budgets. At $\epsilon = 0.1$, our mechanism can estimate λ with the worst absolute error of 10^{-2} on dataset WT. This is nearly 3 orders of magnitudes better than the SAA mechanism.

3.6.4 Parametric Estimations for Weibull Distribution

In this set of experiments, we consider the performance of our proposed mechanisms for parametric models with Weibull distribution. We use the mechanism LSP (Algorithm 5.2) with half of the privacy budget $\epsilon/2$ to estimate parameter p and use the mechanism TLL (Algorithm 5.3) with half of the privacy budget $\epsilon/2$ to estimate parameter λ .

Baselines: We compare our proposed mechanisms with the Laplace mechanism and the *subsample-and-aggregate* (SAA) mechanism.

Evaluation metric: We use the median absolute error (MdAE) for performance measurement.

Hyperparameter settings: For the LSP mechanism (Algorithm 5.2), we set the number of rungs $K = 500$. For the SAA mechanism, we set the number of subsets such that each subset has 500 records on average. For all mechanisms, we set the upper bound 10 for the values of p and λ .

Performance results on estimating the shape parameter p : Figure 3.5 shows the performance results of the Laplace mechanism, the SAA mechanism and our proposed LSP mechanism on estimating the parameter p . The Laplace mechanism performs quite consistently throughout the datasets. The SAA mechanism tends to have better performance on bigger datasets. Meanwhile, the performance of SAA on SB is also about 6 times better than the performance on FL. LSP has the best performance with useful output in datasets, even at very small privacy budget, i.e., 0.05. However, the performance of the LSP mechanism depends on the number of uncensored records more than the actual size of the datasets.

Figure 3.5: Performance results (in MdAE) on estimating the value of the shape parameter p .

Performance results on estimating the scale parameter λ : Figure 3.6 shows the performance results of the Laplace mechanism, the SAA mechanism and our proposed TLL mechanism on estimating the parameter λ . The Laplace mechanism has consistent performance throughout all datasets. The SAA mechanism has improved performance for bigger datasets. Our proposed TLL mechanism outperforms both SAA and Laplace mechanisms on all datasets and privacy budgets. For the smallest performance gap on dataset FL, the TLL mechanism is still 33 times better than the SAA mechanism and 464 times better than the Laplace mechanism.

Figure 3.6: Performance results (in MdAE) on estimating the value of the scale parameter λ .

3.7 Summary

In this chapter, we have proposed differentially private mechanisms for both nonparametric and parametric survival analysis. Our proposed mechanism for nonparametric models allows the data analyst to do estimations (e.g., Kaplan-Meier estimation and Nelson-Aalen estimation) in the traditional ways while guaranteeing privacy protection. For parametric models, we have proposed a mechanism to estimate the parameter of exponential distribution privately. We have also proposed mechanisms for estimating the parameters of Weibull distribution privately. In our experiments, we compare our mechanisms with the state-of-the-art methods on real datasets, and the results have shown that our proposed mechanisms outperform the state-of-the-art methods.

Chapter 4

Bayesian Parametric Survival Analysis

In this chapter, we propose a novel framework for the data privacy problem of estimating the uncertainty with Bayesian parametric models and flexible parametric models. Our proposed Sampling-A-Posterior (SAP) framework publishes a noisy *posterior distribution* which is guaranteed to be differentially private. This is different from the mechanisms in the previous chapter which only publish a point estimate of the model. Posterior distribution is particularly useful for applications which concern about the uncertainty of model predictions, such as medical applications. In the proposed SAP framework, we represent the likelihood function as a linear combination of basis functions and use the K -norm mechanism to publish the coordinate vector of the total likelihood function. The proposed framework can be applied to design differentially private mechanisms for parametric models with Weibull distribution and flexible parametric models in survival analysis. The experimental results show the effectiveness of the proposed framework on real survival datasets.

4.1 Bayesian Inference Framework

We propose to use Bayesian inference to estimate the uncertainty in parametric survival analysis. From a Bayesian inference's perspective, we would like to

know the posterior distribution after observing the dataset modeled by the likelihood function. With the knowledge of the posterior distribution, we are not only able to select a parameter which minimizes a particular loss function, but also able to estimate the uncertainty of the estimated parameter.

We first introduce a generic Bayesian inference framework for parametric and flexible parametric models. The task of statistical modeling is to specify the joint distribution $p(\mathcal{D}, \theta)$ of observed dataset $\mathcal{D}$ and model's parameter θ . Bayesian inference implements this by defining a likelihood function $p(\mathcal{D} \mid \theta)$ and a prior distribution $\pi(\cdot)$. By chain rule, we have:

$$p(\mathcal{D}, \theta) = p(\mathcal{D} \mid \theta) \times \pi(\theta)$$

The likelihood models the relationship between the model's parameter and the observed data records, while the prior distribution is used to specify domain knowledge and regularization for the model. Let $\ell(\theta; \mathcal{D}_i)$ be the log-likelihood function of the i -th data record in the dataset $\mathcal{D}$. Let θ be the parameter of the survival model. Let $\pi(\theta)$ be the prior distribution of θ . By Bayes' theorem, the posterior distribution of θ given the dataset $\mathcal{D}$ is as follows:

$$\sigma(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D}, \theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \theta) \times \pi(\theta)}{p(\mathcal{D})} \quad (4.1)$$

$$\propto \exp \left(\sum_{i=1}^n \ell(\theta; \mathcal{D}_i) \right) \times \pi(\theta) \quad (4.2)$$

where n is the size of dataset $\mathcal{D}$. Considering the survival dataset with right censoring of time values, we have $\mathcal{D}_i = (t_i, d_i)$, $1 \leq i \leq n$, where t_i is the observed time and d_i is the death indicator. $d_i = 0$ indicates that the death time is censored and greater than t_i , while $d_i = 1$ indicates that the death time is equal to t_i . From

that, the log-likelihood function of survival model has a generic form as follows:

$$\begin{aligned} \ell(\theta; \mathcal{D}_i) = & d_i \cdot (\log(h(t_i | \theta)) + \log(S(t_i | \theta))) \\ & + (1 - d_i) \cdot \log(S(t_i | \theta)) \end{aligned} \quad (4.3)$$

4.2 Problem Definitions

We now define formally the problems we aim to address in this chapter.

Problem 4.1: Given a dataset $\mathcal{D}$ and privacy budget ϵ , release a posterior function of a parametric survival model with Weibull distribution such that the released function is ϵ -differentially private.

Solving Problem 4.1 gives us a posterior function of the parametric model, which can be used to estimate the uncertainty of the model.

Problem 4.2: Given a dataset $\mathcal{D}$ and privacy budget ϵ , release a posterior function of a flexible parametric survival model such that the released function is ϵ -differentially private.

The above problem addresses the limitations of Weibull distribution in situations where it is not applicable. Solving this problem allows us to estimate the uncertainty of a parametric model which has much more expressive power in capturing the underlying distribution of survival data.

4.3 The Sampling-A-Posterior Framework

In this section, we propose a novel approach which aims to promise differential privacy for the whole posterior function instead of a single sample as in traditional approaches.

To achieve this, we publish a noisy version of the posterior distribution privately instead of the traditional approach of publishing a private point estimate. We propose the Sampling-A-Posterior (SAP) framework for publishing a noisy

posterior function which guarantees differential privacy. By representing the likelihood function as a linear combination of basis functions, SAP uses the K -norm mechanism to publish the coordinate vector of the total likelihood function relative to the set of basis functions. From the private coordinate vector, we are able to construct the private posterior function. After that, we use an MCMC sampler [23] to draw random samples from the posterior distribution. To the best of our knowledge, this is the first work which is able to guarantee differential privacy for the posterior of survival models.

Our proposed Sampling-A-Posterior (SAP) framework includes three logical steps to guarantee differential privacy for Bayesian inference. They are described below.

4.3.1 Approximating Log-likelihood Function

As likelihood functions are infinite dimensional objects, we first need to represent or approximate it by a finite space. Let's assume that we can derive a functional basis of k functions as follows:

$$B = [b_1(\theta), b_2(\theta), \dots, b_k(\theta)]$$

Then, we can represent the log-likelihood function of a data record as a linear combination of the basis functions as:

$$\ell_i = \ell(\theta; \mathcal{D}_i) = \alpha_1(\mathcal{D}_i)b_1(\theta) + \dots + \alpha_k(\mathcal{D}_i)b_k(\theta) = \langle B, [\ell_i]_B \rangle$$

where $[\ell_i]_B = [\alpha_1(\mathcal{D}_i), \alpha_2(\mathcal{D}_i), \dots, \alpha_k(\mathcal{D}_i)]$ is the coordinate vector of $\ell(\theta; \mathcal{D}_i)$ relative to basis B and $\langle \cdot, \cdot \rangle$ is the inner product operator. As a result, we can

Algorithm 4.1 $\mathcal{A}_{K\text{-Norm}}$: The K-norm Mechanism [54]**Input:** Convex hull K , privacy budget ϵ **Output:** v_{priv}

- 1: Compute $v \leftarrow \sum_{i=1}^n [\ell_i]_B$
- 2: Sample z uniformly at random from K
- 3: Sample $r \sim \text{Gamma}\left(k+1, \frac{1}{\epsilon}\right)$
- 4: Output $v_{priv} \leftarrow v + rz$

represent the posterior as follows:

$$\sigma(\theta \mid \mathcal{D}) \propto \exp\left(\sum_{i=1}^n \sum_{j=1}^k \alpha_j(\mathcal{D}_i) b_j(\theta)\right) \times \pi(\theta) \quad (4.4)$$

$$= \exp\left(\left\langle B, \sum_{i=1}^n [\ell_i]_B \right\rangle\right) \times \pi(\theta) \quad (4.5)$$

4.3.2 Publishing Coordinate Vector of Log-likelihood Function

In this step, we aim to solve the problem of publishing a coordinate vector:

$$\sum_{i=1}^n [\ell_i]_B$$

To do this, we apply the K -norm mechanism [54, 83, 52], which is an optimal differential private mechanism for publishing real-valued vector. Let $[\ell_\gamma]_B = [\alpha_1(\gamma), \dots, \alpha_k(\gamma)]$ be the coordinate vector of $\ell(\theta; \gamma)$, where $\gamma = (t_\gamma, d_\gamma)$ is an arbitrary data record. Let K be the convex hull of the set $S = \{[\ell_\gamma]_B - [\ell_{\gamma'}]_B \mid \forall \gamma, \gamma'\}$ which is the set of all possible differences between values of $\sum_{i=1}^n [\ell_i]_B$ on any two neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$. Algorithm 4.1 shows the pseudo-code of the K -norm mechanism. Line 1 computes the non-private coordinate vector. Line 2 samples uniformly a random vector from the convex hull K . This random vector is then scaled by a random length r (Line 3), which grows with the dimension of v . The output $v + rz$ is proven to be ϵ -differentially private with an error rate of $O\left(\min\left(\frac{k}{\epsilon}, \frac{\sqrt{k \cdot \log(n/k)}}{\epsilon}\right)\right)$ [54].

Here, we illustrate the above steps with the random variable T representing

Figure 4.1: Convex hull K for a parametric model with exponential distribution.

an exponential distribution. The log-likelihood function of a data record $\mathcal{D}_i$ is a function of a scale parameter λ as follows:

$$\ell(\lambda; \mathcal{D}_i) = d_i + \lambda \cdot t_i$$

With a set of basis functions $B = \{b_1(\cdot), b_2(\cdot)\}$, where

$$b_1(\lambda) = 1 \tag{4.6}$$

$$b_2(\lambda) = \lambda \tag{4.7}$$

we can keep all the information needed to determine the posterior distribution. As a result, the coordinate vector of $\ell(\lambda; \mathcal{D}_i)$ relative to B is $[\ell_i]_B = [d_i, t_i]$. Since $0 \leq t_i \leq 1$ and $d_i \in \{0, 1\}$, it is guaranteed that all possible values of $[\ell_\gamma]_B - [\ell_{\gamma'}]_B$ lie on the blue lines in Figure 4.1. Hence, the convex hull K is the shaded region in Figure 4.1, which is easy to sample uniformly at random.

4.3.3 Drawing Random Samples

Having a noisy coordinate vector v_{priv} from Algorithm 4.1, we can use an Markov Chain Monte-Carlo (MCMC) sampler (e.g., the No-U-Turn sampler [56]) to draw

multiple random samples from the noisy posterior:

$$\sigma_{priv}(\theta \mid \mathcal{D}) \propto \exp(\langle B, v_{priv} \rangle) \times \pi(\theta)$$

The random samples are then used to construct the survival function and hazard function, and also their credible intervals.

4.4 Applying SAP Framework to Parametric Models

We apply the proposed SAP framework to parametric models with Weibull distribution and flexible parametric models. For parametric models, we use C-spline basis functions to approximate likelihood functions. For flexible parametric models, we use I-spline to approximate the cumulative hazard function $H(t) = \int_{u=0}^t h(u)du$. Because of its flexibility, the flexible parametric models are useful in cases where the underlying distribution of T is not a Weibull distribution. Hence, it overcomes the limitation of parametric models with Weibull distribution. In both parametric and flexible parametric models, we need to tackle the hard problem of computing convex hull in high dimensional spaces as required by the K -norm mechanism. In this work, we propose to replace the convex hull with convex sets which are easier to compute with an insignificant trade-off in accuracy.

In this section, we first introduce the likelihood function of parametric models with Weibull distribution. Then, we present a private mechanism which uses C-spline to approximate the likelihood function in order to guarantee DP under the proposed SAP framework.

4.4.1 Bayesian Modeling for Parametric Models

In parametric models, the random variable T belongs to a particular family of probability distributions. The most common distributions are exponential distribution and Weibull distribution [78]. In the following, we describe a parametric model with Weibull distribution. The proposed framework can also be applied to other distributions. For Weibull distribution, there are two positive parameters: the shape parameter p and the scale parameter λ . The survival function has the following form:

$$S(t \mid p, \lambda) = e^{-\left(\frac{t}{\lambda}\right)^p}$$

From that, we can derive the hazard function as follows:

$$h(t \mid p, \lambda) = -\frac{dS(t)/dt}{S(t)} = \frac{p}{\lambda} \left(\frac{t}{\lambda}\right)^{p-1}$$

From Equation (4.3), the log-likelihood function for i -th record is:

$$\ell(p, \lambda; \mathcal{D}_i) = d_i [\log(p) - p \cdot \log(\lambda) + (p-1) \log(t_i)] - \left(\frac{t_i}{\lambda}\right)^p$$

Therefore, the total log-likelihood function is as follows:

$$\begin{aligned} \ell(p, \lambda) &= \sum_{i=1}^n \ell(p, \lambda; \mathcal{D}_i) \\ &= (\log(p) - p \cdot \log(\lambda)) \cdot \sum_{i=1}^n d_i \\ &\quad + (p-1) \cdot \left(\sum_{i=1}^n d_i \log(t_i) \right) \\ &\quad - \frac{1}{\lambda^p} \sum_{i=1}^n t_i^p \end{aligned} \tag{4.8}$$

The posterior is $\sigma(p, \lambda \mid \mathcal{D}) \propto \exp(\ell(p, \lambda)) \times \pi(p, \lambda)$. Since p and λ are positive real values, we pick a Half-Cauchy distribution for $\pi(\cdot)$, which is commonly used as a weakly informative prior distribution.

4.4.2 Differentially Private Protection

From Equation (4.8), we can simply pick $b_1(p, \lambda) = \log(p) - p \cdot \log(\lambda)$ and $b_2(p, \lambda) = p - 1$ as basis functions. However, it is non-trivial to represent t_i^p as a linear combination of basis functions. Here, we propose to approximate the function t_i^p with C-spline basis functions. Figure 4.2a shows some examples of t_i^p of a dataset with $t_1 = 0.2, t_2 = 0.5$ and $t_3 = 0.8$. Because of the convexity of t_i^p , we can approximate the function by a set of C-spline (convex spline) basis functions which are also convex functions. Let $C = \{c_1(p), \dots, c_\tau(p)\}$ be the set of C-spline basis functions. Figure 4.2b shows an example of basis functions (see Appendix A for the detailed definitions of these basis functions). From that, the function t_i^p can be approximated as follows:

$$t_i^p = \eta_0 + \sum_{j=1}^{\tau} \eta_j \cdot c_j(p)$$

where $\eta_j, 0 \leq j \leq \tau$ are coordinates of t_i^p relative to the basis C . Given $\eta_j, 0 \leq j \leq \tau$, which are non-negative, we can guarantee that t_i^p is a non-increasing convex function. We can compute $\eta_0, \dots, \eta_\tau$ by using the method of non-negative least squares. Given a set of fixed values $p_1, \dots, p_m$, we have:

$$\eta_0, \dots, \eta_\tau = \arg \min_{\beta \geq 0} \sum_{j=1}^m \left(\beta_0 + \sum_{w=1}^{\tau} \beta_w \cdot c_w(p_j) - t_i^{p_j} \right)^2$$

Given the C-spline approximation of t_i^p , we can represent the total log-likelihood function as a linear combination of the following basis functions:

$$b_1(p, \lambda) = \log(p) - p \cdot \log(\lambda) \quad (4.9)$$

$$b_2(p, \lambda) = p - 1 \quad (4.10)$$

$$b_3(p, \lambda) = -\frac{1}{\lambda^p} \quad (4.11)$$

$$b_{3+j}(p, \lambda) = -\frac{1}{\lambda^p} \cdot c_j(p), \quad 1 \leq j \leq \tau \quad (4.12)$$

(a) t_i^p with three different values $t_i = 0.2$, $t_i = 0.5$ and $t_i = 0.8$. (b) C-spline basis functions of degree 3 with 4 inner knots.

Figure 4.2: An illustration of t_i^p and C-spline basis functions which are used to approximate t_i^p .

The corresponding coordinate vector is $[\ell_i]_B = [d_i, d_i \log(t_i), \eta_0, \eta_1, \dots, \eta_\tau]$.

As required by the K -norm mechanism, we aim to sample uniformly at random from the convex hull of set $S = \{[\ell_\gamma]_B - [\ell_{\gamma'}]_B \mid \forall \gamma, \gamma'\}$. However, we may not have a closed-form representation of S . To overcome this difficulty, we propose a discretization method of time values. Instead of computing the convex hull of the continuous set S , we can compute the convex hull of the discretization of S . If the discretization is fine-grained enough, we could practically achieve the same performance as of the original set S . Let $\Delta = \{\delta_1, \delta_2, \dots, \delta_q\}$ be the set of discrete time values. Then, the set of all possible data records with respect to the discrete time values Δ is:

$$\mathcal{D}_{\text{discrete}} = \{(\delta_i, 1), (\delta_i, 0) \mid i = 1, \dots, q\}$$

Then, we can compute the set of all corresponding coordinate vectors of $\mathcal{D}_{\text{discrete}}$ as follows:

$$[\mathcal{D}_{\text{discrete}}]_B = \{[\mathcal{D}_\gamma]_B \mid \mathcal{D}_\gamma \in \mathcal{D}_{\text{discrete}}\}$$

From that, the discrete version of set S is:

$$S_{\text{discrete}} = \{x - y \mid x, y \in [\mathcal{D}_{\text{discrete}}]_B\}$$

Algorithm 4.2 $\mathcal{A}_{\text{Discrete-Convex-Hull}}$: Computing a Discrete Set of Coordinate Vectors and its Convex Hull

---

Input: Discrete times $\delta_1, \dots, \delta_q$

Output: v

```

1:  $C \leftarrow \emptyset$ 
2: for  $i = 1$  to  $q$  do
3:    $C[2i] \leftarrow$  coordinate vector of  $\ell(p, \lambda; \delta_q, 0)$ 
4:    $C[2i - 1] \leftarrow$  coordinate vector of  $\ell(p, \lambda; \delta_q, 1)$ 
5: end for
6:  $S \leftarrow \emptyset$ 
7: for  $i = 1$  to  $2q$  do
8:   for  $j = 1$  to  $2q$  do
9:      $S \leftarrow S \cup \{C[i] - C[j]\}$ 
10:  end for
11: end for
12:  $K \leftarrow$  compute the convex hull of set  $S$ 
13:  $v \leftarrow$  sample uniformly at random from  $K$ 
14: Output  $v$ 

```

Algorithm 4.2 shows the above steps in pseudo-code. Lines 1-11 compute the set of all possible differences between any two coordinate vectors. Unfortunately, this algorithm also has a major problem. The run-time complexity for computing a convex hull and sampling uniformly at random from its interior grow very quickly with the dimension of the vector space. Therefore, it is impractical to compute a convex hull of a set of thousands of points in a high dimensional space. To circumvent this problem, we propose to approximate the exact convex by a ℓ_1 -norm ball. This comes from an observation that the function t_i^p is bounded by $[0, 1]$, which matches with the range of C-spline basis functions. Therefore, the ℓ_1 -norm of coordinate vector $\|[\ell_i]_B\|_1$ is bounded by a small constant. So, we can achieve a good approximate of the convex hull with a ℓ_1 -norm ball. Moreover, it is guaranteed that sampling uniformly at random from a convex set, which contains the convex hull K , also satisfies differential privacy.

Let t_i^δ be the nearest discrete time of t_i , $t_i^\delta = \delta_{j^*}$, such that $j^* = \arg \min_{j \in \{1, \dots, q\}} |\delta_j - t_i|$. Algorithm 4.3 shows the pseudo-code of the proposed sampling algorithm. Line 1 computes the nonprivate coordinate vector of the dataset $\mathcal{D}$, whose time

Algorithm 4.3 $\mathcal{A}_{\text{Priv-Para-Model}}$: Private Mechanism for Parametric Models**Input:** Set $S_{\text{discrete}} = \{x - y \mid x, y \in [\mathcal{D}_{\text{discrete}}]_B\}$, privacy budget ϵ **Output:** c_{priv}

- 1: $c \leftarrow \sum_{i=1}^n [\ell(p, \lambda; t_i^\delta, d_i)]_B$
- 2: $\Omega \leftarrow \max_{i \in [1, \dots, q]} |\log(\delta_i)|$
- 3: $X[1] \leftarrow \text{Uniform}(-1, +1)$
- 4: $X[2] \leftarrow \text{Uniform}(-\Omega, +\Omega)$
- 5: $\text{radius} \leftarrow \max_{x \in S_{\text{discrete}}} \|x\|_1$
- 6: **for** $i = 1$ to $\tau + 1$ **do**
- 7: $A[i] \sim \text{Lap}(0, 1)$
- 8: **end for**
- 9: $y \sim \text{Exp}(1)$
- 10: $A \leftarrow \frac{A}{y + \|A\|_1}$
- 11: $A \leftarrow \text{radius} \times A$
- 12: $r \sim \text{Gamma}(\tau + 4, \frac{1}{\epsilon})$
- 13: **Output** $c_{\text{priv}} \leftarrow c + r \cdot \begin{bmatrix} X \\ A \end{bmatrix}$

values are discretized. Line 2 computes the limit of $d_i \log(t_i^\delta)$, it is guaranteed that $-\Omega \leq d_i \log(t_i^\delta) - d_j \log(t_j^\delta) \leq +\Omega$. Similarly, the coordinate value $d_i - d_j$ is bounded by $[-1, +1]$. Then, at Lines 3 and 4 the two elements of vector X are sampled uniformly at random from the corresponding boundaries. Line 5 computes the maximum value of ℓ_1 -norms of C-spline coordinate vectors. Lines 6-10 implement the sampling procedure proposed in [12], which guarantees to sample A uniformly at random from ℓ_1 -norm unit ball. Lines 11-13 implement the rest of the K -norm mechanism.

4.5 Applying SAP Framework to Flexible Parametric Models

In this section, we first introduce the statistical models for flexible survival analysis. Then, we apply the proposed SAP framework to design a differentially private mechanism for the models.

4.5.1 Bayesian Modeling for Flexible Parametric Models

Parametric models are limited in their capability to model complex distributions. One possible solution is to use flexible parametric models. Instead of assuming a specific family of distributions for random variable T , flexible parametric models use spline functions to approximate the survival function and hazard function. In this work, we propose to use I-spline to approximate the *cumulative hazard function*:

$$H(t) = \int_{x=0}^t h(x)dx$$

This is because $H(t)$ is a monotonic function which is suited for the monotonic property of I-spline basis functions.

Let $\{b_i(t)\}_{i=1}^k$ be a set of k basis I-spline functions of degree 1. The definition of $b_i(t)$ is as follows:

$$b_i(t) = \int_0^t M_i(u)du$$

where $M_i(u)$ are the basis functions of the non-negative M-spline, of which the definition is given as follows:

$$M_i(x) = \begin{cases} \frac{1}{v_{i+1}-v_i}, & \text{if } v_i \leq x \leq v_{i+1} \\ 0, & \text{if otherwise} \end{cases}$$

where $v_i, 1 \leq i \leq k+1$, are knot locations. Then, the function $H(t)$ can be approximated by:

$$H(t) = \sum_{i=1}^k b_i(t) \cdot \beta_i$$

where $\beta = \{\beta_i\}_{i=1}^k$ are non-negative coefficients. From this, we derive the survival function as follows:

$$S(t_i; \beta) = \exp(-H(t_i)) = \exp\left(-\sum_{j=1}^k b_j(t_i) \cdot \beta_j\right)$$

The corresponding hazard function is as follows:

$$\begin{aligned}
 h(t_i; \beta) &= \frac{dH(t_i; \beta)}{dt} = \sum_{j=1}^k \frac{db_j(t_i)}{dt} \cdot \beta_j \\
 &= \sum_{j=1}^k M_j(t_i) \cdot \beta_j \\
 &= \frac{1}{v_{\alpha_i+1} - v_{\alpha_i}} \cdot \beta_{\alpha_i}
 \end{aligned}$$

where $\alpha_i \in \{1, \dots, k\}$ such that $v_{\alpha_i} \leq t_i \leq v_{\alpha_i+1}$.

From Equation (4.3), the log-likelihood function is:

$$\ell(\beta; \mathcal{D}_i) = d_i \log \left(\frac{\beta_{\alpha_i}}{v_{\alpha_i+1} - v_{\alpha_i}} \right) - \sum_{j=1}^k b_j(t_i) \cdot \beta_j \quad (4.13)$$

As I-spline with degree 1 is chosen for the sake of simplicity in our model, we lost the ability to control the smoothness of the approximated cumulative hazard function. Here, we leverage the advantage of Bayesian inference in order to achieve a similar effect of controlling the smoothness. For Bayesian inference, we can impose constraints on the parameters of our model by choosing an appropriate prior. Moreover, as prior does not relate to any data record in the dataset, we are free to pick the prior we need without compromising the privacy protection. We aim to pick a prior which can constrain the cumulative hazard function so that its derivatives would not fluctuate much between time segments. As a result, we aim to design a prior such that the parameters $\beta_1, \dots, \beta_\tau$ are close to each other as these parameters are proportional to the derivatives of the cumulative hazard function. To this end, we adopt the random walk priors as follows:

$$\gamma_1 \sim \mathcal{N}(0, \sigma_1^2) \quad (4.14)$$

$$\gamma_i \sim \mathcal{N}(\gamma_{i-1}, \sigma_2^2) \quad (4.15)$$

$$\beta_i = \exp(\gamma_i) \quad (4.16)$$

Algorithm 4.4 $\mathcal{A}_{\text{Priv-Flex-Model}}$: Private Mechanism for Flexible Parametric Models

---

Input: Privacy budget ϵ **Output:** c_{priv}

```

1:  $c \leftarrow \sum_{i=1}^n [\ell_i]_B$ 
2: for  $i = 1$  to  $k$  do
3:    $A[i] \sim \text{Lap}(0, 1)$ 
4: end for
5:  $y \sim \text{Exp}(1)$ 
6:  $A \leftarrow \frac{A}{y + \|A\|_1}$ 
7: for  $i = 1$  to  $k$  do
8:    $B[i] \sim \text{Uniform}(-1, 1)$ 
9: end for
10:  $r \sim \text{Gamma}(2k + 1, \frac{1}{\epsilon})$ 
11: Output  $c_{\text{priv}} \leftarrow c + r \cdot \begin{bmatrix} 2A \\ B \end{bmatrix}$ 

```

These priors induce that β_i is non-negative and close to the nearby values β_{i-1} and β_{i+1} .

4.5.2 Differentially Private Protection

From Equation (4.13), we choose a set of $2k$ basis functions as follows:

$$B = \left[\log \left(\frac{\beta_1}{v_2 - v_1} \right), \dots, \log \left(\frac{\beta_k}{v_{k+1} - v_k} \right), -\beta_1, \dots, -\beta_k \right]$$

The corresponding coordinate vector is:

$$[\ell_i]_B = [z_1, \dots, z_k, b_1(t_i), \dots, b_k(t_i)]$$

where $z_j = 1$ if $j = \alpha_i$, and $z_j = 0$ if otherwise. Since $\sum_{j=1}^k |z_j| = 1$, we can approximate the first k coordinates of the convex hull K by sampling a point from ℓ_1 -norm ball of radius 2. Meanwhile, since $\forall j \in \{1, \dots, k\}, 0 \leq b_j(t_i) \leq 1$, we can approximate the last k coordinates of convex hull K by sampling a random point from ℓ_∞ -norm unit ball. Note that there is no need for discretization of time values. Algorithm 4.4 shows the pseudo-code of the mechanism for flexible

parametric models. Line 1 computes the non-private estimate. Lines 2-6 sample a random point from ℓ_1 -norm unit ball [12]. And Lines 7-9 sample a random point from ℓ_∞ -norm unit ball.

4.6 Experimental Evaluation

In this section, we have designed experiments aiming at answering the following research questions:

- What is the trade-off between privacy and accuracy of our proposed SAP framework in estimating the posterior distributions of parametric and flexible parametric models?
- What is the performance of our proposed SAP framework in estimating the underlying distribution of random variable T ?
- What is the performance of our proposed SAP framework in estimating parametric models in comparison with the proposed mechanisms in Chapter 3 and other methods?

4.6.1 Datasets

We use the same four real datasets as in Chapter 3 which were given in Table 3.1. We observed that because the survival times t_i are sparse at high value, it causes unstable experimental results. Therefore, we remove data records whose survival times belong to the 90th percentile to make the results more stable. The survival times are then normalized to the interval $[0, 1]$.

4.6.2 Publishing the Posterior Distribution

In this set of experiments, we aim to measure the performance of our proposed SAP framework in estimating the posterior distribution. For both parametric

and flexible parametric models, we use the proposed SAP framework to estimate a private posterior distribution. Then, we compare the private posterior distribution to the non-private posterior distribution obtained from traditional Bayesian inference.

Evaluation metric: We use the Kullback-Leibler divergence (KLD) [98] to measure the distance between the published posterior and the non-private posterior. The Kullback-Leibler divergence between two probabilities P and Q is defined as follows:

$$D_{KL}(P||Q) = \int_{\theta} \log \left(\frac{dP}{dQ} \right) dP$$

Since we do not have the closed-form representation of the two posteriors, we propose to draw multiple random samples from the two posteriors. Then, we fit the two sets of random samples by two multivariate Gaussian distributions $\mathcal{N}_{\text{priv}}$ and $\mathcal{N}_{\text{nonpriv}}$ respectively. The performance score is the Kullback-Leibler divergence between these two Gaussian distributions:

$$D_{KL}(\mathcal{N}_{\text{priv}}||\mathcal{N}_{\text{nonpriv}}) \quad (4.17)$$

Hyperparameters: For parametric models, we use C-spline of degree 3 with 10 knots, which are equally distributed in the interval $[0, 10]$. The priors for shape and scale parameters are Half-Cauchy(0, 5). For flexible parametric models, we use I-spline with 15 knots, which are equally distributed in the interval $[0, 1]$. The prior of γ_1 (in Equation (4.14)) is $\mathcal{N}(0, 1)$ and the prior of γ_i ($\forall i > 1$) (in Equation (4.15)) is $\mathcal{N}(\gamma_{i-1}, 0.1)$.

Experimental results: We repeat the experiments 1000 times and report the median scores. Figure 4.3 shows that the KLD score gradually decreases as we increase the privacy budget from 0.1 to 1.6. As a result, the private posteriors are shown to be close to the non-private posteriors, especially at high privacy budgets.

Figure 4.3: Performance results (in KLD) of our proposed SAP framework for parametric models and flexible parametric models.

4.6.3 Point Estimations of Parametric Models and Flexible Parametric Models

In this set of experiments, we consider the performance of the proposed SAP framework in point estimation. We first use the Algorithm 4.3 (and Algorithm 4.4, respectively) to estimate the private posterior distribution of parametric models with Weibull distribution (and flexible parametric models, respectively). We then infer a point estimation from the private distribution.

Compared mechanisms: For parametric models with Weibull distribution, we compare the point estimation performance of our proposed framework (SAP) with the baseline *subsample-and-aggregate* mechanism SAA [95]. We also compare SAP with a mechanism, called Local Sensitivity mechanism, which is a combination of the two mechanisms proposed in Chapter 3, i.e., (i) LSP mechanism for estimating p , and (ii) TLL mechanism for estimating λ , with each mechanism spending half of the privacy budget ϵ . For flexible parametric models, we compare the point estimation performance of our proposed framework (SAP) with the baseline *subsample-and-aggregate* mechanism SAA.

Figure 4.4: Performance results (in MdAE) for parametric models with Weibull distribution.

Evaluation metric: To measure the performance, we use the median absolute error (MdAE) which is defined as follows:

$$\text{MdAE} = \text{median}(\{\|x_i - x^*\|_1, i = 1 \dots t\}) \quad (4.18)$$

where x_i is estimated from private mechanisms, x^* is obtained from the traditional maximum likelihood estimation (MLE) approach without privacy protection and $t = 1000$ is the number of tries. Since our proposed SAP framework publishes a posterior distribution, we use *maximum a posteriori estimation* (MAP) [13] as our point estimator to estimate x_i from the posterior.

Experimental results for parametric models with Weibull distribution: Figure 4.4 shows the experimental results on the log-log scale. On all the datasets, our proposed SAP framework consistently achieves good performance, outperforms the baseline SAA mechanism. SAP performs many orders of magnitude

Figure 4.5: Performance results (in MdAE) for flexible parametric models.

better than SAA with MdAE score. SAP also performs closely to that of the Local Sensitivity mechanism. SAP is about 2 times higher than Local Sensitivity in MdAE score. However, our proposed SAP framework publishes the whole posterior distribution, which is much more informative than just a point estimate of the Local Sensitivity mechanism.

Experimental results for flexible parametric models: Figure 4.5 shows the experimental results on the log-log scale. On all the datasets, our proposed SAP framework consistently achieves good performance, outperforms the baseline SAA mechanism. SAP performs 10 times better than SAA with the smallest performance gap on dataset SB at $\epsilon = 0.8$.

Figure 4.6: Estimated distribution of survival functions using our proposed SAP framework (in red) in comparison with that of Kaplan-Meier estimator and its 95% confidence interval (in black).

4.6.4 Estimating the Underlying Distribution of T

In this set of experiments, we aim to investigate the ability of the proposed SAP framework (parametric models with Weibull distribution and the proposed flexible parametric models) in estimating the underlying distribution of random variable T .

Baselines: We use the survival function and its 95% confidence intervals obtained from Kaplan-Meier estimator as the *ground truth* for comparison with the private posterior from the SAP framework.

Experimental results: In Figure 4.6, we show a private posterior distribution of survival functions (in red) using the proposed mechanisms from SAP framework with privacy budget $\epsilon = 1.6$. Figure 4.6 shows that the performance of the proposed mechanism (Algorithm 4.3) for parametric models with Weibull distribution (see Figures 4.6a-4.6d) which does not match well with the ground truth on datasets WT and SB. Whereas the performance of the proposed mechanism (Algorithm 4.4) for flexible parametric models (see Figures 4.6e-4.6h), with the flexibility of spline approximation, matches well with the ground truth on all datasets.

4.7 Summary

In this chapter, we have proposed a novel differential private SAP framework for estimating the posterior of survival models. The proposed framework represents the likelihood function as a linear combination of basis functions. We apply the K -norm mechanism to publish the coordinate vector of the total likelihood function. We then use the MCMC sampler to draw multiple random samples from the private posterior. We have applied the proposed SAP framework to (i) parametric models, in which we use C-spline basis functions to approximate the likelihood function and (ii) flexible parametric models, in which we use I-spline to approximate the cumulative hazard function. Experimental results have shown the effectiveness of the proposed SAP framework.

Chapter 5

Discrete-time Survival Regression

In survival analysis, regression models are used to understand the effects of explanatory variables (e.g., age, gender, weight, etc.) to the survival probability. However, for sensitive survival data such as medical data, there are serious concerns about the privacy of individuals in the dataset when such data is used to fit the regression models. The closest work addressing such privacy concerns is the work on Cox regression [28] which linearly projects the original data to a lower dimensional space. However, the weakness of this approach is that there is no formal privacy guarantee for such a projection. In this chapter, we aim to propose solutions for the regression problem in survival analysis with the protection of differential privacy which is a golden standard of privacy protection in data privacy research. To this end, we extend the *Output Perturbation* and *Objective Perturbation* [24] approaches which are originally proposed to protect differential privacy for the Empirical Risk Minimization (ERM) problem. In addition, we also propose a novel sampling approach based on the Markov Chain Monte Carlo (MCMC) [71] method to practically guarantee differential privacy with better accuracy. We show that our proposed approaches achieve good accuracy as compared to the non-private results while guaranteeing differential privacy for individuals in the private dataset.

5.1 Discrete-time Survival Analysis

In this section, we first introduce the discrete-time model of survival function and hazard function. We then introduce the discrete-time survival regression models.

5.1.1 Discrete-time Survival Model

For discrete-time models, let time be divided into intervals $[a_0, a_1), [a_1, a_2), \dots, [a_{q-1}, a_q]$, $a_0 = 0, a_q = 1$, where q is the number of discrete times. The discrete time t refers to the interval $[a_{t-1}, a_t)$. A discrete random variable T represents the discrete death time. $T = t$ denotes the death within the time interval $t = [a_{t-1}, a_t)$. The characteristic function of T is the discrete hazard function:

$$h(t) = \Pr[T = t \mid T \geq t], \quad t = 1, \dots, q$$

which is the conditional probability for the risk of death in interval t given the survival in all previous intervals. The discrete survival function for reaching interval t is:

$$S(t) = \Pr[T \geq t] = \prod_{s=1}^{t-1} (1 - h(s)) \quad (5.1)$$

Discrete-time datasets are given by $(x_i, \delta_i, t_i), i = 1, \dots, n$, where $t_i = \min(T_i, c_i)$ is the minimum of the survival time T_i and censoring time c_i , and δ_i is the indicator variable for death ($\delta_i = 1$) or censoring ($\delta_i = 0$). When $\delta_i = 0$, the i -th patient is known to survive until time c_i but the survival time T_i is not observed ($T_i > c_i$). x_i is a real vector of explanatory variables (e.g., gender, age, weight, etc.) which affect the survival probability. We assume that x_i is inside the unit-sphere, $\|x_i\| \leq 1$. This is actually a common practice in machine learning. For convenience, we use y_i to refer to the term $(2\delta_i - 1)$ and d_i to refer to the tuple (x_i, y_i, t_i) .

5.1.2 Discrete-time Survival Regression

In this section, we introduce the discrete-time survival regression models which are used to model the relationship between explanatory variables and the hazard rate, i.e., the predictive variable. From that, the subsequent sections will discuss the proposed differentially private approaches to guarantee that the estimated parameters from the regression model satisfy the definition of differential privacy.

Generalized linear model: We model the effects of explanatory variables x_i to the survival probability by using a generalized linear model:

$$g(h(t_i | x_i)) = \gamma(t_i) + x_i' \beta \quad (5.2)$$

where $g(\cdot)$ is the link function, β is the parameter vector representing the effects of explanatory variables and $\gamma(t_i)$ is a time-varying baseline hazard effect.

A commonly used link function in survival probability is the logit link function $g(x) = \text{logit}(x) = \log\left(\frac{x}{1-x}\right)$. The logit link function allows the model to have a nice interpretation of the proportional odds ratio. The other two link functions, which are also used in survival analysis, are the complementary log-log link function $g(x) = \text{cloglog}(x) = \log(-\log(1-x))$, and the probit link function $g(x) = \text{probit}(x) = \Phi^{-1}(x)$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. Interestingly, the complementary log-log link function has the same interpretation of proportional hazard ratio as the Cox regression. We refer interested readers to [8] for more details.

As illustrated in Figure 5.1, the three link functions have similar shapes which lead to similar estimation results. Notably, the logit link function and cloglog link function are almost identical at x near 0. This explains why the output models from the logit link function and cloglog link function are very similar in practice when the number of discrete time q is large, or equivalently, the hazard rate x is small.

Figure 5.1: The illustration of link functions: logit (green), cloglog (blue) and probit (red).

In this work, we have selected the logit link function because it has a bounded derivative for the loss function which is required by our proposed Extended Output Perturbation and Extended Objective Perturbation approaches. Whereas our proposed sampling approach can work with all three link functions.

Baseline hazard effect: We model the baseline hazard effect $\gamma(t)$ using natural cubic spline [49] with e knots equally distributed over the interval $[0, 1]$, $0 = k_1 < k_2 < \dots < k_e = 1$.

Let

$$d_j(t) = \frac{\max(t - k_j, 0)^3 - \max(t - k_e, 0)^3}{k_e - k_j}$$

and

$$b_1(t) = 1, b_2(t) = t, b_{i+2} = d_i(t) - d_{e-1}(t)$$

The baseline hazard effect $\gamma(t)$ is approximated as a linear combination of e basis functions:

$$\gamma(t) = \alpha_1 b_1(t) + \dots + \alpha_e b_e(t)$$

In particular, let $A_i = [b_1(t_i), \dots, b_e(t_i)]'$ and $\alpha = [\alpha_1, \dots, \alpha_e]'$, then we can write

$$\gamma(t_i) = \alpha' A_i$$

5.2 Problem Definitions

Traditionally, we use MLE to estimate parameters α and β in our models. The aim is to maximize the log-likelihood of the observed data. For simplicity, let $f = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $x_i^t = \begin{pmatrix} A_t \\ x_i \end{pmatrix}$. There are two cases: (i) If we observe a death event $d_i = 1$ at t_i , then the likelihood of this event is $S(t_i | f, x_i) \times h(t_i | f, x_i)$. (ii) Otherwise, if we observe a censored event $d_i = 0$ at t_i , then the likelihood of this event is $S(t_i | f, x_i) \times (1 - h(t_i | f, x_i))$. Overall, the log-likelihood function is:

$$\log \mathcal{L}(f) = \sum_{i=1}^n \log \left[h(t_i | f, x_i)^{d_i} (1 - h(t_i | f, x_i))^{1-d_i} S(t_i | f, x_i) \right]$$

Let $y_i = 2d_i - 1$, from Equation (5.1) and Equation (5.2), and substituting $g(x) = \text{logit}(x)$, we can rewrite our problem as:

$$\log \mathcal{L}(f) = - \sum_{i=1}^n \left[\ell_{\text{LR}}(y_i f' x_i^{t_i}) + \sum_{s=1}^{t_i-1} \ell_{\text{LR}}(-f' x_i^s) \right]$$

where $\ell_{\text{LR}}(x) = \log(1 + \exp(-x))$ is the logistic loss function. To further simplify the formula, let $d_i = (x_i, y_i, t_i)$, $i = 1, \dots, n$, and let

$$\ell(f; d_i) = \ell_{\text{LR}}(y_i f' x_i^{t_i}) + \sum_{s=1}^{t_i-1} \ell_{\text{LR}}(-f' x_i^s) \quad (5.3)$$

be the loss function. Then, we get an ERM problem as follows:

$$f^* = \arg \min_f \sum_{i=1}^n \ell(f; d_i) \quad (5.4)$$

Problem 5.1: Given a dataset $\mathcal{D}$ and privacy budget ϵ , release the parameters of the discrete-time regression model such that ϵ -differential privacy is satisfied.

Therefore, our main goal in this chapter is to propose algorithms which protect differential privacy for the parameters f^* in Equation (5.4).

5.3 Extended Output Perturbation Approach

In this section, we present our proposed algorithm which is the extension of the Output Perturbation approach in [24]. For our problem, the loss function is a sum of logistic loss functions instead of a single logistic loss function as in [24]. The proposed algorithm is in fact based on the generalized version of the *Laplace mechanism* [35] which is described as follows: Let $f^* = G(\mathcal{D})$ be the value that we want to guarantee differential privacy. f^* is the result of applying a function G on the private dataset $\mathcal{D}$ (e.g., it is in our case to minimize the objective function). We define the sensitivity of the function G as follows:

$$sen(G) = \max_{\mathcal{D}, \mathcal{D}'} \|G(\mathcal{D}) - G(\mathcal{D}')\|$$

where $\mathcal{D}$ and $\mathcal{D}'$ are two neighboring datasets. Then, the differentially private version of $f^* = G(x)$ is:

$$f_{priv} = f^* + \mu$$

where μ is a noisy random variable with probability density function $\text{pdf}(\mu) \propto \exp(-\epsilon \|\mu\| / sen(G))$.

As required by the Output Perturbation approach, we consider the following regularized objective function:

$$J(f; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \ell(f; d_i) + \frac{\Lambda}{2} \|f\|^2 \quad (5.5)$$

where $\mathcal{D} = \{d_i\}_{i=1}^n$, $\ell(\cdot)$ is the loss function as defined in Equation (5.3) and Λ is the regularization parameter. In this approach, our goal is to compute the sensitivity of:

$$f^* = \arg \min_f J(f; \mathcal{D})$$

Then, we use the sensitivity to control the amount of noise added to f^* .

Algorithm 5.1 $\mathcal{A}_{\text{Ext-Out-Pert}}$: Extended Output Perturbation Approach**Input:** Dataset $\mathcal{D} = \{d_1, \dots, d_n\}$, loss function $\ell(f; d_i)$, privacy budget ϵ **Output:** f_{priv}

- 1: $J(f; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \ell(f; d_i) + \frac{\Lambda}{2} \|f\|^2$
- 2: Minimize $J(f; \mathcal{D})$ by using the BFGS algorithm to get the non-private solution f^*
- 3: Compute $t \leftarrow \frac{\sum_{s=1}^q \sqrt{4 + \|A_s\|^2} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}}{n \cdot \Lambda}$
- 4: Sample a random vector b such that $\text{pdf}(b) \propto \exp\left(-\epsilon \frac{\|b\|}{t}\right)$
- 5: Output $f_{\text{priv}} \leftarrow f^* + b$

5.3.1 Proposed Algorithm

Algorithm 5.1 shows the proposed Extended Output Perturbation approach. It returns a vector f_{priv} as the minimizer of $J(\cdot)$ while guaranteeing differential privacy. At Line 2, we compute the non-private solution $f^* = \arg \min_f J(f; \mathcal{D})$ using the well-known BFGS algorithm [48]. f^* is guaranteed to exist due to the strongly convexity of $J(f; \mathcal{D})$. At Line 3, we compute t which is the sensitivity of f^* . Lines 4-5 add noise to the value of f^* .

In order to sample a random vector b in Algorithm 5.1 from the distribution $\text{pdf}(b) \propto \exp(-\epsilon \|b\|/t)$, we observe that the length of the vector b follows a Gamma distribution:

$$\|b\| \sim \Gamma(d, t/\epsilon)$$

where d is the number of components of b . Thus, in order to sample b we first sample its length $r = \|b\|$ from the Gamma distribution and then sample b as a uniform random point on the surface of a sphere with radius r .

5.3.2 Privacy Guarantee

In order to prove the differential privacy protection, we focus on proving that the sensitivity of f^* at Line 2 in Algorithm 5.1 is equal to the value of t which is computed at Line 3. Here, we use the following Lemma 5.1 derived from [24] to bound the sensitivity of f^* .

Lemma 5.1. Let $G(f)$ and $g(f)$ be two vector-valued functions, which are continuous and differentiable at all points. In addition, let $G(f)$ and $G(f) + g(f)$ be λ -strongly convex. If $f_1 = \arg \min_f G(f)$ and $f_2 = \arg \min_f G(f) + g(f)$, then

$$\|f_1 - f_2\| \leq \frac{1}{\lambda} \max_f \|\nabla g(f)\|$$

From Lemma 5.1, our goal now is to bound the magnitude of the difference in the gradients of the objective function $J(\cdot)$ on any two neighboring datasets.

Lemma 5.2. For any pair of patient's records $d_i = (x_i, y_i, t_i)$ and $d_j = (x_j, y_j, t_j)$, and for any f ,

$$\|\nabla \ell(f; d_i) - \nabla \ell(f; d_j)\| \leq \sum_{s=1}^q \sqrt{\|A_s\|^2 + 4} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}$$

Proof.

$$\begin{aligned} \nabla \ell(f; d_i) &= \nabla \ell_{\text{LR}}(y_i f' x_i^{t_i}) + \sum_{s=1}^{t_i-1} \nabla \ell_{\text{LR}}(-f' x_i^s) \\ &= \frac{-y_i x_i^{t_i}}{1 + \exp(y_i f' x_i^{t_i})} + \sum_{s=1}^{t_i-1} \frac{x_i^s}{1 + \exp(-f' x_i^s)} \end{aligned}$$

Therefore, we can write $\nabla \ell(f; d_i) = \sum_{s=1}^q l_i^s$, where

$$l_i^s = \begin{cases} \frac{x_i^s}{1 + \exp(-f' x_i^s)}, & \text{if } s < t_i \\ \frac{-y_i x_i^{t_i}}{1 + \exp(y_i f' x_i^{t_i})}, & \text{if } s = t_i \\ \vec{0}, & \text{if } s > t_i \end{cases}$$

Similarly, we can also write $\nabla \ell(f; d_j) = \sum_{s=1}^q l_j^s$. Therefore,

$$\nabla \ell(f; d_i) - \nabla \ell(f; d_j) = \sum_{s=1}^q l_i^s - l_j^s$$

We have $|\frac{-y_i}{1+\exp(y_i f'_i x_i^s)}| \leq 1$, $\|x_i\| \leq 1$, $\|x_j\| \leq 1$, for any $s \in \{1 \dots q\}$, we consider four possible cases as follows:

Case 1: if $s < t_i$ and $s < t_j$, then

$$\begin{aligned} \|l_i^s - l_j^s\| &= \left\| \begin{pmatrix} (e_1 - e_2)A_s \\ e_1 x_i - e_2 x_j \end{pmatrix} \right\| \leq \sqrt{\|A_s\|^2 + (\|x_i\| + \|x_j\|)^2} \\ &\leq \sqrt{\|A_s\|^2 + 4} \end{aligned}$$

where $e_1 = \frac{1}{1+\exp(-f'_i x_i^s)}$ and $e_2 = \frac{1}{1+\exp(-f'_j x_j^s)}$.

Case 2: if $s > t_i$ or $s > t_j$, then $\|l_i^s - l_j^s\| \leq \max(\|x_i^s\|, \|x_j^s\|) \leq \sqrt{\|A_s\|^2 + 1} < \sqrt{\|A_s\|^2 + 4}$.

Case 3: if $l_i^s = \frac{-x_i^s}{1+\exp(f'_i x_i^s)}$ and $l_j^s = \frac{x_j^s}{1+\exp(-f'_j x_j^s)}$, then

$$\|l_i^s - l_j^s\| = \left\| - \begin{pmatrix} (e_1 + e_2)A_s \\ e_1 x_i + e_2 x_j \end{pmatrix} \right\| \leq \sqrt{\|2A_s\|^2 + 4}$$

where $e_1 = \frac{1}{1+\exp(f'_i x_i^s)}$ and $e_2 = \frac{1}{1+\exp(-f'_j x_j^s)}$.

Case 4: if $l_i^s = \frac{x_i^s}{1+\exp(-f'_i x_i^s)}$ and $l_j^s = \frac{-x_j^s}{1+\exp(f'_j x_j^s)}$, then $\|l_i^s - l_j^s\| \leq \sqrt{\|2A_s\|^2 + 4}$.

This case is similar to **Case 3**.

We observe that there is at most one value of s , $1 \leq s \leq q$, belonging to **Case 3** or **Case 4** in which $\|l_i^s - l_j^s\| \leq \sqrt{\|2A_s\|^2 + 4}$. Therefore, from the triangle inequality:

$$\left\| \sum_{s=1}^q l_i^s - l_j^s \right\| \leq \sum_{s=1}^q \sqrt{\|A_s\|^2 + 4} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}$$

Therefore, the lemma follows. $\square$

Finally, we can bound the sensitivity of $f^* = \arg \min_f J(f; \mathcal{D})$ by the following lemma.

Lemma 5.3. *The ℓ_2 -sensitivity of $f^* = \arg \min_f J(f; \mathcal{D})$ is at most*

$$\frac{\sum_{s=1}^q \sqrt{4 + \|A_s\|^2} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}}{n\Lambda}$$

Proof. Without loss of generality, we assume that two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ are different at n -th patient with $(x_n, y_n, t_n) \in \mathcal{D}$ and $(x'_n, y'_n, t'_n) \in \mathcal{D}'$.

Let $G(f) = J(f; \mathcal{D})$, $g(f) = J(f; \mathcal{D}') - J(f; \mathcal{D}) = \frac{1}{n}(\ell(f; d'_n) - \ell(f; d_n))$, $f_1 = \arg \min_f J(f; \mathcal{D})$, and $f_2 = \arg \min_f J(f; \mathcal{D}')$. Because $\frac{1}{2}\|f\|^2$ is 1-strongly convex, $G(f) = J(f; \mathcal{D})$ is Λ -strongly convex and $G(f) + g(f) = J(f; \mathcal{D}')$ is also Λ -strongly convex. From Lemma 5.2,

$$\begin{aligned} \|\nabla g(f)\| &= \left\| \frac{1}{n} (\nabla \ell(f; d'_n) - \nabla \ell(f; d_n)) \right\| \\ &\leq \frac{\sum_{s=1}^q \sqrt{4 + \|A_s\|^2} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}}{n} \end{aligned}$$

From Lemma 5.1,

$$\|f_1 - f_2\| \leq \frac{1}{\Lambda} \frac{\sum_{s=1}^q \sqrt{4 + \|A_s\|^2} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}}{n}$$

Therefore, the lemma follows. $\square$

Theorem 5.1. *Algorithm 5.1 is ϵ -differentially private.*

Proof. For any pair of neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ and for any f_{priv} , we have:

$$\frac{\text{pdf}(f_{priv} \mid \mathcal{D})}{\text{pdf}(f_{priv} \mid \mathcal{D}')} = \frac{\text{pdf}(b_1)}{\text{pdf}(b_2)} = \exp(-\epsilon/t(\|b_1\| - \|b_2\|))$$

where b_1 and b_2 are the corresponding noise vectors at Line 4 in Algorithm 5.1 with respect to the datasets $\mathcal{D}$ and $\mathcal{D}'$. If f_1^* (resp., f_2^*) is the solution at Line 2 of Algorithm 5.1 on the dataset $\mathcal{D}$ (resp., $\mathcal{D}'$), then $f_1^* + b_1 = f_2^* + b_2 = f_{priv}$. From

Algorithm 5.2 $\mathcal{A}_{\text{Ext-Obj-Pert}}$: Extended Objective Perturbation Approach

Input: Dataset $\mathcal{D} = \{d_1, \dots, d_n\}$, objective function $J(f; \mathcal{D})$, privacy budget ϵ , parameter Λ

Output: f^*

- 1: $\Delta \leftarrow 0$
- 2: Compute $\epsilon' \leftarrow \epsilon - 2 \sum_{s=1}^q \log \left(1 + \frac{\frac{1}{4} \sqrt{\|A_s\|^2 + 1}}{n(\Lambda + \Delta)} \right)$
- 3: **if** $\epsilon' < \epsilon/2$ **then**
- 4:   Binary search value of Δ such that $2 \sum_{s=1}^q \log(1 + \frac{\frac{1}{4} \sqrt{\|A_s\|^2 + 1}}{n(\Lambda + \Delta)}) = \epsilon/2$ and set $\epsilon' \leftarrow \epsilon/2$
- 5: **end if**
- 6: Compute $t \leftarrow \sum_{s=1}^q \sqrt{4 + \|A_s\|^2} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}$
- 7: Sample a random vector b such that $\text{pdf}(b) \propto \exp(-\epsilon' \|b\|/t)$
- 8: $f^* \leftarrow \arg \min_f J(f; \mathcal{D}) + \frac{1}{n} \langle b, f \rangle + \frac{1}{2} \Delta \|f\|^2$
- 9: Output f^*

Lemma 5.3 and triangle inequality:

$$\|b_1\| - \|b_2\| \leq \|b_1 - b_2\| = \|f_1 - f_2\| \leq t$$

where $t = \frac{\sum_{s=1}^q \sqrt{4 + \|A_s\|^2} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2 + 4}}{n \cdot \Lambda}$. Therefore, $\frac{\text{pdf}(b_1)}{\text{pdf}(b_2)} \leq \exp(\epsilon)$. Thus, Algorithm 5.1 is ϵ -differentially private. $\square$

5.4 Extended Objective Perturbation Approach

In this section, we present our solution based on the Objective Perturbation approach proposed in [24]. Similar to the Extended Objective Perturbation approach, we also consider the objective function as described in Equation (5.5). In this approach, instead of adding noise to the solution of the optimization problem as the output perturbation does, it adds noise to the objective function.

5.4.1 Proposed Algorithm

Algorithm 5.2 shows the solution in pseudo-code. At Line 2, we compute ϵ' which is used to calibrate the magnitude of a random variable b . Here, the

regularization parameter is equal to Λ . At Line 3, if $\epsilon' < \epsilon/2$, then it indicates that Λ is not large enough. In this case, an additional positive regularization parameter Δ is picked to set the value of ϵ' to $\epsilon/2$ (Line 4). At Line 5, we compute t which is the sensitivity of $\nabla J(f; \mathcal{D})$. Line 6 samples a random vector b using the same sampling method described in Subsection 5.3.1. Lines 7-8 return the solution of the noisy objective function using the BFGS optimization algorithm [21].

5.4.2 Privacy Guarantee

In this subsection, we will prove that the probability density of f^* from Algorithm 5.2 satisfies the differential privacy definition.

Theorem 5.2. *Algorithm 5.2 is ϵ -differentially private.*

Proof. The noisy objective function from Algorithm 5.2 is:

$$f^* = \arg \min_f \frac{1}{n} \sum_{i=1}^n \ell(f; d_i) + \frac{1}{n} \langle b, f \rangle + \frac{1}{2} (\Lambda + \Delta) \|f\|^2$$

Due to the convexity of $\ell(\cdot)$, the gradient is zero at the minimal point f^* , equivalently, we have:

$$b = -n(\Lambda + \Delta)f^* - \sum_{i=1}^n \nabla \ell(f^*; d_i)$$

Due to the strongly convexity of the objective function, there is a bijective (injective and surjective) mapping from f to b (denoted as $f \rightarrow b$). Therefore, we can transform the probability density function of random variable f to the probability density function of random variable b by a multiplication factor of the Jacobian determinant [17]. From that, the probability density ratio in differential

privacy can be rewritten as:

$$\frac{\text{pdf}(f \mid \mathcal{D})}{\text{pdf}(f \mid \mathcal{D}')} = \frac{\text{pdf}(b \mid \mathcal{D})}{\text{pdf}(b' \mid \mathcal{D}')} \cdot \frac{|\det(\text{Jacob}(f \rightarrow b \mid \mathcal{D}))|^{-1}}{|\det(\text{Jacob}(f \rightarrow b' \mid \mathcal{D}'))|^{-1}} \quad (5.6)$$

We first bound the ratio of the Jacobian determinants. Without loss of generality, we assume that the two datasets $\mathcal{D}$ and $\mathcal{D}'$ are different at n -th record with $d_n \in \mathcal{D}$ and $d_{n'} \in \mathcal{D}'$. Let

$$A = -\text{Jacob}(f \rightarrow b \mid \mathcal{D}) = n(\Lambda + \Delta)\mathbb{I} + \sum_{i=1}^n \nabla^2 \ell(f^*; d_i)$$

and $E = \nabla^2 \ell(f^*; d_n) - \nabla^2 \ell(f^*; d'_{n'})$, then

$$\frac{|\det(\text{Jacob}(f \rightarrow b \mid \mathcal{D}))|^{-1}}{|\det(\text{Jacob}(f \rightarrow b' \mid \mathcal{D}'))|^{-1}} = \frac{|\det(A + E)|}{|\det(A)|} = |\det(\mathbb{I} + A^{-1}E)|$$

Moreover, $E = \sum_{s=1}^q E_n^s - \sum_{s=1}^q E_{n'}^s$, where

$$E_n^s = \begin{cases} \frac{(x_n^s)(x_n^s)'}{(1+\exp(f'x_n^s))(1+\exp(-f'x_n^s))}, & \text{if } s < t_n \\ \frac{-y_n^2(x_n^s)(x_n^s)'}{(1+\exp(y_n f'x_n^s))(1+\exp(-y_n f'x_n^s))}, & \text{if } s = t_n \\ 0, & \text{if } s > t_n \end{cases}$$

Similarly, we can define $E_{n'}^s$ by replacing n by n' . From [94], for any square matrices A and B , we have:

$$\det(\mathbb{I} + A + B) \leq \det(\mathbb{I} + |A|) \cdot \det(\mathbb{I} + |B|)$$

where $|A| = (A'A)^{\frac{1}{2}}$. Moreover, $A^{-1}E_n^s$ and $A^{-1}E_{n'}^s$ are symmetric. Thus, we have:

$$\det(\mathbb{I} + A^{-1}E) \leq \prod_{s=1}^q \det(\mathbb{I} + A^{-1}E_n^s) \cdot \det(\mathbb{I} + A^{-1}E_{n'}^s)$$

We now prove that $|\det(\mathbb{I} + A^{-1}E_n^s)| \leq 1 + \frac{\frac{1}{4}\sqrt{\|A_s\|^2+1}}{n(\Lambda+\Delta)}$. Because

$$\left| \frac{-y_n^2}{(1 + \exp(y_n f' x_n^s))(1 + \exp(-y_n f' x_n^s))} \right| \leq \frac{1}{4}$$

and E_n^s is either a zero matrix or 1-rank matrix. Hence, the only non-zero eigenvalue of E_n^s , if exists, satisfies $|\lambda_1(E_n^s)| \leq \frac{1}{4}\|x_n^s\| \leq \frac{1}{4}\sqrt{\|A_s\|^2+1}$. As the objective function is $(\Lambda + \Delta)$ -strongly convex, A is a full-rank matrix with each eigenvalue greater than $n(\Lambda + \Delta)$. Therefore, $|\det(\mathbb{I} + A^{-1}E_n^s)| \leq 1 + \frac{\frac{1}{4}\sqrt{\|A_s\|^2+1}}{n(\Lambda+\Delta)}$. Similarly, $|\det(\mathbb{I} + A^{-1}E_{n'}^s)| \leq 1 + \frac{\frac{1}{4}\sqrt{\|A_s\|^2+1}}{n(\Lambda+\Delta)}$. Therefore,

$$\frac{|\det(\text{Jacob}(f \rightarrow b) \mid \mathcal{D})|^{-1}}{|\det(\text{Jacob}(f \rightarrow b') \mid \mathcal{D}')|^{-1}} \leq \exp\left(2 \sum_{s=1}^q \log\left(1 + \frac{\frac{1}{4}\sqrt{\|A_s\|^2+1}}{n\Lambda}\right)\right) \quad (5.7)$$

Next, we bound the ratio of the probability density of random vector b with respect to two neighboring datasets. We have:

$$b - b' = \nabla \ell(f^*; d_n) - \nabla \ell(f^*; d'_n)$$

From Lemma 5.2,

$$\|b - b'\| \leq \sum_{s=1}^q \sqrt{\|A_s\|^2+4} + \max_{s \in \{1, \dots, q\}} \sqrt{\|2A_s\|^2+4}$$

Therefore,

$$\frac{\text{pdf}(b \mid \mathcal{D})}{\text{pdf}(b' \mid \mathcal{D}')} \leq \exp(\epsilon' \|b - b'\|/t) \leq \exp(\epsilon') \quad (5.8)$$

From Equation (5.6), Equation (5.7), Equation (5.8), and $\epsilon = \epsilon' + 2 \sum_{s=1}^q \log\left(1 + \frac{\frac{1}{4}\sqrt{\|A_s\|^2+1}}{n\Lambda}\right)$, the theorem follows. $\square$

5.5 Proposed Sampling Approach

In this section, we propose a solution which guarantees differential privacy by directly sampling a random output from a modified version of the posterior distribution. In this work, we pick a normal distribution as the prior distribution. This is equivalent to using:

$$\mathcal{U}(f; \mathcal{D}) = -\frac{1}{2}\sigma\|f\|^2 - \sum_{i=1}^n \ell(f; d_i)$$

as the utility function in the *exponential mechanism* [76] where the parameter σ is used to control the variance of the prior normal distribution. Then, the differentially private output is sampled from the following distribution:

$$\text{pdf}(f) \propto \exp\left(\frac{\epsilon\mathcal{U}(f; \mathcal{D})}{2\Delta_{\mathcal{U}}}\right)$$

where $\Delta_{\mathcal{U}} = \max_{d(\mathcal{D}, \mathcal{D}')=1, f} \|\mathcal{U}(f; \mathcal{D}) - \mathcal{U}(f; \mathcal{D}')\|$ is the sensitivity of $\mathcal{U}$. The reason why we pick a normal prior distribution instead of a uniform prior distribution is not because our proposed solution requires that in order to guarantee differential privacy, but instead we observe that with a normal prior distribution the sampling algorithm converges better and is more stable.

Moreover, this approach requires the utility function $\mathcal{U}(f; \mathcal{D})$ has to have a bounded sensitivity. However, the loss function $\ell(\cdot)$ is not bounded. Therefore, the function $\mathcal{U}(f; \mathcal{D})$ has unbounded sensitivity. In order to overcome this difficulty, we propose a smooth sanitizer function $C(x)$ which is used to control the maximum value of the loss function $\ell(\cdot)$. The definition of $C(x)$ is given as follows:

$$C_v(x) = v \cdot \tanh\left(\frac{x}{v}\right) \tag{5.9}$$

Algorithm 5.3 $\mathcal{A}_{\text{Sanitized-EXP}}$: Sanitized Loss Mechanism

Input: Dataset $\mathcal{D} = \{d_i\}_{i=1}^n$, loss function $\ell(f; d_i)$, privacy budget ϵ , maximum value v , parameter Λ

Output: f

- 1: $\mathcal{U}(f; \mathcal{D}) = -\frac{1}{2}\sigma\|f\|^2 - \sum_{i=1}^n C_v(\ell(f; d_i))$
- 2: Output a random vector f with the probability density

$$\text{pdf}(f) \propto \exp\left(\frac{\epsilon}{2v}\mathcal{U}(f; \mathcal{D})\right)$$

We now take the composition of $C_v(\cdot)$ with $\ell(f; d_i)$ to have a bounded-sensitivity utility function:

$$\mathcal{U}(f; \mathcal{D}) = -\frac{1}{2}\sigma\|f\|^2 - \sum_{i=1}^n C_v(\ell(f; d_i))$$

We intentionally pick the $\tanh(\cdot)$ function as the sanitizer because it nicely keeps the loss function in its original form when the value of the loss function is near 0. Meanwhile, it deforms the loss function at large values to make the function finite. The advantage of this approach is that the sampled parameter can be arbitrary large while the objective function is kept almost the same around the optimal parameter which maximizes the posterior probability. We present the pseudo-code of our approach in Algorithm 5.3.

Theorem 5.3 (Privacy guarantee). *Algorithm 5.3 is ϵ -differentially private.*

Proof. For two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$,

$$\Delta_{\mathcal{U}} = \max_{f, d(\mathcal{D}, \mathcal{D}')=1} |\mathcal{U}(f; \mathcal{D}) - \mathcal{U}(f; \mathcal{D}')| \leq v$$

Thus, at any point f , we have:

$$\begin{aligned} \frac{\text{pdf}(f \mid \mathcal{D})}{\text{pdf}(f \mid \mathcal{D}')} &= \frac{\exp\left(\frac{\epsilon}{2v}\mathcal{U}(f; \mathcal{D})\right) / \int \exp\left(\frac{\epsilon}{2v}\mathcal{U}(f; \mathcal{D})\right) df}{\exp\left(\frac{\epsilon}{2v}\mathcal{U}(f; \mathcal{D}')\right) / \int \exp\left(\frac{\epsilon}{2v}\mathcal{U}(f; \mathcal{D}')\right) df} \\ &\leq \exp\left(\frac{2\epsilon}{2v} |\mathcal{U}(f; \mathcal{D}) - \mathcal{U}(f; \mathcal{D}')|\right) \\ &\leq \exp(\epsilon) \end{aligned}$$

Therefore, Algorithm 5.3 is ϵ -differentially private. $\square$

Algorithm 5.4 $\mathcal{A}_{\text{pSGLD}}$: pSGLD Sampling Algorithm

Input: Dataset $\mathcal{D} = \{d_i\}_{i=1}^n$, loss function ℓ , privacy parameter ϵ , μ , λ , k , bounded value v and learning rate τ

Output: f^{T+1}

```

1:  $V_0 \leftarrow \vec{0}, f_1 \leftarrow \vec{0}$ 
2: for  $t = 1$  to  $T$  do
3:   Compute  $\epsilon_t \leftarrow t^{-\tau}$ 
4:   Uniformly sample  $\Omega_k^t = \{d_{t_1}, \dots, d_{t_k}\} \subset \mathcal{D}$ 
5:   Compute  $\bar{g}^t = \frac{\epsilon}{2v}(\frac{\sigma f^t}{n} + \frac{1}{k} \sum_{i=1}^k \nabla C_v(\ell(f^t, d_{t_i})))$ 
6:    $V^t \leftarrow \mu V^{t-1} + (1 - \mu)(\bar{g}^t \odot \bar{g}^t)$ 
7:    $G^t \leftarrow 1/(\lambda \mathbb{I} + \text{diag}(\sqrt{V^t}))$ 
8:    $f^{t+1} \leftarrow f^t - \epsilon_t (G^t \cdot n \bar{g}^t) + \mathcal{N}(0, \epsilon_t G^t)$ 
9: end for
10: Output  $f^{T+1}$ 

```

The problem with Algorithm 5.3 is that there is no run-time efficient algorithm to sample the distribution of f exactly. Bassily et al. [15] proposed a polynomial run-time sampling algorithm. However, their proposed algorithm is still impractical due to the high degree of the polynomial run-time complexity and only apply for the log-convex function. Recently, there are new developments [25, 6, 74] in the Markov Chain Monte Carlo (MCMC) method which can be applied to machine learning problems with large datasets. The idea is to construct Markov chain to simulate dynamical systems with stochastic gradients. At each step, we compute the gradient at the current location, then add a controlled amount of noise to the gradient and follow the noisy gradient to a new location. Asymptotically, the stationary distribution of this process converges to the true distribution from which the gradient is computed.

In this work, we propose to use an MCMC sampling algorithm, namely Preconditioned Stochastic Gradient Langevin Dynamics (pSGLD) [71], to approximately sample the posterior distribution. pSGLD is good at sampling variables with difference in scale which is useful for our problem because the parameter α is usually much larger in magnitude than the parameter β (recall that $f = [\alpha, \beta]'$). The pseudo-code of pSGLD is described in Algorithm 5.4. At Line 1, we initialize

Table 5.1: Statistics of the datasets.

Dataset	Size	#uncensored	#explanatory variables
FL	7874	2169	6
WT	21685	18615	3

the values of V_0 and f_1 . Line 3 computes the learning rate ϵ_t . It is required that $\lim_{t \rightarrow \infty} \sum_t \epsilon_t \rightarrow \infty$ and $\lim_{t \rightarrow \infty} \sum_t \epsilon_t^2 < \infty$ to guarantee the convergence. We sample uniformly k records from $\mathcal{D}$ for estimating the average gradient $\bar{g}^t$ (Line 5). We then compute the variance of the gradient at Line 6 ($\odot$ is the element-wise product) and convert it to the preconditioned matrix G^t at Line 7. We update the parameter at Line 8 with a noise variable $\mathcal{N}(0, \epsilon_t G^t)$. It is worth noting that there is a permanent bias in pSGLD due to excluding a correction term in the updating step (Line 8). However, this bias is negligible and excluding the correction term helps to speed up the sampling algorithm which then helps to reduce the finite-sample bias as more steps are executed in a finite amount of time.

5.6 Experimental Evaluation

In this section, we present the results of our experiments on real datasets. We focus on answering the following important research questions:

- What is the trade-off between privacy and accuracy of our proposed approaches as compared to the non-private estimation?
- Does the sampling approach converge to its stationary distribution?

5.6.1 Datasets

In our experiments, we use two datasets: *FLchain dataset* (FL) and *Wichert dataset* (WT) for training regression models. We do not use the other two datasets *time-to-second-birth* (SB) and *time-to-third-birth* (TB) because we observed that their

explanatory variables are not good in predicting survival rate. Table 5.1 gives the statistics of the two datasets used in the experiments.

- *The FLchain dataset* (FL) - The explanatory variables are age, sex, kappa, lambda, creatinine and mgus, where kappa, lambda, creatinine and mgus are different biological measurements related to human blood.
- *The Wichert dataset* (WT) - The explanatory variables are sex, age and wage.

The survival times in these datasets are normalized to the interval $[0, 1]$. All the vectors of the explanatory variables are normalized to have zero mean and fitted inside the unit sphere.

5.6.2 Trade-off between Privacy and Accuracy

In this subsection, we investigate the trade-off between privacy and accuracy in our proposed approaches. We use the proposed approaches (i.e., extended output perturbation Ext-Out-Pert, extended objective perturbation Ext-Obj-Pert and posterior sampling pSGLD) to estimate regression coefficients and use these regression coefficients to evaluate the performance of our approaches.

Compared approaches: We compare the experimental results from four approaches:

- MLE - It refers to the non-private maximum likelihood estimation.
- Ext-Out-Pert - It refers to the extended output perturbation approach (Algorithm 5.1).
- Ext-Obj-Pert - It refers to the extended objective approach (Algorithm 5.2).
- pSGLD - It refers to the posterior sampling approach (Algorithm 5.4).

Performance metric: To evaluate the predictive performance of regression survival models, we perform the prediction task on whether a death event would

Figure 5.2: The performance results (in AUC-ROC) of our proposed approaches with privacy budget ϵ from 0.1 to 1.6.

happen in time interval $[0, t_{med}]$, where t_{med} is the median of survival times in the dataset. We report the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) [90] scores to show the effectiveness of the survival models in predicting death events. Higher AUC-ROC score will indicate a better model in predicting the death events. Moreover, we use a 5-fold cross-validation to get the average performance.

Hyperparameters: We set the number of discrete-time intervals $q = 5$. We use the natural cubic spline with $e = 3$ knots to model the baseline hazard effect. For Algorithm 5.1 and Algorithm 5.2, we set the parameter $\Lambda = 10^{-2}$. For Algorithm 5.4, we set the parameters $\tau = 0.51$, $\lambda = 10^{-5}$ and $\mu = 0.99$. We also set the parameter $\sigma = 10^{-2} \cdot 2v/\epsilon$. At each step of the Markov chain, we randomly pick $k = 200$ records from the dataset to compute the gradient. The loss function is bounded by the value $v = \log(n)$ where n is the size of the dataset.

Experimental results: The results are shown in Figure 5.2. Overall, pSGLD outperforms both Ext-Out-Pert and Ext-Obj-Pert approaches. Moreover, we observe that the accuracy of Ext-Out-Pert and Ext-Obj-Pert does not improve much at high privacy budgets. It is due to the regularization term in the objective function that causes the output parameter moving towards the zero vector instead of the optimal parameter. Because of the regularization effect, Ext-Out-Pert and

Ext-Obj-Pert cannot reach the same performance as MLE at privacy budget $\epsilon = 1.6$. Whereas the proposed sampling approach (pSGLD) does not suffer from such effect. pSGLD has almost the same performance as MLE at privacy budget $\epsilon = 1.6$. At small privacy budget $\epsilon = 0.1$, pSGLD still performs much better than Ext-Out-Pert and Ext-Obj-Pert.

5.6.3 Convergence of the Proposed Sampling Approach

This subsection reports on the convergence of our proposed sampling approach. It aims to check whether the proposed sampling approach converges to the stationary distribution.

Statistical test: We use the mean relative error (MRE) as a statistical measure on the convergence of the sampling approach (Algorithm 5.4). MRE is defined as follows:

$$\text{MRE} = \frac{1}{t} \sum_{i=1}^t \frac{\|f_i - f^*\|}{\|f^*\|} \quad (5.10)$$

where f_i is the parameter vector from the sampling process, f^* is the optimal parameter vector which maximizes the likelihood in non-private setting and t is the number of samples.

Experimental results: We plot the MRE as the function of epochs with three different privacy budgets which is shown in Figure 5.3. Each epoch is a bundle of n steps, where n is the size of the dataset. We observe that after 250 epochs, MRE becomes stable which indicates that the sampling approach converges to its stationary distribution.

Figure 5.3: The convergence (in MRE) of the pSGLD sampling algorithm.

5.7 Summary

In this work, we have proposed solutions for the problem of protecting differential privacy for discrete-time regression models used in survival analysis. In particular, we extend the perturbation approaches to a generalized form in which the loss function is a sum of logistic loss functions. In addition, we propose a sampling approach to practically protect differential privacy by sampling a scaled posterior distribution with the pSGLD sampling algorithm. Experimental results have shown that our proposed approaches are promising for discrete-time survival regression. Though we focus our work on discrete-time survival regression, our proposed approaches can be applied to other problems with similar loss functions as well.

Chapter 6

Continuous-time Survival Regression

In Chapter 5, we have proposed differentially private mechanisms for discrete-time survival models as the alternative models for the Cox regression model. However, the proposed approaches require the discretization of time values which causes loss of information and reduces the predictive performance of the model. To tackle this problem, in this chapter, we propose two private mechanisms for flexible parametric models with continuous-time survival models instead of discrete-time survival models as proposed in the previous chapter. We focus on flexible parametric models which are powerful alternatives to the well-known Cox regression model. For the first approach, we propose a sampling mechanism which guarantees differential privacy for flexible parametric survival models. For the second approach, we propose a mechanism which protects privacy by randomly perturbing the posterior distribution such that in the worst case scenario, when all data records are revealed except the target data record, the random noise added to the posterior would make it extremely difficult to obtain the target data record. The experiments conducted on two real datasets show that our proposed approaches outperform state-of-the-art methods in predicting the survival rate of individuals.

6.1 Continuous-time Survival Regression

We focus on two flexible parametric survival models [11, 70, 93]: (i) the proportional hazard models, and (ii) the proportional odds models. These models are powerful alternatives to the well-known Cox regression model [28, 10, 9]. An advantage of the Cox regression model is that we do not need to define a full likelihood function to estimate the parameters of the model. However, it makes prediction difficult as a full likelihood function is generally needed for predicting the survival rate of a new subject. Moreover, the lack of a full likelihood function is also a major obstacle in developing a differentially private mechanism for the Cox regression model. A partial likelihood function, which is used in the Cox regression model, mixes all the data records together. As such, it is very challenging for developing a differentially private mechanism as we aim to control the impact of each individual data record to the output model. In contrast, flexible parametric models define a full likelihood function in which each data record is independent of the other records in the dataset.

6.1.1 Proportional Hazard Models

Let T be a random variable representing the survival time of a subject. For a continuous-time survival regression model, the survival rate depends on covariates (or explanatory variables) such as gender, age, blood pressure, etc. Let's denote these covariates by a d -dimensional real-valued vector x_i . In the model, the survival function and hazard function depend on x_i . Therefore, they can be denoted as $S(t; x_i)$ and $h(t; x_i)$. From that, we derive a generic log-likelihood function as follows:

$$\ell(\beta; x_i, t_i, d_i) = \log \left(h(t_i; x_i)^{d_i} S(t_i; x_i) \right) \quad (6.1)$$

where β is the parameter of the model, d_i is an indicator variable whose value is equal to 1 when the subject is dead at time t_i , and equal to 0 when the subject is alive at time t_i . Traditionally, maximum likelihood estimator (MLE) is applied to estimate a reasonable value for the parameter β :

$$\hat{\beta} \leftarrow \arg \max_{\beta} \sum_{i=1}^n \ell(\beta; x_i, t_i, d_i)$$

where n is the size of the dataset. However, we cannot use MLE directly to protect the privacy of individuals in the dataset as it does not control the influence of each data record to the estimated parameter $\hat{\beta}$.

The proportional hazard models (which include the well-known Cox regression) assume that each unit increase in the value of an explanatory variable will induce a multiplicative risk of death to the subject:

$$h(t; x_i) = h_0(t) \cdot \exp(\beta^T x_i)$$

where β is the parameter vector of the model and $h_0(t)$ is the *baseline hazard function* (the risk of death with $x_i = 0$). From this assumption, we can derive a partial likelihood function which is a function of β and does not depend on $h_0(t)$. This allows Cox regression to estimate β without the need of specifying the baseline hazard function $h_0(t)$. However, in this chapter we will use a full likelihood function instead of a partial likelihood function, which is very useful in predicting the survival rate of new subjects. To get a full likelihood function, we propose to use *I-spline* [91] to approximate the non-decreasing function $\log(H(t))$ as follows:

$$\log(H(t; x_i)) = \zeta_i = s(t_i; \gamma) + \beta^T x_i$$

where $s(\cdot; \gamma)$ is an I-spline function with parameter γ :

$$s(x; \gamma) = \gamma_0 + \gamma_1 v_1(x) + \gamma_2 v_2(x) + \cdots + \gamma_m v_m(x) \quad (6.2)$$

where $v_j(x)$ is the j -th ($1 \leq j \leq m$) basis function of I-spline. Since the basis functions $v_j(\cdot)$ are non-decreasing functions, with constraints $\forall i > 1 : \gamma_i \geq 0$, it is guaranteed that $s(x; \gamma)$ is a non-decreasing function. Hence, we guarantee that $H(t; x_i)$ is also a non-decreasing function.

From there, we derive the flexible parametric forms of the survival function:

$$S(t; x_i) = \exp(-\exp(\zeta_i)) \quad (6.3)$$

and the hazard function:

$$h(t; x_i) = \frac{ds(t; \gamma)}{dt} \cdot \exp(\zeta_i) \quad (6.4)$$

6.1.2 Proportional Odds Models

Proportional odds models assume that a unit increase of an explanatory variable will induce a multiplicative *odds* of dying:

$$\frac{1 - S(t_i; x_i)}{S(t_i; x_i)} = \frac{1 - S_0(t_i)}{S_0(t_i)} \cdot \exp(\beta^T x_i)$$

Since the baseline cumulative odds function $\frac{1 - S_0(t_i)}{S_0(t_i)}$ is a non-decreasing function, we can also use I-spline to approximate the function:

$$\log \left(\frac{1 - S(t_i; x_i)}{S(t_i; x_i)} \right) = \zeta_i = s(t_i; \gamma) + \beta^T x_i$$

where $s(t_i; \gamma)$ is the I-spline approximate of $\log \left(\frac{1 - S_0(t_i)}{S_0(t_i)} \right)$. As a result, the flexible parametric form of the survival function is:

$$S(t_i; x_i) = \frac{1}{1 + \exp(\zeta_i)} \quad (6.5)$$

Also, the flexible parametric form of the hazard function is:

$$h(t_i; x_i) = \frac{ds(t_i; \gamma)}{dt} \cdot \frac{\exp(\zeta_i)}{1 + \exp(\zeta_i)} \quad (6.6)$$

6.2 Problem Definitions

In this section, we give a formal definition of the problems we aim to address in this chapter.

Problem 6.1: Given a dataset $\mathcal{D}$ and privacy budget ϵ , release the parameters of the proportional hazard models and proportional odds models such that they satisfy ϵ -differential privacy.

This problem focuses on protecting differential privacy for continuous-time survival regression for both proportional hazard models and proportional odds models.

Problem 6.2: Given a dataset $\mathcal{D}$, release the posterior distribution of the parameters of the proportional hazard models and proportional odd models such that the privacy of individuals in the dataset are protected.

This problem addresses the scenarios in which differential privacy cannot practically guarantee privacy with high accuracy for estimations of parameters and their uncertainty.

6.3 Proposed Posterior Sampling Approach

Our first proposed posterior sampling approach (PSA) guarantees differential privacy by sampling a random parameter vector from the posterior distribution of a modified model. This approach is adapted from the sampling approach proposed in the previous chapter for discrete-time survival regression models. Since our models use continuous time values directly, it solves the problem of discretization. In order to guarantee DP, we need to control the contribution of

Figure 6.1: An illustration of the sanitizer functions with a maximum value of 0 and minimum value of -3 . $C_1(\cdot)$ is our proposed sanitizer function, while $C_2(\cdot)$ was previously proposed in Chapter 5.

each data record to the full likelihood function. As the likelihood function of a data record can receive any value in the real domain (i.e. unbounded value), we propose to use a sanitizer function, denoted as $C(\cdot)$, with bounded maximum and minimum values. The sanitizer function $C(\cdot)$ is then combined with the log-likelihood function (denoted as $\ell(\cdot)$) to create a modified log-likelihood function $\ell' = C \circ \ell$. Figure 6.1 illustrates two sanitizer functions with the maximum value of 0 and the minimum value of -3 . Our proposed sanitizer function $C_1(\cdot)$ keeps the log-likelihood function unchanged for values inside the interval $[-3, 0]$, while the sanitizer function $C_2(\cdot)$ proposed in Chapter 5 (Equation (5.9)) deforms the log-likelihood function at all values.

Our differentially private mechanism needs to publish a random sample from the posterior distribution to guarantee DP. Here, we propose to use the relativistic Hamiltonian Monte-Carlo (RHMC) [73] sampler which is designed to be robust to different parameter settings. RHMC uses the relativistic mechanics to create a proposal distribution which can achieve high mixing rate as compared to other Metropolis-Hastings algorithms [55, 51].

Let's denote $\theta = (\beta, \gamma)^T$ as the parameter vector of our model where β is the coefficient vector of the explanatory variables and γ is the coefficient vector of the

spline function $s(\cdot)$. Let $\pi(\theta)$ be the prior distribution of θ . In order to guarantee DP, we use the exponential mechanism [76] as the framework for our approach. The exponential mechanism requires to define a utility function $U(\theta, \mathcal{D})$ and sample a random output from a probability distribution with density:

$$\text{pdf}(\theta) \propto \exp \left(\epsilon \cdot \frac{U(\theta, \mathcal{D})}{2\Delta_U} \right) \quad (6.7)$$

where $\Delta_U = \max_{\theta, \mathcal{D}, \mathcal{D}'} |U(\theta, \mathcal{D}) - U(\theta, \mathcal{D}')|$ for any neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. Then, a random sample of θ can be proven to be ϵ -DP (see Theorem 3.1). Based on this framework, for flexible parametric models, we define the utility function $U(\cdot)$ as follows:

$$C(x; \eta) = \frac{\epsilon}{2\eta} \times \begin{cases} x & \text{if } -\eta \leq x \leq 0 \\ 0 & \text{if } x > 0 \\ -\eta & \text{if } x < -\eta \end{cases} \quad (6.8)$$

We observe that $\Delta_U = \frac{\epsilon}{2}$. Moreover,

$$\begin{aligned} \text{pdf}(\theta) &\propto \exp \left(\epsilon \cdot \frac{U(\theta, \mathcal{D})}{2\Delta_U} \right) \\ &= \exp \left(\sum_{i=1}^n C \left(\ell(\theta; x_i, t_i, d_i) - d_i \cdot \frac{\eta}{2}; \eta \right) \right) \times \pi(\theta) \end{aligned}$$

defines a posterior distribution from the prior $\pi(\theta)$ and the modified log-likelihood function $C(\ell(\theta; x_i, t_i, d_i) - d_i \cdot \frac{\eta}{2}; \eta)$. As illustrated in Figure 6.1, the sanitizer function keeps the shape of the likelihood function unchanged at important regions where likelihood values are high. The scaling factor $\frac{\epsilon}{2\eta}$ guarantees that the modified log-likelihood function is bounded by $[-\epsilon/2, 0]$ which is necessary for the proof of ϵ -DP protection. The log-likelihood $\ell(\theta; x_i, t_i, d_i)$ can be positive when $d_i = 1$, whereas the sanitizer function $C(\cdot; \eta)$ sets all positive values to zero. Therefore, we offset the log-likelihood $\ell(\theta; x_i, t_i, d_i)$ by a constant $-\eta/2$ when

$d_i = 1$ to prevent losing information due to trimming.

Theorem 6.1. *A random sample from the probability distribution*

$$\text{pdf}(\theta, \mathcal{D}) \propto \exp(\epsilon U(\theta, \mathcal{D})/2\Delta_U)$$

where $\Delta_U = \epsilon/2$, guarantees ϵ -DP.

Proof. Considering the probability density ratio of two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, we have:

$$\begin{aligned} \frac{\text{pdf}(\theta, \mathcal{D})}{\text{pdf}(\theta, \mathcal{D}')} &= \frac{\exp(\epsilon U(\theta, \mathcal{D})/2\Delta_U)}{\int_{\theta} \exp(\epsilon U(\theta, \mathcal{D})/2\Delta_U) d\theta} \bigg/ \frac{\exp(\epsilon U(\theta, \mathcal{D}')/2\Delta_U)}{\int_{\theta} \exp(\epsilon U(\theta, \mathcal{D}')/2\Delta_U) d\theta} \\ &= \frac{\exp(U(\theta, \mathcal{D}))}{\int_{\theta} \exp(U(\theta, \mathcal{D})) d\theta} \bigg/ \frac{\exp(U(\theta, \mathcal{D}'))}{\int_{\theta} \exp(U(\theta, \mathcal{D}')) d\theta} \\ &= \frac{\exp(U(\theta, \mathcal{D}))}{\exp(U(\theta, \mathcal{D}'))} \cdot \frac{\int_{\theta} \exp(U(\theta, \mathcal{D}')) d\theta}{\int_{\theta} \exp(U(\theta, \mathcal{D})) d\theta} \\ &= \frac{\exp(U(\theta, \mathcal{D}))}{\exp(U(\theta, \mathcal{D}'))} \cdot \frac{\int_{\theta} \exp(U(\theta, \mathcal{D})) \cdot \frac{\exp(U(\theta, \mathcal{D}'))}{\exp(U(\theta, \mathcal{D}))} d\theta}{\int_{\theta} \exp(U(\theta, \mathcal{D})) d\theta} \\ &\leq \exp(|U(\theta, \mathcal{D}) - U(\theta, \mathcal{D}')|) \cdot \exp(|U(\theta, \mathcal{D}) - U(\theta, \mathcal{D}')|) \\ &\leq \exp(\epsilon) \end{aligned}$$

Therefore, $\text{pdf}(\theta, \mathcal{D}) \propto \exp(\epsilon U(\theta, \mathcal{D})/2\Delta_U)$ guarantees ϵ -DP by definition. $\square$

To sample the posterior distribution, we use the relativistic Hamiltonian Monte-Carlo (RHMC) sampler which is a random walk process whose stationary distribution converges to the target distribution $\text{pdf}(\theta) \propto \exp(\epsilon U(\theta, \mathcal{D})/2\Delta_U)$. The RHMC sampler has been shown to be robust to different parameter settings [73].

Let $T(\theta) = -U(\theta, \mathcal{D})$ be defined as the potential energy function of a simulated physical system with coordinates θ . The RHMC sampler uses the relativistic mechanism to define a kinetic energy function with momentum p as

follows:

$$K(p) = \sum_{j=1}^d m_j c_j^2 \left(\frac{p_j^2}{m_j^2 c_j^2} + 1 \right)^{\frac{1}{2}} \quad (6.9)$$

where j indexes the coordinates of the d -dimensional system, and each coordinate can have its own mass m_j and speed of light c_j . From that, we derive the Hamiltonian of the physical system:

$$H(\theta, p) = T(\theta) + K(p) \quad (6.10)$$

A useful property of the Hamiltonian system is that we can sample the joint distribution (θ, p) from H and then discard p to get the distribution defined by $T(\theta)$. It is useful to sample from H because we can simulate the dynamical system defined by H to jump from one location to a far away location while guaranteeing to have the same total energy. The formula of the Hamiltonian dynamical system is:

$$\begin{aligned} \dot{\theta} &= \frac{\partial H}{\partial p} = M^{-1}(p) \odot p \\ \dot{p} &= -\frac{\partial H}{\partial \theta} = -\nabla T(\theta) \end{aligned}$$

where $M(p_j) = m_j \left(\frac{p_j^2}{m_j^2 c_j^2} + 1 \right)^{\frac{1}{2}}$ can be interpreted as the relativistic mass and $\odot$ is the element-wise multiplication operator. By simulating this system, we can achieve high mixing rate as compared to other Monte-Carlo methods.

Algorithm 6.1 is the pseudo-code of the RHMC sampling algorithm. We execute T random walks and output a random sample at the final step. In each walk, we first randomly sample the initial momentum p_0 whose coordinates are drawn from the symmetric hyperbolic distribution defined by Equation (6.9). Then, at Lines 3-7, we simulate the dynamics of the physical system from the initial state (θ_0, p_0) to a new state at time step L . At Line 8, we flip the sign of

Algorithm 6.1 $\mathcal{A}_{\text{RHMC}}$: Relativistic Hamiltonian Monte-Carlo Sampling

Input: Dataset $\mathcal{D} = \{x_i, d_i, t_i\}_{i=1}^n$, potential energy $T(\cdot)$, discretization constant δ , number of simulation steps L

Output: A random sample from the posterior

```

1: for  $i = 1$  to  $T$  do
2:    $p_0[j] \sim$  a symmetric hyperbolic distribution ( $1 \leq j \leq d$ )
3:   for  $t = 0$  to  $L - 1$  do
4:      $p_{t+1/2} \leftarrow p_t - \frac{1}{2}\delta \nabla T(\theta_t)$ 
5:      $\theta_{t+1} \leftarrow \theta_t + M^{-1}(p_{t+1/2}) \odot p_{t+1/2}$ 
6:      $p_{t+1} \leftarrow p_{t+1/2} - \frac{1}{2}\delta \nabla T(\theta_{t+1})$ 
7:   end for
8:    $\theta_L \leftarrow -\theta_L$ 
9:    $q \leftarrow \min(1, \exp(-H(\theta_L, p_L) + H(\theta_0, p_0)))$ 
10:   $\text{coin} \sim \text{Bernoulli}(q)$ 
11:  if  $\text{coin} == 1$  then
12:     $\theta_0 \leftarrow \theta_L$ 
13:  end if
14: end for
15: Output  $\theta_0$ 

```

θ obtained at time step L to guarantee that our discrete integral in Lines 3-7 is reversible, which is a requirement for the accept/reject procedure in the subsequent steps to work properly. At Lines 9-13, we accept the new value of θ with probability $\min(1, \exp(-H(\theta_L, p_L) + H(\theta_0, p_0)))$, which should always be near 1 from the invariant in total energy of a Hamiltonian dynamical system. In other words, we would accept a new value with high probability. This explains why the RHMC sampler is very efficient.

6.4 Proposed Posterior Perturbing Approach

A disadvantage of our proposed posterior sampling approach is that we can only get one random sample from the posterior distribution. Consequently, we cannot determine the credible intervals of the estimated parameters. For medical applications, it is very important to have accurate predictions and also to know the confidence (or credible intervals) of the estimated parameters. Additionally,

for small datasets (e.g., rare disease survival datasets), differentially private protection becomes meaningless as we need to consume a very large privacy budget to obtain a reasonably good model. In order to tackle these problems, we propose the posterior perturbing approach (PPA) which allows learning a perturbed version of the posterior distribution. The significant trade-off for accuracy and confidence is that the output now does not guarantee differential privacy. However, it is able to guarantee that it would be extremely difficult to recover the target data record from the posterior even in the worst case scenario when all other data records in the dataset are exposed. In our proposed approach, we first sanitize the log-likelihood function by composing $C(\cdot)$ to the log-likelihood function. Then, to protect the individuals in the dataset in the worst case scenario, we add random functions to the posterior distribution. The noisy functions added to the posterior prevent an adversary to recover the target data record. We can then use the RHMC sampler to draw multiple random samples to estimate the credible intervals of the parameters.

In this approach, we add two kinds of random functions to the posterior:

$$\text{pdf}(\theta) \propto \exp \left(\sum_{i=1}^n C \left(\ell(\theta; x_i, t_i, d_i) - d_i \cdot \frac{\eta}{2}; \eta \right) + \log(\pi(\theta)) + R_1(\theta) + R_2(\theta) \right)$$

where (i) $R_1(\cdot)$ is a model-independent random function generated by averaging many feed-forward neural networks whose weights are randomly assigned, and (ii) $R_2(\cdot)$ is a model-dependent random function which is the summation of many likelihood functions whose data records are randomly generated. Sampling random neural networks $R_1(\cdot)$ is a practical solution to the problem of randomly sampling generic non-linear functions. Meanwhile, the random function $R_2(\cdot)$, created by random data records, is indistinguishable from the true likelihood functions. In this approach, we also use the sanitizer function for the

(a) The sanitized log-likelihood function of a data record. We can easily observe the directional feature of the likelihood.

(b) The same log-likelihood function after adding a noisy prior. We can see that the directional feature is hidden by the noisy prior.

Figure 6.2: Log-likelihood function before and after adding a noisy prior.

likelihood function except that the scaling factor is set to 1:

$$C(x; \eta) = \begin{cases} x & \text{if } -\eta \leq x \leq 0 \\ 0 & \text{if } x > 0 \\ -\eta & \text{if } x < -\eta \end{cases} \quad (6.11)$$

Figure 6.2 illustrates the privacy protection in action in the worst case scenario when all the data records are revealed except a single target record. Figure 6.2a is a likelihood function before adding noisy functions, while Figure 6.2b is the same function after adding noisy functions. In the worst case scenario, when the adversary knew all the records except the target data record, the adversary could simply subtract the Gaussian prior and all the likelihoods of the known data records from the posterior to receive the likelihood of the target data record. If no noisy function is added to the posterior, what the adversary will observe is shown in Figure 6.2a in which the directional feature of the log-likelihood function can be recognized easily. From the directional information, the adversary can easily recover the data record. On the contrary, if we add noisy functions to the posterior, the resultant function after subtraction is the mixing

of the true likelihood function formed by the target record and the noisy random functions as shown in Figure 6.2b. We would like to scramble the posterior enough so that the adversary will not be able to recover the target data record from the noisy function. Interestingly, the summation $\log(\pi(\theta)) + R_1(\theta) + R_2(\theta)$ can be interpreted as a noisy prior distribution of our model which in this case helps protect the privacy of data records.

One may argue that some unique features of the perturbed posterior function in Figure 6.2b can be exploited to fully recover the target data record. In fact, we will not publish the whole perturbed posterior function. We keep all the fake data records and neural networks' weights in private and only evaluate the posterior function at some finite points to provide enough information needed by the RHMC sampler. As such, only a finite number of random samples, which are the results of the sampling process, is published. Therefore, we *hypothesize* that a single bounded-value likelihood function in the presence of a noisy enough prior will not be able to imprint distinguishable statistical features on the results of the sampling process.

Remark: This approach of perturbing the posterior is also helpful in the case when data is distributed across many parties. In this case, each party adds a private noisy function to the posterior. Then, a secure multi-party network protocol can be deployed which allows at each step of the sampling process all the values and gradients needed by the sampler can be aggregated together. The secure protocol will keep the values and gradients of each party private, and only the aggregated values are known. Here, we will not examine this case further.

Figure 6.3: A feed-forward neural network whose network weights are randomly initialized.

6.4.1 Random Neural Networks

As shown in Figure 6.3, we design a feed-forward neural network whose input neurons are the parameter θ and the output neuron is a real-valued noisy likelihood. The first layer of hidden neurons aggregates values from the input layer and passes the results through the $\sin(\cdot)$ function. Then, the second hidden layer aggregates the outputs from the first layer and passes them through the $\tanh(\cdot)$ function. At the output layer, all the values from the second hidden layer are aggregated and passed to the $\tanh(\cdot)$ function. As a result, the output value is in the range of $[-1, 1]$. Let's denote $f_i(\theta, w_i)$ as an instance of our network architecture whose network weights w_i are randomly assigned. Then, the proposed approach adds the following random function to the posterior:

$$R_1(\theta) = \eta \cdot \frac{1}{t} \sum_{i=1}^t f_i(\theta, w_i) \quad (6.12)$$

where t is the number of random neural networks. R_1 is the scaled average of all output values from t random neural networks. Therefore, it is guaranteed that $R_1(\theta)$ is bounded by $[-\eta, +\eta]$ which has the same magnitude as a single sanitized likelihood function. The random neural networks $R_1(\theta)$ added to the posterior function are intentionally designed to scramble the likelihood function

locally, that explains the reason why we pick the periodic function $\sin(\cdot)$ as the non-linear function in the first hidden layer. It allows perturbing the posterior function with roughly the same magnitude of noise at every point in the parameter space. The objective is to make it difficult to recover the data record from locally observed features of the perturbed posterior.

6.4.2 Model-dependent Noisy Prior

We can observe from Section 6.1 that the likelihood functions of the proportional hazard models and proportional odds models are functions of the dot product between the data record and the parameter vector. This explains why the directional feature can clearly be seen in Figure 6.2a. In order to hide such *global* directional feature (which can be easily exploited to recover the data record), we propose a random procedure that adds a model-dependent noise to the posterior. The procedure has two steps: (i) We generate multiple fake data records, which could be random records from a plausible generative model or perturbed public data records whose privacy protections are not necessary. (ii) We pass these fake records to our sanitized likelihood function to generate plausible random likelihoods. These random functions are then added to the posterior:

$$R_2(\theta) = \sum_{i=1}^k C \left(\ell(\theta; x_i, t_i, d_i) - d_i \cdot \frac{\eta}{2}; \eta \right) \quad (6.13)$$

where (x_i, t_i, d_i) ($1 \leq i \leq k$) are fake records. As a result, which is illustrated in Figure 6.2b, the directional feature is scrambled in such a way that when enough faked random records are added to the posterior we cannot detect the feature created by the target data record anymore.

6.5 Experimental Evaluation

In this section, we have designed different sets of experiments for answering the following research questions:

- What is the empirical trade-off between privacy and the predictive performance of the proposed posterior sampling approach as compared to the baseline approach?
- What is the effect of the noisy prior distribution to the performance of the proposed posterior perturbing approach when we increase the number of fake records?
- Does sanitizing the likelihood function cause a noticeable impact on the performance of the survival models?

Subsections 6.5.2 and 6.5.3 will answer the above questions accordingly.

6.5.1 Datasets

In our experiments, we use two survival datasets, namely Wichert dataset (WT) and FLchain dataset (FL), which are also used in the experiments of Chapter 5. The survival times are normalized to the interval $[0, 1]$. All the values of the explanatory variables are normalized to have zero mean and unit variance.

6.5.2 Proposed Posterior Sampling Approach

In this subsection, we evaluate the performance of the proposed posterior sampling approach (PSA) under 5 different privacy budgets 0.05, 0.1, 0.2, 0.4 and 0.8. We use the proposed posterior sampling approach to randomly sample a private parameter vector and use this vector for performance evaluation.

Performance metric: To evaluate the predictive performance of the flexible parametric survival models, we perform the prediction task on whether a death event

would happen in time interval $[0, t_{med}]$, where t_{med} is the median of survival times in the dataset. We report the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) scores to show the effectiveness of the survival models in predicting death events. Moreover, we use a 5-fold cross-validation to get the average performance.

Hyperparameters: We use the natural cubic spline with 5 knots (3 internal knots) to model the baseline functions. Knots are set at locations which represent 0, 25, 50, 75, 100 percentiles of the survival times. We choose a sanitizer function with $\eta = \log(n)$ which is conceptually equivalent to the likelihood of one particular data record randomly picked from the dataset. All the priors are normal distributions (or half-normal distributions for non-negative parameters) with variance $\sigma^2 = 3$. For the RHMC sampling algorithm (Algorithm 6.1), we set the discretization parameter $\delta = 0.001$ and execute $L = 100$ discrete steps of the Hamiltonian dynamics for each random walk. For each coordinate j , we set the mass $m_j = 1$ and the speed of light $c_j = 1$ ($1 \leq j \leq d$).

Baselines: We use the discrete-time sampling approach pSGLD proposed in Chapter 5 as the baseline for comparison. We implement two discrete-time models with two link functions: (i) the *cloglog* link function which corresponds to our proportional hazard model and (ii) the *logit* link function which corresponds to our proportional odds model. We compare these models to our proposed models respectively. For pSGLD approach, we discretize the time values into 10 discrete values which are equally distributed..

Experimental results: Figure 6.4 shows the results of our proposed model (PSA) in comparison with the baseline (pSGLD) and the results from the non-private maximum likelihood estimator (MLE) (which is the best we can achieve without privacy protection). In both datasets, our PSA model achieves reasonably good performance even at small privacy budgets $\epsilon = 0.05$ and $\epsilon = 0.1$. With large

Figure 6.4: Performance results (in AUC-ROC) of the posterior sampling approach (PSA) with different privacy budgets ϵ .

privacy budgets $\epsilon = 0.4$ and $\epsilon = 0.8$, the performance is very close to the non-private MLE model. The PSA model outperforms the baseline pSGLD model in all configurations, especially at small privacy budgets.

6.5.3 Proposed Posterior Perturbing Approach

In this set of experiments, we consider the performance of our proposed posterior perturbing approach (PPA) for cloglog link function (referred to as PPA (Hazard)) and log link function (referred to as PPA (Odds)) when adding fake records into the dataset. We use the posterior mean (by averaging random samples drawn from the posterior distribution) as the point estimation for performance evaluation.

Baseline: We use the best results from non-private maximum likelihood estimation (MLE) for both cloglog link and logit link functions as the baseline for the performance comparison.

Figure 6.5: Performance results of the posterior perturbing approach (PPA) with different numbers of fake records.

Performance metric: We use AUC-ROC scores to measure the predicting performance of the approaches.

Hyperparameters: We fix the number of random neural networks to 50 as we observed that the predicting performance is not affected much by varying this number. The weights of the neural networks are randomly sampled from a standard normal distribution. Also, all the fake data records are generated from a standard normal distribution except the survival time which is generated from a uniform distribution. The number of fake records are 50, 100, 200, 400 and 800.

Experimental results: Figure 6.5 shows the performance results in AUC-ROC scores. When less than 400 fake records are used, the performance is very close to non-private MLE’s performance. The performance drops significantly at 800 fake records. For dataset FL, there is not much difference between the predicting performance using cloglog link function (i.e., PPA (Hazard)) and the predicting performance using logit link function (i.e., PPA (Odds)). For dataset WT, the performance of the PPA (Odds) model with logit link function is better than that of the PPA (Hazard) model with cloglog link function. Figure 6.5 also helps to answer the impact of the sanitizer function. We observe that with a small number of fake records (e.g., 50-100 fake records), there is virtually no drop in the performance of the sanitized models as compared to the best results from the non-private MLE. This observation implies that, practically, there is no noticeable

Figure 6.6: Estimated credible intervals of the model's parameters.

impact on the performance for sanitizing the likelihood function.

Besides point estimation, we can also estimate the uncertainty, i.e. the credible intervals, of the model's parameters from the posterior. Figure 6.6 shows the estimated credible intervals. The boxes determine the 50% credible intervals of the parameters, whereas the whiskers determine the 95% credible intervals of the parameters. These estimated credible intervals are very useful for scenarios in which we want to estimate the uncertainty of the model's predictions.

6.6 Summary

In this chapter, we have proposed two approaches to the privacy protection problem of continuous-time survival regression models. The posterior sampling approach promises differential privacy by publishing a random sample from the posterior distribution of a modified model with the likelihood function transformed by a specific sanitizer function. The sanitizer function is designed to

---

keep the shape of the likelihood function unchanged when the likelihood values are high. The posterior perturbing approach aims at tackling the situation when the accuracy and confidence of the predictions are very important. First, we sanitize the likelihood function. Then, we add random functions from two random sources: (i) random neural networks whose network weights are randomly assigned, and (ii) random likelihood functions from fake input data records. These noisy functions make it difficult for an adversary to learn the true data record of a target subject even in the worst case scenario when all other data records in the dataset are already revealed to the adversary. The experimental results have shown the effectiveness of the two proposed approaches.

Chapter 7

Local Differential Privacy for User Data Collection

In this chapter, we address the privacy preservation problems in local differential privacy (LDP). We first introduce the definition of local differential privacy and then propose private mechanisms for publishing numeric values and categorical values in the generic settings. We apply these mechanisms to the problem of training empirical risk minimization models in machine learning. Our experimental results show that our solutions outperform other methods on real datasets.

7.1 Privacy in User Data Collection

In the LDP setting, a *curator* collects data from a set of *users*, and computes statistical models of the collected data. The goal is to maximize the accuracy of these statistical models while preserving the privacy of the users. Formally, let n be the total number of users, and u_i ($1 \leq i \leq n$) denote the i -th user. Each user u_i 's private data is represented by a tuple t_i , which contains d attributes $a_1, a_2, \dots, a_d$. These attributes can be either numerical or categorical. Without loss of generality, we assume that each numeric attribute has a domain $[-1, 1]$, and each categorical attribute with k distinct values has a discrete domain $\{1, 2, \dots, k\}$.

To protect privacy, each user u_i first perturbs her tuple t_i using a randomized *perturbation function* f . Then, she sends the perturbed data $f(t_i)$ to the curator instead of her true data record t_i . The perturbation function determines the privacy/utility trade-off. As an extreme case, if $f(t_i) = t_i$ (i.e., no perturbation), the curator obtains perfect utility since it computes the statistical models based on the exact data. However, the privacy of the users is completely lost as the curator receives their sensitive data. The other extreme is that f simply outputs a random tuple regardless of t_i , which leads to the highest level of privacy and zero utility, i.e., the curator learns nothing about the users' data.

Given a privacy budget ϵ that controls the privacy-utility trade-off, we require that f satisfies ϵ -local differential privacy (ϵ -LDP) [45], defined below.

Definition 7.1 (ϵ -Local Differential Privacy). *A randomized function f satisfies ϵ -local differential privacy if and only if for any two input tuples $t, t' \in \text{Dom}(f)$ and for any possible output t^* of f , we have:*

$$\Pr[f(t) = t^*] \leq e^\epsilon \cdot \Pr[f(t') = t^*]$$

Basically, local differential privacy is a special case of differential privacy where the random perturbation is performed by the users, not by the curator. In other words, the curator never possesses the exact private data of any user. According to the above definition, the curator, who receives the perturbed tuple t^* , cannot distinguish whether the true tuple is t or another tuple t' with high confidence (controlled by parameter ϵ), regardless of the background information of the curator. This approach provides plausible deniability to the user. Note that in ϵ -LDP, since random perturbation is done for each user, it is possible to achieve *personalized privacy protection* by using different values of the privacy budget ϵ at different users, depending on their respective privacy requirements. In this research, we assume a universal ϵ for the ease of presentation and analysis.

7.2 Problem Definitions

In the LDP setting, the curator wants to collect useful information from the users but cannot access to users' information directly. Therefore, the curator requests each user to publish a randomized version of the user's data, and the curator will infer useful information from these randomized data. There are two essential pieces of information which the curator wants to know: (i) the mean values of numeric attributes and (ii) the frequencies (or histogram) of categorical attributes. The formal problem statements are given below.

Problem 7.1: Given a set of n users, each keeps a private tuple of d numeric attributes $t = (a_1, a_2, \dots, a_d)$ where $a_i \in [-1, 1]$. Let $\mathcal{D}$ be the collection of all these tuples, $\mathcal{D} = (t_1, t_2, \dots, t_n)$, and ϵ be the privacy budget for each user. Estimate the value of $\bar{v} = \frac{1}{n} \sum_{i=1}^n v_i$ and guarantee that each user's output satisfies ϵ -LDP.

Problem 7.2: Given a set of n users, each keeps a private tuple of d numeric and categorical attributes. Estimate the frequency of categorical attributes and mean of numeric attributes such that each user's output satisfies ϵ -LDP.

These two problems are fundamental problems in local differential privacy. From this, we can tackle more complicated problems. In this research, we consider the application of local differential privacy in training machine learning models privately. In particular, we consider the training problem under the empirical risk minimization framework. The formal problem statement is given below.

Problem 7.3: Given a set of n users, each keeps a tuple $x = (a_1, \dots, a_d)$ of d attributes and a predicting value y , where $a_i \in [-1, 1]$ ($i = 1, \dots, n$) and $y \in [-1, 1]$. Train machine learning models under the empirical risk minimization framework while protecting each user by LDP with ϵ privacy budget.

7.3 Estimating Means and Frequencies

This section investigates the design of the perturbation function f to support accurate estimation of mean values of numeric attribute and frequencies of categorical attributes (i.e., Problem 7.1 and Problem 7.2). For ease of exposition, Section 7.3.1 considers the case when all attributes $a_1, a_2, \dots, a_d$ in the users' data have a numeric domain $[-1, 1]$ (Problem 7.1). After that, Section 7.3.2 extends our work to the case when both numeric and categorical attributes are present (Problem 7.2).

7.3.1 Estimating Mean Values for Numeric Attributes

Given a tuple t_i containing d numeric attributes, a naive design of the perturbation function f is to apply the Laplace mechanism. In particular, let $t_i^* = f(t_i)$ be the perturbed tuple of user u_i , we have:

$$\forall 1 \leq j \leq d, t_i^*[a_j] = t_i[a_j] + \text{Lap}\left(\frac{2d}{\epsilon}\right)$$

Once the curator receives all perturbed tuples, it simply computes their average $\frac{1}{n} \sum_{i=1}^n t_i^*[a_j]$ as an estimate of the mean of a_j . Clearly, this estimate is unbiased, since the injected Laplace noise $\text{Lap}\left(\frac{2d}{\epsilon}\right)$ in each $t_i^*[a_j]$ has zero mean. Meanwhile, it is easy to calculate that the expected error incurred by this estimator is $O\left(\frac{d}{\epsilon\sqrt{n}}\right)$, which is linear to the number of attributes d and, thus, could be excessively large when there are many attributes. Note that this is a fundamental problem that also exists in the traditional differential privacy setting, when publishing statistics for multiple independent attributes. Perhaps rather surprisingly, this problem has not received much attention in the differential privacy literature. The first and only solution we are aware of is proposed by Duchi *et al.* [32, 31] under the local differential privacy setting given below.

Algorithm 7.1 Duchi *et al.*'s Method [32, 31]**Input:** Tuple $t_i \in [-1, 1]^d$ and privacy budget ϵ **Output:** Tuple $t_i^* \in \{-B, B\}^d$ 1: Generate a random tuple $v \in \{-1, 1\}^d$ by sampling each $v[a_j]$ independently

from the following distribution: $\Pr[v[a_j] = x] = \begin{cases} \frac{1}{2} + \frac{1}{2}t_i[a_j], & \text{if } x = 1 \\ \frac{1}{2} - \frac{1}{2}t_i[a_j], & \text{if } x = -1 \end{cases}$

2: Let T^+ (resp. T^-) be the set of all tuples $t^* \in \{-B, B\}^d$ such that $t^* \cdot v > 0$ (resp. $t^* \cdot v \leq 0$)3: Sample a Bernoulli variable u that equals 1 with $e^\epsilon / (e^\epsilon + 1)$ probability4: **if** $u = 1$ **then**5:   Output a tuple uniformly at random from T^+ 6: **else**7:   Output a tuple uniformly at random from T^- 8: **end if**

Duchi *et al.*'s method: Algorithm 7.1 shows the pseudo-code of Duchi *et al.*'s method. The authors claim that this method satisfies ϵ -local differential privacy, yields unbiased estimates for the mean value of each attribute, and incurs $O(\sqrt{d \log d} / (\epsilon \sqrt{n}))$ expected error for each attribute, which is proven to be asymptotically optimal. As we explain later, all three claims are incorrect, i.e., their method can violate differential privacy, lead to a biased estimate and incur a much higher amount of errors. It takes as input the exact tuple $t_i \in [-1, 1]^d$ of user u_i and a privacy budget ϵ , and outputs a perturbed vector $t_i^* \in \{-B, B\}^d$, where B is a constant decided by d and ϵ . Note that the output is binary for each attribute, i.e., it is either B or $-B$. Therefore, it suffices for each user to transmit only one bit for each attribute to the curator.

Upon receiving the perturbed tuples, the curator simply computes the average value for each attribute over all users, and outputs these averages as the estimates of the mean values for their corresponding attributes. Next, we focus on the calculation of B , which is rather complicated. Essentially, B is a scaling factor to ensure that the expected value of a perturbed attribute is the same as

that of the exact attribute value. First, we calculate:

$$C_d = \begin{cases} 2^{d-1}, & \text{if } d \text{ is odd} \\ 2^{d-1} - \frac{1}{2} \binom{d}{d/2}, & \text{otherwise} \end{cases} \quad (7.1)$$

Then, B is calculated by:

$$B = \begin{cases} \frac{2^d + C_d \cdot (e^\epsilon - 1)}{\binom{d-1}{(d-1)/2} \cdot (e^\epsilon - 1)}, & \text{if } d \text{ is odd} \\ \frac{2^d + C_d \cdot (e^\epsilon - 1)}{\binom{d-1}{d/2} \cdot (e^\epsilon - 1)}, & \text{otherwise} \end{cases} \quad (7.2)$$

Duchi *et al.* showed that $\frac{1}{n} \sum_{i=1}^n t_i^*[a_j]$ is an unbiased estimator of the mean of a_j , and

$$\mathbb{E} \left[\max_{j \in [1, d]} \left| \frac{1}{n} \sum_{i=1}^n t_i^*[a_j] - \frac{1}{n} \sum_{i=1}^n t_i[a_j] \right| \right] = O \left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{n}} \right) \quad (7.3)$$

which is asymptotically optimal [32].

Problems in Duchi *et al.*'s method and a possible fix: We implemented and evaluated Duchi *et al.*'s method, but found that whenever the number d of attributes is even, the method yields a biased estimation of the mean of each attribute and incurs significant error. Then, we also found that it violates differential privacy when d is even. To illustrate, consider that $d = 2$ and we have an input tuple $t_i = (1, 1)$, i.e., $t_i[a_1] = t_i[a_2] = 1$. Then, Line 1 in Algorithm 7.1 would generate a tuple $v = (1, 1)$. Let B be as defined in Equation (7.2), and T^+ and T^- be as defined in Line 2 in Algorithm 7.1. It can be verified that T^+ and T^- contain 1 and 3 tuples respectively, with

$$T^+ = \{ (B, B) \}, \text{ and}$$

$$T^- = \{ (-B, -B), (-B, B), (B, -B) \}$$

Then, by Lines 3-8 in Algorithm 7.1, the method outputs (B, B) with $\frac{e^\epsilon}{e^\epsilon + 1}$ probability. In contrast, each tuple in T^- has only $\frac{1}{3e^\epsilon + 3}$ probability to be output.

Now consider another input tuple $t'_i = (-1, -1)$. It follows that, for t'_i , the algorithm outputs (B, B) with only $\frac{1}{3e^\epsilon + 3}$ probability. As a consequence,

$$\begin{aligned} \Pr[f(t_i) = (B, B)] &= 3e^\epsilon \cdot \Pr[f(t'_i) = (B, B)] \\ &> e^\epsilon \cdot \Pr[f(t'_i) = (B, B)] \end{aligned}$$

which indicates that the algorithm does not satisfy ϵ -differential privacy.

We find that the above problem is caused by Line 3 in Algorithm 7.1, in that the Bernoulli variable u is incorrectly defined for the case when d is even. To address the problem, one possible fix we found is to re-define u as a Bernoulli variable such that:

$$\Pr[u = 1] = \frac{e^\epsilon \cdot C_d}{(e^\epsilon - 1)C_d + 2^d}$$

It can be shown that, with this revised choice of u , Algorithm 7.1 achieves ϵ -differential privacy and ensures the error bound in Equation (7.3).

Proposed method: In what follows, we present an algorithm for perturbing a tuple that is conceptually simpler than Duchi *et al.*'s method, but achieves the same privacy assurance and asymptotic error bound. Furthermore, our experiments (in Section 7.5) show that the algorithm slightly outperforms Duchi *et al.*'s method in terms of the empirical accuracy of the estimated means of numeric attributes. Additionally, our method is more efficient. In particular, each user only needs to transmit one bit to the curator, which is clearly optimal. Our algorithm is inspired by an existing approach [14] for publishing categorical data, which we will discuss in Section 7.3.2.

Algorithm 7.2 shows the pseudo-code of our method. Given a tuple $t_i \in [-1, 1]^d$, the algorithm returns a perturbed tuple t_i^* that has non-zero value on only one attribute a_j ($j \in \{1, \dots, d\}$). Specifically, a_j is selected uniformly at

Algorithm 7.2 $\mathcal{A}_{\text{Numeric-LDP}}$: Proposed Method for Handling Numeric Attributes**Input:** Tuple $t_i \in [-1, 1]^d$ and privacy budget ϵ **Output:** Tuple $t_i^* \in \left\{ -\frac{e^\epsilon+1}{e^\epsilon-1}d, 0, \frac{e^\epsilon+1}{e^\epsilon-1}d \right\}^d$

- 1: Let $t_i^* = (0, 0, \dots, 0)$
- 2: Sample j uniformly at random from $\{1, 2, \dots, d\}$
- 3: Sample a Bernoulli variable u such that $\Pr[u = 1] = \frac{t_i[a_j] \cdot (e^\epsilon - 1) + e^\epsilon + 1}{2e^\epsilon + 2}$
- 4: **if** $u = 1$ **then**
- 5: $t^*[a_j] = \frac{e^\epsilon+1}{e^\epsilon-1} \cdot d$
- 6: **else**
- 7: $t^*[a_j] = -\frac{e^\epsilon+1}{e^\epsilon-1} \cdot d$
- 8: **end if**
- 9: Output t_i^*

random from all d attributes of t_i , and $t_i^*[a_j]$ is sampled from the following distribution:

$$\Pr[t_i^*[a_j] = x] = \begin{cases} \frac{t_i[a_j] \cdot (e^\epsilon - 1) + e^\epsilon + 1}{2e^\epsilon + 2}, & \text{if } x = \frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d \\ \frac{-t_i[a_j] \cdot (e^\epsilon - 1) + e^\epsilon + 1}{2e^\epsilon + 2}, & \text{if } x = -\frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d \end{cases} \quad (7.4)$$

Observe that in the above method, the output t_i^* contains only one non-zero value, for the randomly chosen attribute a_j . This value is binary; hence, the user u_i only needs to transmit 1 bit to the curator indicating its sign, and the latter can re-scale it using parameters ϵ and d . Further, as we show below the correctness of this method does not depend on the choice of a_j as long as it is chosen uniformly at random. Therefore, the value of j can be obtained, e.g., using a public source of random numbers such as a hash value of the user's ID. Therefore, the communication overhead between each user and the curator is exactly 1 bit.

The following lemmas establish the theoretical guarantees of Algorithm 7.2.

Lemma 7.1. *Algorithm 7.2 satisfies ϵ -local differential privacy.*

Proof. Let t^* be an output of Algorithm 7.2, and a_j be the only attribute such that $t^*[a_j] \neq 0$. Let t and t' be any two tuples, and u (resp. u') be the Bernoulli

variable generated in Line 3 of Algorithm 7.2 given t (resp. t') as the input. In the following, we focus on the case when $t^*[a_j] = \frac{e^\epsilon + 1}{e^\epsilon - 1}d$; the case when $t^*[a_j] = -\frac{e^\epsilon + 1}{e^\epsilon - 1}d$ can be analyzed in a similar manner.

By Algorithm 7.2, we have:

$$\begin{aligned} \frac{\Pr[t^* \mid t]}{\Pr[t^* \mid t']} &= \frac{1/d \cdot \Pr[u = 1 \mid t]}{1/d \cdot \Pr[u' = 1 \mid t']} \leq \frac{\max_t \Pr[u = 1 \mid t]}{\min_{t'} \Pr[u' = 1 \mid t']} \\ &= \frac{\max_{t[a_j] \in [-1, 1]} (t[a_j] \cdot (e^\epsilon - 1) + e^\epsilon + 1)}{\min_{t'[a_j] \in [-1, 1]} (t'[a_j] \cdot (e^\epsilon - 1) + e^\epsilon + 1)} = e^\epsilon \end{aligned}$$

This completes the proof. $\square$

Lemma 7.2. *Let t_i^* be the output of Algorithm 7.2 given an input tuple t_i . Then, for any $j \in \{1, \dots, d\}$, $\mathbb{E}[t^*[a_j]] = t[a_j]$.*

Proof. By Equation (7.4),

$$\begin{aligned} \mathbb{E}[t^*[a_j]] &= \Pr \left[t^*[a_j] = \frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d \right] \cdot \frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d \\ &\quad + \Pr \left[t^*[a_j] = -\frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d \right] \cdot \left(-\frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d \right) \\ &\quad + \Pr [t^*[a_j] = 0] \cdot 0 \\ &= \frac{2t[a_j] \cdot (e^\epsilon - 1)}{2e^\epsilon - 2} = t[a_j] \end{aligned}$$

$\square$

By Lemma 7.2, the server can use $\frac{1}{n} \sum_{i=1}^n t_i^*[a_j]$ as an unbiased estimator of the mean of a_j . The following lemma shows the accuracy guarantee of this estimator.

Lemma 7.3. *For any $j \in \{1, \dots, d\}$, let $Z[a_j] = \frac{1}{n} \sum_{i=1}^n t_i^*[a_j]$ and $X[a_j] = \frac{1}{n} \sum_{i=1}^n t_i[a_j]$.*

With at least $1 - \beta$ probability,

$$\max_{j \in \{1, \dots, d\}} |Z[a_j] - X[a_j]| = O \left(\frac{\sqrt{d \log(d/\beta)}}{\epsilon \sqrt{n}} \right)$$

Proof. First, observe that for any $i \in \{1, \dots, n\}$ and any $j \in \{1, \dots, d\}$, the variance of $t_i^*[a_j] - t_i[a_j]$ equals:

$$\begin{aligned} \text{Var}[t_i^*[a_j] - t_i[a_j]] &= \text{Var}[t_i^*[a_j]] = \mathbb{E}[(t_i^*[a_j])^2] - (\mathbb{E}[t_i^*[a_j]])^2 \\ &= \frac{1}{d} \left(\frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d \right)^2 - (t_i[a_j])^2 \leq \left(\frac{e^\epsilon + 1}{e^\epsilon - 1} \right)^2 \cdot d \end{aligned}$$

By Bernstein's inequality,

$$\begin{aligned} &\Pr[|Z[a_j] - X[a_j]| \geq \lambda] \\ &\leq 2 \cdot \exp \left(- \frac{n\lambda^2}{\frac{2}{n} \sum_{i=1}^n \text{Var}[t_i^*[a_j] - t_i[a_j]] + \frac{2}{3}\lambda \cdot \frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot 2d} \right) \\ &= 2 \cdot \exp \left(- \frac{n\lambda^2}{2d \cdot (O(1/\epsilon^2) + \lambda \cdot O(1/\epsilon))} \right) \end{aligned}$$

By the union bound, there exists $\lambda = O \left(\frac{\sqrt{d \log(d/\beta)}}{\epsilon \sqrt{n}} \right)$ such that $\max_{j \in [1, d]} |Z[a_j] - X[a_j]| < \lambda$ holds with at least $1 - \beta$ probability. $\square$

7.3.2 Estimating Frequencies for Categorical Attributes

We now focus on the case where each user's data record contains not only numeric attributes but also categorical ones. For each categorical attribute, the curator aims to build an accurate histogram containing the frequency estimate for each possible value in the attribute's domain.

Randomized response for binary attributes: For a single binary attribute (e.g., WiFi on/off), it suffices to use the classic randomized response method [100] (also used in Rappor [45]) to estimate the distribution of users. Specifically, suppose that the domain of the binary attribute (say a_j) contains two possible values, -1 and $+1$. Each user u_i reports her true answer $t_i[a_j]$ with probability p , and a random answer with probability $1 - p$. The latter has the same probability to be -1 and $+1$; hence, its expected value is zero. Therefore, the expected value

for u_i 's reported value is $p \cdot t_i[a_j]$. Thus, we can obtain an unbiased estimate by multiplying the reported value by a scaling factor $c_\epsilon = 1/p$.

Meanwhile, comparing u_i 's true attribute value and her reported one, the two are the same with probability $p + (1 - p)/2$, and they are different with probability $(1 - p)/2$. By definition, ϵ -local differential privacy requires that $\frac{p+(1-p)/2}{(1-p)/2} \leq e^\epsilon$. The equality holds when $p = \frac{e^\epsilon - 1}{e^\epsilon + 1}$. We thus arrive at the following unbiased mechanism that satisfies ϵ -LDP: each user u_i reports $c_\epsilon \cdot t_i[a_j] = \frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot t_i[a_j]$ with probability $p + (1 - p)/2 = \frac{e^\epsilon}{e^\epsilon + 1}$, and $-c_\epsilon \cdot t_i[a_j]$ otherwise (i.e., with probability $\frac{1}{e^\epsilon + 1}$).

Once the curator receives all reported values for attribute a_j , it computes the average over all users, which is an estimate of the mean value $\mathbb{E}[a_j]$ for a_j . Since a_j can be either $+1$ or -1 , the percentage of users with $+1$ (resp. -1) is $\frac{1+\mathbb{E}[a_j]}{2}$ (resp. $\frac{1-\mathbb{E}[a_j]}{2}$).

Bassily and Smith's method: The problem is more complicated when the categorical attribute a_j contains $k > 2$ possible values. In this situation, the curator aims to build a histogram that contains the estimated frequency for each of the k possible values. The current state-of-the-art method to our knowledge is by Bassily and Smith [14], shown in Algorithm 7.3, which is proven to satisfy ϵ -LDP and achieve an optimal asymptotical error bound. There are two main ideas in this method. First, the authors assume that the number of possible values k in the categorical attribute is far larger than the number of users n ; hence, the method applies random projection to reduce the dimensionality from k to m . The value of m is chosen carefully so as to obtain the asymptotically optimal error bound. This step essentially transforms the categorical attribute into m binary ones.

Specifically, the random projection is done with a $m \times k$ matrix Φ in which each element is randomly set to either $+\frac{1}{\sqrt{m}}$ or $-\frac{1}{\sqrt{m}}$ with equal probability. This ensures that (i) the inner product of any column in Φ with itself is 1 (which is where the absolute value of each element $\frac{1}{\sqrt{m}}$ comes from) and (ii) the inner product of two different columns in Φ has zero expected value, since the signs

Algorithm 7.3 Bassily and Smith's Method [14]

Input: $t_i[a_j] \in \{1, \dots, k\}$ each user u_i , privacy budget ϵ , confidence of the error bound β

Output: Frequency estimate for each of the k values in attribute a_j

- 1: Compute $\gamma = \sqrt{\frac{\log(2k/\beta)}{\epsilon^2 n}}$ and $m = \frac{\log(k+1) \log(2/\beta)}{\gamma^2}$
- 2: Generate random matrix $\Phi \in \left\{ \pm \frac{1}{\sqrt{m}} \right\}^{m \times k}$
- 3: **for** $i = 1$ to n **do**
- 4:   User u_i : draw $s \sim \text{Uniform}(\{1, 2, \dots, m\})$
- 5:   User u_i : draw $t \sim \text{Bernoulli}(\frac{e^\epsilon}{e^\epsilon + 1})$
- 6:   User u_i : **if** $t = 1$ **then** $\alpha = c_\epsilon m \Phi[s, t_i[a_j]]$ **else** $\alpha = -c_\epsilon m \Phi[s, t_i[a_j]]$, where $c_\epsilon = \frac{e^\epsilon + 1}{e^\epsilon - 1}$
- 7:   User u_i : submit (s, α) , which represents a k -dimensional vector z_i where the s -th entry is α and the other entries are 0
- 8: **end for**
- 9: Compute $\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$
- 10: **for** $l = 1$ to k **do**
- 11:   Estimate the frequency of the l -th value by the inner product of the l -th column of Φ and $\bar{z}$
- 12: **end for**
- 13: Output k frequency estimates obtained above

are randomly generated. Each user u_i 's attribute value $t_i[a_j]$ is then transformed to m binary values by taking the $t_i[a_j]$ -th column in Φ .

The second idea is for each user to randomly pick one of the converted m binary attributes, and report a randomized response using the method described earlier. Note that the randomized response needs to be scaled by a factor of m , since each of the m binary attributes has probability $1/m$ to be chosen. The curator collects the average for each of the m binary attributes, which is stored as a vector $\bar{z}$. To obtain the frequency estimate of a particular attribute value l , the method takes the inner product of $\bar{z}$ and the l -th column of Φ , which can be proven to yield an unbiased frequency estimate for the l -th value in a_j . Meanwhile, the frequency estimate is proven to be within $O\left(\frac{\sqrt{\log(k/\beta)}}{\epsilon\sqrt{n}}\right)$ error with probability $1 - \beta$, where β is an input to the algorithm [14].

Proposed method for a single categorical attribute: Our proposed method generally follows Bassily and Smith’s method to estimate value frequencies for a categorical attribute. However, we found that although Bassily and Smith’s method achieves optimal asymptotic accuracy, in practice its accuracy tends to be unstable, especially for relatively small categorical domains. The reason is that the random projection matrix Φ introduces considerable noise. In particular, the inner product of two different columns is often non-zero unless k is very large, which is magnified by a large number of users n . Hence, we propose an alternative solution that obtains higher accuracy when $k = o(n)$. In particular, instead of generating random matrix Φ (size $m \times k$ where $m = O(n)$), we construct a binary matrix of size $k \times k$ satisfying that any two column vectors are always orthogonal.

Proposed method for multiple numeric and categorical attributes: Bassily and Smith’s method is limited to a single categorical attribute. To extend it to multiple categorical attributes, a straightforward approach is to apply the method once for each attribute separately. In that case, however, the privacy budget ϵ needs to be divided among all attributes, so as to ensure ϵ -LDP as a whole. Without loss of generality, assume that we have d categorical attributes, and we assign ϵ/d budget to each of them. Then, the amount of noise incurred by Bassily and Smith’s method on each attribute is increased d times to $O\left(\frac{d\sqrt{\log(d/\beta)}}{\epsilon\sqrt{n}}\right)$, which is unsatisfactory when d is large.

Another approach for extension is to (i) convert the d categorical attributes into a “composite” attribute whose domain equals to the Cartesian product of the individual attribute domains, and then (ii) apply Bassily and Smith’s method on the composite attribute. This, however, only allows us to (accurately) derive the frequency each composite value (i.e., a combination of values from all d individual attributes), but does not provide a quality estimation of the frequency of each individual value. Note that these limitations are not specific to Bassily and Smith’s method. To the best of our knowledge, there is no existing work

(including Rappor) that are designed to handle multiple categorical attributes, let alone a mixture of numeric and categorical ones.

In our method, we use a simple and elegant solution to handle multiple attributes: for each numerical attribute, the curator estimates its mean value; for each categorical attribute, the curator estimates its value frequencies. In particular, given d attributes $a_1, a_2, \dots, a_d$, the solution asks each user to perform the following:

1. Draw j uniformly at random from the set $\{1, 2, \dots, d\}$.
2. If a_j is a numeric attribute, then submit a noisy version of $t[a_j]$ computed as in Lines 3-7 of Algorithm 7.2.
3. Otherwise (i.e., a_j is a categorical attribute), compute (s, α) as in Lines 4-8 of Algorithm 7.3, then submit $(s, d \cdot \alpha)$ to represent a d -dimensional vector where the s -th entry is $d \cdot \alpha$ and all other entries are zero.

The above solution satisfies ϵ -LDP, which follows from the fact that (i) each user randomly selects one attribute to submit, and (ii) the algorithm used for submitting the selected attribute is ϵ -differentially private. For each numeric attribute, it is easy to see that the solution provides the same accuracy guarantee as Algorithm 7.2, since both methods handle numeric attributes in exactly the same way. The following lemma states the accuracy guarantee of our solution for categorical attributes.

Lemma 7.4. *For each categorical attribute a_j with a domain $\{1, 2, \dots, k\}$, let x_l be the frequency of the l -th value of a_j , and y_l be the estimation of x_l returned by our solution. With at least $1 - \beta$ probability,*

$$\max_{l \in [1, k]} |y_l - x_l| = O \left(\frac{\sqrt{d \log(k/\beta)}}{\epsilon \sqrt{n}} \right)$$

Proof. Consider any user with a tuple t . With respect to a_j , our solution can be regarded as a method that (i) outputs nothing with $d - 1/d$ probability, and

(ii) with the remaining $1/d$ probability, applies Bassily and Smith's algorithm on a_j and scale its output up by d times. It follows that the variance of each of our frequency estimators for a_j is d times that of Bassily and Smith's estimator. Based on the analysis in [14], it can be shown that the variance of Bassily and Smith's estimator is $O\left(\frac{1}{n\epsilon^2}\right)$. Therefore, the variance of each of our frequency estimators is $O\left(\frac{d}{n\epsilon^2}\right)$. Combining this with Bernstein's inequality and the union bound, it can be proven that with at least $1 - \beta$ probability, the error in any k of our estimators is $O\left(\frac{\sqrt{d \log(k/\beta)}}{\epsilon \sqrt{n}}\right)$. $\square$

By Lemma 7.4, when there exist multiple categorical attributes, the error incurred by our approach is a factor of $O(\sqrt{d})$ smaller than that of a solution that repeatedly applies Bassily and Smith's method on each categorical attribute.

Generation of orthogonal binary vectors: Algorithm 7.4 allows us to generate a set of orthogonal binary vectors which can be in our mechanism as the replacement for random vectors. The algorithm starts by creating a set $\mathcal{S}$ of two initial orthogonal vectors $[1, -1]$ and $[1, 1]$ (Line 1). Let's consider each iteration by Lines 2-6 in the algorithm. We first initialize a new empty set $\mathcal{S}'$. Then, for each element v in the current $\mathcal{S}$, we construct two new elements in the set $\mathcal{S}'$ by (i) concatenating v with itself $v \parallel v$, and (ii) concatenating v with its negative $v \parallel (-v)$. Finally, we assign $\mathcal{S}'$ to $\mathcal{S}$. Therefore, for each iteration, the size of set $\mathcal{S}$ is double. The process is terminated when size of $\mathcal{S}$ exceeds the input d . Lemma 7.5 shows that the result of this algorithm is an orthogonal set of binary vectors.

Lemma 7.5. *The set $\mathcal{S}$ returned from Algorithm 7.4 is an orthogonal set.*

Proof. We prove this lemma by induction. For the base case, observe that the initial value of $\mathcal{S} = \{[1, -1], [1, 1]\}$ is an orthogonal set. Now assume that $\mathcal{S} = \{v_i\}$ is an orthogonal set. We will prove that $\mathcal{S}' = \{v_i \parallel v_i, v_i \parallel (-v_i)\}$ is also an orthogonal set. For any $v'', v' \in \mathcal{S}'$, consider the inner product $\langle v'', v' \rangle$, there are two general cases: (i) $\langle v'', v' \rangle = \langle v_i \parallel \pm v_i, v_j \parallel \pm v_j \rangle$, and (ii) $\langle v'', v' \rangle = \langle v_i \parallel$

Algorithm 7.4 Generating the Orthogonal Set**Input:** Dimensionality $d = 2^k$ **Output:** Set S of d vectors

```

1:  $S \leftarrow \{[1, -1], [1, 1]\}$ 
2: for  $|S| < d$  do
3:    $S' = \emptyset$ 
4:   for  $v \in S$  do
5:      $S' \leftarrow S' \cup \{v \parallel v, v \parallel (-v)\}$ 
6:   end for
7:    $S \leftarrow S'$ 
8: end for
9: Output  $S$ 

```

$v_i, v_i \parallel -v_i\rangle$. In both cases, the inner product equals zero. Thus, the lemma is proven. $\square$

7.4 Empirical Risk Minimization

This section investigates building a large class of machine learning models that can be expressed as empirical risk minimization under ϵ -local differential privacy. In particular, we focus on three common types of learning tasks: linear regression, logistic regression, and SVM classification. Section 7.4.1 introduces the basic approaches for training these models, while Section 7.4.2 discusses our proposed improved method.

7.4.1 Basic Methods

Suppose that each user u_i has a pair (x_i, y_i) , where $x_i \in [-1, 1]^d$ and $y_i \in [-1, 1]$ (for linear regression) or $y_i \in \{-1, 1\}$ (for logistic regression and SVM classification). Let $\ell(\cdot)$ be a *loss function* that (i) maps a d -dimensional *parameter vector* β into a real number, and (ii) is parameterized by x_i and y_i . We aim to identify a parameter vector β^* such that:

$$\beta^* = \arg \min_{\beta} \frac{1}{n} \left(\sum_{i=1}^n \ell(\beta; x_i, y_i) \right) + \frac{\lambda}{2} \|\beta\|_2^2$$

where $\lambda > 0$ is a regularization parameter. We consider three specific loss functions:

- Linear regression: $\ell(\beta; x_i, y_i) = (x_i^T \beta - y_i)^2$
- Logistic regression: $\ell(\beta; x_i, y_i) = \log(1 + e^{-y_i x_i^T \beta})$
- SVM (hinge loss): $\ell(\beta; x_i, y_i) = \max\{0, 1 - y_i x_i^T \beta\}$

For convenience, we define

$$\ell'(\beta; x_i, y_i) = \ell(\beta; x_i, y_i) + \frac{\lambda}{2} \|\beta\|_2^2$$

One of the most common solutions to compute β^* is *stochastic gradient descent* (SGD). It starts from an initial parameter vector β_0 , and iteratively updates it into $\beta_1, \beta_2, \dots$ based on the following equation:

$$\beta_{k+1} = \beta_k - \gamma_k \cdot \nabla \ell'(\beta_k; x, y)$$

where (x, y) is the tuple of a randomly selected user, $\nabla \ell'(\beta_k; x, y)$ is the gradient of ℓ' at β_k , and γ_k is a constant typically set to $O(1/\sqrt{k})$. It terminates when the difference between β_{k+1} and β_k is sufficiently small.

Under our problem setting, however, $\nabla \ell'$ is not directly available to the curator and needs to be collected in a private manner. Towards this end, existing work [53, 32] has suggested that the curator may ask the selected user in each iteration to submit a noisy version of $\nabla \ell'$, by using the Laplace mechanism or Duchi *et al.*'s method (i.e., Algorithm 7.1). We can straightforwardly improve these existing approaches by perturbing $\nabla \ell'$ using Algorithm 7.2 instead. However, we observe that such a solution is insufficient for our target application, as we will explain in Section 7.4.2.

7.4.2 Proposed Improvements

We observe in our experiments that the aforementioned SGD approach yields rather inaccurate results, due to the noise injected into the gradient $\nabla \ell'$ returned by each user. In particular, if each user applies Algorithm 7.2 to compute $\nabla \ell'(\beta_k; x, y)$, the amount of noise in $\nabla \ell'$ is $O\left(\frac{\sqrt{d \log d}}{\epsilon}\right)$, which is excessively large given that $x \in [-1, 1]^d$. To address this issue, we adopt *mini-batch* gradient descent instead of SGD. That is, for each iteration of the algorithm, we involve a group G of users, and ask each of them to submit a noisy version of the gradient. After that, we update the parameter vector β_k with the mean of the noisy gradients, i.e.,

$$\beta_{k+1} = \beta_k - \gamma_k \cdot \frac{1}{|G|} \sum_{i=1}^{|G|} \nabla \ell_i^*$$

where $\nabla \ell_i^*$ is the noisy gradient submitted by the i -th user in G . This helps because the amount of noise in the average gradient is $O\left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{|G|}}\right)$, which could be acceptable if $|G| = \Omega(d \log d / \epsilon^2)$.

However, when d is sizable, $|G| = \Omega(d \log d / \epsilon^2)$ is large. As a consequence, when we allow each user to participate in at most one iteration of the algorithm, the maximum number of iterations (i.e., $n/|G|$) is small. In that case, the algorithm may terminate prematurely and return an inferior parameter vector. One may attempt to mitigate this problem by allowing each user to be involved in $m > 1$ iterations, but it would further increase the amount of noise in each gradient returned. To explain this, suppose that the i -th ($i \in \{1, \dots, m\}$) gradient returned by the user satisfies ϵ_i -differential privacy. By the composition property of differential privacy [76], if we are to enforce ϵ -differential privacy for the user's data, we should have $\sum_{i=1}^m \epsilon_i \leq \epsilon$. Consider that we set $\epsilon_i = \epsilon/m$. Then, the amount of noise in each gradient becomes $O\left(\frac{m \sqrt{d \log d}}{\epsilon}\right)$. As such, the acceptable mini-batch size becomes $|G| = \Omega(m^2 d \log d / \epsilon^2)$, which is m^2 times the acceptable size when we allow each user to participate in at most one iteration. It then follows that the total number of iterations in the algorithm is inverse

proportional to $1/m$, i.e., setting $m > 1$ only degrades the performance of the algorithm.

For linear regression, instead of increasing m , we propose to apply dimensionality reduction on each user's data, so as to reach an acceptable size of mini-batches. Specifically, the curator first generates a random $r \times d$ matrix P where $r < d$ and each entry has an equal probability to be assigned $1/d$ or $-1/d$. Then, the curator shares P with all users, and asks each user to convert her tuple (x_i, y_i) into a reduced tuple (x'_i, y_i) , where $x'_i = Px$. In other words, we project $\{x_i\}$ into a random r -dimensional sub-space, and such a projection is known to preserve several important characteristics of the original data [3]. It can be verified that $x'_i \in [-1, 1]^r$.

Subsequently, each user uses the reduced tuple (x'_i, y_i) to participate in the mini-batch gradient descent algorithm. In other words, each noisy gradient $\nabla \ell_i^*$ returned by the user is r -dimensional instead of d -dimensional. As such, the average noisy gradient obtained from a mini-batch G of users has an error of $O\left(\frac{\sqrt{r \log r}}{\epsilon \sqrt{|G|}}\right)$ instead of $O\left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{|G|}}\right)$. Accordingly, the acceptable mini-batch size is reduced to $|G| = \Omega(r \log r / \epsilon^2)$.

Algorithm 7.5 shows the pseudo-code of our mini-batch gradient descent method. The curator first generates a random $r \times d$ matrix P , and maintains a r -dimensional parameter vector α (Lines 1-3). We use α instead of β to denote the parameter vector to avoid confusion on its dimensionality. After that, whenever a user with a tuple (x_i, y_i) comes online, she obtains P and the current α from the curator (Line 6). Then, the user computes a reduced tuple (x'_i, y_i) , as well as the gradient $\nabla_i = \nabla l'(\alpha; x'_i, y_i)$ (Line 7). If any entry of ∇_i is larger than 1 (resp. smaller than -1), then the user resets the entry to 1 (resp. -1) (Lines 8-9). This ensures that $\nabla_i \in [-1, 1]^d$, so that it can be a valid input to Algorithm 7.2. After that, the user computes a noisy gradient ∇_i^* using Algorithm 7.2, submits it to the curator, and then logs off (Line 10).

The curator computes the average noisy gradient from every g users (where g

Algorithm 7.5 $\mathcal{A}_{\text{ERM-LDP}}$: An Iterative Method for Empirical Risk Minimization**Input:** Positive integer r , mini-batch size g , privacy budget ϵ **Output:** Parameter vector $\alpha \in \mathbb{R}^r$

```

1: Generates a random  $r \times d$  matrix  $P$  each entry has an equal probability to
    $1/d$  or  $-1/d$ 
2: Initialize a counter  $k = 0$ , and learning rate  $\gamma$ 
3: Initialize a  $r$ -dimensional vector  $\nabla = [0, 0, \dots, 0]$ 
4: for true do
5:   if a user with a tuple  $(x_i, y_i)$  comes online then
6:     Send  $P$  to the user
7:     The user computes  $x'_i = Px_i$ , and derives  $\nabla_i = \nabla l'(\alpha; x'_i, y_i)$ 
8:     if  $\nabla_i \notin [-1, 1]^r$  then
9:       The user projects  $\nabla_i$  onto  $[-1, 1]^r$ 
10:    end if
11:    The user applies Algorithm 7.2 on  $\nabla_i$ , and submits a noisy gradient  $\nabla_i^*$ 
12:     $k = k + 1$ , and  $\nabla = \nabla + \nabla_i^*$ 
13:    if  $k \bmod g = 0$  then
14:       $\nabla = \frac{\nabla}{g}$ , and  $\gamma = 1/\sqrt{k/g}$ 
15:       $\alpha = \alpha - \gamma \cdot \nabla$ 
16:    end if
17:    if  $k$  is sufficiently large or  $\alpha$  changes sufficiently small in the last update
      then
18:      break
19:    end if
20:  end if
21: end for
22: Output  $\alpha$ 

```

---

is an input parameter), and updates the parameter vector α accordingly (Lines 11-14).

When the update to α is sufficiently small or when a sufficiently large number of users have participated, the curator terminates the algorithm (Lines 15-16).

7.5 Experimental Evaluation

In this section, we have designed experiments for answering the following research questions:

- What is the trade-off between privacy of users and accuracy in estimating means and frequencies?

Table 7.1: Statistics of the datasets.

Dataset	Size	#categorical	#numerical
US	9M	17	6
BR	4M	12	6

- What is the trade-off between privacy of users and accuracy in training linear regression, logistic regression and SVM models?

7.5.1 Datasets

We use two public datasets extracted from the *Integrated Public Use Microdata Series* [58], US and BR, which contain census records from the United States and Brazil, respectively.

- US - It contains 9M records and 23 attributes, among which 6 are numeric (e.g., age) and 17 are categorical (e.g., gender).
- BR - It contains 4M records and 18 attributes, among which 6 are numeric and 12 are categorical.

Table 7.1 gives a summary of the two datasets. Both datasets contain a numeric attribute “total income”, which is used as the dependent attribute in linear regression, logistic regression, and SVM. We normalize the domain of each numeric attribute into $[-1, 1]$.

7.5.2 Estimating Means and Frequencies

In the first set of experiments, we consider the task of collecting a noisy tuple from each user to estimate the mean of each numeric attribute and the frequency of each categorical value.

Baseline: As mentioned previously, none of the existing solutions can directly support this task, since they are designed for either numeric or categorical attributes, but not both. To address this issue, we construct a baseline method

(referred to as *Hybrid*) by combining the best existing solutions as follows. Let t be a tuple with d_n numeric attributes and d_c categorical attributes. Given t and a privacy budget ϵ , *Hybrid* first constructs a d_n -dimensional tuple t' that contains all numeric values in t , and then releases a noisy version of t' by invoking Duchi *et al.*'s method (see Subsection 7.3.1) on t' , using a privacy budget of $d_n\epsilon/d$. After that, for each categorical value in t , *Hybrid* applies Bassily and Smith's method (see Section 7.3.2) to release a noisy version with a privacy budget of ϵ/d . By the composition property of differential privacy [76], *Hybrid* ensures ϵ -LDP. Intuitively, *Hybrid* is a best-effort approach to incorporate two state-of-the-art methods that are designed only for numeric attributes (i.e., Duchi *et al.*'s method) and a single categorical attribute (i.e., Bassily and Smith's method), respectively.

Performance metric: We use ℓ_∞ -norm error as the performance metric for estimating means and frequencies. The ℓ_∞ -norm error is defined as follows:

$$\ell_\infty \text{ error} = \|x_i - x^*\|_\infty$$

where x_i is the estimated vector and x^* is the true vector.

Experimental results: We apply our solution in Section 7.3 and *Hybrid* on both US and BR to generate noisy tuples. We then use the noisy tuples to estimate the mean of each numeric attribute. Figure 7.1a (resp. 7.1b) shows the results on US (resp. BR) as a function of the privacy budget ϵ . Our solution slightly outperforms *Hybrid*, regardless of the dataset used and the value of ϵ . We also note that, compared with *Hybrid* (which applies Duchi *et al.*'s method to handle numeric attributes), our solution has a much lower communication cost on each user (since it only requires each user to transfer 1 bit), and is much simpler.

We also use the noisy tuples to estimate the frequency of each value in each categorical domain in US and BR. Figures 7.1c and 7.1d illustrate the results as ϵ varies. Our method is significantly more accurate than *Hybrid* in all cases, and its error is only around 1/4 of the error incurred by *Hybrid*. This is consistent with

Figure 7.1: Experimental results on estimating means and frequencies.

the analysis in Section 7.3.2 that (i) our method has $O\left(\frac{\sqrt{d \log k}}{\epsilon \sqrt{n}}\right)$ estimation error, and (ii) repeatedly applying Bassily and Smith’s method on each categorical leads to $O\left(\frac{d \sqrt{\log k}}{\epsilon \sqrt{n}}\right)$ error.

7.5.3 Empirical Risk Minimization

In the second set of experiments, we consider linear regression, logistic regression, and SVM classification on US and BR. For both datasets, we use the numeric attribute “total income” as the dependent variable, and all other attributes as independent variables. Following the standard practice, we transform each categorical attribute a_j with k values into $k - 1$ binary attributes with a domain $\{-1, 1\}$, such that (i) the l -th ($l < k$) value in a_j is represented by a 1 on the l -th binary attribute and a -1 on each of the remaining $k - 2$ attributes, and (ii) the

k -th value in a_j is represented by -1 on all binary attributes. After this transformation, the dimensionality of US (resp. BR) becomes 85 (resp. 95). For logistic regression and SVM (which require the dependent variable to be binary), we convert “total income” into a binary attribute by mapping all positive values (in the normalized domain $[-1, 1]$) to 1, and all non-positive values to 0.

Compared methods: We evaluate four methods as follows:

- Non-Private - It is a non-private version of SGD.
- Private SGD - It is a private version of SGD that involves one user in each iteration, and asks the user to submit a noisy gradient using Duchi *et al.*'s method.
- Private MGD - It is our proposed method with mini-batch gradient descent (MGD), which involves g users in each iteration, and uses the average noisy gradients of those users (generated with Algorithm 7.2) to update the parameter vector.
- Private MGD-DR - It is our proposed method with mini-batch gradient descent and dimensionality reduction (DR), which is an improved version of MGD that projects the users' data onto an r -dimensional subspace before the learning task.

Performance metrics: For linear regression, we use *mean squared error* (MSE) to measure the performance of models. The mean squared error is defined as:

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

where y_i is the true regression value, $\hat{y}_i$ is the predicted regression value and m is the size of the test set. For logistic regression and SVM, we use *misclassification*

Figure 7.2: Experimental results for linear regression.

rate to measure the performance of models. Misclassification rate is defined as:

$$\text{misclassification rate} = \frac{\#\{y_i \neq \hat{y}_i, i = 1 \dots m\}}{m}$$

where y_i is the true classification value, $\hat{y}_i$ is the predicted value and m is the size of the test set. On each dataset, we use 10-fold cross-validation to evaluate the performance of each method.

Hyperparameters: Based on the analysis in Section 7.4, we set the mini-batch size for MGD and MGD-DR to $g = \max\{n/1000, 2d \log d / \epsilon^2\}$ and $g = \max\{n/1000, 2r \log r / \epsilon^2\}$, respectively. The number $n/1000$ in batch size formula is to ensure our batch size isn't too small in case of large ϵ . In addition, we set $r = 20$. For all methods, we set the regularization factor $\lambda = 10^{-4}$.

Experimental results: Figure 7.2 shows the *mean squared error* (MSE) of the linear regression model generated by each method, under various values of ϵ . Private SGD incurs prohibitive errors in all cases, due to a large amount of noise in the gradient that it obtains in each iteration. MGD alleviates this issue with mini-batches, but still provides unsatisfactory accuracy. The reason, as we mentioned in Section 7.4.2, is that MGD requires using a large mini-batch size g when d is large, which in turn leads to a small total number of iterations and degrades its performance. MGD-DR overcomes the drawback of MGD by incorporating dimensionality reduction to reduce the required mini-batch size, and hence, it is

Figure 7.3: Experimental results for logistic regression and SVM.

able to achieve an accuracy that is close to the non-private SGD. However, this method is only applicable for linear regression, experimental evidents show that it isn't useful for logistic regression and SVM.

Figures 7.3 illustrate the *misclassification rate* of each method for logistic regression and SVM. Overall, our experimental results demonstrate the effectiveness of mini-batches and dimensionality reduction for empirical risk minimization under local differential privacy.

7.6 Summary

In this chapter, we have presented our work on local differential privacy. We propose a mechanism for private estimation of the mean value of numeric vectors. We have proved that our mechanism is asymptotic optimal in both accuracy and run-time complexity. In addition, we also propose a mechanism for estimating the frequencies of categorical values privately. It has also been proven to

---

be asymptotic optimal on accuracy and run-time complexity. We have applied these mechanisms to the problem of training machine learning models using empirical risk minimization framework. The experimental results have shown that our mechanisms outperform other approaches.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this thesis, we discuss the data privacy problem in the setting where the data analysis results on the private dataset are published in the public domain. We claim that differential privacy framework has many advantages compared to other privacy preservation techniques. We start by introducing the previous work on formalizing the privacy notion. Then, we introduce the definition and building blocks of differential privacy based on the idea that privacy of an individual is protected if the presence or absence of this individual in the dataset does not significantly change the probability distribution of the data analysis output.

After reviewing the related work, we then investigate the application of differential privacy to two important areas: survival analysis and user data collection. We investigate privacy for survival analysis which uses personal health records to estimate the survival rate of patients in medical research. Moreover, we also investigate local differential privacy which allows users to keep their own private data while the curator is still able to estimate useful information.

For survival analysis, we have investigated the following mechanisms and approaches:

- For nonparametric model, we have proposed a private mechanism for two

popular nonparametric estimators, namely Kaplan-Meier estimator and Nelson-Aalen estimator.

- For parametric models, we have proposed a simple private mechanism for accurately estimating the parameter of the exponential distribution. In addition, we have also proposed a private mechanism based on the local sensitivity concept for estimating the parameters of the Weibull distribution.
- For estimating uncertainty in parametric models, we have proposed a private framework which allows learning the posterior function. Moreover, we have applied the proposed framework to parametric models with Weibull distribution and flexible parametric models.
- We have proposed three private approaches for estimating the discrete-time survival regression model, namely extended output perturbation approach, extended objective perturbation approach, and posterior sampling approach.
- We have proposed the posterior sampling approach for continuous-time survival regression model. In addition, we have also proposed the posterior perturbing approach which supports a relaxation of differential privacy for scenarios in which differential privacy is impractical.

For user data collection, we have investigated the following mechanisms and approaches:

- We have proposed a mechanism which allows each user to publish a numeric vector.
- We have proposed a mechanism which allows each user to publish a randomized vector of categorical data.

- We have applied the proposed mechanisms to supervised learning problems under the empirical risk minimization framework.

Although our proposed posterior sampling approach and posterior perturbing approach have been applied only to survival regression, they can also be applied for other statistical models. However, the model must have a full likelihood function which is differentiable at any data point. Moreover, data points in the dataset are assumed to be independent from each other to guarantee that the total log-likelihood is the sum of the log-likelihood of each data point.

8.2 Future Work

In this research, we have proposed different mechanisms and approaches for parametric and nonparametric survival analysis, discrete-time and continuous-time survival regression, and local differential privacy. The following subsections discuss possible directions for future research work.

8.2.1 Differentially Private Cox Regression

In this thesis, we have proposed alternative models for Cox regression. Cox regression lacks a full likelihood function which leads to a partial likelihood objective function in which all the data records are mixed together. It would be an interesting research problem to design a differentially private mechanism for Cox regression directly. Because of its nature, a private mechanism for Cox regression may open up other opportunities in privatizing statistical models whose relationships between data points are much more complicated.

8.2.2 Differential Privacy for Bayesian Inference

Our proposed posterior sampling mechanism, which is used for survival regression, can also be applied to other machine learning models. To guarantee differential privacy, our proposed posterior sampling mechanism uses a finite likelihood function. It is an interesting research problem to find other methods and approaches which guarantee differential privacy without the finite constraint on the likelihood function. Moreover, we have also proposed mechanisms for publishing the posterior distribution of parametric survival models. One possible future extension is to learn the posterior distribution under the protection of differential privacy in a generic setting. In order to achieve this, we may need to use variational methods (such as mean field approximation [20] and others [66, 67]) to approximate the posterior distribution and then sample from the space of all possible approximations to guarantee differential privacy.

8.2.3 Local Differential Privacy

We have proposed mechanisms for local differential privacy which mainly aim to estimate the mean and frequency of data. It would be an interesting research direction to design local differentially private mechanisms for more complex statistical models such as graph models. This research direction may greatly improve the accuracy of the training model. Recently, there are many attempts [61, 89, 18] to use cryptographic techniques (e.g., multi-party computation) to achieve a similar effect as local differential privacy, i.e., protecting privacy without the need of a central server. One possible future research work is to combine cryptography techniques and differential privacy techniques to learn survival models from multiple data sources (e.g., multiple hospitals) without the need of sharing patients' data between these data sources.

8.2.4 Differential Privacy for Deep Learning Models

Currently, deep learning is one of the most important research areas in AI. Guaranteeing differential privacy for deep learning models is an important open research problem. As deep learning models require large datasets for training, it is important to consider the privacy aspects of these deep models. A private mechanism for deep learning models can illustrate the generalization property of deep models as differential privacy is related to generalization [85, 87]. Guaranteeing differential privacy is equivalent to not remembering any particular data point which can prevent overfitting for deep models. The current deep models use stochastic gradient descent for training which is an iterative process of going over the dataset many times. One may take advantage of recent developments in relaxation of differential privacy, namely *concentrated differential privacy* [42, 22, 79], which behaves better than differential privacy for iterative process such as stochastic gradient descent.

Appendix A

Spline Basis Functions

We give the definition of M-spline basis functions as follows. Given a set of knots $t_1 \leq t_2 \leq \dots \leq t_n$, where n is the number of free parameters. Let's define $t_{n+i} = t_i$ for $i = 1 \dots k$, where k is the order of M-spline. The M-spline basis functions are defined recursively as follows. For $k = 1$,

$$M_i(x \mid 1, t) = \begin{cases} \frac{1}{t_{i+1} - t_i} & \text{if } t_i \leq x \leq t_{i+1} \\ 0 & \text{if otherwise} \end{cases} \quad (\text{A.1})$$

For $k > 1$,

$$M_i(x \mid k, t) = \frac{k[(x - t_i)M_i(x \mid k - 1, t) + (t_{i+k} - x)M_{i+1}(x \mid k - 1, t)]}{(k - 1)(t_{i+k} - t_i)} \quad (\text{A.2})$$

I-spline basis is defined as follows:

$$I_i(x \mid k, t) = \int_{t_1}^x M_i(u \mid k, t) du$$

Figure A.1 shows an example of I-spline basis functions.

From Equation (A.1) and Equation (A.2), we can easily prove that $M_i(x \mid k, t)$ are non-negative. Therefore, $I_i(x \mid k, t)$ are non-decreasing functions. Then, C-spline basis functions are defined as:

$$C_i(x \mid k, t) = \int_x^{t_n} I_i(x \mid k, t)$$

Figure A.1: I-spline basis functions of order 3 with 5 knots.

Since our purpose is to approximate the functions of form a^x , the above definition of C-spline is different from the standard definition which is defined as follows:

$$C_i(x \mid k, t) = \int_{t_1}^x I_i(x \mid k, t)$$

Bibliography

- [1] Odd Aalen. “Nonparametric estimation of partial transition probabilities in multiple decrement models”. In: *The Annals of Statistics* (1978), pp. 534–545.
- [2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. “Deep learning with differential privacy”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 308–318.
- [3] Dimitris Achlioptas. “Database-friendly random projections”. In: *PODS*. 2001.
- [4] Charu C Aggarwal. “On k-anonymity and the curse of dimensionality”. In: *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment. 2005, pp. 901–909.
- [5] Rakesh Agrawal, Ramakrishnan Srikant, and Dilys Thomas. “Privacy preserving OLAP”. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM. 2005, pp. 251–262.
- [6] Sungjin Ahn, Anoop Korattikara Balan, and Max Welling. “Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring.” In: *ICML*. 2012.
- [7] Murray Aitkin and David Clayton. “The fitting of exponential, Weibull and extreme value distributions to complex censored survival data using GLIM”. In: *Applied Statistics* (1980), pp. 156–163.
- [8] Paul D Allison. “Discrete-time methods for the analysis of event histories”. In: *Sociological methodology* 13 (1982), pp. 61–98.

- [9] Per Kragh Andersen, Knut Borch-Johnsen, Torsten Deckert, Anders Green, Philip Hougaard, Niels Keiding, and Svend Kreiner. "A Cox regression model for the relative mortality and its application to diabetes mellitus survival data". In: *Biometrics* (1985), pp. 921–932.
- [10] Per Kragh Andersen and Richard David Gill. "Cox's regression model for counting processes: a large sample study". In: *The annals of statistics* (1982), pp. 1100–1120.
- [11] Therese M-L Andersson, Paul W Dickman, Sandra Eloranta, Mats Lambe, and Paul C Lambert. "Estimating the loss in expectation of life due to cancer using flexible parametric survival models". In: *Statistics in medicine* 32.30 (2013), pp. 5286–5300.
- [12] Franck Barthe, Olivier Guédon, Shahar Mendelson, Assaf Naor, et al. "A probabilistic approach to the geometry of the ℓ_p^n -ball". In: *The Annals of Probability* 33.2 (2005), pp. 480–513.
- [13] Robert Bassett and Julio Deride. "Maximum a posteriori estimators as a limit of Bayes estimators". In: *Mathematical Programming* (2016), pp. 1–16.
- [14] Raef Bassily and Adam Smith. "Local, Private, Efficient Protocols for Succinct Histograms". In: *STOC*. 2015, pp. 127–135.
- [15] Raef Bassily, Adam Smith, and Abhradeep Thakurta. "Private empirical risk minimization: Efficient algorithms and tight error bounds". In: *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE. 2014, pp. 464–473.
- [16] Roberto J Bayardo and Rakesh Agrawal. "Data privacy through optimal k-anonymization". In: *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*. IEEE. 2005, pp. 217–228.
- [17] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

- [18] Vincent Bindschaedler, Shantanu Rane, Alejandro E Brito, Vanishree Rao, and Ersin Uzun. “Achieving differential privacy in secure multiparty data aggregation protocols on star networks”. In: *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*. ACM. 2017, pp. 115–125.
- [19] Graeme Blair, Kosuke Imai, and Yang-Yang Zhou. “Design and analysis of the randomized response technique”. In: *Journal of the American Statistical Association* 110.511 (2015), pp. 1304–1319.
- [20] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational inference: A review for statisticians”. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877.
- [21] Charles George Broyden. “The convergence of a class of double-rank minimization algorithms 1. general considerations”. In: *IMA Journal of Applied Mathematics* 6.1 (1970), pp. 76–90.
- [22] Mark Bun and Thomas Steinke. “Concentrated differential privacy: Simplifications, extensions, and lower bounds”. In: *Theory of Cryptography Conference*. Springer. 2016, pp. 635–658.
- [23] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. “Stan: A probabilistic programming language”. In: *Journal of statistical software* 76.1 (2017).
- [24] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. “Differentially private empirical risk minimization”. In: *The Journal of Machine Learning Research* 12 (2011), pp. 1069–1109.
- [25] Tianqi Chen, Emily Fox, and Carlos Guestrin. “Stochastic gradient hamiltonian monte carlo”. In: *International Conference on Machine Learning*. 2014, pp. 1683–1691.

- [26] Tingting Chen and Sheng Zhong. "Privacy-preserving models for comparing survival curves using the logrank test". In: *Computer methods and programs in biomedicine* 104.2 (2011), pp. 249–253.
- [27] Enrico Colosimo, Flávio Ferreira, Maristela Oliveira, and Cleide Sousa. "Empirical comparisons between Kaplan-Meier and Nelson-Aalen survival function estimators". In: *Journal of Statistical Computation and Simulation* 72.4 (2002), pp. 299–308.
- [28] David R Cox. "Regression models and life-tables". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1972), pp. 187–220.
- [29] Tore Dalenius. "Towards a methodology for statistical disclosure control". In: *statistik Tidskrift* 15.429-444 (1977), pp. 2–1.
- [30] Angela Dispenzieri, Jerry A Katzmann, Robert A Kyle, Dirk R Larson, Terry M Therneau, Colin L Colby, Raynell J Clark, Graham P Mead, Shaji Kumar, L Joseph Melton, et al. "Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population". In: *Mayo Clinic Proceedings*. Vol. 87. 6. Elsevier. 2012, pp. 517–523.
- [31] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. "Local Privacy, Data Processing Inequalities, and Minimax Rates". In: *CoRR abs/1302.3203* (2013).
- [32] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. "Privacy Aware Learning". In: *J. ACM* 61.6 (2014), 38:1–38:57.
- [33] John C Duchi, Michael Jordan, Martin J Wainwright, et al. "Local privacy and statistical minimax rates". In: *FOCS*. 2013, pp. 429–438.
- [34] Cynthia Dwork. "Differential privacy". In: *Automata, languages and programming*. Springer, 2006, pp. 1–12.
- [35] Cynthia Dwork. "Differential privacy: A survey of results". In: *Theory and applications of models of computation*. Springer, 2008, pp. 1–19.

- [36] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. "Our data, ourselves: Privacy via distributed noise generation". In: *Advances in Cryptology-EUROCRYPT 2006*. Springer, 2006, pp. 486–503.
- [37] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *TCC*. 2006, pp. 265–284.
- [38] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating noise to sensitivity in private data analysis". In: *Theory of cryptography*. Springer, 2006, pp. 265–284.
- [39] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. "On the complexity of differentially private data release: efficient algorithms and hardness results". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM. 2009, pp. 381–390.
- [40] Cynthia Dwork and Kobbi Nissim. "Privacy-preserving datamining on vertically partitioned databases". In: *Advances in Cryptology-CRYPTO 2004*. Springer. 2004, pp. 528–544.
- [41] Cynthia Dwork and Aaron Roth. "The algorithmic foundations of differential privacy". In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407.
- [42] Cynthia Dwork and Guy N Rothblum. "Concentrated differential privacy". In: *arXiv preprint arXiv:1603.01887* (2016).
- [43] Cynthia Dwork and Adam Smith. "Differential privacy for statistics: What we know and what we want to learn". In: *Journal of Privacy and Confidentiality* 1.2 (2010), p. 2.

- [44] Khaled El Emam and Fida Kamal Dankar. "Protecting privacy using k-anonymity". In: *Journal of the American Medical Informatics Association* 15.5 (2008), pp. 627–637.
- [45] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response". In: *CCS*. 2014, pp. 1054–1067.
- [46] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. "Privacy preserving mining of association rules". In: *Information Systems* 29.4 (2004), pp. 343–364.
- [47] Giulia C. Fanti, Vasyl Pihur, and Úlfar Erlingsson. "Building a RAPPOR with the Unknown: Privacy-Preserving Learning of Associations and Data Dictionaries". In: *CoRR* abs/1503.01214 (2015).
- [48] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.
- [49] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin, 2001.
- [50] Glenn Fung, Shipeng Yu, Cary Dehing-Oberije, Dirk De Ruyscher, Philippe Lambin, Sriram Krishnan, and R Rao Bharat. "Privacy-Preserving Predictive Models for Lung Cancer Survival Analysis". In: *Practical Privacy-Preserving Data Mining* (2008), p. 40.
- [51] Wally R Gilks, NG Best, and KKC Tan. "Adaptive rejection Metropolis sampling within Gibbs sampling". In: *Applied Statistics* (1995), pp. 455–472.
- [52] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. "Differential privacy for functions and functional data". In: *Journal of Machine Learning Research* 14.Feb (2013), pp. 703–727.

- [53] Jihun Hamm, Adam C. Champion, Guoxing Chen, Mikhail Belkin, and Dong Xuan. "Crowd-ML: A Privacy-Preserving Learning Framework for a Crowd of Smart Devices". In: *ICDCS*. 2015, pp. 11–20.
- [54] Moritz Hardt and Kunal Talwar. "On the geometry of differential privacy". In: *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM. 2010, pp. 705–714.
- [55] W Keith Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: (1970).
- [56] Matthew D Hoffman and Andrew Gelman. "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.
- [57] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays". In: *PLoS Genet* 4.8 (2008), e1000167.
- [58] IPUMS. *Integrated Public Use Microdata Series*. <https://www.ipums.org>.
- [59] Lorentz M Irgens. "The Medical Birth Registry of Norway. Epidemiological research and surveillance throughout 30 years". In: *Acta obstetricia et gynecologica Scandinavica* 79.6 (2000), pp. 435–439.
- [60] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. "Discrete distribution estimation under local privacy". In: *arXiv preprint arXiv:1602.07387* (2016).
- [61] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. "Differentially private multi-party computation." In: *CISS*. 2016, pp. 128–132.

- [62] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. “Differentially Private Multi-party Computation: Optimality of Non-Interactive Randomized Response”. In: *arXiv preprint arXiv:1407.1546* (2014).
- [63] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. “Extremal mechanisms for local differential privacy”. In: *NIPS*. 2014, pp. 2879–2887.
- [64] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. “The Composition Theorem for Differential Privacy”. In: *Proceedings of the 32nd International Conference in Machine Learning(ICML)*. 2015, pp. 1376–1385.
- [65] Edward L Kaplan and Paul Meier. “Nonparametric estimation from incomplete observations”. In: *Journal of the American statistical association* 53.282 (1958), pp. 457–481.
- [66] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. “Improved variational inference with inverse autoregressive flow”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 4743–4751.
- [67] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).
- [68] David G Kleinbaum and Mitchel Klein. *Survival analysis*. Springer, 1996.
- [69] Robert A Kyle, Terry M Therneau, S Vincent Rajkumar, Dirk R Larson, Matthew F Plevak, Janice R Offord, Angela Dispenzieri, Jerry A Katzmann, and L Joseph Melton III. “Prevalence of monoclonal gammopathy of undetermined significance”. In: *New England Journal of Medicine* 354.13 (2006), pp. 1362–1369.
- [70] Paul C Lambert and Patrick Royston. “Further development of flexible parametric models for survival analysis”. In: *Stata Journal* 9.2 (2009), p. 265.

- [71] Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. “Pre-conditioned stochastic gradient Langevin dynamics for deep neural networks”. In: *arXiv preprint arXiv:1512.07666* (2015).
- [72] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. “t-closeness: Privacy beyond k-anonymity and l-diversity”. In: *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE. 2007, pp. 106–115.
- [73] Xiaoyu Lu, Valerio Perrone, Leonard Hasenclever, Yee Whye Teh, and Sebastian Vollmer. “Relativistic Monte Carlo”. In: *AISTATS*. 2017.
- [74] Yi-An Ma, Tianqi Chen, and Emily Fox. “A complete recipe for stochastic gradient MCMC”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 2917–2925.
- [75] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. “l-diversity: Privacy beyond k-anonymity”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007), p. 3.
- [76] Frank McSherry and Kunal Talwar. “Mechanism Design via Differential Privacy”. In: *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. Providence, RI: IEEE, Oct. 2007. URL: <https://www.microsoft.com/en-us/research/publication/mechanism-design-via-differential-privacy/>.
- [77] Adam Meyerson and Ryan Williams. “On the complexity of optimal k-anonymity”. In: *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM. 2004, pp. 223–228.
- [78] Rupert G Miller Jr. *Survival analysis*. Vol. 66. John Wiley & Sons, 2011.
- [79] Ilya Mironov. “Renyi differential privacy”. In: *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*. IEEE. 2017, pp. 263–275.

-
- [80] Yves-Alexandre de Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. "Unique in the crowd: The privacy bounds of human mobility". In: *Scientific reports* 3 (2013).
- [81] Arvind Narayanan and Vitaly Shmatikov. "Robust de-anonymization of large sparse datasets". In: *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE. 2008, pp. 111–125.
- [82] Wayne Nelson. "Theory and applications of hazard plotting for censored failure data". In: *Technometrics* 14.4 (1972), pp. 945–966.
- [83] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. "The geometry of differential privacy: the sparse and approximate cases". In: *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM. 2013, pp. 351–360.
- [84] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. "Smooth sensitivity and sampling in private data analysis". In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM. 2007, pp. 75–84.
- [85] Kobbi Nissim and Uri Stemmer. "On the generalization properties of differential privacy". In: *CoRR*, abs/1504.05800 (2015).
- [86] Christine M O’Keefe, Ross Stewart Sparks, Damien McAullay, and Bronwyn Loong. "Confidentialising survival analysis output in a remote data access system". In: *Journal of Privacy and Confidentiality* 4.1 (2012), p. 6.
- [87] Luca Oneto, Sandro Ridella, and Davide Anguita. "Differential privacy and generalization: Sharper bounds with applications". In: *Pattern Recognition Letters* 89 (2017), pp. 31–38.
- [88] Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data". In: *Proceedings of the International Conference*

- on Learning Representations*. 2017. URL: <https://arxiv.org/abs/1610.05755>.
- [89] Martin Pettai and Peeter Laud. "Combining differential privacy and secure multiparty computation". In: *Proceedings of the 31st Annual Computer Security Applications Conference*. ACM. 2015, pp. 421–430.
- [90] David Martin Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: (2011).
- [91] James O Ramsay et al. "Monotone regression splines in action". In: *Statistical science* 3.4 (1988), pp. 425–441.
- [92] Jason T Rich, J Gail Neely, Randal C Paniello, Courtney CJ Voelker, Brian Nussenbaum, and Eric W Wang. "A practical guide to understanding Kaplan-Meier curves". In: *Otolaryngology—Head and Neck Surgery* 143.3 (2010), pp. 331–336.
- [93] Patrick Royston, Paul C Lambert, et al. "Flexible parametric survival analysis using Stata: beyond the Cox model". In: (2011).
- [94] E Seiler and B Simon. "An inequality among determinants". In: *Proceedings of the National Academy of Sciences of the United States of America* 72.9 (1975), p. 3277.
- [95] Adam Smith. "Efficient, differentially private point estimators". In: *arXiv preprint arXiv:0809.4794* (2008).
- [96] Latanya Sweeney. "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.
- [97] Darrel R Thoman, Lee J Bain, and Charles E Antle. "Inferences on the parameters of the Weibull distribution". In: *Technometrics* 11.3 (1969), pp. 445–460.

- [98] Tim Van Erven and Peter Harremos. "Rényi divergence and Kullback-Leibler divergence". In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 3797–3820.
- [99] Yue Wang, Xintao Wu, and Donghui Hu. "Using Randomized Response for Differential Privacy Preserving Data Collection." In: *EDBT/ICDT Workshops*. Vol. 1558. 2016.
- [100] Stanley L. Warner. "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias". In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69.
- [101] Laura Wichert and Ralf A Wilke. "Simple non-parametric estimators for unemployment duration analysis". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 57.1 (2008), pp. 117–126.
- [102] Shipeng Yu, Glenn Fung, Romer Rosales, Sriram Krishnan, R Bharat Rao, Cary Dehing-Oberije, and Philippe Lambin. "Privacy-preserving cox regression for survival analysis". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2008, pp. 1034–1042.
- [103] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. "Private release of graph statistics using ladder functions". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM. 2015, pp. 731–745.
- [104] Yu Zhu and Lei Liu. "Optimal randomization for privacy preserving data mining". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2004, pp. 761–766.