ON PRIVACY RISK OF RELEASING DATA AND MODELS

ASHISH DEEPAK DANDEKAR

(B. Tech in Computer Engineering, College of Engineering, Pune)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2019

PhD Supervisor:

Associate Profeesor Stéphane Bressan

Examiners:

Dr Ng See Kiong

Associate Professor Xiao Xiaokui

Professor Albert Bifet *(Laboratoire Traitement ET Communication De L'information (LTCI))*

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Ashish Deepak Dandekar

May 2019

Acknowledgments

When I was in school, I had read quotes such as: “Alone we can do so little; together we can do so much!”. I felt the gravity of the quote when I started retrospecting my life in Singapore. There are so many people whose encouragement has culminated in this dissertation. This acknowledgement is a small note of thanks to all of them for being with me on my journey.

To begin with, I thank my supervisor, Stéphane Bressan, for being an unwavering source of motivation for me. There were umpteen moments when I would go to his cabin without any interesting results and hence without any hope. He always showed me how an uninteresting result might give rise to something interesting, something that you would not have thought of had you not conducted the experiment in the first place. He taught me to be objective and critical of the research findings. I am sure that I have given him a hard time proofreading my drafts, especially with the usage of ‘the’. My deepest gratitude goes to him for making my scientist dream a reality.

I would like to thank Talel Abdessalem for hosting me as a visiting researcher at Télécom ParisTech and offering his valuable feedback on my works. I am thankful to my colleagues in A-star and Singtel lab for giving me an exposure to real-world problems faced by industries. I am also thankful to the School of Computing for offering me a full-time Teaching Assistant position that helped me in fulfil my passion for teaching. I am equally thankful to my reviewers for their thorough feedback.

Research lab is the first home of a PhD student. I would like to thank Remmy, Deb, Agus, Naheed, Liu Qing, Fajrian, Thomas for sharing

my research efforts. I would like to offer special thanks to Deb, who with his strong theoretical background, and Remmy, who with his efficient programming skills, complemented my research ideas. There were a number of people in the School of Computing whom I got acquainted with over the years. I cannot forget morning lunches with Raj, Oana, Ayush, Suparna, Aseem and my afternoon tea with Astha and Sakshi. My special thanks go to Akanksha, and by transitivity to Tapeesh, for being my goto persons not only for brainstorming ideas but also for venting out the frustration in research at large.

UTown was my second home after the lab. I would like to thank my roommates Hung and Anisur for making my hostel a homely place. Aseem - a quintessentially Indian neighbour with whom I could talk at length on diverse issues over a cup of chai, Omkar, Apurva and Suparna - my stress busters, Neeraj, Karthik, Sanket, Pushkar - my study-roommates! I cannot thank anyone less for making my hostel memories golden.

There were friends, across the sea, who were always reachable to me via WhatsApp or Skype. I like to thank Rohit, Sneha, Gayatri, Antoine, Neha, Charanya, Yash, Vishwanjali, Srinath, Monika, Rushikesh, Apurva for pushing me hard to get the PhD.

When I was coming to Singapore, I was worried. It was the first time I was living away from my parents, away from my family. I met Sushant, a person with a "healthy" body and a genial face, who was sitting next to me on my first flight to Singapore. Later he introduced me to Pulkit and Sajitha. They are my family in Singapore. They have given me support in every possible way, whenever I needed it. I always look up to Sushant and Sajitha for their unfaltering determination towards work. It was Pulkit's compassionate friendship that offered me solace in life. I have relished my days in Singapore in the company

of Neeraj-Madhura, Pushkar-Ketki, Pranali and the trio - my fellow globetrotters! What have I not done with them? Endless card games, food-explorations, weekend walks, culinary improvisations and many more things. I am indebted by what they have done for me. It is very hard to even imagine my life without them.

Sometimes when the life is in doldrums, a few things miraculously happen and save you from the chaos. Two things happened to me. First one is the serendipitous encounter with Chaitanya. He taught me to live with a free spirit. His words have always been an inspiration to me. The second one is classical music. I accidentally met Chandranath Bhattacharya, my Guruji. His mellifluous notes have not only made my life musical but also offered me the inner peace.

Last but not the least! There are a few people whom you thank for the sake of formality. Expression of gratitude can never lessen the debt that we owe to them. They are parents and siblings. I thank Aai, Baba and Dada for having faith in me. If it wasn't for your support, I would not have been what I am today. In the end, I am thankful to the God for giving me an affectionate family.

Abstract

The earliest reference to secure data exchange can be traced back to the Sumerian civilisation but privacy concerns developed as early as the late 19th century when the US census bureau considered the publication and exchange of data. Contents of the census dataset are privy to the citizens. Release of census dataset, rather any dataset, for the research purpose increases the risk of breach of privacy.

We begin our work by studying statistical disclosure based risk assessment that was traditionally carried out by census bureaus. Statistical disclosure based risk assessment relies on various scenarios that are hand-crafted by domain experts, thereby restricting the assessment to those scenarios. Synthetically generated datasets, which preserve the relationships among attributes from the original dataset, do not contain any data-point that relates to a data-point in the real world. Therefore, it is convenient to release synthetic datasets in the research community.

We conduct experiments on traditional techniques of partially and fully synthetic dataset generation. Although fully synthetic dataset generation techniques safeguard against the identity disclosure, they often result in a large drop in the utility of the dataset. Additionally, the traditional techniques use discriminative machine learning models that generally capture one particular relationship among the attributes. Therefore, they fail to capture the true generative process of the data at large. We complement our experiments by using a generative model to generate synthetic data. We adapt and extend Latent Dirichlet Allocation, which is a generative model, to handle spatiotemporal data. We use the proposed adaptation to analyse trav-

elling patterns of the commuters in the public transportation network of Singapore. We also use the same model to release synthetically generated travel records of commuters.

Use of machine learning models for generating synthetic datasets does not completely nullify the risk of breach of privacy. Machine learning models leak information about the training datasets from their outputs. We use differential privacy to provide privacy guarantees for machine learning models. We consider a scenario wherein a business entity monetises their dataset by releasing the machine learning models, trained on the dataset, with differential privacy guarantees. We study this scenario in light of the two classes of machine learning models: parametric models and non-parametric models. We understand that the parametric models can be released by publishing parameters of the model whereas the non-parametric models require the release of training data along with the parameters. Therefore, we suggest the use of machine learning as a service for non-parametric models whereas we release the parameters of parametric models. We use the functional mechanism to provide privacy guarantees for a parametric model such as linear regression and its regularised variants. We use functional perturbation to provide privacy guarantees for non-parametric models such as kernel density estimator, support vector machine and Gaussian process regression.

We find two difficulties while using differential privacy in a real-world setting. Firstly, the privacy level ϵ in differential privacy, which encodes the privacy guarantee, is too abstract to be actionable in a business setting. Secondly, the privacy level in differential privacy is an upper bound on the worst case loss of privacy by using a machine learning model. The worst case guarantee, at times, leads to a severe loss in utility of the machine learning model. In order to solve the latter difficulty, we propose privacy at risk that is a probabilistic

extension of differential privacy. It provides confidence bounds on the privacy level by accounting for various sources of randomness. We solve the former problem by proposing a cost model that bridges the gap between the privacy level and the compensation budget estimated by a GDPR compliant business entity. The proposed cost model also helps in balancing the privacy-utility tradeoff.

In conclusion, we study the problem of data privacy and propose solutions in two different aspects. Synthetic dataset generation lies at the heart of the first aspect and differential privacy lies at the heart of the second.

Contents

List of Figures

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

The earliest available reference to secure data exchange dates back to 3500BC when Sumerian farmers were using clay tokens to seal the earthen pots [1]. Farmers used to go to Ziggurat to safely store their grain into earthen pots that were sealed by unique clay tokens made for every farmer. Every farmer was given a copy of the clay token that was used for sealing his earthen pot. In the following year, the farmers used to go back to Ziggurat to claim their own share of the seed grains by matching their tokens with tokens on the sealed pots. Later, Sumerians invented the art of writing and cuneiform took the place of clay tokens. With the spread of art of writing, we observe various references for data exchange and book-keeping in multiple ancient civilisations such as Egyptian and Roman civilisations. All historical references point to different ways of exchanging data among different parties and the security measures used while exchanging the data. But, it was not until the 19th century when data processing became a necessity due to increasing volumes of data.

In 1880, US census bureau estimated that it would take 8 years to process and analyse the data that they had collected for the 1880 national census [2]. With increasing population, the processing time was bound to increase in the following census events. In 1881, Herman Hollerith, an employee of the US census bureau, invented a punch card based tabulating-machine that reduced the processing

[1] https://medium.com/@cryptoleaf/a-potted-history-of-money-and-computation-63b86c2bd46

[2] https://www.weforum.org/agenda/2015/02/a-brief-history-of-big-data-everyone-should-read/

time of 8 years to 3 months. Census data processing marked the beginning of not only data storage systems but also large-scale data processing.

Census dataset contains information that spans topics such as health, education, marital status, employment status about the citizens. Analysis of the census data helps a government in designing future policies that improve the quality of life of the citizens by solving existing problems. Census dataset is also used by non-governmental organisations, educational institutes and businesses for research purposes. The information in any census dataset is privy to every individual in the dataset and hence its usage raises the concerns related to leakage of private information of the citizens.

Traditionally, the census bureau used to employ pre-emptive techniques that would reduce the chance of statistical disclosure from the released dataset. The techniques included value aggregation, value suppression and noise addition. Value aggregation reduces the granularity of an attribute. For instance, aggregation of numeric attributes uses mean or median values whereas aggregation of categorical attributes merges different categories in the attributes. Value suppression redacts the values of attributes that are deemed to be sensitive from the dataset. For instance, HIPAA [U.S04] uses a list of 18 attributes that any HIPAA compliant company has to remove before releasing any health-related dataset. Noise addition adds random noise to the values of attributes. Effectiveness of these *statistical disclosure control* based techniques was evaluated by estimating the risk of disclosure under the assumption of the external information possessed by an intruder [WDW12]. The risk assessment was carried out under the scenarios that were crafted by the domain experts.

We do not find any record of blatant privacy violation in the dataset release until the late 1990s. In 1998, Latanya Sweeney, then an MIT graduate student, showed a promising attack on the privacy by re-identifying the medical records of the governor of Massachusetts [Swe97]. She did so by linking two publicly available datasets, anonymised medical records and the voter list, on three attributes

namely zip-code, birthdate and sex. In 2000, she further showed that 87% of the Americans can be uniquely identified based on the same three attributes. This attack catalysed the need for quantifiable privacy definition in place of the scenario based risk assessment. It led to the proposition of privacy definitions of k-anonymity [Swe02], l- diversity [MGKV06] and t-closeness [LLV07]. These definitions are known under an umbrella term of *anonymisation* based techniques for the release of datasets.

Inception of the World Wide Web in the 90s gave birth to a new source of data. Companies started amassing data at an unprecedented scale and at the granularity as fine as the time spent by a user on a certain webpage. Collected data was analysed to model the profile of a user and send her the advertisements that catered to her profile. Until today, the targetted advertising [ISVB05] is the business model that drives the internet.

There are two ways for a company to earn money using the collected dataset. Firstly, the company sells samples drawn from the collected dataset. This way resembles the way of releasing datasets by the census bureau. In order to avoid disclosure, the company needs to enforce the same pre-emptive techniques used by the census bureau before releasing the samples. Secondly, the company processes the collected data and sells the results. This way is inspired by the plethora of research in machine learning. Machine learning models, when applied to the collected dataset, learn various latent patterns in it. Latent patterns encode relationships among various attributes in the dataset. These relationships offer insights that are valuable to design business strategies. Thus, results of the processed dataset are an equally valuable source of money. The use of machine learning models as a processing tool gives rise to two different scenarios. Firstly, it is possible to release the model as a summary of the latent patterns. Secondly, it is possible to provide the model as a service to a user wherein the model would respond to a query from the user.

Releasing the results without any measures to protect privacy of the data-points in

the underlying data is as harmful as releasing the dataset itself. Researchers have shown successful attacks on the machine learning models that serve as the pieces of evidence of leakage of information through the results [SSSS17, HSR+08, FJR15]. In the recent lawsuit on Facebook, where Facebook failed to compel Cambridge Analytica to delete all traces of data including its derivatives [3], led to 134 billion dollars drop in its market share. In 2006, the New York Times successfully recovered the identity of a woman using the publicly released web search query dataset by AOL [4]. In 2008, Narayan et al. [NS08] were able to re-identify several users in the publicly released anonymised Netflix dataset by linking user data to the publicly available IMDb dataset. These attacks have increased the concerns of privacy of the users while releasing the datasets as well as the results after processing the datasets.

Due to these hard evidences of breaches of privacy, organisations need to provide quantifiable privacy guarantees while monetising collected datasets in any way. Since the last decade, differential privacy has become the prevalent privacy definition for machine learning models. Researchers have developed privacy-preserving mechanisms that introduce randomness in either machine learning models or their outputs [DR+14, CMS11, ZZX+12, HRW12]. Under appropriate calibration of parameters, these privacy-preserving mechanisms satisfy differential privacy. One has to take appropriate care when applying privacy-preserving mechanisms for releasing the results of the machine learning model. The privacy-preserving mechanisms that introduce calibrated noise in the results before releasing them, suffer from degradation in the privacy guarantees with increasing number of queries. When a machine learning model is provided as a service, a user can query such a service with any query as many times as she desires. Therefore, naive use of differentially private privacy-preserving mechanism may lead to poor privacy guarantees that are not robust against the number of queries.

[3] https://www.theguardian.com/uk-news/2018/may/06/cambridge-analytica-kept-facebook-data-models-through-us-election

[4] https://www.nytimes.com/2006/08/09/technology/09aol.html

In the age of computers, digital signatures have taken place of the clay tokens of the Sumerian era. They are safeguarding against the security issues while exchanging the data among different parties. But at the same time, availability of datasets and availability of computational power to process them have given rise to a sophisticated problem of data privacy. People do want to share their data with companies to enjoy the services that are catered to their individual tastes. But at the same time, they do not want to get singled out from the population. This is the problem of data privacy - ensuring the privacy of individuals while learning from the data.

1.1 Challenges and contribution

A company can monetise data that it collects in two ways. Either it earns money by selling samples drawn from the collected data or it earns money by selling machine learning models trained on the collected data. The first way results in blatant privacy violation whereas the second way leaks the information about training data through the model and the outputs. Therefore, in either of the cases, there is a risk of breach of privacy of the users whose information is present in the collected dataset. In this thesis, we address the concern of privacy risk in two cases namely, while releasing the datasets and while releasing the machine learning models.

Privacy risk of releasing datasets

Unrestricted availability of the datasets is important for researchers to evaluate their strategies to solve the research problems. In order to effectively learn latent patterns, researchers require access to good quality datasets. Many times, good quality datasets are proprietary and they are not freely available. Additionally, even if a business entity decides to make them available, it must comply with the prevalent privacy laws. Generally, privacy laws require redaction or recoding of

sensitive attributes in the dataset before it is publicly released. Such pre-emptive measures alter the patterns in the dataset and hence reduce its usability.

Synthetic dataset generation is an attractive solution to these problems. Since the data-points in a synthetic dataset do not trivially relate to any data-point in reality, synthetic datasets address the problem of data privacy in a unique way that does not require any explicit privacy assessment. Synthetic datasets are generated by ensuring preservation of the desired patterns from the original datasets. Thus, synthetic datasets provide a way to release good quality datasets with a lowered risk of breach of privacy.

We start the exploration of synthetic dataset generation by conducting experiments on traditional techniques from the statistical research community. In the early 90s, Rubin [Rub93] proposed a procedure to synthetically generate any sensitive attribute in the dataset. The procedure makes use of sampling from the posterior distribution of sensitive attribute given the rest of the attributes in the dataset. Reiter et al. [Rei03] extended the idea to fully and partially synthetic datasets. They use machine learning models to generate the values of sensitive attributes in the dataset. We extend these experiments with the use of neural networks to generate synthetic datasets. Fully synthetic datasets are generated by synthetically generating all attributes in the dataset one by one. Partially synthetic datasets are generated by synthetically generating values for sensitive attributes of only those data-points that bear a higher risk of disclosure.

Through these experiments, we learn that the fully synthetic dataset generation suffers from a few drawbacks. Although fully synthetic datasets safeguard against the risk of identity disclosure, utility of these datasets is very poor. One by one synthesis of the attributes leads to this observation. Quality of data for the attribute that is synthesised in the end of a sequence heavily relies on the quality of the data synthesised before it. Additionally, these statistical techniques use discriminative machine learning models to synthesise the data. Discriminative models are machine learning models that predict a particular attribute in the

dataset given rest of the attributes in the data. Therefore, they fail to generalise the true data generation at large.

We alleviate these problems by fitting generative models on a dataset. A generative model simulates a generative process for the data that is designed by taking into account various factors that contribute to data generation. It models each of these factors by a probability distribution whose parameters are learned for a given dataset. In the end, the generative model learns a joint distribution over all of the factors, which accounts for interdependence among them. Each of these factors can be fine-tuned to synthesise data with the desired quality.

We develop two generative models in two use cases. Firstly, we adapt and extend Latent Dirichlet Allocation, which is a widely used generative model in topic modelling, to handle spatiotemporal data. We use the adaptation to analyse travelling patterns of commuters in the public transportation network of Singapore. We further use the same model to synthetically generate datasets of travel records of the commuters. Secondly, we propose a recurrent neural network based model that learns latent patterns in annotated sequence data. We use the proposed model to synthetically generate datasets of news headlines for a specified topic[5].

Privacy risk of releasing machine learning models

Machine learning models learn latent patterns in data. These latent patterns provide insights into various relationships in the data that are very helpful to businesses to design their strategies. An organisation, which collects data from users, can sell machine learning models trained on the collected data instead of releasing the dataset. Researchers show that machine learning models leak information about training dataset from their parameters as well as outputs. Thus, the use of machine learning models without any quantifiable privacy guarantees

[5]Generation of news headlines does not directly bear any risk of disclosure of sensitive information. Hence, in the light of theme of this thesis, we provide this work in Appendix B

increases the risk of breach of privacy of the data-points in the training datasets.

We use differential privacy as privacy definition to provide privacy guarantees for machine learning algorithms. We study differential privacy of two classes of machine learning models namely parametric models and non-parametric models. A parametric model assumes a parametric formula for the relationship among different variables whereas non-parametric model does not make such an assumption. Parametric models can be released by releasing this parametric form since these parameters are sufficient to compute the output. Non-parametric models need to release training dataset along with the parameters of the model. We suggest presenting non-parametric models as a service, which is an alternative to release of training datasets.

In the case of parametric models, we can not control the number of evaluations once we release the parameters of the model. In the case of non-parametric models, which we provide a service to users, we can not control the number of evaluations of model. Therefore, in either of the cases, we do not have control over the number of evaluations of the model. Sequential composition theorem of differential privacy states that the privacy guarantee of a randomised algorithm that explicitly introduces noise in the output of a query degrades with the number of queries. Thus, we need to take extra care while providing robust privacy guarantees for machine learning algorithms.

We propose the use of the functional mechanism and functional perturbation to provide differential privacy guarantees for parametric models and non-parametric models respectively. The functional mechanism introduces noise in the loss function of a machine learning model whereas functional perturbation perturbs the expansion of machine learning model in an appropriate functional space. We instantiate the functional mechanism for linear regression and its regularised variants. We instantiate the use of functional perturbation for various kernel non-parametric models such as kernel density estimation, kernel SVM and Gaussian process regression. We perform experiments on real-world US census dataset.

Through these experiments, we observe two main difficulties with differential privacy guarantees in a real-world setting. Firstly, the privacy guarantee in differential privacy is too abstract to be actionable in a business setting. Secondly, the privacy guarantee of differential privacy is an upper bound on the worst case loss of privacy. This worst-case guarantee, at times, requires a large amount of perturbation that leads to a severe drop in the utility of the algorithm. We solve the latter difficulty by proposing *privacy at risk* that is a probabilistic extension of differential privacy. It provides confidence bounds on the differential privacy by accounting for various sources of randomness. We instantiate privacy at risk for widely used Laplace mechanism and provide analytical results. We solve the former problem by proposing a cost model that bridges the gap between differential privacy guarantee and the compensation budget estimated by a GDPR [gdp16] compliant business entity. We apply the proposed model to privacy at risk and show how a business entity is able to save money without compromising privacy under a realistic scenario. Additionally, the proposed cost model also helps in balancing the privacy-utility tradeoff.

1.2 Structure

As the title of the thesis reads, we divide main part of this thesis into three parts. The first part discusses privacy risk under release of the datasets. In this part, we present our work on generative models and synthetic dataset generation. The second part discusses privacy under release of machine learning models. We present our work on differential private machine learning models in this part. Figure 1.1 presents the overall structure of the this thesis.

We begin our thesis with a brief background of machine learning and data privacy in Chapter 2. Expert readers may skip this chapter. In the first part, we address the privacy risk while releasing the dataset by releasing synthetic datasets. Specifically, we propose the use of machine learning models that are trained on the private data to generate synthetic datasets. We experiment with

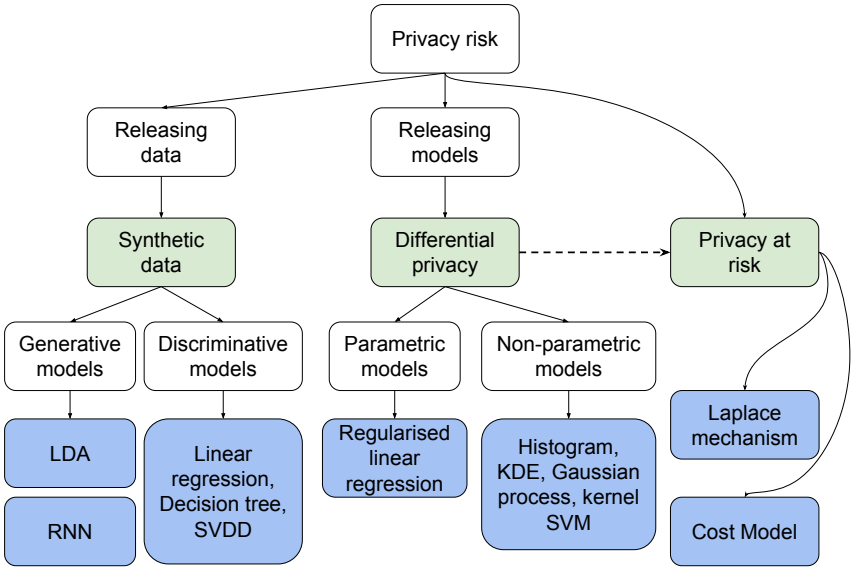


Figure 1.1: Structure of the thesis.

two classes of machine learning models, namely, discriminative and generative machine learning models. In Chapter 3, we explicate the need of synthetic data generation. It is followed by comparative evaluation of traditional synthetic dataset generation techniques that use discriminative machine learning models in Chapter 4. In Chapter 5, we illustrate an application of synthetic data generation using a generative model. We adapt and extend Latent Dirichlet Allocation to handle spatiotemporal data and hence use it to generate travelling records of commuters in the public transportation network of Singapore.

In the second part, we address the privacy risk while releasing models by providing provable differential privacy guarantees. Specifically, we provide differential privacy guarantee for a selection of parametric as well as non-parametric machine learning models. In Chapter 6, we illustrate the use of Function mechanism [ZZX+12] to provide privacy guarantee for regularised linear regression. In Chapter 7, we illustrate the use of functional perturbation [HRW13a] to provide four non-parametric machine learning models.

In the third part, we propose privacy at risk in Chapter 8 in the spirit of making

differential privacy amenable to a business setting. We demonstrate privacy at risk for the Laplace mechanism. We also propose a cost model of the compensation budget and illustrate its usage for a hypothetical but realistic scenario. In Chapter 9, we conclude our work and provide future directions in the research.

Chapter 2

Background

2.1 Machine learning

Machine learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty [Mur12a]. It is possible to represent the dataset as well as the patterns in the dataset as mathematical sets. Let f be an abstract function that maps the set of all possible datasets to the set of patterns. The machine learning problem is to learn this mapping.

Specifically, let $\mathcal{D}$ denote a universe of different datasets. Let, $\mathcal{H}$ denote a set of functions, called as *hypotheses set*. Each function in $\mathcal{H}$ maps any dataset in $\mathcal{D}$ to a set of patterns corresponding to the dataset. Let, $L : (\mathcal{H} \times \mathcal{D}) \rightarrow \mathbb{R}$ a measure that quantifies the goodness of fit of a function in the hypotheses set to any dataset in $\mathcal{D}$. Thus, for a given dataset $D \in \mathcal{D}$, called as the *training dataset*, the machine learning problem is an optimisation problem that yields the optimal $h^* \in \mathcal{H}$ that minimises or maximises L.

Based on the type of datasets and the hypotheses set, the machine learning problem is studied under different categories. *Supervised learning* is performed on a labelled dataset whereas *unsupervised learning* is performed on an unlabelled dataset. In *parametric learning*, the size of the hypotheses set does not change with the number of data-points in the dataset whereas in *non-parametric learning* the hypotheses set grows with the number of data-points. The probabilistic

treatment to the machine leaning problem yields two classes of models, namely *discriminative models* and *generative models*. We now explain each of these categories with their merits and demerits.

2.1.1 Supervised and unsupervised learning

Suppose that each dataset in $\mathcal{D}$ is a labelled dataset. Each dataset $D \in \mathcal{D}$ is represented as a set of n data-points $\{(x_i, y_i)|x_i \in X, y_i \in Y\}_{i=1}^{n}$ for some sets X and Y. Thus, $D \subset X \times Y$. Supervised learning problem is to a learn a function $f : X \to Y$ that fits well to a specified dataset. Variant of the problem with Y as a continuous variable, i.e. $Y \subset \mathbb{R}^d$ for $d > 0$, is known as *regression* whereas variant of the problem with Y as a categorical variable, i.e. $Y = \{1, ..., C\}$ with $C \in \mathbb{Z}^{>=1}$, is known as *classification*.

Example 1 (**Linear regression**). *Linear regression is a supervised learning problem that works on the datasets with $X = \mathbb{R}^d$ for some $d > 0$ attributes and $Y = \mathbb{R}$. Linear regression gets its name from the choice of function $f = \omega^T x$ where $\omega, x \in \mathbb{R}^d$. Thus, the hypothesis set $\mathcal{H}$ for linear regression is the set of all possible linear combinations of d attributes. Linear regression uses mean squared loss as the measure for goodness of fit. It estimates ω^* that minimises over the mean squared loss, calculated using Equation 2.1, for a given training dataset D.*

$$L(\omega, D) = \frac{1}{N} \sum_{i=1}^{N} (\omega^T x_i - y_i)^2 \tag{2.1}$$

Now suppose that we do not have access to a labelled datasets. Each dataset $D \in \mathcal{D}$ is represented as a set n data-points $\{x_i|x_i \in X\}_{i=1}^{n}$ for some set X. Let, K be an unknown number of patterns that we hypothesise to exist in a dataset. Unsupervised learning problem is to learn a function $f : X \to \{1, ..., K\}$. The goal of the unsupervised learning problem is to learn latent patterns in the dataset; hence, it is sometimes called as *knowledge discovery* [Mur12b].

Example 2 (**Kernel density estimation**). *Kernel density estimation (KDE) is an unsupervised learning problem that works on any unlabelled dataset in $\mathbb{R}^d$.*

KDE estimates the probability density function of a dataset D in $\mathbb{R}^d$. For a dataset with n data-points, it assumes a kernel κ_θ, which is a known probability distribution with its hyper-parameter θ, centred at every data-point. Probability of any new data-point is the average of probability of the new data-point being associated to each kernel κ_{θ_i}. Thus, the hypotheses set $\mathcal{H}$ for KDE consists of n-tuples, each tuple representing the values of n hyper-parameters. KDE uses likelihood of the dataset as the measure of goodness of fit. It estimates $\Theta^* \in \mathbb{R}^n$ that maximises the likelihood of the given dataset D using Equation 2.2.

$$L(\Theta, D) = \frac{1}{n} \sum_{x \in D} \mathbb{P}(x) = \frac{1}{n} \sum_{x \in D} \sum_{i=1}^{n} \kappa_{\theta_i}(x - x_i) \tag{2.2}$$

In summary, the supervised learning problem works on labelled datasets whereas the unsupervised learning problem works on unlabelled datasets. The supervised learning problem usually have a quantifiable measure to check the goodness of fit. Due to unavailability of the labels, unlabelled datasets generally lack the existence of the ground truth to validate the goodness of the fit. For instance, *clustering* is an unsupervised learning problem that clusters the data-points by discovering the latent patterns in the dataset. The dataset lacks the ground-truth for the patterns since the patterns are not known apriori. Hence, at times one needs to consult with the domain experts to analyse the patterns discovered by the unsupervised learning problem. Finding an objective and quantitative metric measure of goodness of fit for any general unsupervised learning problem is still a research question.

2.1.2 Parametric and non-parametric learning

The mapping learned by any machine learning problem is governed by a set of parameters. For instance, the mapping $f(x) = ax^2 + bx + c$ has three parameters: a, b and c. A machine learning problem with the fixed number of parameters is called as the parametric learning problem whereas a machine learning problem for which the number of parameters increases with the size of the data-points in

the training dataset is called as the non-parametric learning problem.

Linear regression problem in the Example 1 is a parametric learning problem. The hypothesis set is $\mathcal{H} = \{\omega | \omega \in \mathbb{R}^d\}$. Thus, each function in the hypothesis set has d parameters. The number of parameters is fixed to d irrespective of the data-points in the training dataset, which is used to minimise Equation 2.1. Kernel density problem in the Example 2 is a non-parametric learning problem. The hypothesis set is $\mathcal{H} = \{(\theta_1, \theta_2, ..., \theta_n) | \theta_i \in \mathbb{R}, 1 \leq i \leq n\}$. Thus each function in the hypotheses set has n parameters, where n is the number of data-points in the training dataset.

In summary, parametric learning problem uses the hypotheses set that is not as complex as the hypotheses set used by the non-parametric learning problem. Parametric learning problems are more efficient and more easy to interpret than the non-parametric learning problems. On the one hand, non-parametric learning problems, such as kernel density estimation, K-means clustering and Gaussian process, require access to the entire training dataset to compute the result for any input. On the other hand, parametric learning problems do not require training dataset after the parameters are learnt. Despite these demerits of non-parametric learning problems, the capability of fitting large number of mappings gives them a cutting edge over parametric learning problems. Parametric learning problems lack this capability due to assumption of a certain parametric form. For instance, linear regression problem is not able to give accurate results if the relationship in the dataset is inherently non-linear.

2.1.3 Probabilistic perspective

Probabilistic perspective of machine learning assumes that observed data is sampled from an unknown probability distribution and a hypothesis set comprises of family of probability distributions. Learning involves estimation of the parameters of probability distribution using the training dataset. There are two classes of machine learning models based on the assumption of probability distribution.

They are: *discriminative models* and *generative models.*

Let us reconsider the setting of a universe of labelled datasets $\mathcal{D}$ from supervised machine learning models. Discriminative models learn the conditional probability distribution $\mathbb{P}(Y|X)$. By virtue of the choice of this hypothesis, discriminative models are used for either predicting or classifying Y attribute of a data-point for a given set of attributes X.

Example 3 (**Logistic regression**). *In its simple setting, logistic regression is a binary classifier. Let us reconsider labelled datasets from the case of supervised models. Every dataset is of the form* $D = \{(x_i, y_i)|x_i \in \mathbb{R}^d, y_i \in \{0, 1\}\}_{i=1}^n$. *Logistic regression is a parametric model with parameters* $\omega \in \mathbb{R}^d$. *Logistic regression assumes parametric form of the following probability distribution*

$$\mathbb{P}(y = 1|\omega, x) \triangleq sigm(w^t x) = \frac{1}{1 + e^{-\omega^t x}} \tag{2.3}$$

Logistic regression estimates the parameters ω *by maximising likelihood, computed using the Equation 2.3 over the specified training dataset.*

Generative models learn the joint distribution $\mathbb{P}(X, Y)$ by assuming a prior distribution over a set of attributes $\mathbb{P}(Y)$ and a conditional probability distribution $\mathbb{P}(X|Y)$. Generative models are used in two way. Firstly, generative models are used to generate synthetic data by sampling values of Y attributes from $\mathbb{P}(Y)$ followed by sampling values of Y from $\mathbb{P}(X|Y)$. Secondly, generative models are used for either predicting or classifying Y attribute for any specified values of X attributes by means of the Bayes rule.

Example 4 (**Naive Bayes classifier**). *Let us reconsider labelled datasets from the case of supervised models. Every dataset is of the form* $D = \{(x_i, y_i)|x_i \in \mathbb{R}^d, y_i \in \{1, ..., C\}\}_{i=1}^n$ *with* $C > 0$. *Naive Bayes model assumes a prior distribution over class labels* $\mathbb{P}(Y)$ *and a conditional probability distribution* $\mathbb{P}(X|Y)$. *It uses Bayes rule in Equation 2.4 to compute the class label for a specified data-point* x.

$$\mathbb{P}(y = c_i|x) \propto \mathbb{P}(x|y = c_i)\mathbb{P}(y = c_i) \tag{2.4}$$

Naive Bayes model outputs the class label that maximises the probability in Equation 2.4.

Generative models tend to generalise more than discriminative models due to the assumption of the prior distribution. By means of the conditional distribution, $\mathbb{P}(X|Y)$, generative models learn the distribution of data x under various labels. Discriminative classifiers tend to learn class boundaries whereas generative classifiers tend to learn distribution of individual classes. Hence, generative models are able to generate data for a specified class label.

2.2 Data privacy

Word 'privacy' in its ordinary usage as well as its legal usage carries multitudes of meanings. The earliest reference dates back to 1890 when Warren and Brandies [WB90] defined it to be "the right to be left alone". Starting from the 20th Century, U. S. Census Bureau accentuated the need for large scale data analysis, which led to refinements in the notion of privacy. Data analysis tools find insights in the data that can be used for the betterment of society or for designing strategies for the businesses. Naturally, data analysis tools enable analysts to learn something about the data, which was not known to them before analysing it. Thus, inappropriate use of data analysis tools may lead to a breach of privacy of individuals.

The first reference to the notion of a privacy breach through the results of data analyses dates back in 1977. Delanius [Dal77] describes: "If a release of the statistics S makes it possible to determine the value of confidential data more accurately than is possible without access to S, a *disclosure* has taken place". There are two kinds of disclosures: *identity disclosure* and *attribute disclosure*. Identity disclosure occurs if the released statistics makes it possible to identify the presence of an individual in the dataset. Attribute disclosure occurs if the released statistics makes it possible to identify a certain feature of an individual

in the dataset. Disclosure of either of these kinds leads to the breach of privacy of the individual. Legislative entities design privacy laws in order to reduce the risk of breach of privacy of individuals. An organisation has to comply with prevalent privacy laws while releasing datasets or statistics computed on the datasets. The compliance involves either removal of sensitive features from datasets [U.S04] or assessment of disclosure risk.

2.2.1 Statistical disclosure control

Statistical disclosure control [WDW12] (also known as *statistical disclosure limitation*) is an umbrella term for a group of techniques employed by an organisation to reduce the risk of disclosure while releasing the dataset. Assessment of disclosure risk involves scenarios that hand-crafted by domain experts. These scenarios make assumptions about the auxiliary knowledge about the released datasets that might be available through other data sources. Risk of identity or attribute disclosure is computed under assumptions made in the scenarios. Statistical disclosure control techniques are divided into two types: perturbative techniques and non-perturbative techniques. Traditionally, perturbative techniques involve recoding, suppression, noise addition to the values of attributes whereas non-perturbative techniques include shuffling, generalisation. These traditional techniques depend not only on the type of attribute of data but also depends on the specifics of data. For instance, perturbative techniques need to treat categorical and numerical attributed differently.

In order to avoid the dependence on datasets, syntactic definitions such as k-anonymity [Swe02], l-diversity [MGKV06] and t-closeness [LLV07] are proposed. Syntactic definitions split attributes of data into two categories: *identifying attributes* and *sensitive attributes*. Identifying attributes are the attributes of data-points that are available through sources other than the released dataset whereas sensitive attributes are the attributes that are unique to the released dataset. A dataset satisfies k-anonymity if it can be partitioned in a way that

every partitions contains at least k-data-points that share the same values for identifying attributes. An intruder who is interested in the record of a particular person ends up with at least k records with values of identifying attribute same as the values of identifying attributes of the person of interest. Thus, k-anonymity reduces the risk of identity disclosure. If all data-points in an equivalence class of k-anonymised dataset bears same values for sensitive variables, then it leads to attribute disclosure of these data-points. l-diversity enforces l representative values of sensitive attributes in every equivalence class in the dataset. t-closeness further enforces the distance between distributions of values of the sensitive attributes in equivalence classes and the entire dataset to be at most t. t-closeness alleviates the attack on l-diversity due to skewed distribution of sensitive value in the anonymised datasets [LLV07].

2.2.2 Differential privacy

Differential privacy [DKM$^+$06] is a privacy definition to provide provable bounds on the privacy loss from a randomised function. It quantifies the privacy loss in terms of the indistinguishability between the outputs when any pair of *neighbouring datasets* are provided as inputs to a randomised function. We now describe the formalism of differential privacy along with a few important results.

We consider a universe of datasets $\mathcal{D}$. Two datasets of equal cardinality x and y are said to be *neighbouring datasets* if they differ in one data point. A pair of neighbouring datasets is denoted by $x \sim y$. *Queries* are the functions that map a given dataset to a desired domain. Many a times, the desired domain is the set of real numbers. Queries with the set of real numbers as a domain are called as *numeric queries*. For instance, the sum query returns sum of the values of a specified attribute in a dataset. Many queries that we use in our day-to-day life on the datasets are *deterministic* queries, i.e. there exists a fixed output for a given input dataset. In order to achieve a differential privacy guarantee, a *privacy-preserving mechanism*, which is a randomised algorithm, explicitly adds

random noise drawn from a specified probability distribution to the query. Thus, a privacy-preserving mechanism, $\mathcal{M}(f, \Theta)$, for a query f and a set of parameters Θ of the given noise distribution, is a function that maps a dataset into a real vector, i.e. $\mathcal{M}(f, \Theta) : \mathcal{D} \to \mathbb{R}^k$. We denote a privacy-preserving mechanism as $\mathcal{M}$, when the query and the parameters are clear from the context.

Definition 1 (**Differential privacy [DR+14]**). *A privacy-preserving mechanism $\mathcal{M}$, equipped with a query f and with parameters Θ, is (ϵ, δ)-differentially private if for all $Z \subseteq Range(\mathcal{M})$ and $x, y \in \mathcal{D}$ such that $x \sim y$:*

$$\mathbb{P}(\mathcal{M}(f, \Theta)(x) \in Z) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(f, \Theta)(y) \in Z) + \delta \tag{2.5}$$

$(\epsilon, 0)$-differential privacy is often referred as ϵ-differential privacy.

Privacy level ϵ quantifies the privacy guarantee provided by ϵ-differential privacy. If the outputs of a privacy-preserving mechanism are highly indistinguishable, inequality in Equation 2.5 tends to become an equality. Therefore, a smaller value of ϵ, provides higher privacy than its larger values.

Note. Under differential privacy guarantee, inclusion or exclusion of any datapoint has a bounded impact on the output of a privacy-preserving mechanism. Hence, differential privacy quantifies the risk of identity disclosure. The privacy level ϵ is an upper bound on the privacy under the consideration of all possible pairs of neighbouring datasets and all possible outputs. Unlike techniques of statistical disclosure control, differential privacy does not require hand-crafted scenarios.

Important results

We now present a few mathematical results that are direct consequences of Definition 1.

- **Non-uniqueness of the privacy level.** Any ϵ-differentially private privacy-preserving mechanism satisfies ϵ'-differential privacy for any $\epsilon' \geq \epsilon$.

- **Sequential composition [DR$^+$14].** Let us consider a set of privacy-preserving mechanisms $\{\mathcal{M}_i(f_i, \Theta_i) : \mathcal{D} \rightarrow R_i\}_{i=1}^n$. Each of these privacy-preserving mechanisms satisfy ϵ differential privacy. Sequential composition of these mechanisms $\mathcal{M} : \mathcal{D} \rightarrow (R_1 \times R_2 ... \times R_n)$ satisfies $(n\epsilon)$-differentially private.

- **Parallel composition [DR$^+$14].** Let us consider a set of privacy-preserving mechanisms $\{\mathcal{M}_i(f_i, \Theta_i) : \mathcal{D}_i \rightarrow R_i\}_{i=1}^n$ where $\mathcal{D}_i's$ are disjoint partitions over $\mathcal{D}$. Each of these privacy-preserving mechanisms satisfy ϵ differential privacy. Parallel composition of these mechanisms $\mathcal{M} : (\mathcal{D}_i \times \mathcal{D}_2 ... \times \mathcal{D}_n) \rightarrow (R_1 \times R_2 ... \times R_n)$ satisfies ϵ-differentially privacy.

- **Privacy level of deterministic functions.** Definition 1 necessitates $\mathcal{M}$ to be a randomised function. For a deterministic function, it is always possible to find two neighbouring datasets that output two different values. Probability of obtaining a certain output from a deterministic function is either zero or one. The upper bound on the privacy level ϵ for deterministic functions is infinity, which is synonymous to "no differential privacy". Therefore, deterministic functions do not satisfy differential privacy for any non-trivial privacy level.

Privacy-preserving mechanism

In order to satisfy ϵ-differential privacy, the parameters of a privacy-preserving mechanism require calculated calibration. The amount of noise required to achieve a specified privacy level depends on the query. If the output of the query does not change drastically for two neighbouring datasets, then lesser amount of noise is required to achieve a given privacy level. The measure of maximum amount of fluctuation in the output of queries for pair of neighbouring datasets is called the *sensitivity* of the query.

Definition 2 (**Sensitivity [DR$^+$14]**). *The sensitivity of a query* $f : \mathcal{D} \rightarrow \mathbb{R}^k$

is defined as

$$\Delta_f \triangleq \max_{\substack{x,y\in\mathcal{D}\\ x\sim y}} \|f(x)-f(y)\|_1.$$

Laplace mechanism is one of the widely used privacy-preserving mechanisms. It adds scaled noise sampled from a calibrated Laplace distribution to the numeric query. In order to satisfy ϵ-differential privacy, the calibration of the Laplacian noise depends on the sensitivity of the query and the desired privacy level ϵ.

Definition 3 (**Laplace distribution [PP02]**). *The Laplace distribution with mean zero and scale $b > 0$ is a probability distribution with probability density function*

$$\mathrm{Lap}(b) \triangleq \frac{1}{2b}\exp\left(-\frac{|x|}{b}\right),$$

where $x \in \mathbb{R}$. We write $\mathrm{Lap}(b)$ to denote a random variable $X \sim \mathrm{Lap}(b)$

Definition 4 (**Laplace mechanism [DR$^+$14]**). *Given any function $f : \mathcal{D} \to \mathbb{R}^k$ and any $x \in \mathcal{D}$, the Laplace Mechanism is defined as*

$$\mathcal{L}_\epsilon^{\Delta_f}(x) \triangleq \mathcal{M}\left(f,\ \frac{\Delta_f}{\epsilon}\right)(x) = f(x) + (L_1, ..., L_k),$$

where L_i is drawn from $\mathrm{Lap}\left(\frac{\Delta_f}{\epsilon}\right)$ and added to the i^{th} component of $f(x)$.

Theorem 1 (**[DR$^+$14]**). *The Laplace mechanism, $\mathcal{L}_{\epsilon_0}^{\Delta_f}$, is ϵ_0-differentially private.*

Note that for the value of sensitivity Δ smaller than Δ_f, $\mathcal{L}_{\epsilon_0}^{\Delta}$ is not ϵ_0-differentially private.

Part I

On privacy risk of releasing data

In this part, we present our works on synthetic dataset generation. Firstly, we present the comparative evaluation of synthetic dataset generation techniques namely fully synthetic dataset generation and partially synthetic dataset generation. Followed by that, we present two works on generative models. The first work synthetically generates travelling records of commuters in a public transportation network. The second work synthetically generates news headlines for a specified topic. Works in this part are published at following venues:

- Ashish Dandekar, Remmy A. M. Zen, and Stéphane Bressan. A comparative study of synthetic dataset generation techniques. In *Database and Expert Systems Applications- 29th International Conference, DEXA 2018, Regensburg, Germany, Proceedings, Part II, pages 387-395*

- Ashish Dandekar, Remmy A. M. Zen, and Stéphane Bressan. Comparative evaluation of data generation methods. In *Deep Learning Security Workshop, Singapore, December 2017. (Poster)*

- Ashish Dandekar, Stéphane Bressan, Talel Abdessalem, Huayu Wu, Wee Siong Ng. Detecting communities of commuters: graph based techniques versus generative models. In *On the Move to Meaningful Internet Systems: OTM 2016 Conferences, CoopIS 2016, Rhodes, Greece, Proceedings, pages 482-502*

- Ashish Dandekar, Stéphane Bressan, Talel Abdessalem, Huayu Wu, Wee Siong Ng. Trajectory simulation in communities of commuters. *In International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016, Jakarta, Indonesia, Proceedings, pages 39-42 (Invited paper)*

Chapter 3

Synthetic dataset generation

Unrestricted availability of datasets is important for researchers to evaluate their strategies to solve their research problems. While it is important to make datasets publicly accessible, it is equally important to protect the privacy of the respective data owners. Synthetic datasets that preserve utility while protecting the privacy of the data-owners stands as a midway solution to this problem.

In this chapter, we provide motivation for using the synthetic datasets in order to avoid identity disclosure. Machine learning models that are trained on the private data retain latent properties in the dataset. Therefore, we use machine learning models to generate synthetic datasets. We outline traditional techniques of fully and partially synthetic datasets from statistical research community. We defer implementation of these methods to Chapter 4 and Chapter 5.

3.1 Introduction

There exists issue of data privacy whenever organisations need to exchange data with each other. Exchanged data may contain private information of stakeholders of one organisation that other the organisation may exploit. Therefore, organisations traditionally use statistical disclosure control based techniques to redact sensitive information in the private data and exchange *de-identified data*. We mainly observe two difficulties in the application of statistical disclosure control based techniques on the datasets.

Firstly, statistical disclosure control based techniques suffer from the loss in utility. Kifer and Gehrke [KG06] observe that k-anonymity and l-diversity do not preserve utility in the data. Sweeney [Swe01] mentions that meaningful solutions of k-anonymised datasets are application specific. They also depend on the user-specific preferences. Most of these anonymisation [LDR06] algorithms use statistical disclosure control techniques such as suppression and generalisation. There have been follow-up works [Iye02, Sam01] that alter suppression techniques to improve utility that is specific to the application in the study. Despite these efforts anonymisation algorithms are not efficient enough to use them on large datasets.

Secondly, effectiveness of statistical disclosure control based techniques heavily relies on the scenarios that delineate assumptions about auxiliary knowledge of an intruder about the dataset. Such a scenario based evaluation is not complete. Some other datasets, which are publicly released after publication of the dataset of interest, may reveal information about the current dataset. An intruder may exploit this auxiliary information to perform an attack on the dataset. Thus, use of statistical disclosure techniques creates a never-ending arms race between de-identification and re-identification[1] [BDR18].

Thus, statistical disclosure control based techniques suffer from not only poor efficiency but also lack of generality in the disclosure risk assessment. We suggest the use of synthetic datasets to tackle privacy while sharing data. Synthetic datasets puts a full-stop to the arms race between de-identification and re-identification [BDR18]. Data-points in a synthetic dataset do not correspond any data-point in a real dataset, from which the synthetic dataset is generated. Therefore, an intruder, even with the complete knowledge of identifying attributes in the dataset, fails to disclose identity of any user. Additionally, synthetic datasets are generated in a way that ensures the utility in the data.

[1] Act of an intruder who successfully discloses information from a de-identified dataset is called *re-identification*.

Organisations use machine learning algorithms to analyse their data. We propose to use the same machine learning algorithms, which are trained on private data to learn latent patterns int the data, to generate synthetic datasets. In this introductory chapter, we provide a summary of traditional synthetic dataset generation methods from the statistical literature. We defer the experiments on a selection of machine learning models to Chapter 4 and Chapter 5. In Chapter 4, we use four discriminative models to comparatively evaluate the tradition techniques synthetic dataset generations. In Chapter 5, we illustrate an application that uses a generative model to generate dataset. Particularly, we adapt Latent Dirichlet Allocation to synthetically generate travelling records of commuters in the public transportation network of Singapore.

3.2 Related work

Synthetic dataset generation work stems from the early works of data imputation to fill in the missing values in the surveys [Rub86]. In [Rub93], Rubin proposes a procedure to generate fully synthetic dataset that uses multiple imputation technique to synthetically generates values for a set of attributes for all datapoints in the dataset. Although it is advantageous to synthetically generate values for all datapoints, it is not always a necessity. Partially synthetic datasets, proposed by Little [Lit93], are generated by synthetically generating the values of the attributes that are sensitive to public disclosure. Various dataset synthesisers such as decision tree [Rei05b, CR10, Dre10] have been used to generate fully and partially synthetic datasets. Drechsler et al. [DR11] have performed an empirical comparative study between different dataset synthesisers. Comparison between fully and partially synthetic datasets can be found in [DBR08]. Recently, Nowok et al. [NRD16] have created an R package *synthpop* that provides basic functionalities to generate synthetic datasets and perform statistical evaluation.

Effectiveness of the synthetic datasets lies in the amount of utility they retain from the original dataset. Most of the works [Rei03, Rei05b, CR10, DR11] use

statistical methods of estimation for the evaluation of utility. They use estimators of mean and variance to calculate confidence intervals. Regression analyses are used to test whether the relationships among different variables are preserved. Aside from these analysis specific measures, Woo et al. [WROK09] and Karr et al. [KKO+06] have proposed global measures such as Kullback-Leibler (KL) divergence, extension of propensity score and cluster analysis based measure.

One of the prime motivations behind publicly releasing synthetic dataset instead of original datasets is to maintain the privacy of the data owners. In [Rei05a], Reiter introduces formalism to compute risk of disclosure in synthetically generated datasets using multiple imputation. The same formalism has been used in [Rei05b, DR11] to evaluate the risk of disclosure. For further details, readers are requested to refer to [Dre11].

3.3 Synthetic dataset generation using multiple imputation

3.3.1 Multiple imputation

Consider a dataset of size n sampled from a population of size N. Let Y_{nobs} denote subset of attributes in the dataset whose values are either missing for some data-points or sensitive towards the public disclosure. Rubin [Rub86] proposes to synthetically generate values for Y_{nobs} given the knowledge of rest of the attributes in the dataset, say Y_{obs}.

Let, $\mathcal{M}$ be a dataset synthesiser that generates values for an attribute Y_i given the information about rest of the attributes, denoted as Y_{-i}. With the help of $\mathcal{M}$, an imputer independently synthesises values of Y_{nobs} m times and releases m synthetic datasets $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^m\}$. In order to synthesise multiple sensitive attributes, we follow the procedure presented in [Dre10]. It defines an order on the attributes that are to be synthesised. Values of the first attribute are

synthesised by training the dataset synthesiser on the original dataset. For any later attributes, the dataset synthesiser is trained on the dataset with synthetic values of the attributes preceding it. Interested readers can refer to [Rei05b] for a detailed discussion on choosing the order of synthesis.

The reason behind for releasing m different datasets and combining estimators on each dataset is two folds. Firstly, there is randomness in the dataset due to sampling from the population. Secondly, there is randomness in the dataset due to imputed values. In order two capture these variabilities, framework of multiple imputation proposes the release of m datasets.

3.3.2 Fully synthetic dataset generation

Consider a dataset of size n sampled from a population of size N. Suppose that an imputer knows the values of a set of variables X for the entire population and values for rest of the variables, Y, only for a selected small sample. Let, Y_{inc} and Y_{exc} denote values of variables which are included in the sample and excluded from the sample respectively. The imputer synthetically generates values of Y_{exc} using a dataset synthesizer $\mathcal{M}$ trained on Y_{inc} and X. This synthesis is equivalent to performing multiple imputation with Y_{exc} as Y_{nobs} and Y_{inc} as Y_{obs}. Publicly released datasets, $\mathcal{D}$, comprise of m samples selected synthetically generated population. In order to statistically estimate an attribute Q, we use the estimators of mean and variance presented in [Rei03]. Figure 3.1 shows schematic diagram of fully synthetic dataset generation.

Theoretically, fully synthetic datasets provides 100% guarantee against the identity disclosure [Rub93]. Since $n << N$, it is less probable to have record from the original sample in the final dataset. Final datasets are sampled from synthetic population datasets in which $N - n$ records are synthetically generated.

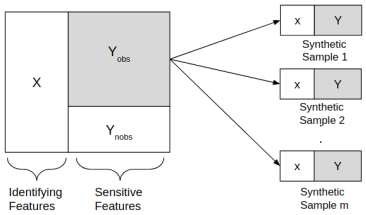


Figure 3.1: Schematic diagram for fully synthetic dataset generation.

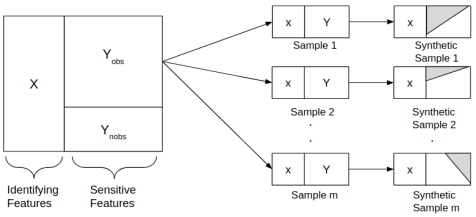


Figure 3.2: Schematic diagram for partially synthetic dataset generation.

3.3.3 Partially synthetic dataset generation

Let S be a dataset of size n sampled from a population of size N. In order to protect the sensitive information, an imputer decides to alter values of a set of attributes, Y, for a subset of data-points in S. Let Z be a binary vector of size n. Z_i takes value one if Y values of the i^{th} datapoint are to be synthetically generated and Z_i takes value zero if values of Y attributes are not be altered.

Let $Y_{syn} = \{Y_i | \forall i, Z_i = 1\}$ and $Y_{org} = \{Y_i | \forall i Z_i = 0\}$. We generate m partially synthetic datasets by multiple imputation. In this case, Y_{syn} are the data-points, with missing values, that we synthetically generate by training a dataset synthesiser on the available data, i.e Y_{org}. This synthesis is equivalent to performing multiple imputation with Y_{syn} as Y_{nobs} and Y_{org} as Y_{obs}. Publicly released datasets, $\mathcal{D}$, comprise of m datasets sampled from the population wherein values of attributes in Y are synthetically generated, as specified by Z, for each of the dataset. In order to statistically estimate an attribute Q, we use the estimators of mean and variance presented in [RRR03]. Figure 3.2 shows schematic diagram of fully synthetic dataset generation.

3.4 Privacy risks for synthetic data

Traditional methods of fully and synthetic data generation need a data synthesiser that outputs values of a sensitive variable given values of identifying

variables as inputs. Discriminative machine learning models learn parameters of posterior distribution of response attribute given predictor attributes. Thus, discriminative models befit the task of data synthesisers with sensitive attribute as the response attribute and identifying attributes as the predictor attributes. A generative machine learning model assumes a data generation process that prescribes assumptions regarding various dependencies among attributes of the data. It uses these dependencies to fit a probability distribution over the entire training data. Thus, generative machine learning models are a natural choice for synthetic dataset generation.

On the one hand, traditional methods generate a synthetic data-point by using values of identifying attributes of a real data-point to generate values for its sensitive attributes. Every synthetic data-point has one to one correspondence with the a real data-point. Therefore, synthetic datasets generated using the traditional techniques bear identity disclosure risk as well as attribute disclosure risk. On the other hand, generative models generate data in a truly synthetic fashion. They synthetically generate values for every attribute of any data-point. Such a data-point does not belong to any data-point in the original dataset. Therefore, synthetic datasets generated using generative models do not bear identity disclosure risk.

Having said that, synthetic dataset generation is not the panacea for the problem of data privacy. Firstly, although synthetic datasets provide protection against identity disclosure, they may contain a data-point with values of attributes that exactly match a data-point in the real world. Overfitting of a data synthesiser to the training data increases the probability of such an event. Discriminative models tend to overfit data more that generative models [Mur12a]. Therefore, overfit discriminative models used for generating synthetic attributes in a dataset would increase the probability of such events. Secondly, synthetic datasets do not provide a strong privacy guarantee against attribute disclosure. A data analyst can use a good quality synthetic dataset to learn relationships among the attributes and predict the sensitive attribute of a known point.

Chapter 4

A comparative study of synthetic dataset generation techniques

In this chapter, we comparatively evaluate fully synthetic dataset generation and partially synthetic dataset generation using different dataset synthesisers, namely, linear regression, decision tree, random forest and neural network. We comparatively evaluate effectiveness, as utility preserved and risk of disclosure, and efficiency of the synthetic dataset generation techniques. Given the trade-off between the efficiency and effectiveness, we observe that decision trees are not only efficient but also competitively effective compared to other dataset synthesisers.

4.1 Introduction

On the one hand, the philosophy of open data dictates that if the valuable datasets are made publicly available, the problems can be crowdsourced in the expectation to obtain the best possible solution. On the other hand, business organisations have their concerns regarding the public release of the datasets which may lead to the breach of private and sensitive information of stakeholders. In order to mitigate the risk of breach of privacy, agencies employ different techniques such as reordering or recoding of sensitive variables, shuffling values among different records. In order to have faithful analyses, publicly released datasets need to retain the utility from the original dataset. But standard disclosure control techniques result in the loss of utility [KG06]. Synthetic datasets provide a

midway in this privacy-utility tradeoff. Synthetic datasets are generated in a way such that it ensures retention of utility. At the same time, synthetic datasets do not contain any real-world data-point and hence safeguard against the risk of identity disclosure.

In this chapter various discriminative models as data synthesisers to comparatively evaluate fully and partially synthetic dataset generation techniques described in Chapter 5. We conduct experiments on census datasets. We extend the experiments in [DR11] by using neural networks to synthesise data.

4.2 Discriminative data synthesisers

We discuss different dataset synthesisers, which are discriminative models, that we use for the comparative study. In order to synthesise an attribute Y_i we train a dataset synthesiser to train on the information about rest of attributes Y_{-i}.

We use **linear regression** [Mur12a] as a baseline dataset synthesiser. In order to generate synthetic data, for every dataset and for every attribute Y_i to be synthetically generated, we learn the parameters of the regression model using the dataset with attributes in Y_{-i}. We generate values by sampling from a Gaussian distribution with a constant variance and the mean as determined parameters of regression.

We use the technique, proposed by Reiter [Rei05b], that uses **classification and regression tree** [Mur12a] to generate partially synthetic datasets. The procedure starts with building a decision tree using the values of the attributes that are available in the dataset Y_{-i}. In order to synthesise the value of an attribute Y_i for a data-point j, we trace along branches of the tree using the known attributes of j until we reach the leaf node. Let L_j be the set of values of Y_i in the leaf node. For a categorical attribute Y_i, Reiter proposes Bayesian bootstrap sampling to choose m different values. For a continuous attribute Y_i, we fit a kernel density estimator over the values in L_j and sample m values from

the estimate.

We use the technique, proposed by Caiola et al. [CR10], that uses **random forest** [Mur12a] to generate partially synthetic datasets. In order to synthesise values for a certain attribute Y_i, they train a fixed number decision trees on random samples of training dataset Y_{-i}. For a categorical attribute, the collection of results from constituent decision tree forms a multinomial distribution. m values are sampled from this distribution as the synthetic values for Y_i. For a continuous attribute, they propose use of a kernel density estimator over the results from decision trees and sample values from the estimator.

We use **neural network** [Mur12a] that learns an abstract function mapping an input to the corresponding output as a data synthesiser. If we consider K-class classification problem, the output layer of a neural network comprises of K nodes, with each node representing the probability of the respective class being the output of the model. We treat every attribute as a categorical variable. In order to synthesise value of an attribute Y_i, we train a neural network using features in Y_{-i}. We sample a value for attribute Y_i using the output layer as a multinomial distribution.

4.3 Empirical evaluation

4.3.1 Dataset and experimental setup

Dataset. We conduct experiments on a microdata sample of US Census in 2000 provided by IPUMS International [IPU09]. The dataset consists of 1% sample of the original census data. It spans over 1.23 million households with records of 2.8 million people. It has several attributes of which not every single attribute is reported by all of the people. In order to avoid these discrepancies in the data, we consider the records of 316,276 heads of households as the population. Table 4.1 shows the set of features which we consider for experiments.

Table 4.1: Schema for the census dataset.

Attribute Name	Variable Type	Notes
House Type	Categorical	
Family Size	Ordinal	
Sex	Categorical	
Age	Ordinal	5.9% have age less than 26
Marital Status	Categorical	
Race	Categorical	
Educational Status	Categorical	
Employment Status	Categorical	
Income	Ordinal	7.13% have income more than 70000
Birth Place	Categorical	

Experimental setup. All programs are run on Linux machine with quad core 2.40GHz Intel® Core i7™processor with 8GB memory. The machine is equipped with two Nvidia GTX 1080 GPUs. Python® 2.7.6 is used as the scripting language.

4.3.2 Metrics of evaluation

Utility evaluation

Utility of the generated dataset needs to be evaluated at two different levels. Firstly, we need to evaluate differences between the distribution of values of original attribute and synthetically generated attributes. Secondly, we need to evaluate the difference between the quality of a statistical estimator for an attribute on synthetic dataset and original data.

Let $y \in Y$ be any attribute that we synthetically generate from an original dataset. We calculate the similarity between the overall distribution of values of y by calculating **normalised KL-divergence** between the distribution of

values of y in population and the distribution of synthetically generated values. For m synthetic datasets, we consider mean of the normalised KL-divergence over individual datasets. Closer the value to 1, more similar the synthetically generated values are to the original values.

Karr et al. [KKO+06] develop a mechanism based on **overlap** between confidence intervals to evaluate the effectiveness of a statistical estimator. We estimate mean and variance of y using the point estimators described Section 3.3. We construct a 95% confidence interval around the estimator. Let (L_s, U_s) be confidence interval for synthetically generated y and (L_o, U_o) be interval from original data. We compute intersection of these intervals denoted as (L_i, U_i). The overlap utility measure is calculated using Equation 4.1.

$$I = \frac{(U_i - L_i)}{2(U_o - L_o)} + \frac{(U_i - L_i)}{2(U_s - L_s)} \tag{4.1}$$

If the intervals are similar to each other, we say that the synthetic dataset generation procedure preserves the utility. If the utility is preserved, the value of I is close to one and $I = 0$ refers to dissimilar confidence intervals.

Privacy risk evaluation

We follow the procedure in Reiter [DR08, Rei03] to estimate the **risk of disclosure** in the synthetically generated datasets. We assume that the intruder has complete information about an auxiliary variable, say region of birth, which is not a sensitive variable. Let $\mathbf{t}$ be a vector of information possessed by an intruder. For every data-point j in the dataset, the intruder calculates the probability of the datapoint j being the record of interest. In Equation 4.2, $N_{(\mathbf{t},i)}$ denotes the number of records in dataset D^i that match target. $\mathbb{I}(Y_j^i = \mathbf{t})$ is the identity function that equals to 1 if j^{th} data-point in the dataset D^i matches $\mathbf{t}$ otherwise 0.

$$Pr(J = j|D, \mathbf{t}) \;=\; \frac{1}{m}\sum_{i=1}^{m}\frac{1}{N_{(\mathbf{t},i)}}\mathbb{I}(Y_j^i = \mathbf{t}) \tag{4.2}$$

The intruder selects data-points with maximum probability value. This process is repeated for every target data-point in $\mathbf{t}$. In order to evaluate the risk of disclosure, we calculate *true match rate (true MR)* and *false match rate (false MR)* as defined in [Rei03, DR08]. Smaller the true match rate, better is the performance of a dataset synthesiser.

For a data-point $j \in \mathbf{t}$, an intruder may find multiple data-points with the same value of maximum probability. Let, R denotes the set of datapoints in $\mathbf{t}$ for which only one data-point in the dataset is matched with highest probability. Set R can be decomposed into two mutually exhaustive sets T and F that denote the set of data-points with true matches and false matches respectively.

$$\begin{aligned}\text{true match rate} &= \frac{|T|}{|\mathbf{t}|}\\ \text{false match rate} &= \frac{|F|}{|R|}\end{aligned} \tag{4.3}$$

4.3.3 Results analysis

The process starts by drawing 1% sample from the population, which we treat as the original dataset. We synthetically generate values for two attributes: income and age, in the same order. We generate 5 synthetic datasets for each original dataset. We repeat this procedure for 500 original datasets and mean of various metrics over 500 iterations is reported. In order to generate partially synthetic datasets, we need to define the cutoffs for the values of attribute that determine quantify the sensitivity of the attribute towards disclosure. We consider datapoints that have more than 70000$ income value and less than 26 age value to be the ones with sensitive information.

Utility evaluation results for the *age* attribute for partially synthetic datasets and fully synthetic datasets are presented in Table 4.2 and Table 4.3 respectively. We

Table 4.2: Evaluation of utility for partially synthetic datasets generated using different dataset synthesisers.

Feature	Data synthesisers	Original Sample Mean	Partially Synthetic Data		
			Synthetic Mean	Overlap	Norm KL Div.
Income	Linear Regression	27112.61	27117.99	0.98	0.54
	Decision Tree	27143.93	27131.14	0.94	0.53
	Random Forest	27107.04	27254.38	0.95	0.58
	Neural Network	27069.95	27370.99	0.81	0.54
Age	Linear Regression	49.83	24.69	0.50	0.55
	Decision Tree	49.83	49.83	0.90	0.56
	Random Forest	49.82	49.74	0.95	0.56
	Neural Network	49.87	49.78	0.90	0.56

Table 4.3: Evaluation of utility for fully synthetic datasets generate using different dataset synthesisers.

Feature	Data synthesisers	Original Sample Mean	Fully Synthetic Data		
			Synthetic Mean	Overlap	Norm KL Div.
Income	Linear Regression	27112.61	27074.80	0.52	0.55
	Decision Tree	27081.45	27091.02	0.55	0.58
	Random Forest	27107.04	28720.93	0.54	0.64
	Neural Network	27185.26	26694.54	0.54	0.94
Age	Linear Regression	49.83	-192.21	0.50	0.56
	Decision Tree	49.83	49.83	0.56	0.56
	Random Forest	49.82	46.25	0.68	0.57
	Neural Network	49.76	54.32	0.75	0.99

observe that although two techniques show comparable values of synthetic means, the technique of partially synthetic dataset generation shows greater extent of the overlap Partially synthetic dataset generation does not replace all values of the attributes in the sample. Therefore, we observe higher overlap for partially synthetic datasets. We also observe a large deviation in the sample mean of from its original mean in case of linear regression. Linear regression in the absence of any regularization suffers from overfitting [Mur12a]. Due to the order of synthesis, linear regression model is fit on the synthetically generated values of *income* while synthesising value for *age*. Thus, it overfits the synthetic data and fails to capture exact distribution of values in the original dataset. Decision tree and

Table 4.4: Evaluation of risk of disclosure for different dataset synthesisers under the scenario described in Section 4.3.3.

Dataset synthesisers	Target is in the sample		Target may be in the sample	
	True MR	False MR	True MR	False MR
Linear Regression	0.06	0.82	0.00	0.00
Decision Tree	0.18	0.68	0.00	0.99
Random Forest	0.35	0.50	0.00	0.99
Neural Network	0.03	0.92	0.00	0.99

other models are not prone to overfitting the training dataset and hence do not show such a degradation in utility.

In order to evaluate the risk of disclosure, we require a scenario. We assume that an intruder is interested in people who are born in US and have income more than $250,000$\$. All these people are the targets of the intruder. Intruder tries to match every single target with the records in the released datasets. We consider two records perfectly match if the people representing the records are born in US, they have income more than $250,000$\$ and the age of the person in dataset in within the tolerance of 2 compared to target person.

Two cases arise in the evaluation. For a given target, the intruder may or may not know if the target person is included in the released sample. We observe that, in the case when the intruder does not have certainty about inclusion of target in the sample, risk of disclosure is the least. In most of the cases, the targets might not be present in the released sample which leads to true match rate of 0. Observing the results for the case when a target is present in the sample, we see that neural networks comparatively offer better performance than rest of the dataset synthesisers.

We present the results of comparative efficiency of both these techniques using different dataset synthesisers in Table 4.5. We observe that the neural networks achieve the low risk of disclosure at the cost of a higher running time than the time taken by linear regression or decision trees.

Table 4.5: Comparative evaluation of efficiency of synthetic dataset generation techniques for different data synthesisers. Each cell shows the running time required, in seconds, to generate 5 synthetic datasets.

Dataset synthesiser	Partially Synthetic Dataset Generation	Fully Synthetic Dataset Generation
Linear Regression	0.040	0.068
Decision Tree	0.048	0.533
Random Forest	3.350	103.543
Neural Network	0.510	55.26

4.4 Discussion

In this work, we comparatively evaluate fully and partially synthetic dataset generation techniques using different dataset synthesisers, namely linear regression, decision tree, random forest and neural network. We comparatively evaluate effectiveness, in terms of utility preservation and risk of disclosure, and efficiency of these techniques. The analysis shows that decision trees stand as a good dataset synthesiser given its high effectiveness compared to other data synthesisers. This observation agrees with the result in [DR11].

We use a well-structured dataset in this work. Many real-world datasets do not have a well defined structure. For instance, social network datasets or the datasets generated from the readings collected by different sensors. For such datasets, we require to design machine learning models that are catered not only to the format of the data but also to the application at hand.

Publications

Work in this chapter is part of following publications:

- Ashish Dandekar, Remmy A. M. Zen, and Stéphane Bressan. A comparative study of synthetic dataset generation techniques. In *Database and*

Expert Systems Applications- 29th International Conference, DEXA 2018, Regensburg, Germany, Proceedings, Part II, pages 387-395

- Ashish Dandekar, Remmy A. M. Zen, and Stéphane Bressan. Comparative evaluation of data generation methods. In *Deep Learning Security Workshop, Singapore, December 2017. (Poster)*

Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd.

Chapter 5

Generating travelling records of commuters

In this chapter, we synthetically generate travelling records of commuters that are consistent with the travelling patterns of real commuters in the world. We do so by finding communities of commuters based on their travelling patterns. We adopt and extend Latent Dirichlet Allocation (LDA) to find communities in the spatiotemporal data of commuters. We generate synthetic travelling records by performing a random walk on the commuter transportation graph that we build for every community of commuters.

5.1 Introduction

Urban planning, development and management authorities and stakeholders need to understand the mobility of urban dwellers in order to manage the sociological, economic and environmental issues created by the continuing growth of cities and urban population. The study of human mobility concerns with answering the question of where was a particular user(who) was at what time and for which purpose[GL15]. Such kind of data about urban dwellers is available for study via automated fare collection (AFC) system which smart cities employ for their public transportation networks. The system facilitates commuters with the cards equipped with electronic chips which track the timestamped origin and destination location of the commuters and accordingly calculates the fare for the journey. This sequence of tappings on a card of a particular commuter defines the mobility of the commuter in the city in terms of where the commuter

was at what time. A community of commuters is a group of users of a public transportation network who share similar mobility patterns as recorded by their automated fare collection cards.

In this chapter, we propose an approach based on generative models to find communities of commuters from the spatiotemporal information stored on automated fare collection cards. We consider timestamped visits of the commuters as observations and form communities based on the latent topics, spatial and temporal, found by the generative model. We show that such a method of community discovery is not only efficient but also effective compared to community detection techniques. We use the learned community structure to synthesise the commuter transportation graph for every community. We further propose a mechanism that uses random walk on the graph along with latent topics learned from LDA to generate trajectories for a synthetic individual. Experiments show that simulated trajectories conform to the underlying hidden community structure.

The rest of the chapter is organised as follows. In Section 5.2 describes the generative models to find communities based on the travelling patterns of commuters. We present the experiments and the evaluation in Section 5.3. Additionally, Section 5.3 describes the dataset and the data pre-processing steps. In Section 5.4, we present related work pertinent to community discovery in the public transportation network. We conclude the paper by discussing the work underway in Section 5.5.

5.2 Generative data synthesisers

5.2.1 Terminology

Although a single commuter may possess two or more cards that she can use interchangeably, we treat each unique card as a single **commuter**. Let $\mathcal{C}$ denotes the set of all commuters, i.e. a set of distinct card numbers in the dataset. A commuter can visit any location l from a finite set of locations $\mathcal{L}$ at any time t

Table 5.1: List of notations.

Symbol	Description
$\mathcal{C}$	Set of commuters
$\mathcal{L}$	Set of locations
$\mathcal{T}$	Set of time segments
K	Total number of topics
$\mathcal{M}_c$	Mobility of a commuter c
$\mathcal{M}_c^t$	Temporal Mobility of a commuter c
$\mathcal{M}_c^l$	Spatial Mobility of a commuter c
α	Dirichlet prior over topics
β	Dirichlet prior over locations
γ	Dirichlet prior over time-segments
θ_c	Topic distribution of a commuter c
ϕ_k	Location distribution of a topic k
ψ_k	Time distribution of a topic k
z	Latent topic
t	Timestamp
l	Location

from a set of time segments $\mathcal{T}$.

The list of spatio-temporal visits done by a commuter as recorded on the corresponding card is defined as **mobility** of the commuter. For instance mobility of a commuter c, denoted as $\mathcal{M}_c$, is a multiset $\{l - t | l \in \mathcal{L}, t \in \mathcal{T}\}$. Each element of $\mathcal{M}_c$ is called as a **visit** of a commuter c. Sometimes, we want to focus on either spatial or temporal mobility of a commuter instead of her spatio-temporal visits. We denote spatial mobility of a commuter c as $\mathcal{M}_c^l$ which is a multiset $\{l | l \in \mathcal{L}\}$. Similarly, temporal mobility of a commuter c is denoted as $\mathcal{M}_c^t$ which is a multiset $\{t | t \in \mathcal{T}\}$. Table 5.1 lists the notation which are used throughout the paper.

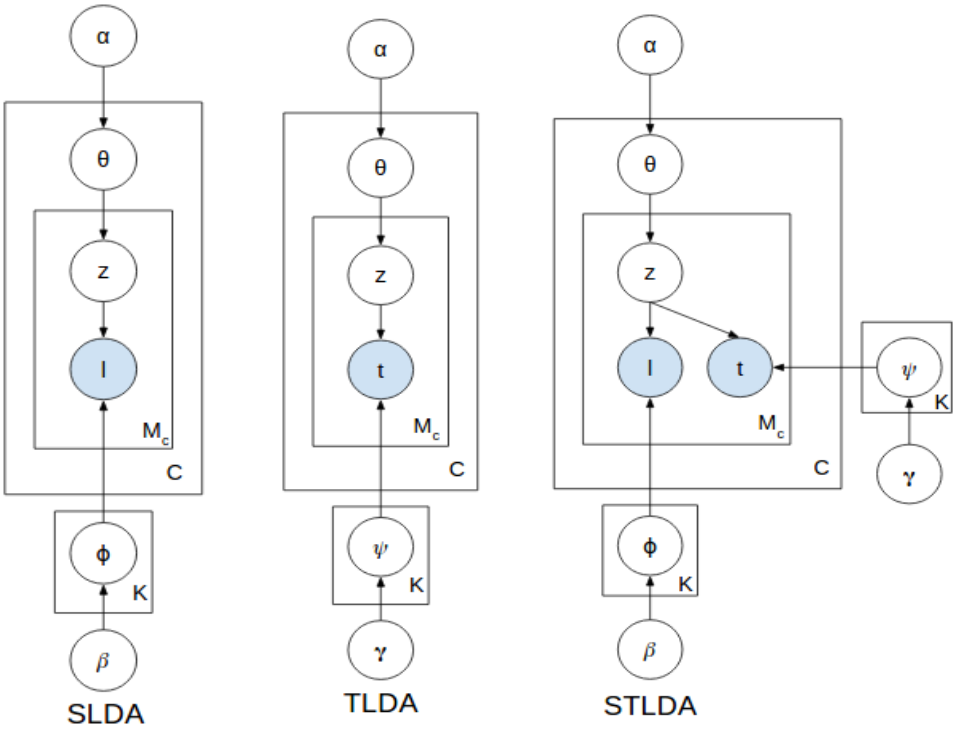


Figure 5.1: Plate diagrams of adaptations of LDA for spatiotemporal data.

5.2.2 Latent Dirichlet Allocation (LDA)

The probabilistic graphical model, LDA [BNJ03], finds latent topics in the text corpora. LDA assumes every document as a distribution over K topics, K being an input to the model, and each of the K topics as the distribution over words. This makes LDA a probabilistic clustering technique that considers overlapping clusters. The assumption of LDA that a document is a probability distribution of over the topics befits the problem at our hand. Intuitively, a commuter belongs to different social communities in the city. So, a commuter with his mobility can be seen as a distribution over K communities. Similarly, each community can be seen as distribution over the visits. Inspired by this analogy, we adopt and extend LDA to different generative models.

5.2.3 Spatiotemporal adaptations of LDA

SLDA and TLDA

In LDA, there is only one observed variable *viz.* word in the document. So if we consider document as either spatial or temporal visits of a commuter, we can directly adopt LDA to find latent topics. Considering, every commuter as a document, a bag of locations which a commuter visits, we find communities that share an overlap among the locations they visit. Therefore intuitively, this *spatial* adoption of LDA - SLDA, shown in the Figure 5.1, captures spatially cohesive topics. Similarly, if we consider every commuter as a document, a bag of timestamps at which a commuter reaches certain locations, we find communities that share an overlap among the periods of time at which commuters travel. This *temporal* adoption of LDA, TLDA, is also shown in Figure 5.1. Depending on the segmentation of time variable, topics found by TLDA carry different semantics. If we fold time on hourly basis everyday, then we find daily temporal patterns of commuters such as people travelling in the afternoon, working people who go to work in the morning and get back to home in the evening, etc. If we fold time on weekly basis, accordingly, we find weekly temporal patterns.

The generative process for SLDA for a commuter c is as follows:

1. For every topic k:
   (a) Sample $\phi_k \sim Dirichlet(\beta)$
2. For every commuter $c \in \mathcal{C}$:
   (a) Sample $\theta \sim Dirichlet(\alpha)$
   (b) For every visit v by c:
      i. Sample $z \sim Multinomial(\theta)$
      ii. Sample $l \sim Multinomial(\phi_z)$

In case of TLDA, we have similar generative process except that we sample

time $t \sim Mult(\psi_z)$ where ψ_z is the distribution over time unlike ϕ_z which is the distribution over locations.

STLDA

As the words are the only observables in the original LDA model, we can not directly use it to find the communities with spatial and temporal overlap. We extend the LDA model, to spatio-temporal LDA (STLDA), by assuming independent sampling distributions for both location and time for every visit of a commuter. Commuters go to a certain place at a certain time for a definite purpose. For instance, if location l_1 is office place of a commuter c then it is highly improbable that he visits it at night or after working hours. With this intuition, we assign a single topic to each spatiotemporal visit of a commuter which leads to the assignment of the same topic to both location and time of travel in the visit. Figure 5.1 shows the plate diagram for STLDA.

The generative process for STLDA for a commuter c is as follows:

1. For every topic k:
   (a) Sample $\phi_k \sim Dirichlet(\beta)$
   (b) Sample $\psi_k \sim Dirichlet(\gamma)$
2. For every commuter $c \in \mathcal{C}$:
   (a) Sample $\theta \sim Dirichlet(\alpha)$
   (b) For every visit v by c:
      i. Sample $z \sim Multinomial(\theta)$
      ii. Sample $l \sim Multinomial(\phi_z)$
      iii. Sample $t \sim Multinomial(\psi_z)$

Inference

For learning parameters of our models, we use Gibbs sampling based inference of LDA as presented in [GS04]. The backbone framework for Gibbs sampling for all of the proposed models remain the same except the update equation which changes for every model. Algorithm 5.1 shows Gibbs sampling based inference of STLDA. SLDA and TLDA follow the same algorithm except that we ignore timestamp and location in their sampling procedures respectively.

Update equations are given as follows:

$$
\begin{aligned}
P(z_i = k|\bar{z}_{\neg i}, \bar{l}) &\propto \frac{n^{(l)}_{kl,\neg i} + \beta_l}{\sum_{l=1}^{\mathcal{L}} n^{(l)}_{kl,\neg i} + \beta_l} \left(n^{(k)}_{c,\neg i} + \alpha_k\right) && \text{(SLDA)} \\
P(z_i = k|\bar{z}_{\neg i}, \bar{t}) &\propto \frac{n^{(t)}_{kt,\neg i} + \gamma_t}{\sum_{t=1}^{\mathcal{T}} n^{(t)}_{kt,\neg i} + \gamma_t} \left(n^{(k)}_{c,\neg i} + \alpha_k\right) && \text{(TLDA)} \\
P(z_i = k|\bar{z}_{\neg i}, \bar{l}, \bar{t}) &\propto \frac{n^{(l)}_{kl,\neg i} + \beta_l}{\sum_{l=1}^{\mathcal{L}} n^{(l)}_{kl,\neg i} + \beta_l} \frac{n^{(t)}_{kt,\neg i} + \gamma_t}{\sum_{t=1}^{\mathcal{T}} n^{(t)}_{kt,\neg i} + \gamma_t} \left(n^{(k)}_{c,\neg i} + \alpha_k\right) && \text{(STLDA)}
\end{aligned}
\tag{5.1}
$$

where $n^{(z)}_c$ denotes the number of times a visit with a topic z is observed for a commuter c, $n^{(l)}_{zl}$ denotes the number of times a topic z has been assigned to a location l and $n^{(t)}_{zt}$ denoted the number of times a topic z has been assigned to a time segment t. $\neg i$ denotes removal of i^{th} visit from the mobility. Parameters of the generative model are given by:

$$
\begin{aligned}
\theta_{c,k} &= \frac{n^k_c + \alpha_k}{\sum_{k=1}^{K} n^k_c + \alpha_k} \\
\phi_{k,l} &= \frac{n^k_{zl} + \beta_l}{\sum_{l=1}^{\mathcal{L}} n^k_{zl} + \beta_l} \\
\psi_{k,t} &= \frac{n^k_{zt} + \gamma_t}{\sum_{t=1}^{\mathcal{T}} n^k_{zt} + \gamma_t}
\end{aligned}
\tag{5.2}
$$

Algorithm 5.1: Gibbs sampling based inference for STLDA

Output : ϕ, θ, ψ

Zero all count variables $n_c^{(z)}, n_c, n_{zl}^{(l)}, n_{zl}, n_{zt}^{(t)}, n_{zt}$
for *all commuters* $c \in [1, 2, ..., \mathcal{C}]$ **do**
  for *all spatiotemporal words* $w \in [1, 2, ..., \mathcal{M}_c]$ **do**
    Sample topic $z \sim Mult(1/K)$
    Increment document-topic count $n_c^{(z)} + 1$
    Increment document-topic sum $n_c + 1$
    Increment topic-location count $n_{zl}^{(w_l)} + 1$
    Increment topic-location sum $n_{zl} + 1$
    Increment topic-time count $n_{zt}^{(w_t)} + 1$
    Increment topic-time $n_{zt} + 1$
  end
end
while *not converged* **do**
  for *all commuters* $c \in [1, 2, ..., \mathcal{C}]$ **do**
    for *all spatiotemporal words* $w \in [1, 2, ..., \mathcal{M}_c]$ **do**
      $k =$ currently assigned topic for w
      Decrement counts $n_m^{(k)}, n_m, n_{kl}^{(w_l)}, n_{kl}, n_{kt}^{(w_t)}, n_{kt}$ by 1
      $Z =$ sample new topic according to Equation 5.1
      Increment counts $n_m^{(z)}, n_m, n_{zl}^{(l)}, n_{zl}, n_{zt}^{(t)}, n_{zt}$ by 1
    end
  end
end
Compute ϕ, θ, ψ according to Equation 5.2
return ϕ, θ, ψ;

5.2.4 Generating travelling records of commuters

After running SLDA on the corpus of mobilities, we learn θ_c - topic distribution of each commuter c and ϕ_k - location distribution of each topic k. Let $\mathcal{C}_k = \{c | c \in \mathcal{C}, \theta_c^k \geq \tau\}$ denote a community of commuters with topic k, for some threshold $\tau \in [0, 1]$. It can be observed that $\mathcal{C} = \cup_{i=1}^{K} \mathcal{C}_i$. So, the communities of commuters

partition the set of commuters $\mathcal{C}$ in K partitions.

Using the partitions of commuters generated by SLDA, for each topic, we generate a weighted directed graph of locations using the mobilities of those users. For a given topic k, the graph comprises of locations as vertices and an edge is added between two vertices if a commuter from the partition $\mathcal{C}_k$ has commuted between these locations.

Formally, for a given topic k, we create a weighted directed graph $G_k(V_k, E_k)$, called as Commuter Transportation Graph, where $V_k = \mathcal{L}$ and $E_k = \{(v_1, v_2) \mid v_1, v_2 \in V, \exists\ c \in \mathcal{C}_k\ v_1, v_2 \in \mathcal{M}_c\}$. Each edge is weighted by number of commutations observed between two locations in the community $\mathcal{C}_k$.

We perform a random walk on the graph to generate travelling records of a synthetic user. For a given topic k, so as to pick a starting point and we then non-uniformly sample a location from the location distribution ϕ_k for topic k. Next location is chosen by following one of the outward edges of the previous node in the commuter transportation graph depending on the weight. If a node does not have any outgoing edge then the next location is again sampled from ϕ_k. We repeat the same procedure until we generate the desired number of visits for a synthetic individual. The procedure to generate n visits for a synthetic commuter in a community z is given in Algorithm 5.2.

5.3 Experimental evaluation

In this section we evaluate performance of the proposed method on real dataset - a snapshot of Singapore public transportation automated fare system. We present the experimental evaluation of the synthetically generated travelling records. We also conduct qualitative evaluation as well as comparative performance evaluation of the spatiotemporal models. We present these auxiliary experiments in Appendix A.

Algorithm 5.2: Algorithm for generating travelling records typical to a given community

Input : Community z, ϕ_z, G_z
Output : Trajectory l

$l \leftarrow []$
$l_0 \sim Multinomial(\phi_z)$
for $i = 1$ *to* $n - 1$ **do**
  if *(outdegree*$(l_{(i-1)} \in V_z) = 0$*)* **then**
    $l_i \sim Multinomial(\phi_z)$
  end
  else
    Choose l_i as one of the out-neighbors of $l_{(i-1)}$ with probability proportional to weight of the edge $(l_i, l_{(i-1)})$ in G_z
  end
end
return l

5.3.1 Dataset and experimental setup

Dataset. The dataset comprises of tappings made by commuters of public transportation system consisting of MRT/LRT stations and bus stops, identified by the cards' unique identifier. Please refer to Table 5.2 for the detailed dataset schema. Figure 5.2 shows statistics of commuters over 25 days of a month. One can identify recurrent pattern in the travels. We expect the majority of commuters to travel between home and work place during weekdays and between home and shopping or leisure centres during weekends. The pattern seems to break on 25^{th} which is a public holiday in Singapore.

We prepare two datasets comprising 41000 commuters over 4985 locations, The first dataset spans over weekdays in a typical week. The second dataset spans over one weekend. Time is considered at the granularity of an hour on every day. In this way, a document, after these pre-processing steps, is a bag of geospatial

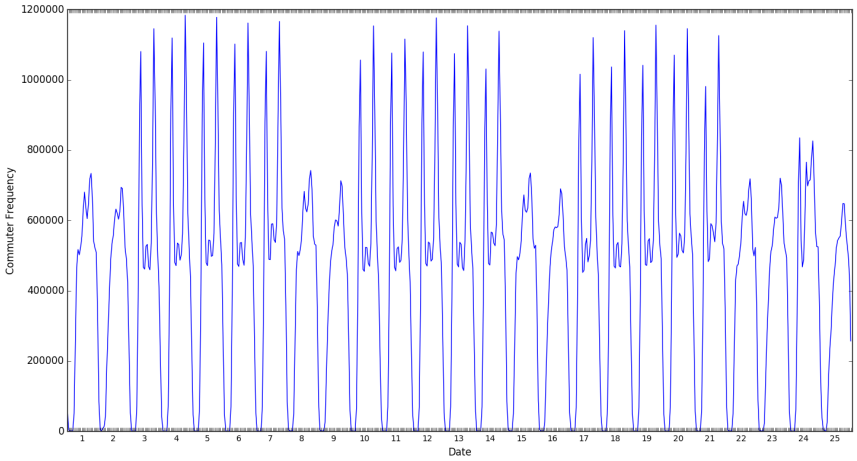


Figure 5.2: Statistics of commuters over a 25 consecutive days in the EZ-link card dataset.

timestamps. A typical word looks like - "56-20"- which means location 56 at time 20:00.

Experimental setup. All programs are run on a Linux machine with quad core 2.40GHz Intel® Core i7™processor and 8GB memory. Python® 2.7.6 is used as a scripting language. We have used the Java library, *jLDADMM*, developed by Dat et al. [Ngu15] for finding the latent topics using LDA[1] and modified the source to adapt to proposed extension of LDA.

We perform 2000 cycles of Gibbs sampling iteration to obtain the communities by LDA. Every commuter is assigned to the community with the maximum probability.

5.3.2 Results

Privacy risk. We want to evaluate risk of identity disclosure. Since the proposed generative models work on "Bag of Words" assumption, every travelling record is considered as multiset of locations. In order to compute overlap among the

[1] `http://jldadmm.sourceforge.net/`

Table 5.2: Schema for the EZ-link card dataset.

Field	Description
Card_Number_E	ID of the EZ-link card
Transport_Mode	Bus, MRT or LRT
Entry_Date	Date of the tap-in
Entry_Time	Time of the tap-in
Exit_Date	Date of the tap-out
Exit_Time	Time of the tap-out
Payment_Mode	Mode of the payment
Commuter_Category	Category of the card
Origin_Location_ID	Location ID of the tap-in
Destination_Location_ID	Location ID of the tap-out

generated travelling records and travelling records in the training dataset, we use Jaccard similarity [Mur12a] as the metric. Value of Jaccard similarity lies in between 0 and 1. Jaccard similarity of 0 denotes highest dissimilarity whereas value of 1 denotes perfect overlap. Therefore, smaller values of Jaccard similarity bear lower risk of identity disclosure.

We synthetically generate travelling records for 1000 commuters in every community. For every synthetically generated record, we compute Jaccard similarity with every travelling in the training dataset. Thus, we obtain a probability distribution over the Jaccard similarity for every topic. In Figure 5.3, we plot the probability distributions for a randomly chosen community. We observe that more than 80% of times synthetically generated travelling records have Jaccard similarity of smaller than 0.2. We make the same observation for rest of the communities. Thus, synthetically generated records using the proposed method bear lower risk of identity disclosure.

Utility. Location probability distributions, ϕ_is, capture the community structure of commuters. We use them to predict the community of synthetically generated

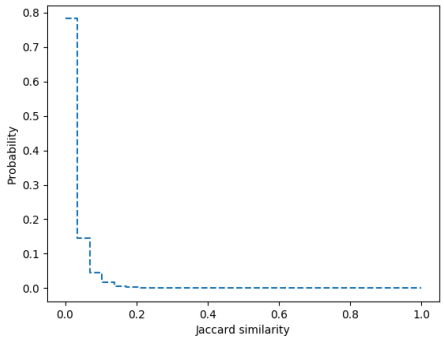


Figure 5.3: Privacy risk evaluation of synthetic travelling records.

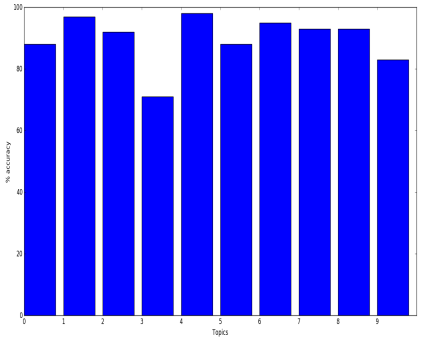


Figure 5.4: Utility evaluation of synthetic travelling records.

travelling records. We vectorise a generated travelling record as $|\mathcal{L}|$-dimensional vector, say $\bar{t}$. Inner product of $\bar{t}$ with ϕ_i denotes the probability of the travelling record to be in community i. So the community for travelling record $\bar{t}$ is predicted as follows:

$$k_{\bar{t}} = \underset{i \in [1..K]}{\operatorname{argmax}} \phi_i \cdot \bar{t}$$

Synthetically generated travelling records have a good utility if they are classified in a community that is same as the community provided as an input to the generation algorithm. We use classification accuracy as the metric of utility. Higher the classification accuracy, higher is the utility of the proposed data generation method.

We set threshold τ as 0.4 and classify commuters in the respective communities. We use these communities to construct the commuter transportation graph. For each community we generate 1000 trajectories each of length 10. We then compute accuracy of the proposed method for each community as the fraction of trajectories out of 1000 that are correctly predicted to be from the same community. Figure 5.4 shows the result of the experiment. We observe that the proposed method generate travelling records that are typical to the communities of commuters.

5.4 Related Work

Related work spans three different domains of research, namely *urban computing*, *Latent Dirichlet Allocation (LDA)* and *community detection*, as well as their mutual overlap.

Urban Computing [ZCWY14] is a process of acquisition, integration, and analysis of big and heterogeneous data generated by diverse sources in urban spaces, such as sensors, devices, vehicles, buildings, and humans, to tackle the major issues that cities face. It also helps understanding the latent trends and foresee the development of the city. Public transport data has been studied by authors for the betterment of the mobility of the citizens inside the cities. In [LC11a, LC11b, LQC12], London public transport has been widely studied for exploring travelling behaviours, minimising travelling time and finding communities of citizens. Ferris et al. [FWB10] develop an app *OneBusAway* to reduce the waiting time of commuters by providing real-time bus arrival information in King county, Washington. To identify tourists from the daily commuters of the public transportation services Xue et al. [XWC+14] devise a method and tested it on the data from Singapore. Montis et al. [DMCC13] find communities of commuters to help in demarcation of the sub-regions in Sardinia.

Blei et al. [BNJ03] propose **Latent Dirichlet Allocation** (LDA) - a soft clustering technique used for finding latent topics by intuitively capturing the co-occurrence of the words in the textual corpora. Till the date, it is a widely used technique for topic discovery. In [YZX12, HJE13], authors alter the original model to handle geospatial data. LDA has been used to find the group of places in cities using the check-in data from LBSN Foursquare users [LJJ12, JTC12, CVSG13]. They show that LDA has the ability to cluster the places based on the hidden user interests than their geographical proximity. In the GeoFolk model proposed by Sizov [Siz10], author handles the spatial aspect by considering longitude and latitude to be sampled independently from uniform distributions. Hu et al. [HJE13] model longitude and latitude as bivariate Gaussian distribution. Yin

et al. [YCH+15] extend joint spatial modelling by incorporating time of visit and textual description of Point of Interests into the model. Liu et al. [LEHC15] propose two models to explore dependence of time on spatial visits of users. Unlike these works, the present work focuses on the comparative study of generative models and graph based techniques of community discovery. Further, we use the generative process of the proposed model to produce synthetic data.

To the best of our knowledge, we are the first ones to propose generative model for detecting communities of commuters in the public transportation networks.

5.5 Discussion

In topic modelling community, LDA is still used as the state-of-the-art technique for the topic discovery. But morphing the data at hand in the form of documents and words, we have exploited the caveats of LDA finding the communities of commuters. We have showed that adaptations of LDA are efficient techniques for capturing the travelling patterns of the commuters. We also conducted a preliminary disclosure risk evaluation and showed that the generated travelling records do not bear high risk of identity disclosure.

The generative models in this work assume "Bag of Words" model. Due to the independence among observations within a travelling record, the proposed models do not capture the true nature of trajectories. We use Recurrent Neural Network based model, which accounts for the interdependence among different observations, to generate news headlines. News headlines do not bear any risk due to identity disclosure. Therefore, we defer this work to Appendix B in the interest of the central theme of the thesis. Use of Jaccard similarity to assess the risk of identity disclosure befits the "Bag of words" model. We require more sophisticated risk metrics such as edit distance or Dynamic Time Warping (DTW) to evaluate the disclosure risk for the trajectories of users.

Publications

Work in this chapter is part of the following publications.

- Ashish Dandekar, Stéphane Bressan, Talel Abdessalem, Huayu Wu, Wee Siong Ng. Detecting communities of commuters: graph based techniques versus generative models. In *On the Move to Meaningful Internet Systems: OTM 2016 Conferences, CoopIS 2016, Rhodes, Greece, Proceedings, pages 482-502*
- Ashish Dandekar, Stéphane Bressan, Talel Abdessalem, Huayu Wu, Wee Siong Ng. Trajectory simulation in communities of commuters. *In International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016, Jakarta, Indonesia, Proceedings, pages 39-42 (Invited paper)*

Acknowledgement

This research is funded by research grant R-252-000-622-114 by Singapore Ministry of Education Academic Research Fund (project 251RES1607 - "Janus: Effective, Efficient and Fair Algorithms for Spatio-temporal Crowdsourcing") and is a collaboration between the National University of Singapore, Télécom ParisTech and Singapore Agency for Science, Technology and Research.

Part II

On privacy risk of releasing models

In this part, we present our works on differentially private machine learning models. First work instantiates the use functional mechanism in the release of regularised linear regression model. Second work instantiates the use of functional perturbation to release various non-parametric models as a service. Works in this part are published at following venues:

- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Differential privacy for regularised linear regression. In *Database and Expert Systems Applications- 29th International Conference, DEXA 2018, Regensburg, Germany, Proceedings, Part II, pages 483-491*
- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Privacy as a service: Publishing data and models (Demo paper). In *24th International Conference on Database Systems for Advanced Applications, DASFAA 2019, Chiang Mai, Thailand.*
- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Evaluation of differentially private non-parametric machine learning models. In *Database and Expert Systems Applications- 30th International Conference, DEXA 2019, Linz, Austria (Under review)*

Chapter 6

Differential privacy for regularised linear regression

In this chapter, we present an ϵ-differentially private functional mechanism for the variants of regularised linear regression. We empirically and comparatively analyse its effectiveness. We empirically show that an ϵ-differentially private functional mechanism causes more error than the non-private linear regression models whereas their performances are comparable. We also discuss caveats in the functional mechanism, such as non-convexity of the noisy loss function, which causes instability in the results of differentially private linear regression models

6.1 Introduction

Recent attacks on machine learning models increase the concern for the privacy of users and promotes a need to protect it [GP17]. Different attack models, such as the membership inference attack [SSSS17] and the white-box and the black-box attacks [SRS17] are designed and studied by researchers to explore and to expose the vulnerability of machine learning models. Leveraging the knowledge of such attacks, a malevolent data analyst can breach the privacy of a machine learning model by inferring the values of some attributes of the data points that are used to train the model [TZJ+16]. Specifically under the Machine Learning as a Service (MLaaS) model [RGC15] the user cannot control the data leak caused by the machine learning model catered by the service provider [SRS17]. Therefore,

there is a need for privacy guarantees for machine learning models, especially when they are trained on the data that contains highly sensitive attributes.

Dwork et al. [DR+14] proposed a probabilistic framework, called *differential privacy*, to quantify privacy (Definition 1). Differential privacy provides a framework to design privacy inducing mechanisms [DMNS06a, ZZX+12] that introduce noise at different stages of a machine learning model, for instance in the output of the model [DMNS06a] or in the input of the model [Lei11], to make it as differentially private as required.

In this chapter, we study differential privacy inducing mechanisms for *linear regression* models under the release of the model itself. Linear regression models constitute a family of widely used machine learning models in such eclectic domains as econometrics [Int78], medicine [Aal93, SMT09] and policy making [CV05]. Linear regression is used for the task of predicting an attribute, called as the *response variable*, of a dataset given the rest of the attributes, called the *predictor variables*, of the same dataset. The linear regression models assume the response variable is a linear combination of the predictor variables. This is called the linear function hypothesis and is the basis of any general linear model [Mur12a]. Any linear regression algorithm finds the parameters of the linear combination for a given *training dataset*. The optimal set of parameters minimizes a *loss function*, such as root mean square error, for the *training dataset* (Section 6.3.1). Following the *training step*, these models are used for predicting the unknown response variable for the known predictor variables of the *testing dataset*. Linear regression models also assume that the predictor variables are statistically independent to each other [Mur12a]. In contrary, real world datasets often contain attributes that are correlated to each other. Such datasets violate the independence assumption, and yield unstable and overfitted solutions of linear regression. regularisation terms are added to the loss function of linear regression to overcome these problems (Section 6.3.2). Regularisation terms are weighted function of the parameters of the linear regression. *LASSO* [Tib96], *ridge* [HK70], and *elastic net* [ZH05] are three variants of regularised linear regression. LASSO,

ridge and elastic net add regularisation terms proportional to L_1 norm, L_2 norm and a convex combination of L_1 and L_2 norms respectively.

Differential privacy inducing mechanisms, such as the *functional mechanism* [ZZX+12], are studied and developed for the linear regression models. The functional mechanism [ZZX+12] adds a calculated amount of noise in the loss function of the linear regression model (Section 6.4). This mechanism provides a formal privacy guarantee for the linear regression model trained using the noisy loss function. In order to establish the privacy guarantee it leverages the probabilistic framework of ϵ-differential privacy [DR+14].

In this chapter, we empirically and comparatively evaluate (Section 6.5) the functional mechanism for linear regression, and three of its regularised variants, namely, LASSO, ridge, and elastic net. We comparatively evaluate the performance of these four mechanisms and their differentially private variants on two datasets with different correlations and sparsity. We observe that the functional mechanism applied to the regularised linear regression yields similar performance results and that the private linear regression models perform worse than the non-private linear regression models. We compare the effectiveness of the functional mechanism with an *input perturbation mechanism* [Lei11]. For a given privacy level, ϵ, we empirically show that the functional mechanism is more effective than [Lei11]. We extend the analysis in [ZZX+12] to empirically study the robustness of the functional mechanism. The key observation in our experiments is that all the private linear regression models are unstable. Our analysis (Section 6.5.3) shows that the reason for such an instability is inherent to the functional mechanism. The functional mechanism does not necessarily preserve the convexity properties of the loss function after the addition of noise. This potential non-convexity causes instability in the optimisation. In reference to these experimental evidences, we conclude by puting forth (Section 6.6) the need of designing a differentially private mechanism that produces a convex noisy loss function in order to provide both stable and private output for linear regression models.

6.2 Related work

Linear regression [Mur12a] is a fundamental yet a widely used [Aal93, CV05, Int78, SMT09] machine learning model. Variants of linear regression, Ridge [HK70] and LASSO [Tib96], are used to reduce correlation in the data features and to avoid overfitting. Elastic net [ZH05] regression uses convex combination of regularisation terms that are used in Ridge and LASSO. For a detailed presentation and discussion of regularisation and regression analysis, interested readers can refer to [Mur12a].

Differential Privacy [DR$^+$14] is a probabilistic framework that quantifies the privacy of a randomised function or algorithm. Existing deterministic machine learning models can be randomised by introducing calibrated random noise. The resultant randomised *mechanism* can be shown to satisfy constraints of differential privacy. Dwork et. al. propose the Laplace mechanism [DMNS06a], which perturbs the output of a machine learning model by explicitly adding scaled random noise from the Laplace distribution. The Gaussian mechanism [DR$^+$14] and the K-norm mechanism [HT10] are differentially private mechanisms that are also based on the idea of output perturbation with noise from different distributions. Lei [Lei11] proposes differentially private M-estimators, which perturbs the histogram of input data using a scaled noise and further uses the noisy histogram to train the models. Zhang et. al. [ZZX$^+$12] propose a differentially private *functional mechanism* that adds a properly scaled Laplace noise to the coefficients of loss function in the polynomial basis. Hall et.al. [HRW13a] also propose a differentially private *functional mechanism* that adds a properly scaled noise drawn from the Gaussian process to the coefficients of loss function in the kernel basis.

Zhang et. al. instantiate their functional mechanism on linear regression and logistic regression. In order to alleviate the non-convexity caused in the loss function due to addition of random noise, they use ridge regularised linear and logistic regressions. Yu et. al. [YRUF14] achieve differential privacy in the elastic

net logistic regression by controlling the coefficient of regularisation term. The regularisation term in their proposal is inversely proportional to the number of datapoints. It causes reduction in regularisation as the number of datapoints increases. Therefore, their proposed mechanism is not applicable for large datasets. Talwar et. al. [TTZ15] propose a differentially private variant of Frank-Wolfie optimisation algorithm to perform LASSO regression. This method adds noise in the optimisation algorithm instead of adding it to the objective function.

In this work, we empirically evaluate the effectiveness of using the functional mechanism [ZZX+12] for linear regression and regularised regression with ridge, LASSO and elastic net regulariser. Unlike the output perturbation mechanisms [DMNS06a, DR+14, HT10], which study differential privacy under the release of the outputs of the models, we evaluate the differential privacy of linear regression under the release of parameters of the model [Lei11, ZZX+12].

6.3 Background

Let, $\mathcal{D}$ denotes a universe of d-dimensional real-valued datapoints and the corresponding real-valued responses. An element from this universe can be represented by a pair $D = (X, y)$ where $X \in \mathbb{R}^{n \times d}$ denotes a data matrix whose each row corresponds to a d-dimensional datapoint, x_i and $y \in \mathbb{R}^n$ denotes the response vector in one-to-one correspondence to n datapoints. We use $\|\cdot\|_p$ to represent L_p norm of a vector. A dataset D is split into two disjoint sets of datapoints: training dataset, T and validation dataset V.

6.3.1 Linear regression

Regression is a kind of predictive machine learning model that learns function $f : \mathbb{R}^d \to \mathbb{R}$ parameterised by a set of parameters of θ. Let, $l_\theta : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ denotes loss function that quantifies the loss in prediction for a given datapoint x_i and the corresponding response y_i due to parameter θ. Learning comprises of

the training step wherein one estimates parameters θ which minimise the loss over the training dataset. Mathematically,

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(x_i,y_i)\in T} l_\theta(x_i, y_i) \tag{6.1}$$

Linear regression uses a linear hypothesis to map predictor variable to the corresponding response. In matrix notation, linear regression is parameterised by $\theta \in \mathbb{R}^d$ such that $X\theta = y$. In order to find the optimal value of θ, training step in linear regression minimises *mean squared loss* over the training data as defined in Equation 6.2.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(x_i,y_i)\in T} (x_i^t\theta - y_i)^2 = \underset{\theta}{\operatorname{argmin}} (X\theta - y)^2 \tag{6.2}$$

Linear regression problem can be analytically solved by differentiating Equation 6.2 and equating it to zero. The analytical solution is[1]:

$$\theta^* = (X^tX)^{-1}X^ty \tag{6.3}$$

6.3.2 Regularised linear regression

Properties of the coefficient of quadratic term in Equation 6.3, *viz.* X^tX, determine the convexity of the optimisation problem. If the matrix X^tX does not possess a full rank due to correlation among different data dimensions, it does not remain a positive-definite matrix and induces non-convexity in the corresponding optimisation problem. Non-convex optimisation problems do not guarantee a unique global minimum.

In order to alleviate these irregularities, a constant term is added to the diagonal elements of the matrix X^tX which forcefully makes it a full rank matrix. Under such a transformation, the linear regression optimisation problem transforms into

[1] X^t denotes the transpose of X.

the optimisation problem for *ridge regression* [HK70] as stated in Equation 6.4. Closed form solution for the ridge regression is presented in Equation 6.5.

$$\theta^* = \underset{\theta}{\operatorname{argmin}}(X\theta - Y)^2 + \lambda\|\theta\|_2^2 \tag{6.4}$$

$$\theta^* = (X^t X + \lambda \mathbb{I}_d)^{-1} X^t y \tag{6.5}$$

Addition of the regularisation term that is proportional to L_1 norm of parameter yields a variant of linear regression called as *LASSO regression* [Tib96]. LASSO regression is used for obtaining *sparse* solution for the regression problem. Since a sparse solution contains many zeros, only the features that have non-zero coefficients contribute towards generating the response. Due to this property, LASSO regression is also used for variable selection. Unlike linear regression and ridge regression, we do not have a closed form solution for LASSO regression due to piece-wise differentiability of L_1 norm.

6.4 Functional mechanism for regularised linear regression

Unlike output perturbation mechanisms, Functional mechanism [ZZX+12] introduces random noise in the loss function of a machine learning algorithm. optimisation of such a noisy loss function leads to the parameters which are different than true optimal parameters. In this way, we indirectly get noisy outputs from the machine learning model without explicitly adding noise to the outputs themselves. In this section, we elucidate the details related to the functional mechanism.

6.4.1 Sensitivity calculation

By Stone-Weierstrass theorem, any infinitely differentiable function can be expanded in terms of its polynomial basis for some $J \in [0, \infty]$ as given in Equa-

tion 6.6 where Φ_j denotes the set of polynomials with degree j and $\lambda_{t\phi}$ denote respective coefficients.

$$f(t,\omega) = \sum_{j=0}^{J} \sum_{\phi \in \Phi_j} \lambda_{t\phi} \phi(\omega) \tag{6.6}$$

For a given machine learning model, the loss function l_θ can be expanded in the polynomial basis as a function of parameter θ as given in Equation 6.7 where $t = (x, y)$ denotes a data-point in training dataset T.

$$l(T,\theta) = \sum_{t \in T} \sum_{j=0}^{J} \sum_{\phi \in \Phi_j} \lambda_{t\phi} \phi(\theta) \tag{6.7}$$

Lemma 1. *[ZZX+12] L_1 sensitivity of the loss function of a machine learning model is given by:*

$$\begin{aligned} \Delta_l &= \max_{D, D' \in \mathcal{D}, D \sim D'} \| l(D,\theta) - l(D'\theta) \|_1 \\ &= 2 \max_t \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \| \lambda_{t\phi} \|_1 \end{aligned}$$

where $t = (x, y)$ denotes a data-point in D or D'.

6.4.2 Functional mechanism

Using Lemma 1 and the Laplace mechanism, Zhang et. al. [ZZX+12] devise an algorithm that adds noise to the loss function. For the sake of completeness and notation consistency, we present the algorithm in Algorithm 6.1.

Theorem 2. *Algorithm 6.1 is satisfies ϵ-differential privacy.*

Proof. Proof is available in [ZZX+12]. □

6.4.3 Case: Elastic net regression

Now we apply the idea of functional mechanism to a specific case of machine learning model, namely linear regression. We know that linear regression models

Algorithm 6.1: Functional Mechanism [ZZX+12]

Input : Training dataset T, Loss function l_θ, Privacy level ϵ
Output : Parameters θ^*

$\Delta_l = 2 \max_t \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \|\lambda_{t\phi}\|_1$
for $j \in [0, 1, ..., J]$ **do**
  for $\phi \in \Phi_j$ **do**
    $\lambda_\phi \leftarrow \sum_{t \in T} \lambda_{t\phi} + \text{Lap}(\frac{\Delta_l}{\epsilon})$
  end
end
Let, the noisy loss function $l'(T, \theta) \leftarrow \sum_{j=0}^{J} \sum_{\phi \in \Phi_j} \lambda_\phi \phi(\theta)$
Compute $\theta^* = \text{argmin}_\theta \, l'(T, \theta)$
return θ^*

use *squared error* as the loss function, which is minimised to find the parameters. We expand the loss function, stated in Equation 6.2, as a polynomial in the parameters, θ, for a given data matrix X and the corresponding response vector y.

$$l(T, \theta) = \theta^t (X^t X)\theta - 2\theta^t (X^t y) + y^T y \tag{6.8}$$

Elastic net regression [ZH05] adds the regularisation term that is a convex combination of L_1 regularisation term and L_2 regularisation term. The optimisation problem for elastic net regression with functional mechanism is given in Equation 6.9. $l'(T, \theta)$ denotes the noisy loss function obtained by applying Algorithm 6.1 on the loss function for linear regression, as stated in Equation 6.8.

$$\theta^* = \underset{\theta}{\text{argmin}} \, l'(T, \theta) + \lambda(\alpha \|\theta\|_2^2 + (1 - \alpha)\|\theta\|_1) \tag{6.9}$$

If we put $\alpha = 1$ in the Equation 6.9, the objective function reduces to the objective function of functional mechanism with ridge regression. If we put $\alpha = 0$ in the Equation 6.9, the objective function reduces to the objective function of

functional mechanism LASSO regression. Therefore, elastic net regularisation stands as a bridge between these two variants.

As stated in the Section 6.3, a regularised linear regression incorporates a regularisation term, which is a term proportional to norm of the parameters, in the objective function. The sensitivity of a regularised objective function remains same as the sensitivity of an non-regularised objective function due to independence of the regularisation term on the dataset. Therefore, calculations in Algorithm 6.1 do not change under the regularisation. We use Lemma 1 to calculate sensitivity of the loss function stated in Equation 6.8.

Lemma 2. *Assuming that all features of data-points are normalised such that each of the feature value lies in* $[-1, 1]$*,* L_1 *sensitivity of the loss function in Equation 6.8 is given by:*

$$\Delta_l = 2(d^2 + 2d + 1)$$

6.5 Empirical performance evaluation

We comparatively and empirically evaluate functional mechanism for regularised linear regressions: namely ridge, LASSO, and elastic net. We present the result analysis in this section.

6.5.1 Datasets and experimental setup

Census dataset. We conduct experiments on a microdata sample of US Census in 2000 provided by IPUMS International [IPU09]. The census dataset consists of 1% sample of the original census data. It spans over 1.23 million households with records of 2.8 million people. It has several attributes of which not every single attribute is reported by all of the people. In order to avoid the discrepancies in the data, we consider $316,276$ records of the heads of households in our dataset. Each record has 9 attributes, namely, *Age, Gender, Race, Marital Status, Family Size, Education, Employment Status, House type, Income.* Regression analysis is

performed using *Income* as the response variable and the rest of the attributes as predictor variables.

Wine quality testing dataset. Correlation among different attributes of a dataset adds training bias in the effectiveness of the model. In order to derive unbiased inferences from the results, we use an another dataset: wine quality testing dataset [CCA+09]. The dataset comprises 4898 records of white wine samples wherein each record has 12 attributes, namely, *Fixed Acidity, Volatile Acidity, Citric Acid, Residual Sugar, Chlorides, Free Sulfur Dioxide, Total Sulfur Dioxide, Density, pH, Sulphates, Alcohol, and Quality.* Regression analysis is performed using *Quality* as the response variable and the rest of the attributes as predictor variables.

Experimental setup. All programs are run on Linux machine with 12-core 3.60GHz Intel® Core i7™processor with 64GB memory. Python® 2.7.6 is used as the scripting language. We use SCS [OCPB16] solver, that is available in CVXPY [DB16] package, to find the solution for piecewise differentiable objective functions of LASSO and elastic net regression.

We report the results as the aggregates over 50 experimental runs. For every experimental run, we randomly hold out 20% of the data for testing and use the rest 80% of the data for training regression models. We normalize each of these features such that their values lie in $[-1, 1]$. We use *root mean squared error (RMSE)* as the metric to comparatively evaluate effectiveness. We report RMSE for a trained model calculated on the validation dataset as given in the Equation 6.10. For given value of ϵ, the model with smallest value of RMSE is the most effective model.

$$\text{RMSE}_\theta = \sqrt{\frac{\sum_{(x_i,y_i)\in V}(x_i^t\theta - y_i)^2}{|V|}} \tag{6.10}$$

We comparatively evaluate eight regression problems: linear regression (LR), ridge regression (RG), LASSO regression (LS), elastic net regression (EN), and

their private versions, the ones that are obtained applying functional mechanism to each of these four. For brevity, we call the regression model obtained using the functional mechanism as *functional regression*. For every regularised regression model, we set the regularisation coefficient, λ, by performing cross-validation on the respective non-private versions of regularised regression and choosing the value that results in smallest in testing error. We use the same regularisation coefficient for the private version.

6.5.2 Results

Figure 6.1 shows the boxplot of functional elastic net regression for different values of ϵ's for the census dataset. We note the presence of a large number of outliers in the result. We observe similar results for the rest of the functional regressions. Use of *mean* as an aggregate affects the analysis of the results due to sensitivity of mean to outliers in the data. In order to avoid this bias due to the outliers, we choose *median* as an aggregate to report the results.

Comparative evaluation of functional mechanism for regularised linear regression. Figure 6.2 and Figure 6.3 show the comparative evaluation of the variants of regularised linear regression for the wine quality dataset and the census dataset respectively. In the plot, solid line represents median over 50 experimental runs and the shaded region covers RMSE values that lie between 20^{th} and 80^{th} percentile. Smaller values of ϵ's induce higher noise in the function, which in turn results in higher privacy. Therefore, we observe higher RMSE for smaller values of ϵ's. As the value of ϵ increases, the effectiveness of the functional regression approaches the the effectiveness of the non-private counterpart.

In order to understand the magnitude of instability, we plot the variance of different regressions in Figure 6.5. We observe that the non-private models show vary small amount of variance. As the value of ϵ increases, the amount of noise that is added in the loss function reduces. Reduction in the noise results in lower RMSEs and lower variance in RMSE. We do not observe the same trend in

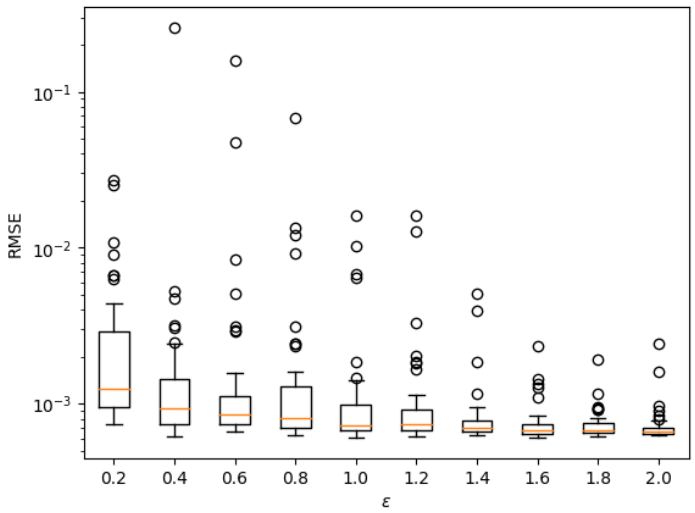


Figure 6.1: Boxplot of RMSE of elastic net ridge regression with functional mechanism for varying ϵ on the census dataset.

functional linear regression. In the absence of regularisation, noisy loss function of linear regression becomes highly non-convex. Therefore, the results show high instability.

Comparative evaluation of functional mechanism and input perturbation mechanism. We, also, compare effectiveness of the functional mechanism with an *input perturbation mechanism*, differentially private M-estimators (DPME) [Lei11]. Discretisation of large number of variables leads to a large discrete space that causes prohibitive computation cost. Due to concentration of data around subsets of features, a large discrete space also leads to sparse histograms [Lei11]. In order to alleviate the sparsity, we follow [Lei11] and evaluate the performance on a simpler regression model. We show the comparative study on the census dataset where we predict *Income* of a person using *Age, Gender, Race* and *Education Status* as the predictors [2]. The results are presented in Figure 6.5b. Solid lines represent the *mean* RMSE over 50 experimental runs. For a given value of ϵ, we observe that the functional mechanism provides lower RMSE for all regularised linear regressions. M-estimator is a robust statistic [HRRS11].

[2]The wine quality testing dataset comprises of 4898 data-points over 12 continuous attributes. Due to the small size of the dataset, it generates highly sparse histograms in the large space. Therefore, for the sake of interpretability, we present the results for the census dataset.

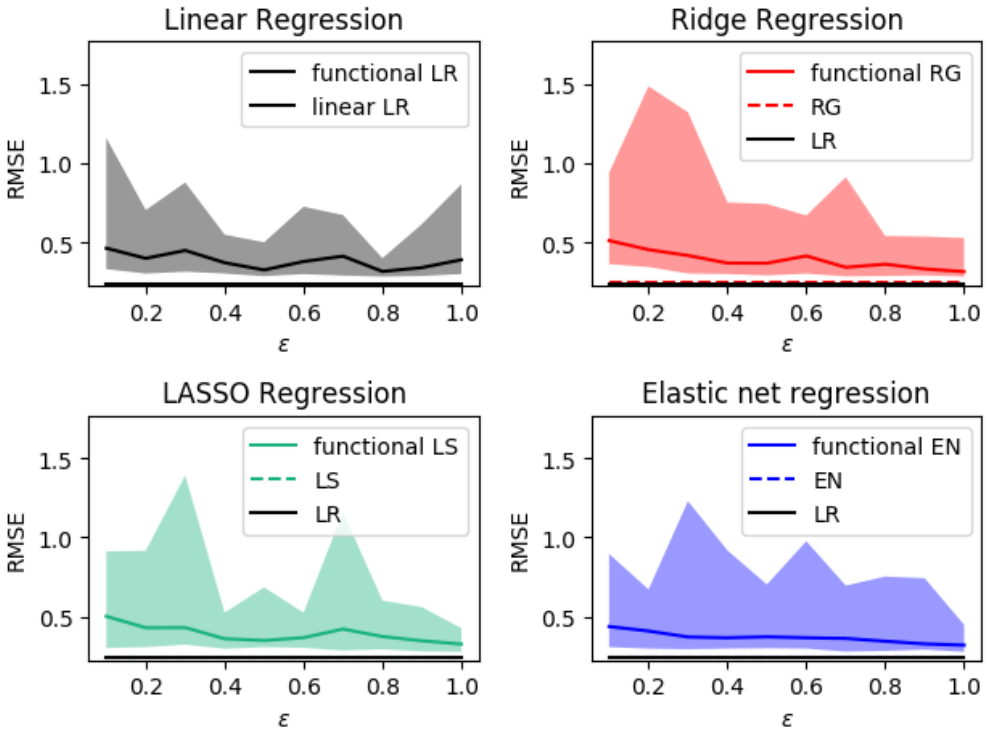


Figure 6.2: Comparative evaluation of regularised regressions onr the wine quality-testing dataset.

Therefore, we observe the stability in the performance of DPME as compared to the results presented in Figure 6.3 and Figure 6.2.

Comparative evaluation of functional mechanism and objective perturbation mechanism. We compare the effectiveness of the functional mechanism with objective perturbation mechanism by Kifer et al. [KST12]. Unlike the functional mechanism, which perturbs Taylor coefficients of the loss function, the objective perturbation mechanism adds an explicit random noise term in the objective function. Figure 6.5a shows the comparative evaluation of differentially private ridge regression on the wine quality testing dataset. Solid lines represent mean over 50 experimental runs whereas the shaded region shows errors that are one standard deviation away from the means. We observe that the functional mechanism has better utility than the objective perturbation for a specified value of the privacy level. Despite its utility, the functional mechanism shows a higher variance in the RMSE than the objective perturbation mechanism. Kifer et al.

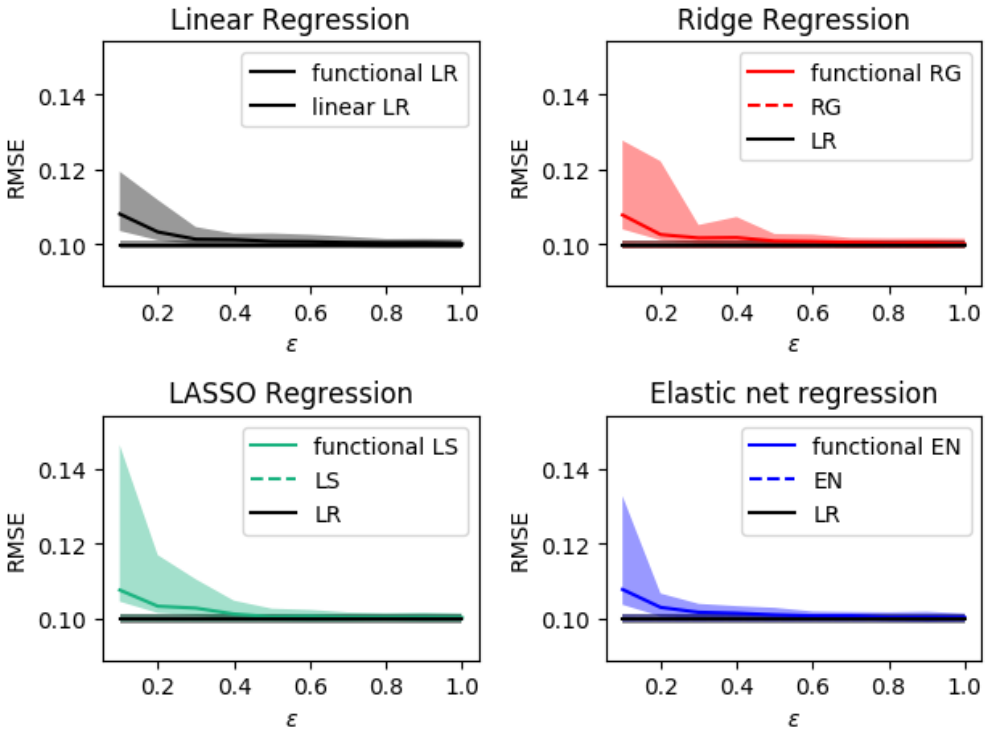
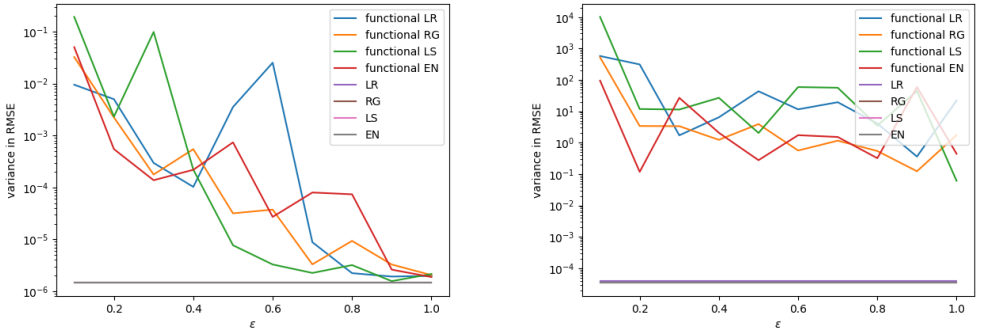


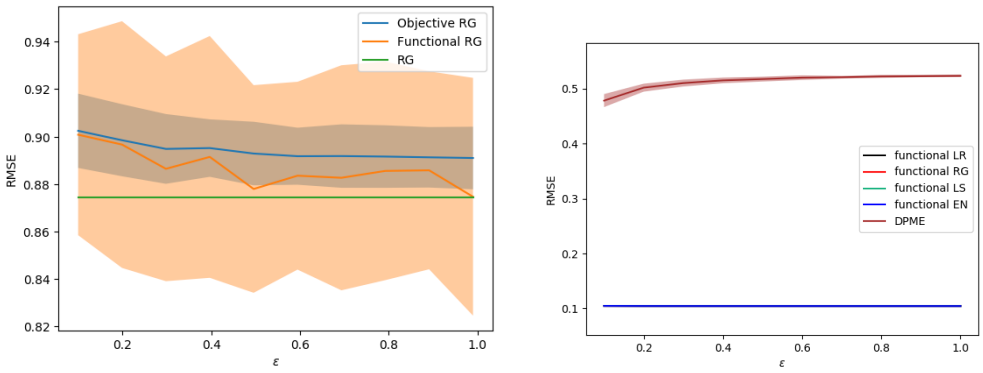
Figure 6.3: Comparative evaluation of regularised regressions on the census dataset.



(a) Census Dataset **(b)** Wine quality testing dataset

Figure 6.4: Variance in RMSE for varying values of privacy level ϵ's for regularised regressions.

introduce convexity preserving additive noise terms in the objective function of a machine learning model whereas the functional mechanism does not ensure convexity of the loss function after the perturbation. Therefore, the functional



(a) Objective perturbation mechanism **(b)** Input perturbation mechanism

Figure 6.5: Variance in RMSE for varying values of ϵ's for regularised regressions.

mechanism may result in a local optimal solution. Such local optimal solutions result in the higher variance in RMSE that is observed in case of the functional mechanism.

6.5.3 Stability analysis

One observation that is common in all the empirical evaluations is the instability in the results. In order to achieve privacy, we, indeed, need to introduce noise in the output so that it does not reveal the true output of the model. But excessive noise deteriorates the effectiveness, and hence the utility, of the machine learning model. We find that the reasons for this instability are rooted in the functional mechanism itself.

Symmetric noise. The coefficient of the quadratic term in Equation 6.8, X^tX, is a symmetric matrix. It loses the symmetric property after adding random noise from the Laplace distribution. A standard way to make a given matrix A symmetric is to use $(A + A^t) * 0.5$. This way of symmetrisation of noisy X^tX indirectly incurs addition two Laplace random variables. Addition of two Laplace random variable does not follow Laplace distribution. Therefore, in order to maintain the integrity of the functional mechanism, we can not make X^tX

symmetric in the conventional way.

Convexity of the objective function. Linear regression works on the assumption that the features of data are independent of each other. Independence among the features makes X^tX a positive definite matrix. Positive definite matrices make the optimisation convex and guarantees optimality of the solution. Noisy loss function fails to guarantee convexity of the objective problem, and hence the optimality of the solution. A similar observation is made by Lei [Lei11] while perturbing the histograms of input data by adding the calibrated noise. In order to make the objective function convex, Zhang et. al. [ZZX+12] calculate the spectral decomposition of X^tX and consider the projection of parameters onto the eigenspace spanned by eigenvectors with positive eigenvalues. They do not provide any analytical justification which guarantees differential privacy after pruning the non-positive eigenspace.

Differential privacy of the optimisation algorithm. Functional mechanism proves that the loss function generated by any two neighbouring datasets satisfies differential privacy. Composition of a differentially private function with a *deterministic* function, called as *post-processing* [DR+14], remains differentially private. An optimisation problem solver calculates an approximate solution when the objective function is not convex. Therefore, differential privacy of a loss function is not preserved by the optimisation algorithm itself.

6.6 Discussion

In this work, we use functional mechanism that presents differentially private linear regression models. We empirically and comparatively evaluate the effectiveness of the private and non-private versions of linear regression, LASSO, ridge and elastic net on the census and the wine quality datasets. For a given privacy level, ϵ, we observe that the functional mechanism is more effective than DPME [Lei11] for the regularised linear regression. We extend the analysis in [ZZX+12] to

empirically study the robustness of the functional mechanism. We invariably observe that the private versions are less effective than their non-private counterparts. The key observation from these experiments is that all these private regularised regression methods are equally unstable, and private linear regression is comparatively more unstable.

The reason for the instability is rooted in the loss of convexity of the perturbed objective function. There are pieces of work that shed light on the convexity issue. Nozari et al. [NTC18] derive constraint on the parameters of Laplace distribution so that the perturbed function remains bounded. They propose post-processing steps that ensures convexity of the objective function. We plan to incorporate such conditions in our future works.

Publications

Work in this chapter is part of the following publications.

- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Differential privacy for regularised linear regression. In *Database and Expert Systems Applications- 29^{th} International Conference, DEXA 2018, Regensburg, Germany, Proceedings, Part II, pages 483-491*
- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Privacy as a service: Publishing data and models (Demo paper). In *24^{th} International Conference on Database Systems for Advanced Applications, DASFAA 2019, Chiang Mai, Thailand.*

Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd.

Chapter 7

Evaluation of differentially private non-parametric machine learning models

In this chapter, we study how to provide differential privacy guarantees for non-parametric models. This cannot be achieved by perturbation of the output but requires perturbation of the model functions. We show how to apply the perturbation to the model functions of histogram, kernel density estimator, kernel SVM and Gaussian process regression in order to provide (ϵ, δ)-differential privacy. We evaluate the trade-off between the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms on benchmarks and real-world datasets.

7.1 Introduction

Organisations are amassing data at an unprecedented scale and granularity. They release either the raw data or the machine learning models that are trained on the raw data. All machine learning models do not fit the choice of releasing only the models. A parametric machine learning model [Mur12a] assumes a parametric model function[1] that maps a new data to the corresponding output. A non-parametric machine learning model [Mur12a] does not assume a parametric model function but calculates some form of correlation between a new data and the

[1]Model function refers to the mapping from input to output that is learned by the corresponding machine learning algorithm.

training data to compute the corresponding output. For instance, kernel density estimation [Par62] computes the probability density of a new data by assimilating the probabilities of the new data originating from the probability distributions centred at every data-point in the training data. Kernel SVM [BGV92] and Gaussian process regression [Ras04] compute kernel Gram matrix between the new data and the training data. Thus, while releasing parametric machine learning models requires the release of the parameters of the model function, releasing non-parametric machine learning models requires the release of the training dataset along with the parameters. An alternative to the release of the training dataset is utilising non-parametric models as a service. While using a non-parametric model as a service, user would send a new data to the model to obtain the output of estimation, prediction, or classification.

Publication of raw data without any processing leads to a violation of the privacy of users [gdp16]. Not only raw data but also publication of a 'non-private' machine learning model as a service leads to a violation of the privacy of users. For instance, experiments in [SSSS17] show that models created using popular machine-learning-as-a-service platforms, such as Google and Amazon, can leak identity of a data-point in the training dataset with accuracy up to 94%. In order to reduce the risk of breach of privacy, we need to take preemptive steps and provide quantifiable privacy guarantees for the released machine learning model. Differential privacy [DR$^+$14] is one of such privacy definitions to quantify the privacy guarantees.

We study how to provide differential privacy guarantees for non-parametric models as a service. This cannot be achieved by the output perturbation using typical differential privacy mechanisms, such as Laplace mechanism and Gaussian mechanism [DR$^+$14]. Due to sequential composition [DR$^+$14] of differential privacy, the privacy guarantee of a mechanism linearly degrades with the number of times the noise is added from a given noise distribution. Output perturbation requires addition of calibrated noise in the output for every new data input. Therefore, it suffers from the degradation of privacy guarantee. When machine

learning is provided as a service one can not limit the number of queries. Once the noise is added to a model function, further evaluations are performed on the noisy model. We adopt the functional perturbation proposed in [HRW13a] in order to provide a robust privacy guarantee. Functional perturbation adds a scaled noise sampled from a Gaussian process to the function. [HRW13a] proves that an appropriate calibration of this mechanism provides (ϵ, δ)-differential privacy. We show how to calibrate the functional perturbation for histogram, kernel density estimator, kernel SVM, and Gaussian process regression. We evaluate the trade-off between the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms on benchmarks and US census dataset.

Our contribution is twofold. Firstly, we show that functional perturbation is a viable alternative to output perturbation to provide privacy guarantees for machine learning models as a service. We also hypothesise as well as experimentally validate that output perturbation is less effective than functional perturbation for a given privacy level and a given test set. Additionally, output perturbation is not directly applicable for machine learning models with categorical outputs, such as classification, where functional perturbation operates naturally. Secondly, we show the practical step to perturb the model functions of histogram, kernel SVM, Gaussian process regression, and the kernel density estimator. We evaluate the trade-off between the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms for US census dataset [IPU09] and a comprehensive range of benchmarks. The results validate that the error decreases for non-parametric machine learning as a service with increase in the size of training dataset and privacy parameters ϵ and δ.

7.2 Related work

Most of the big technology companies offer machine learning as a service on their cloud platforms, such as Google's Cloud Machine Learning Engine[2], Microsoft's Azure Learning Studio[3], Amazon's Machine Learning on AWS[4], IBM's Bluemix[5]. These apps provide machine learning models as easy to use APIs for data scientists. Cloud services also provide storage space to host training datasets. For an extensive survey of such platforms, readers can refer to [Pop16].

Privacy of machine learning models is a well-studied topic. Ateniese et al. [AMS+15] show the ability to learn statistical information about the training data through parameters of a trained machine learning model. They show a successful attack on support vector machine and hidden Markov model. Homer et al. [HSR+08] identify the presence of a certain genome in a publicly released highly complex genomic mixture microarray dataset. They do so by comparing distributions of genomes from the released sample to available statistics of the population. Fredrikson et al. [FLJ+14] propose the model inversion attack on machine learning models wherein they learn some sensitive attribute in the training dataset. Given black-box access to the model and access to demographic information about patients, they successfully learn genomic markers of patients. In the follow-up work, Fredrikson et al. [FJR15] show instantiation of a successful model inversion attack on decision trees and neural networks that are implemented on machine learning as a service platform. Shokri et al. [SSSS17] propose membership inference attack that infers the presence of a data-point in the training dataset based on the outputs machine learning models. They perform attacks on classification models provided by commercial platforms from Google and Amazon. We have enlisted the attacks that are pertinent to research in this work. For an extensive survey

[2] `https://cloud.google.com/ml-engine/`

[3] `https://azure.microsoft.com/en-us/services/machine-learning-studio/`

[4] `https://aws.amazon.com/machine-learning/`

[5] `https://www.ibm.com/cloud/`

of attacks on various machine learning models, readers can refer to [DSSU17].

Differential privacy [Dwo06] has become a popular privacy definition to provide privacy guarantees for machine learning algorithms. Researchers have devised privacy-preserving mechanisms to provide differential privacy guarantees for linear regression [CMS11, ZZX+12], logistic regression [CM09, YRUF14], support vector machines [RBHT12], deep learning [ACG+16, SSSS17]. Chaudhury et al. [CMS11] propose differentially private empirical risk minimisation, which lies at the heart of training of machine learning models. They propose output perturbation and objective perturbation. These mechanisms are helpful for releasing parametric machine learning models. Zhang et al. [ZZX+12] propose the functional mechanism that introduces noise in the loss function of a machine learning model. The functional mechanism is useful for parametric machine learning models that estimate parameters of the model by minimising its loss function. Hall et al. [HRW13a] propose the use of functional perturbation that induced noise in the coefficient of expansion of a function in a functional basis. Functions of non-parametric models that use kernels lie in the RKHS spanned by the kernel. Therefore, it is possible to apply the functional perturbation to provide privacy guarantees for non-parametric models. Smith et al. [SZL16] apply functional perturbation by Hall et al. to provide differential privacy Gaussian process regression. Aldá and Rubinstein [AR17] propose Bernstein mechanism that provides a differentially private way of releasing functions of machine learning models in a non-interactive way. Balog et al. [BTS17] provide a functional perturbation that ensures the closure of functions in a finite dimensional RKHS under the appropriate perturbations. Nozari et al. [NTC18] propose a functional perturbation algorithm that is catered to distributed machine learning task.

Yu et al. [YJV06] propose a privacy-preserving method of releasing kernel SVM, which is a non-parametric model. Their work is focused on the distributed machine learning wherein the data is horizontally partitioned and stored on the different nodes. They devise an encryption based method to securely compute

Gram matrix for the specified kernel. We are interested in the non-distributed learning with purely privacy based mechanisms. Therefore, this work is orthogonal to the work at hand.

Jain and Thakurta [JT13] propose three ways, which are interactive, non-interactive and semi-interactive, of using machine learning models with differential privacy guarantees. This work is an instance of interactive use of non-parametric machine learning model wherein we provide differential privacy guarantees using the functional perturbation proposed in the work of Hall et al. [HRW13a].

7.3 Methodology

In this section, we discuss release of trained machine learning models. We argue that non-parametric machine learning models need to be released as a service to users. We further instantiate functional perturbation by Hall et al. [HRW13a], which provides (ϵ, δ)-differential privacy guarantee, to four non-parametric models.

7.3.1 Non-parametric machine learning models as a service

Jain and Thakurta [JT13] propose three ways in which an organisation can use machine learning models. Firstly, they propose a non-interactive model release wherein an organisation releases a model with quantifiable privacy guarantees. Non-interactive model release is plausible for parametric machine learning models since values of the parameters are sufficient to compute outputs for a new data. Non-parametric machine learning models require training dataset along with the parameters to compute outputs for new data. Secondly, they propose a semi-interactive model release wherein an organisation releases model that provides quantifiable privacy guarantees for a specified set of test data. A priori knowledge of test data is not an assumption that can be realised in every business scenario. Lastly, they propose interactive model release wherein an organisation provides machine learning model as a service. It keeps trained model on the server and

users send queries to the server. For non-parametric models, release of training dataset violates the privacy of the users. Therefore, interactive model release, i.e. release of machine learning as a service is a viable alternative.

Differential privacy, as defined in Definition 1, is a privacy definition for randomised algorithms. In order to provide quantifiable differential privacy guarantees, we need to introduce randomisation while using machine learning models as a service. A privacy-preserving mechanism introduces randomisation to avoid the release of true outputs. Under appropriately calibrated randomisation, privacy-preserving mechanisms provide differential privacy guarantees.

Firstly, randomisation can be introduced by adding an appropriately calibrated random noise to the output of the query. These privacy-preserving mechanisms are called as *output perturbation mechanisms*. For instance, Laplace mechanism [DR$^+$14] adds noise drawn from Laplace distribution whereas Gaussian mechanism [DR$^+$14] adds noise drawn from Gaussian distribution to the output of the model. Multiple evaluations of such mechanisms result in a sequential composition [DR$^+$14]. Privacy guarantee of the sequential composition of privacy-preserving mechanisms linearly degrades with the number of evaluations of privacy preserving mechanisms. Secondly, randomisation can be introduced by adding an appropriately calibrated random noise to the model function. Unlike output perturbation mechanisms, which add calibrated noise to every output of the query, the privacy-preserving mechanisms that perturb functions are *one-shot* privacy-preserving mechanisms. They add calibrated noise in a function leading to change in its functional form. The noisy functional form is used for computing outputs. Therefore, functional perturbation does not suffer from the degradation in the differential privacy guarantee with increasing the number of queries.

When a machine learning model is provided as a service, one cannot strictly control the number of times a user accesses the service. Therefore, we choose functional perturbation based privacy-preserving mechanism. Hall et al.[HRW13a] propose the functional mechanism that adds calibrated noise to the expansion

of the model function in an appropriate functional basis. Functions of non-parametric machine learning models, especially the ones that use kernels, lie in Reproducing Kernel Hilbert Space (RKHS) [SS98] associated with the kernel. Thus, RKHS readily provides a functional basis for functions of non-parametric models. Zhang et al. [ZZX+12] propose the functional mechanism that adds calibrated noise to the loss function of machine learning model. Loss functions are akin to parametric models that train their parameters using some appropriate loss function. Therefore, we choose to use functional perturbation as proposed by Hall et al. [HRW13a] to provide differential privacy guarantees for non-parametric models released as a service.

7.3.2 Functional perturbation in RKHS

Hall et al. [HRW13a] propose a mechanism that provides a calibrated functional perturbation that provides quantifiable (ϵ, δ)-differential privacy guarantee. We briefly explain functional perturbation of a function that lies in a reproducing kernel Hilbert space (RKHS).

Suppose that a function $f : \mathbb{R}^d \to \mathbb{R}$ lies in RKHS, $\mathcal{H}_k$, associated with a kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. For a given dataset $D = \{x_i\}_{i=1}^n$ where each $x_i \in \mathbb{R}^d$, let $\{k(\cdot, x_i)\}_{i=1}^n$ denotes a basis of $\mathcal{H}_k$. In this basis, any function $f \in \mathcal{H}_k$ is expanded as:

$$f(\cdot) = \sum_{i=1}^{n} w_i^f k(\cdot, x_i)$$

where each $w_i^f \in \mathbb{R}$. Inner product between two functions $f, g \in \mathcal{H}_k$ is defined as:

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i^f w_j^g k(x_i, x_j)$$

The inner product is used to define norm of any function in $\mathcal{H}_k$ as $\|f\|_{\mathcal{H}_k} = \sqrt{\langle f, f \rangle}$.

Functional perturbation adds calibrated noise sampled from a Gaussian process

to a model function. Gaussian process uses the kernel that is associated with RKHS where the function lies. We formally define functional perturbation in Definition 5.

Definition 5 (**Functional perturbation** [HRW13a])**.** *Let f_D denotes a model function, whose parameters (or hyperparameters) are estimated on a dataset $D \in \mathcal{D}$. Assume that f_D lies in a reproducing kernel Hilbert space, $\mathcal{H}_k$, with an associated kernel k. Functional perturbation is a privacy-preserving mechanism that perturbs f_D as follows:*

$$f'_D = f_D + \Delta \frac{c(\delta)}{\epsilon} G. \tag{7.1}$$

where G is a sample path of a Gaussian process with mean zero and covariance function k and $\Delta, \epsilon, \delta > 0$.

Functional perturbation in Definition 5 satisfies (ϵ, δ)-differential privacy when parameters are calibrated as $\Delta \geq \max_{D,D'} \|f_D - f_{D'}\|_{\mathcal{H}_k}$ and $c(\delta) \geq \sqrt{2 \log \frac{2}{\delta}}$. Δ is *sensitivity* of the functions in RKHS $\mathcal{H}_k$. Sensitivity is the maximum deviation of model functions that are trained on any two neighbouring datasets D and D'. In order to apply functional perturbation for machine learning tasks, we need to compute the sensitivity of respective model functions.

7.3.3 Applications to non-parametric machine learning models

We now illustrate application of functional perturbation for four non-parametric machine learning models. We use non-parametric models that are based on kernel methods. For such non-parametric models, model functions lie in the RKHS associated with the specified kernel.

Histogram. Histogram [Mur12a] is used for solving discretised probability density estimation problem. It discretises the domain of a given dataset into a finite number of *bins*. Each bin defines an interval in the domain of the training dataset. Probability of a data-point inside an interval is commensurate to the

number of training data-points that lie in the interval.

For a fixed number of bins b, histogram is a vector in $\mathbb{R}^b$. Therefore, we can consider histogram estimation as a function $f : \mathcal{D} \to \mathbb{R}^b$ wherein $\mathcal{D}$ is a universe of datasets. Let, $\{e_i\}_{i=1}^b$ be the standard basis of Euclidean space $\mathbb{R}^b$. Standard basis spans RKHS associated with the dot product kernel, i.e. $k(x, y) = x^T y$. For a pair of neighbouring datasets, L_1 norm between two histograms is two in the case when the distinct data-points occupy two different bins. Therefore, sensitivity of the histogram function is 2. Let, f_D denotes the histogram for a dataset D with the number of bins b. Thus, the functional perturbation of Equation 7.1 for histograms takes the form

$$f'_D = f_D + \frac{2}{n}\frac{c(\delta)}{\epsilon}G.$$

Kernel density estimation. Kernel density estimation [Mur12a] is a probability density estimation problem that estimates the probability density function of a training dataset. It assumes a probability density function centred at every data-point in the training dataset. Probability of a new data-point is computed as weighted average of the probabilities computed using the probability densities centred at every data-point.

We consider the kernel function namely Gaussian kernel that outputs values in the range $[0, 1]$. It acts as a probability density function. Let k denotes a Gaussian kernel with bandwidth h. Estimate of the probability density function for a dataset $D = \{x_i\}_{i=1}^n$ for the Gaussian kernel k is presented as

$$f_D(\cdot) = \frac{1}{n}\sum_{x_i \in D} k(\cdot, x_i) = \frac{1}{n}\sum_{x_i \in D} \frac{1}{(2\pi h^2)^d} \exp\left(-\frac{\langle \cdot, x_i \rangle}{2h^2}\right).$$

Hall et al. [HRW13a] compute the sensitivity Δ of kernel density estimator with a Gaussian kernel as $\frac{\sqrt{2}}{n(2\pi h^2)^{d/2}}$. Thus, from Equation 7.1 the functional

perturbation for kernel density estimate with Gaussian kernel is

$$f'_D = f_D + \left(\frac{\sqrt{2}}{n(2\pi h^2)^{d/2}}\right)\frac{c(\delta)}{\epsilon}G.$$

Gaussian process regression. Gaussian process [Ras04] is a collection of Gaussian random variables such that any subset follows a multivariate Gaussian distribution. Covariance function for the multivariate Gaussian distribution is calculated using a kernel function k. Gaussian process regression outputs a response sampled from posterior distribution of a test data-point given the training dataset. Mean function $\bar{f_D}$ and variance function $Var(f_D)$ of the posterior distribution computed on a training dataset D are given in Equation 7.2.

$$\begin{aligned}\bar{f_D}(\cdot) &= \sum_{d_i \in D}\sum_{d_j \in D}(K_D + \sigma_n^2\mathbb{I})^{-1}_{ij} y_j k(\cdot, x_i) \\ Var(f_D)(\cdot) &= k(\cdot,\cdot) - \sum_{d_i \in D}\sum_{d_j \in D}(K_D + \sigma_n^2\mathbb{I})^{-1}_{ij} k(\cdot, x_i)k(\cdot, x_j) \end{aligned} \tag{7.2}$$

K_D is the Gram matrix computed using kernel k on the training dataset and d is the dimension of each training data-point.

Smith et al. [SZL16] use the functional perturbation to provide differential privacy guarantee to Gaussian process regression. Equation 7.2 shows that the posterior covariance function does not require responses y_j's in the training data. Since only the responses are sensitive towards the disclosure, Smith et al. [SZL16] proposed to perturb only the posterior mean function. Since the sensitivity of the posterior mean function with Gram matrix K_D is $d\|(K_D + \sigma_n^2\mathbb{I})^{-1}\|_\infty$, they apply the functional perturbation to the posterior mean function as

$$\bar{f_D}' = \bar{f_D} + (d\|(K_D + \sigma_n^2\mathbb{I})^{-1}\|_\infty)\frac{c(\delta)}{\epsilon}G.$$

Kernel support vector machine. Support vector machine (SVM) [CV95] is used for solving a classification problem. SVM outputs the class label of a data-point that is specified as the input. Linear SVM is a parametric machine

learning model whereas kernel SVM is a non-parametric machine learning model.

Let us consider a data-point $d = (x, y)$ where $x \in \mathbb{R}^d$ are the predictors and $y \in \{-1, 1\}$ is the associated class label. Let $\mathcal{D}$ denotes universe of datasets with n data-points each. We fit a support vector machine classifier with a kernel k on a training dataset $D \in \mathcal{D}$ with n data-points. Kernel support vector machine assumes the form $f(\cdot) = \langle w, \phi(\cdot) \rangle$ where $w \in \mathbb{R}^F$ and $\phi : \mathbb{R}^d \to \mathbb{R}^F$. w is estimated by solving the optimisation problem in Equation 7.3. In Equation 7.3, C denotes the regularisation constant and l denotes the loss function.

$$\max_{w \in \mathbb{R}^F} \frac{\|w\|^2}{2} + C \sum_{d \in D} l(y_i, f_D(x_i)) \tag{7.3}$$

Using *hinge loss*, $l_{hinge}(x, y) = max(0, xy)$, as the loss function we obtain a closed form solution. It is presented in Equation 7.4. In the solution, α^*'s are called support vectors that are solutions to *dual* of the optimisation problem in Equation 7.3.

$$w_D = \sum_{i=1}^{n} \alpha_i^* y_i k(\cdot, x_i) \tag{7.4}$$

Hall et al. [HRW13a] compute the sensitivity of the minimisers of regularised functionals in RKHS. Equation 7.3 represents an instance of the same problem. Since the sensitivity of w_D is $\frac{2C}{n}$, following Equation 7.1 the functional perturbation for kernel SVM takes the form

$$w'_D = w_D + \left(\frac{2C}{n}\right) \frac{c(\delta)}{\epsilon} G.$$

7.4 Empirical performance evaluation

In this section, we present effectiveness and efficiency evaluation of functional perturbation for four non-parametric models, *viz.* histogram, kernel density estimation (KDE), Gaussian process regression (GP regression) and kernel support vector machine (kernel SVM), as a service. We comparatively evaluate output perturbation and functional perturbation mechanism. We observe that output

perturbation mechanism are less effective than functional perturbation mechanism for a specified setting of differential privacy parameters.

7.4.1 Datasets and experimental setup

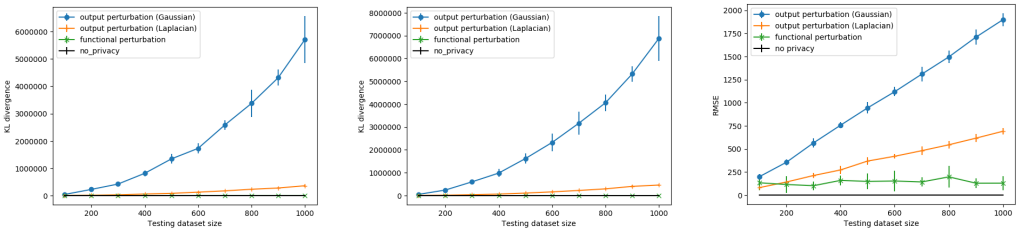
Real world dataset. We conduct experiments on a subset of the 2000 US census dataset provided by Minnesota Population Center in its Integrated Public Use Microdata Series [IPU09]. The census dataset consists of 1% sample of the original census data. It spans over 1.23 million households with records of 2.8 million people. The value of several attributes is not necessarily available for every household. We have therefore selected $212,605$ records, corresponding to the household heads, and 6 attributes, namely, *Age, Gender, Race, Marital Status, Education, Income*. We treat this dataset as the population from which we draw samples of desired sizes.

Benchmark datasets. For histogram and kernel density estimation, we follow Hall et al. [HRW13a] and synthetically generate a dataset from a known probability distribution. We generate 5000 points from a Gaussian distribution with mean and variance of 2 and 1.3 respectively. For Gaussian process regression, we follow Smith [SZL16] and use !Kung San woman demographic dataset [How67]. It comprises of heights and ages of 287 women. For kernel SVM, we use Iris dataset [DKT17]. It comprises of three species of Iris flower with four attributes: length and width of sepal and petal.

Experimental setup. All experiments are run on Linux machine with 12-core 3.60GHz Intel® Core i7™processor with 64GB memory. Python® 2.7.6 is used as the scripting language. We use RBF kernel for the experiments. Hyperparameters of the kernel are tuned by performing cross-validation on respective dataset.

7.4.2 Evaluation metrics

We perform experiments on four non-parametric models solving the problems of estimation, prediction, and classification. Therefore, we use different metrics



(a) Histogram (b) KDE (c) GP regression

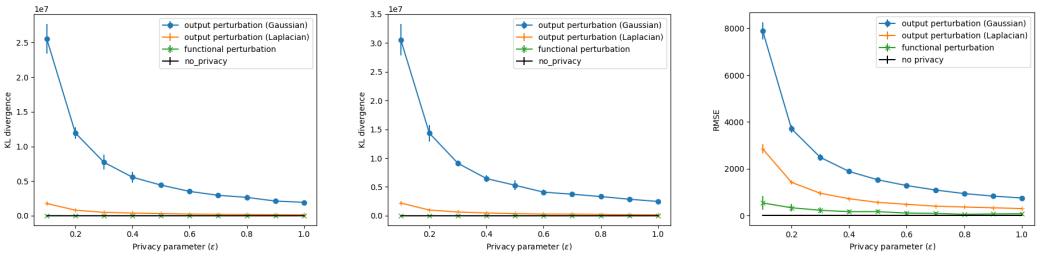
Figure 7.1: Comparative evaluation of functional and output perturbation mechanisms for varying size of the test datasets. We compare (0.4, 0.001)-differentially private functional perturbation, (0.4, 0.001)-differentially private Gaussian mechanism and (0.4, 0.0)-differentially private Laplace mechanism.

of effectiveness for the evaluation. Histogram and kernel density estimation are used for estimating probability density of a given data-point and we use Kullback-Leibler divergence (KL divergence) as the metric of effectiveness. Gaussian process regression is used for predicting real-valued attribute and we use root mean squared error (RMSE) as the metric of effectiveness. Kernel SVM is used for classification and we use classification error as the metric of effectiveness. Smaller the value of any of these metrics higher is the effectiveness of the model. In order to evaluate efficiency, we compute query execution time, i.e. the time required to compute output of the model.

7.4.3 Results

In this section, we present the results on the real-world census dataset.

Effectiveness evaluation. We start by the comparative evaluation of the functional perturbation and output perturbation mechanisms, namely Gaussian mechanism and Laplace mechanism. Output perturbation mechanisms are not directly applicable for machine learning models with categorical outputs, such as SVMs. Therefore, we perform comparative study for histograms, KDE and GP regression. We also plot the effectiveness of the model without any application of privacy-preserving mechanism. We denote it by “no privacy”. In case of



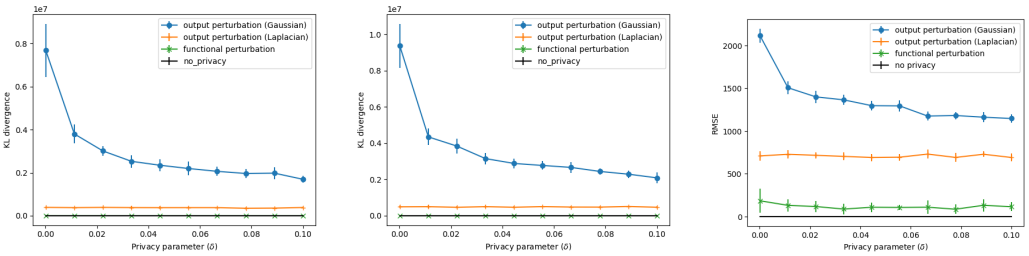
(a) Histogram (b) KDE (c) GP regression

Figure 7.2: Comparative evaluation of functional and output perturbation mechanisms for varying privacy parameter ϵ. We use datasets of size 5000 to train the models and set $\delta = 0.001$.

histogram and KDE, we do not have the true distributions of the attributes from the census dataset. Therefore, we compute effectiveness by computing KL divergence between functionally perturbed estimators and their non-private counterparts.

In Figure 7.1, we comparatively evaluate effectiveness for varying size of testing datasets. Across three models, we observe that effectiveness of the output perturbation mechanisms degrades as the testing dataset size increases. We do not observe such a phenomenon with the functional perturbation. Due to sequential composition [DR+14], privacy guarantee of output perturbation mechanisms linearly degrades with the number of evaluations. In order to attain differential privacy with specified privacy parameters, output perturbation mechanisms introduce higher amount of noise for testing datasets of large sizes. Higher amount of noise results in reduction in the effectiveness.

In Figures 7.2 and 7.3, we comparatively evaluate effectiveness for varying privacy parameters ϵ and δ respectively. Across three models, we observe that the effectiveness of the output perturbation mechanisms increases as values of privacy parameters increase. Privacy parameter ϵ quantifies the privacy guarantee of differential privacy. Higher values of ϵ provides weaker privacy guarantees. Weaker privacy guarantees require less amount of noise and hence yield higher effectiveness. Privacy parameter δ is a quantifier of the extent of slack provided



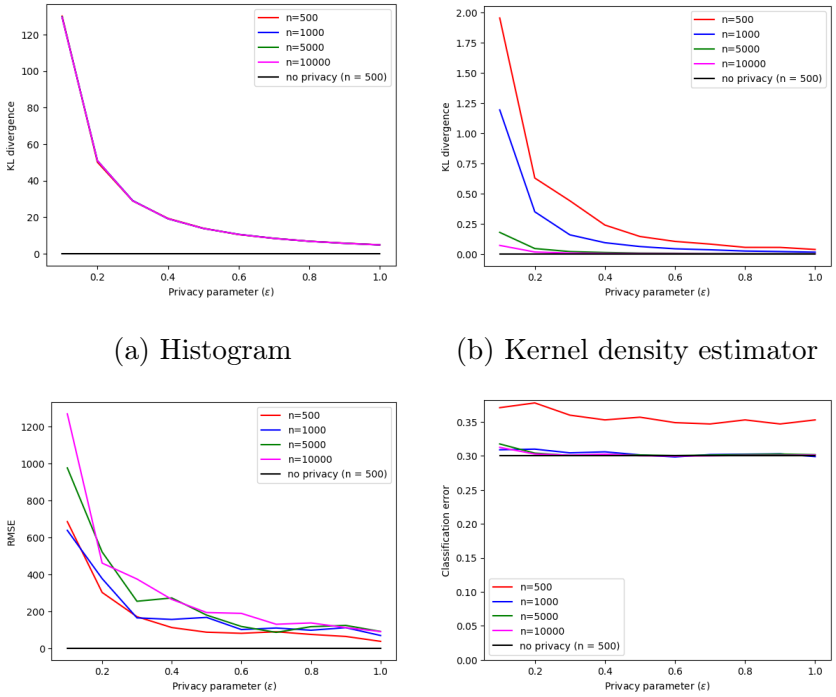
(a) Histogram (b) KDE (c) GP regression

Figure 7.3: Comparative evaluation of functional and output perturbation mechanisms for varying privacy parameter δ. We use datasets of size 5000 to train the models and set $\epsilon = 0.4$.

in the privacy guarantee of ϵ-differential privacy. In order to provide a robust differential privacy guarantee, we require the value of δ to be as small as possible. Thus, with increasing value of δ the amount of perturbation in the function reduces and hence, the effectiveness increases.

We continue our evaluation of functional perturbation for four non-parametric models on the census dataset. In Figure 7.4, we present the effectiveness as privacy parameter ϵ varies between 0 to 1 keeping $\delta = 0.0001$ for different sizes of training dataset sizes. We observe that effectiveness of the models increases with increasing the size of dataset. The reason for this is twofold. Firstly, effectiveness of non-parametric models increases with increasing size of the training dataset [Mur12a]. Secondly, closer inspection of equations of functional perturbation for each of the four models tells that the amount of noise is inversely proportional to the number of training data-points. Thus, the functional perturbation adds lesser amount of noise for specified privacy parameters as the size of training dataset increases. We make similar observations while evaluating the effectiveness under variation in privacy parameter δ for a fixed value of ϵ.

Efficiency evaluation. In Figure 7.5(a), we plot the query execution time that is the time required to compute the output for non-parametric models as a service, on a dataset of size 5000 with varying privacy levels. For a given non-parametric model, we observe that query execution time does not depend on the value of



(a) Histogram

(b) Kernel density estimator

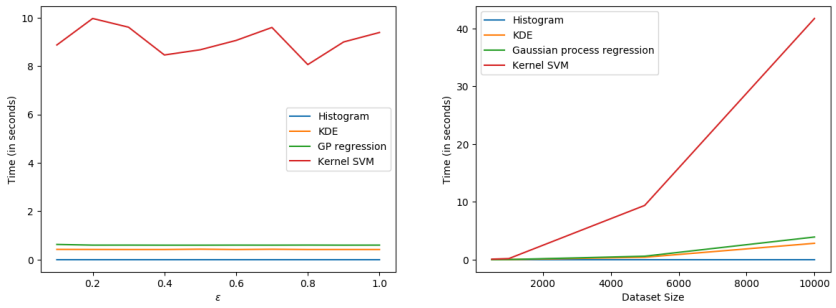
(c) Gaussian process regression

(d) Kernel SVM

Figure 7.4: Variation in the utility as the privacy parameter ϵ changes for datasets of varying sizes. Experiments are carried out with $\delta = 0.0001$ on the census dataset.

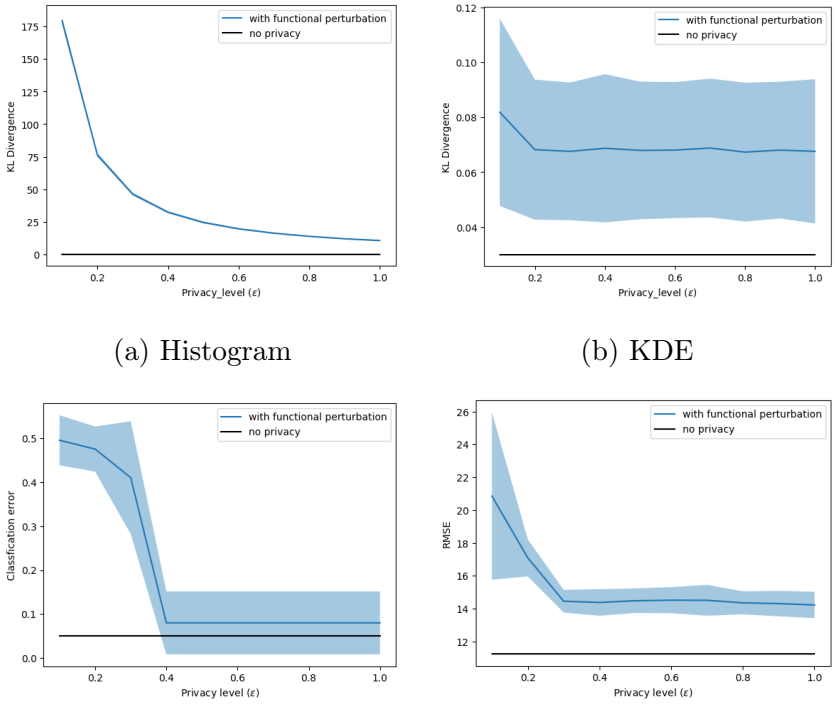
the privacy parameter ϵ. Functional perturbation involves sampling a path from the Gaussian process with zero mean function and covariance function computed using the kernel function used in the non-parametric model. The computation of covariance functions requires a significant computational time. This computation time is not affected by any particular value of privacy level. We make similar observation for varying the privacy parameter δ.

In Figure 7.5(b), we plot query evaluation time for varying size of the training datasets. For this experiment, we set privacy parameters ϵ and δ to 0.2 and 0.001 respectively. We observe that evaluation time increases with increasing size of the training dataset. Large training datasets require large amount of correlations to be computed for every new data-point. Therefore, larger training datasets incur higher amount of time.



(a) Privacy parameter (b) Training dataset size

Figure 7.5: Evaluation of efficiency of functional perturbation on four non-parametric machine learning models. Figure (a) plots query execution time versus privacy level. Figure (b) plots query execution time versus training dataset size. For these experiments, we set $\delta = 0.001$. We set $\epsilon = 0.2$ for the plot in Figure (b).



(a) Histogram (b) KDE

(c) Gaussian process regression (d) Kernel SVM

Figure 7.6: Variation in the utility as the privacy level changes for datasets of varying sizes. Experiments are carried out with $\delta = 0.0001$ on the benchmark datasets.

Experiments on benchmark datasets. In order to have reproducible results, we also conduct experiments on the datasets that are either synthetic or publicly

available. We observe results that are consistent with the results on the real-world dataset.

In Figure 7.6, we present effectiveness of functional perturbation technique on the *benchmark datasets*. We perform 10 experimental runs for each value of the privacy level. Solid lines in Figure 7.4 show mean effectiveness whereas shaded region covers values that are one standard deviation away from the mean. We invariably observe that effectiveness of the models increases when we increase the privacy level in the functional perturbation. Our observation for the other experiments on the benchmark datasets are consistent with the observations that we make for the same experiment on the census datasets.

7.5 Discussion

We show that functional perturbation is not only pragmatic for releasing machine learning models as a service but also yields higher effectiveness than output perturbation mechanisms for specified privacy parameters. We show how to apply functional perturbation to the model functions of histogram, kernel density estimator, kernel SVM and Gaussian process regression in order to provide (ϵ, δ)-differential privacy. We evaluate the tradeoff between the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms for a real-world dataset as well as a selection of benchmarks.

We are now studying functional perturbation for non-parametric machine learning methods such as k-nearest neighbour density estimation and kernel Bayesian optimisation. We are also interested in studying a step by step functional perturbation method that perturbs a model function in adaptive way balancing the specified privacy and utility requirements.

Publication

Work in this chapter is part of the following publications.

- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Privacy as a service: Publishing data and models (Demo paper). In *24th International Conference on Database Systems for Advanced Applications, DASFAA 2019, Chiang Mai, Thailand.*

- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Evaluation of differentially private non-parametric machine learning models. In *Database and Expert Systems Applications- 30th International Conference, DEXA 2019, Linz, Austria (Under review)*

Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd.

Part III

Privacy at risk

In this part, we present our work privacy at risk and a cost model that help addressing the practical issues while using differential privacy guarantees to release machine learning models. This theoretical work is submitted at the following venue:

- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Differential privacy at risk. Submitted in *Journal of Privacy and Confidentiality (Under review)*

Chapter 8

Differential privacy at risk

In this chapter, we propose *privacy at risk* that quantifies the probability with which a privacy-preserving mechanism satisfies differential privacy for a specified privacy level. The probabilistic quantification depends on two sources of randomness, the explicit randomness induced by the noise distribution and the implicit randomness induced by the data-generation distribution, and the coupling between the two. We instantiate privacy at risk for the Laplace mechanism and provide analytical results. We demonstrate the applicability of privacy at risk in two decision-making problems. Firstly, we empirically illustrate how a data steward finds the compromise between the privacy level and the desired utility requirement by implementing Laplace mechanism for ridge regression. Secondly, we propose a cost model that bridges the gap between the privacy level and the compensation budget estimated by a GDPR compliant business entity. The cost model for privacy at risk being a convex function of the privacy level leads to a unique privacy at risk level that minimises the compensation budget. Thus, privacy at risk not only quantifies the privacy level for a given privacy-preserving mechanism but also facilitates decision-making in problems that focus on the privacy-utility trade-off and the compensation budget minimisation.

8.1 Introduction

The privacy level ϵ in ϵ-differential privacy [DR$^+$14] quantifies an upper bound of the worst case privacy loss incurred by a privacy-preserving mechanism. Generally,

a privacy-preserving mechanism *sanitises* the results by adding the calibrated amount of random noise to the results [DR+14, CMS11]. The calibration of noise depends on the sensitivity of the query and the specified privacy level. In a real-world setting, a data steward has to specify a privacy level that balances requirements of the users and monetary constraints of the business entity.

Abowd et al. [GAP18] report the issues that have been observed in deploying differential privacy as the privacy definition by the US census bureau [Abo18]. They highlight the lack of formal methods to choose the privacy level. Additionally, analytical computation of the sensitivity for any general query is not a simple task. For instance, the sensitivity of the optimal solution of an empirical risk minimisation problem involves rigorous and non-trivial analytical calculations [CMS11].

We propose *privacy at risk* that quantifies the probability with which a privacy-preserving mechanism satisfies differential privacy for a specified privacy level (Section 8.2). The probabilistic quantification depends on two sources of randomness, the explicit randomness induced by the noise distribution and the implicit randomness induced by the data-generation distribution, and the coupling between the two. In its most general setting, privacy at risk is computed over the output space obtained by applying the privacy-preserving mechanism on the data-generation distribution. Thus, the proposed definition unifies and extends probabilistic differential privacy [MKA+08] and random differential privacy [HRW12]. Probabilistic differential privacy takes into account the explicit randomness whereas random differential privacy takes into account the implicit randomness.

In order to analytically compute privacy at risk, we need to have at hand analytical forms of both the implicit and explicit sources of randomness as well as of the query. From the analytical form of the data-generation distribution and of the query, we could try and derive the distribution of the sensitivity of the query. From the analytical forms of the sensitivity distribution and of the

noise distribution we could try and derive the distribution of the privacy loss guaranteed by the calibrated noise. While this may be possible in some cases, for instance when we consider a Gaussian or Gaussian mixture generative process for the datasets and a simple enough query such as a non-regularised linear regression and Gaussian noise, it is generally challenging.

We instantiate privacy by risk to three cases illustrative of both the interest of the notion from an application perspective and its technical facets from a computation perspective for the widely used Laplace mechanism [DMNS06a] that adds Laplacian noise.

Firstly, we compute privacy at risk under the effect of explicit randomness induced by the Laplacian noise (Section 8.3.1). In order to quantify this effect, we calculate the overlap between differently parameterised Laplace distributions under the constraint of the sensitivity of the query. This calculation leads us to the use of Bessel-K distribution in our analytical result.

Secondly, we compute the privacy at risk under the effect of implicit randomness induced by the data-generation distribution (Section 8.3.2). In order to quantify this effect, we statistically estimate the sensitivity of the query, which we call as the *sampled sensitivity* of the query, with the help of an empirical distribution over the sensitivities. The empirical distribution is generated using the sensitivities computed for the neighbouring datasets sampled from the data-generation distribution. We use DKW inequality [M$^+$90] to quantify the closeness of the sampled sensitivity to the true sensitivity of the query.

Lastly, we compute privacy at risk under the effect of both explicit and implicit randomness induced by the Laplace mechanism operating on the dataset that is modelled by a data-generation distribution (Section 8.3.3). In order to quantify this effect, we revise the analytical results in the earlier cases in the light of the output space that is obtained by applying the privacy-preserving mechanism on the data-generation distribution. We find that the effect of the noise distribution, which is quantified by the Bessel-K distribution, is coupled with the effect of the

data-generation distribution, which is quantified by the parameters in the DKW inequality. This coupling is evident in the analytical result that we derive for this case.

We demonstrate the applicability of privacy at risk in two decision-making problems. *First application* discusses how a data steward applies privacy at risk to choose a privacy level ϵ with confidence level γ (Section 8.4.1). We empirically illustrate design methods for the three cases for ridge regression with the Laplace mechanism [LNR+17]. In each of the three cases, we illustrate how a data steward finds the balance between the privacy level and the desired utility requirement. *Second application* discusses how a GDPR [gdp16] compliant business entity applies privacy at risk to minimise the compensation budget that it needs to maintain to pay back to the stakeholders in the unfortunate event of a personal data breach (Section 8.4.2). We propose a cost model that bridges the gap between the privacy level in differential privacy and the compensation budget as it is estimated by business entities. We adapt the proposed model for privacy at risk. The corresponding model for privacy at risk is a convex function of the privacy level. Hence, it leads to a unique privacy at risk level that minimises the compensation budget. We further illustrate a realistic scenario of a GDPR compliant health centre in a university that biannually publishes the health statistics of the staff using the Laplace mechanism. The illustration shows that the use of privacy at risk as a quantifier of privacy instead of *pure* differential privacy yields a significant reduction in the compensation budget.

In conclusion, the benefits of privacy at risk are twofold. It not only quantifies the privacy level for a given privacy-preserving mechanism but also facilitates decision-making in problems that focus on the privacy-utility trade-off and the compensation budget minimisation.

8.2 Privacy at risk

The parameters of a privacy-preserving mechanism are calibrated using the privacy level and the sensitivity of the query. A data steward needs to choose appropriate privacy level (the choice of an actual privacy level by a data steward in regard to her business requirements is a non-trivial task [LC11c]). Recall that the privacy level in the definition of differential privacy corresponds to the worst case privacy loss. Business users are however used to taking and managing risk.

For instance, risk analysts use *Value at Risk* [Jor00] to quantify the loss in investments for a given portfolio and an acceptable confidence bound. Motivated by the formulation of *Value at Risk*, we define *privacy at risk* as a privacy definition. For a given privacy-preserving mechanism, privacy at risk defines the privacy level ϵ with a confidence level γ. For the sake of clarity, we refer to this privacy level ϵ as the privacy at risk level.

Definition 6 (**Privacy at risk**). *For a given data generating distribution $\mathcal{G}$, a privacy-preserving mechanism $\mathcal{M}$, equipped with a query f and with parameters Θ, satisfies (ϵ, γ)-privacy at risk, if for all $Z \subseteq Range(\mathcal{M})$ and x, y sampled from $\mathcal{G}$ such that $x \sim y$:*

$$\mathbb{P}\left[\log\left|\frac{\mathbb{P}(\mathcal{M}(f,\Theta)(x) \in Z)}{\mathbb{P}(\mathcal{M}(f,\Theta)(y) \in Z)}\right| > \epsilon\right] \leq \gamma, \tag{8.1}$$

where the outer probability is calculated with respect to the probability space $\text{Range}(\mathcal{M} \circ \mathcal{G})$ *obtained by applying the privacy-preserving mechanism $\mathcal{M}$ on the data-generation distribution $\mathcal{G}$.*

If a privacy-preserving mechanism is ϵ_0-differentially private for a given query f and parameters Θ, its privacy at risk level coincides with the privacy level ϵ_0 with confidence level 1. Our interest is to study the effect of both the randomness induced by the noise and that of the data-generation distribution to help a data steward calibrate the privacy-preserving mechanism as per a desired privacy level and a desired confidence level.

Unifying Probabilistic and Random Differential Privacy. Interestingly, privacy at risk unifies the notions of probabilistic [MKA+08] and random [HRW12] differential privacy by accounting for both sources of randomness in a privacy-preserving mechanism. Probabilistic differential privacy [MKA+08] incorporates the explicit randomness of the noise distribution of the privacy-preserving mechanism. In probabilistic differential privacy, the outer probability is computed over the sample space of Range($\mathcal{M}$) and all datasets are equally probable. Thus, if we consider a uniform data-generation distribution in Equation 8.1, Definition 6 leads to the definition of probabilistic differential privacy. Random differential privacy [HRW12] incorporates the implicit randomness of the data-generation distribution. In random differential privacy, the outer probability is calculated with respect to the support of the data-generation distribution $\mathcal{G}$. Thus, if we consider a specific data-generation distribution $\mathcal{G}$ in Equation 8.1, Definition 6 leads to the definition of random differential privacy.

Now, we instantiate privacy at risk for the Laplace mechanism for three cases: two cases involving two sources of randomness and third case involving the coupled effect. Three different cases correspond to three different interpretations of the confidence level, represented by the parameter γ, corresponding to three interpretation of the support of the outer probability in Definition 6. In order to highlight this nuance, we denote the confidence levels corresponding to the three cases and their three sources of randomness as γ_1, γ_2 and γ_3, respectively.

8.3 Privacy at risk for Laplace mechanism

In this section, we instantiate Privacy at Risk for widely used Laplace mechanism (Definition 4) in three cases. In case of explicit randomness, we obtain a closed form solution that relates privacy at risk level to the respective confidence level. In the rest of the cases, we derive an upper bound on the confidence level.

8.3.1 The case of explicit randomness

In this section, we study privacy at risk of the Laplace mechanism by considering effect of the explicit randomness induced by the Laplacian distribution. We assume that the sensitivity of the query is known a priori. We study the privacy at risk at a given confidence level, γ_1, on the noise distribution induced by the Laplace mechanism.

For a Laplace mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_f}$ calibrated with sensitivity Δ_f and privacy level ϵ_0, we present the analytical formula relating privacy at risk level ϵ and the confidence level γ_1 in Theorem 3.

Theorem 3. *The confidence level $\gamma_1 \in [0, 1]$ of achieving a privacy at risk level $\epsilon \geq 0$ by a Laplace Mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_f}$ is given by*

$$\gamma_1 = \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \epsilon_0)},$$

where T is a random variable dependent on the Laplace noise $\text{Lap}(\frac{\Delta_f}{\epsilon_0})$*, and follows the* $\text{BesselK}\left(k, \frac{\Delta_f}{\epsilon_0}\right)$ *distribution.*

The result of Theorem 3 can be analytically expressed as the result of Corollary 1 by using the properties of the *Bessel K-distribution.*

Corollary 1. *The analytical form of the confidence level γ for a Laplace mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_f}$ operating on queries with k-dimensional output and Privacy at risk ϵ is:*

$$\gamma_1 = \frac{{}_1F_2(\frac{1}{2}; \frac{3}{2} - k, \frac{3}{2}; \frac{\epsilon^2}{4})\sqrt{\pi}4^k\epsilon - {}_1F_2(k; \frac{1}{2} + k, k + 1; \frac{\epsilon^2}{4})2\epsilon^{2k}\Gamma(k)}{{}_1F_2(\frac{1}{2}; \frac{3}{2} - k, \frac{3}{2}; \frac{\epsilon_0^2}{4})\sqrt{\pi}4^k\epsilon_0 - {}_1F_2(k; \frac{1}{2} + k, k + 1; \frac{\epsilon_0^2}{4})2\epsilon_0^{2k}\Gamma(k)}, \tag{8.2}$$

where ${}_1F_2$ is the normalised regularised hypergeometric function [AD10].

Thus, we obtain an analytical closed-form formula to compute γ_1 for a given Laplace mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_f}$ and a privacy at risk level ϵ.

Figure 8.1 shows the plot of the privacy at risk level against confidence level for different values of k and for a Laplace mechanism $\mathcal{L}_{1.0}^{1.0}$. As the value of k increases,

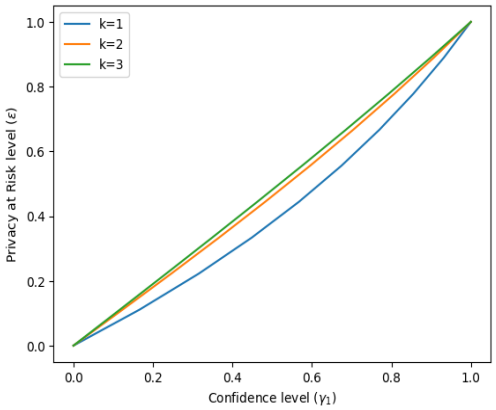


Figure 8.1: Privacy at risk level ϵ for varying confidence level γ_1 and for dfiff-ferent dimensions of $Range(f)$ for the Laplace mechanism $\mathcal{L}^{1.0}_{1.0}$.

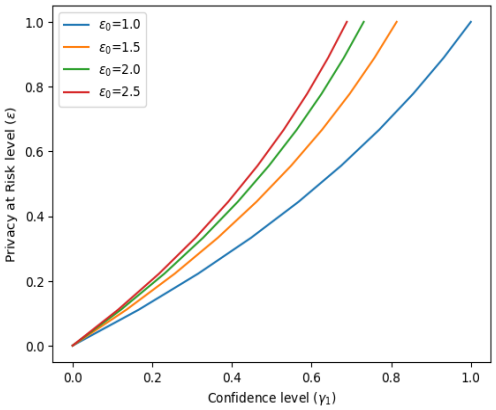


Figure 8.2: Privacy at risk level ϵ for varying confidence level γ_1 for Laplace mechanism $\mathcal{L}^{1.0}_{\epsilon_0}$ for $k = 1$ with different ϵ_0.

the amount of noise added in the output of numeric query increases. Therefore, for a fixed value of the confidence level, the privacy at risk level increases with the value of k.

The analytical formula giving γ_1 as a function of ϵ is a bijection. We need to invert it to obtain the privacy at risk level ϵ for a given confidence level γ_1. However the analytical closed form for such an inverse function is not explicit. We use a numerical approach to compute privacy at risk level for a given confidence level from the analytical formula of Corollary 1. This corresponds to inverting the analytical formula for γ_1 by reading the relevant plot in Figure 8.1 from the y-axis to the x-axis.

Let us now present the details of the derivation of the analytical formula for the confidence level in Theorem 3 and Corollary 1.

Although a Laplace mechanism $\mathcal{L}^{\Delta f}_{\epsilon}$ induces higher amount of noise on average than a Laplace mechanism $\mathcal{L}^{\Delta f}_{\epsilon_0}$ for $\epsilon < \epsilon_0$, there is a non-zero probability that $\mathcal{L}^{\Delta f}_{\epsilon}$ induces noise commensurate to $\mathcal{L}^{\Delta f}_{\epsilon_0}$. This non-zero probability guides us to calculate the confidence level γ_1 for the privacy at risk level ϵ. In order to get an intuition, we illustrate the calculation of the overlap between two Laplace

distributions as an estimator of similarity between the two distributions.

Definition 7 (**Overlap of Distributions [PP02]**). *The overlap, O, between two probability distributions P_1, P_2 with support $\mathcal{X}$ is defined as*

$$O = \int_{\mathcal{X}} \min[P_1(x), P_2(x)]\ dx.$$

Lemma 3. *The overlap O between two probability distributions,* $\text{Lap}(\frac{\Delta_f}{\epsilon_1})$ *and* $\text{Lap}(\frac{\Delta_f}{\epsilon_2})$*, such that $\epsilon_2 \leq \epsilon_1$, is given by*

$$O = 1 - (\exp(-\mu\epsilon_2/\Delta_f) - \exp(-\mu\epsilon_1/\Delta_f)),$$

where $\mu = \frac{\Delta_f \ln(\epsilon_1/\epsilon_2)}{\epsilon_1 - \epsilon_2}$.

Using the result in Lemma 3, we note that the overlap between two distributions with $\epsilon_0 = 1$ and $\epsilon = 0.6$ is 0.81. Thus, $\mathcal{L}^{\Delta_f}_{0.6}$ induces noise that is more than 80% times similar to the noise induced by $\mathcal{L}^{\Delta_f}_{1.0}$. Therefore, we can loosely say that at least 80% of the times a Laplace Mechanism $\mathcal{L}^{\Delta_f}_{1.0}$ will provide the same privacy as a Laplace Mechanism $\mathcal{L}^{\Delta_f}_{0.8}$.

Although the overlap between Laplace distributions with different scales offers an insight into the relationship between different privacy level and the privacy at risk level, it does not capture the constraint induced by the *sensitivity*. For a given query f, the amount of noise required to satisfy differential privacy is commensurate to the sensitivity of the query. This calibration puts a constraint on the noise that is required to be induced on a pair of neighbouring datasets. We state this constraint in Lemma 4, which we further use to prove that the Laplace Mechanism $\mathcal{L}^{\Delta_f}_{\epsilon_0}$ satisfies (ϵ, γ_1)-privacy at risk.

Lemma 4. *For a Laplace Mechanism $\mathcal{L}^{\Delta_f}_{\epsilon_0}$, the difference in the absolute values of noise induced on a pair of neighbouring datasets is upper bounded by the sensitivity of the query.*

Proof. Suppose that two neighbouring datasets x and y are given input to a numeric query $f : \mathcal{D} \to \mathbb{R}^k$. For any output $z \in \mathbb{R}^k$ of the Laplace Mechanism

$\mathcal{L}_{\epsilon_0}^{\Delta_f}$,

$$\sum_{i=1}^{k}(|f(y_i)-z_i|-|f(x_i)-z_i|) \leq \sum_{i=1}^{k}(|f(x_i)-f(y_i)|)$$
$$\leq \Delta_f.$$

We use triangular inequality in the first step and Definition 2 of sensitivity in the second step. □

We write $\text{Exp}(b)$ to denote a random variable sampled from an *exponential distribution* with scale $b > 0$. We write $\text{Gamma}(k, \theta)$ to denote a random variable sampled from a *gamma distribution* with shape $k > 0$ and scale $\theta > 0$.

Lemma 5 (**[PP02]**). *If a random variable X follows Laplace Distribution with mean zero and scale b, $|X| \sim \text{Exp}(b)$.*

Lemma 6 (**[PP02]**). *If $X_1, ..., X_n$ are n i.i.d. random variables each following the Exponential Distribution with scale b, $\sum_{i=1}^{n} X_i \sim Gamma(n, b)$.*

Lemma 7. *If X_1 and X_2 are two i.i.d. $Gamma(n, \theta)$ random variables, the probability density function for the random variable $T = |X_1 - X_2|/\theta$ is given by*

$$P_T(t) = \frac{2^{2-n} t^{n-\frac{1}{2}} K_{n-\frac{1}{2}}(t)}{\sqrt{2\pi}\Gamma(n)}$$

where $K_{n-\frac{1}{2}}$ is the modified Bessel function of second kind.

Proof. Let X_1 and X_2 be two i.i.d. $\text{Gamma}(n, \theta)$ random variables. Characteristic function of a Gamma random variable is given as

$$\phi_{X_1}(z) = \phi_{X_2}(z) = (1 - \iota z\theta)^{-n}.$$

Therefore,

$$\phi_{X_1 - X_2}(z) = \phi_{X_1}(z)\phi^*_{X_2}(z) = \frac{1}{(1 + (z\theta)^2)^n}$$

Probability density function for the random variable $X_1 - X_2$ is given by,

$$\begin{aligned} P_{X_1-X_2}(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-\mathrm{i}zx}\phi_{X_1-X_2}(z)dz \\ &= \frac{2^{1-n}|\frac{x}{\theta}|^{n-\frac{1}{2}}K_{n-\frac{1}{2}}(|\frac{x}{\theta}|)}{\sqrt{2\pi}\Gamma(n)\theta} \end{aligned}$$

where $K_{n-\frac{1}{2}}$ is the Bessel function of second kind. Let, $T = |\frac{X_1-X_2}{\theta}|$. Therefore,

$$P_T(t) = \frac{2^{2-n}t^{n-\frac{1}{2}}K_{n-\frac{1}{2}}(t)}{\sqrt{2\pi}\Gamma(n)}$$

We denote this probability distribution as BesselK(n, θ). □

Lemma 8. *If X_1 and X_2 are two i.i.d. Gamma(n, θ) random variables and $|X_1 - X_2| \leq M$, then $T' = |X_1 - X_2|/\theta$ follows Truncated BesselK(n, θ, M) distribution with probability density function:*

$$P_{T'}(t') = \frac{P_T(t')}{P_T(T \leq M)},$$

where P_T is the probability density function of BesselK(n, θ).

Lemma 9. *For Laplace Mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_f}$ with query $f : \mathcal{D} \to \mathbb{R}^k$ and for any output $Z \subseteq Range(\mathcal{L}_{\epsilon_0}^{\Delta_f})$, $\epsilon \leq \epsilon_0$,*

$$\gamma_1 \triangleq \mathbb{P}\left[\log\left|\frac{\mathbb{P}(\mathcal{L}_{\epsilon_0}^{\Delta_f}(x) \in Z)}{\mathbb{P}(\mathcal{L}_{\epsilon_0}^{\Delta_f}(y) \in Z)}\right| \leq \epsilon\right] = \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \epsilon_0)},$$

where T follows BesselK$(k, \Delta_f/\epsilon_0)$.

Proof. Let, $x \in \mathcal{D}$ and $y \in \mathcal{D}$ be two datasets such that $x \sim y$. Let $f : \mathcal{D} \to \mathbb{R}^k$ be some numeric query. Let $\mathbb{P}_x(z)$ and $\mathbb{P}_y(z)$ denote the probabilities of getting the output z for Laplace mechanisms $\mathcal{L}_{\epsilon_0}^{\Delta_f}(x)$ and $\mathcal{L}_{\epsilon_0}^{\Delta_f}(y)$ respectively. For any

point $z \in \mathbb{R}^k$ and $\epsilon \neq 0$,

$$\begin{aligned}\frac{\mathbb{P}_x(z)}{\mathbb{P}_y(z)} &= \prod_{i=1}^{k} \frac{\exp\left(\frac{-\epsilon_0|f(x_i)-z_i|}{\Delta_f}\right)}{\exp\left(\frac{-\epsilon_0|f(y_i)-z_i|}{\Delta_f}\right)} \\ &= \prod_{i=1}^{k} \exp\left(\frac{\epsilon_0(|f(y_i)-z_i| - |f(x_i)-z_i|)}{\Delta_f}\right) \\ &= \exp\left(\epsilon\left[\frac{\epsilon_0 \sum_{i=1}^{k}(|f(y_i)-z_i| - |f(x_i)-z_i|)}{\epsilon\Delta_f}\right]\right). \end{aligned} \tag{8.3}$$

By Definition 4,

$$(f(x)-z), (f(y)-z) \sim \text{Lap}(\Delta_f/\epsilon_0). \tag{8.4}$$

Application of Lemma 5 and Lemma 6 yields,

$$\sum_{i=1}^{k} (|f(x_i)-z_i|) \sim \text{Gamma}(k, \Delta_f/\epsilon_0). \tag{8.5}$$

Using Equations 8.4, 8.5, Lemma 4 and Lemma 8, we get

$$\left(\frac{\epsilon_0}{\Delta_f}\sum_{i=1}^{k} |(|f(y_i)-z| - |f(x_i)-z|)|\right) \sim \text{TruncatedBesselK}(k, \Delta_f/\epsilon_0, \Delta_f). \tag{8.6}$$

since, $\sum_{i=1}^{k} |(|f(y_i)-z| - |f(x_i)-z|)| \leq \Delta_f$. Therefore,

$$\mathbb{P}\left(\left[\frac{\epsilon_0}{\Delta_f}\sum_{i=1}^{k} |(|f(y_i)-z| - |f(x_i)-z|)|\right] \leq \epsilon\right) = \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \epsilon_0)}, \tag{8.7}$$

where T follows $\text{BesselK}(k, \Delta_f/\epsilon_0)$. Analytically,

$$\begin{aligned}\mathbb{P}(T \leq x) \propto_1 & \; F_2(\frac{1}{2}; \frac{3}{2}-k, \frac{3}{2}; \frac{x^2}{4})\sqrt{\pi}4^k x \\ & - {}_2F_2(k; \frac{1}{2}+k, k+1; \frac{x^2}{4})x^{2k}\Gamma(k),\end{aligned}$$

where ${}_1F_2$ is the regularised generalised hypergeometric function [AD10]. From

Equation 8.3 and 8.7 we obtain,

$$\mathbb{P}\left[\log\left|\frac{\mathbb{P}(\mathcal{L}_{\epsilon_0}(x)\in S)}{\mathbb{P}(\mathcal{L}_{\epsilon_0}(y)\in S)}\right|\leq\epsilon\right]=\frac{\mathbb{P}(T\leq\epsilon)}{\mathbb{P}(T\leq\epsilon_0)}.$$

□

This completes the proof of Theorem 3.

Corollary 2. *Laplace Mechanism* $\mathcal{L}_{\epsilon_0}^{\Delta_f}$ *with* $f:\mathcal{D}\to\mathbb{R}^k$ *is* (ϵ,δ)*-probabilistically differentially private where*

$$\delta=\begin{cases}1-\frac{\mathbb{P}(T\leq\epsilon)}{\mathbb{P}(T\leq\epsilon_0)} & \epsilon\leq\epsilon_0\\ 0 & \epsilon>\epsilon_0\end{cases}$$

and T *follows* $\text{BesselK}(k,\Delta_f/\epsilon_0)$.

For the case $k=1$, the computation of the confidence level is comparatively straightforward because it only involves *Laplace* and *exponential distributions*, and does not require *gamma* and *BesselK-distribution.* Additionally, there is a closed form for the inverse of Equation 1 in Corollary 1. In this case, privacy at risk level is given by the following result.

Result for real-valued query. Privacy at risk level ϵ for a Laplace Mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_f}$ for a query $f:\mathcal{D}\to\mathbb{R}$ and a confidence level γ_1 is given by Equation 8.8.

$$\epsilon=\ln\left(\frac{1}{1-\gamma_1(1-e^{-\epsilon_0})}\right) \tag{8.8}$$

Comment on ϵ_0. For $k=1$, Figure 8.2 shows the plot of privacy at risk level ϵ versus confidence level γ_1 for the Laplace mechanism $\mathcal{L}_{\epsilon_0}^{1.0}$. As the value of ϵ_0 increases, the probability of Laplace mechanism generating higher value of noise reduces. Therefore, we observe that for a fixed confidence level, privacy at risk level increases with the value of ϵ_0. The same observation is made for $k>1$.

8.3.2 The case of implicit randomness

In this section, we study privacy at risk of the Laplace mechanism by considering effect of the implicit randomness induced by the data-generation distribution. We do not assume the knowledge of the sensitivity of the query. We study privacy at risk at a given confidence level, γ_2, on the sensitivity distribution computed using the data-generation distribution.

If we have access to an analytical form of the data-generation distribution and to the query, we could analytically derive the sensitivity distribution for the query. In general, we have access to the datasets, but not the data-generation distribution that generates them. Although we know the query, it rarely takes an analytical form suitable for the derivations needed. We, therefore, statistically estimate sensitivity by constructing an empirical distribution. We call the sensitivity value obtained for a given confidence level from the empirical cumulative distribution of sensitivity the *sampled sensitivity* (Definition 8). However, the value of sampled sensitivity is not the exact value of the sensitivity for a specified confidence level. In order to capture this additional uncertainty introduced by the estimation from the empirical sensitivity rather than the sensitivity distribution, we compute a lower bound on the accuracy of this estimation. This lower bound yields a probabilistic lower bound on the specified confidence level. We refer to it as *empirical privacy at risk*. For a given confidence level γ_2, we denote by $\hat{\gamma_2}$ the confidence level for empirical privacy at risk.

For the Laplace mechanism $\mathcal{L}_\epsilon^{\Delta_{S_f}}$ calibrated with sampled sensitivity Δ_{S_f} and privacy at risk level ϵ, we evaluate the confidence level $\hat{\gamma_2}$. We present the result in Theorem 4.

Theorem 4. *Analytical bound on the confidence level for empirical privacy at risk, $\hat{\gamma_2}$, for Laplace mechanism $\mathcal{L}_\epsilon^{\Delta_{S_f}}$ with privacy at risk level ϵ and sampled sensitivity Δ_{S_f} for a query $f : \mathcal{D} \to \mathbb{R}^k$ is*

$$\hat{\gamma_2} \geq \gamma_2(1 - 2e^{-2\rho^2 n}) \tag{8.9}$$

where n is the number of samples used for estimation of the sampled sensitivity and ρ is the accuracy parameter. γ_2 denotes the confidence level for the privacy at risk.

The accuracy parameter ρ represents the closeness between the empirical cumulative distribution of the sensitivity to the true cumulative distribution of the sensitivity. The lower the value of the accuracy, the closer is the empirical cumulative distribution to the true cumulative distribution. Figure 8.3 shows the plot of number of samples as a function of the confidence interval and the accuracy parameter. We observe that as the value of the accuracy reduces the number of samples in order to achieve the same confidence level exponentially increases.

Let us now present the details of the derivation of the analytical bound on the confidence level for empirical privacy risk in Theorem 5.

If the analytical form of the data-generation distribution is not known a priori, the empirical distribution of sensitivity can be estimated in two ways. The first way is to fit a known distribution on the available data and later use it to build an empirical distribution of the sensitivities. The second way is to sub-sample from a large dataset in order to build an empirical distribution of the sensitivities. In both of these ways, the empirical distribution of sensitivities captures the inherent randomness in the data-generation distribution. The first way suffers from the goodness of the fit of the known distribution to the available data. An ill-fit distribution does not reflect the true data-generation distribution and hence introduces errors in the sensitivity estimation. Since the second way involves subsampling, it is immune to this problem. The quality of sensitivity estimates obtained by sub-sampling the datasets depend on the availability of large population to sample from.

Let, $\mathcal{G}$ denotes the data-generation distribution that is realised using either of the two ways. We adopt the procedure in [RA17] to sample two neighbouring datasets with p data points each. We sample $p-1$ data points from $\mathcal{G}$ that are

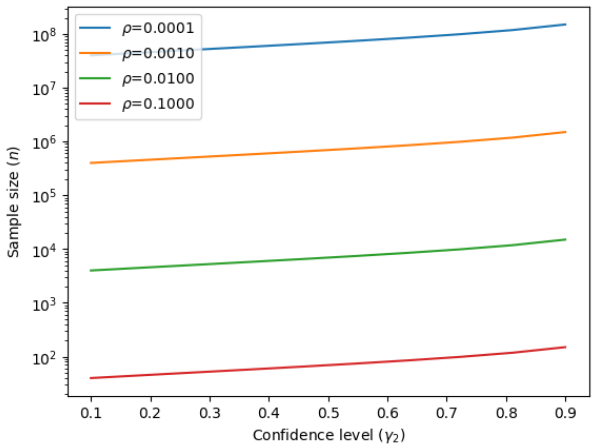


Figure 8.3: Number of samples n required to estimate the sampled sensitivity for varying confidence levels γ_2 for different accuracy parameters ρ.

common to both of these datasets. We sample two more data points from $\mathcal{G}$ and allot one data point to each of the two datasets.

Let, $S_f = \|f(x) - f(y)\|_1$ denotes the sensitivity random variable for a given query f, where x and y are two neighbouring datasets sampled from $\mathcal{G}$. Using n pairs of neighbouring datasets sampled from $\mathcal{G}$, we construct the empirical cumulative distribution, F_n, for the sensitivity random variable.

Definition 8 (**Sampled sensitivity**). *For a given query f and for a specified confidence level γ_2, sampled sensitivity, Δ_{S_f}, is defined as the value of sensitivity random variable that is estimated using its empirical cumulative distribution function, F_n, constructed using n pairs of neighbouring datasets sampled from the data-generation distribution $\mathcal{G}$.*

$$\Delta_{S_f} \triangleq F_n^{-1}(\gamma_2)$$

If we knew analytical form of the data generation distribution, we could analytically derive the cumulative distribution function of the sensitivity, F, and find the sensitivity of the query as $\Delta_f = F^{-1}(1)$. Therefore, in order to have the sampled sensitivity close to the sensitivity of the query, we require the empirical cumulative

distributions to be close to the cumulative distribution of the sensitivity. We use this insight to derive the analytical bound in the Theorem 4.

Lemma 10. *For Laplace Mechanism $\mathcal{L}_\epsilon^{\Delta_{S_f}}$ with sampled sensitivity Δ_{S_f} of a query $f : \mathcal{D} \to \mathbb{R}^k$ and for any output $Z \subseteq Range(\mathcal{L}_\epsilon^{\Delta_{S_f}})$,*

$$\mathbb{P}\left[\log\left|\frac{\mathbb{P}(\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}(x) \in Z)}{\mathbb{P}(\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}(y) \in Z)}\right| \leq \epsilon\right] \geq \gamma_2(1 - 2e^{-2\rho^2 n})$$

where n is the number of samples used to find sampled sensitivity and $\rho \in [0, 1]$ is a accuracy parameter. The outer probability is calculated with respect to support of the data-generation distribution $\mathcal{G}$.

Proof. Let, x and y be any two neighbouring datasets sampled from the data generating distribution $\mathcal{G}$. Let, Δ_{S_f} be the sampled sensitivity for query $f : \mathcal{D} \to \mathbb{R}^k$. Let, $\mathbb{P}_x(z)$ and $\mathbb{P}_y(z)$ denote the probabilities of getting the output z for Laplace mechanisms $\mathcal{L}_\epsilon^{\Delta_{S_f}}(x)$ and $\mathcal{L}_\epsilon^{\Delta_{S_f}}(y)$ respectively. For any point $z \in \mathbb{R}^k$ and $\epsilon \neq 0$,

$$\begin{aligned}
\frac{\mathbb{P}_x(z)}{\mathbb{P}_y(z)} &= \prod_{i=1}^{k} \frac{\exp\left(\frac{-\epsilon|f(x_i)-z_i|}{\Delta_{S_f}}\right)}{\exp\left(\frac{-\epsilon|f(y_i)-z_i|}{\Delta_{S_f}}\right)} \\
&= \exp\left(\frac{\epsilon \sum_{i=1}^{k}(|f(y_i) - z_i| - |f(x_i) - z_i|)}{\Delta_{S_f}}\right) \\
&\leq \exp\left(\frac{\epsilon \sum_{i=1}^{k}|f(y_i) - f(x_i)|}{\Delta_{S_f}}\right) \\
&= \exp\left(\frac{\epsilon\|f(y) - f(x)\|_1}{\Delta_{S_f}}\right) \qquad (8.10)
\end{aligned}$$

We used triangle inequality in the penultimate step.

Using the trick in [RA17], we define following events. Let, $B^{\Delta_{S_f}}$ denotes the set of pairs neighbouring dataset sampled from $\mathcal{G}$ for which the sensitivity random variable is upper bounded by Δ_{S_f}. Let, $C_\rho^{\Delta_{S_f}}$ denotes the set of sensitivity random variable values for which F_n deviates from the unknown cumulative distribution of S, F, at most by the accuracy value ρ. These events are defined

in Equation 8.11.

$$
\begin{aligned}
B^{\Delta_{S_f}} &\triangleq \{x, y \sim \mathcal{G} \text{ such that } \|f(y) - f(x)\|_1 \leq \Delta_{S_f}\} \\
C_\rho^{\Delta_{S_f}} &\triangleq \left\{ \sup_{\Delta} |F_S^n(\Delta) - F_S(\Delta)| \leq \rho \right\}
\end{aligned} \tag{8.11}
$$

$$
\begin{aligned}
\mathbb{P}(B^{\Delta_{S_f}}) &= \mathbb{P}(B^{\Delta_{S_f}} | C_\rho^{\Delta_{S_f}}) \mathbb{P}(C_\rho^{\Delta_{S_f}}) + \mathbb{P}(B^{\Delta_{S_f}} | \overline{C\rho^{\Delta_{S_f}}}) \mathbb{P}(\overline{C_\rho^{\Delta_{S_f}}}) \\
&\geq \mathbb{P}(B^{\Delta_{S_f}} | C_\rho^{\Delta_{S_f}}) \mathbb{P}(C_\rho^{\Delta_{S_f}}) \\
&= F_n(\Delta_{S_f}) \mathbb{P}(C_\rho^{\Delta_{S_f}}) \\
&\geq \gamma_2 \cdot (1 - 2e^{-2\rho^2 n})
\end{aligned} \tag{8.12}
$$

In the last step, we use the definition of the sampled sensitivity to get the value of the first term. The last term is obtained using DKW-inequality [M$^+$90] where the n denotes the number of samples used to build empirical distribution of the sensitivity, F_n.

From Equation 8.10, we understand that if $\|f(y) - f(x)\|_1$ is less than or equals to the sampled sensitivity then the Laplace mechanism $\mathcal{L}_\epsilon^{\Delta_{S_f}}$ satisfies ϵ-differential privacy. Equation 8.12 provides the lower bound on the probability of the event $\|f(y) - f(x)\|_1 \leq \Delta_{S_f}$. Thus, combining Equation 8.10 and Equation 8.12 completes the proof. □

8.3.3 The case of explicit and implicit randomness

In this section, we study the combined effect of both explicit randomness induced by the noise distribution and implicit randomness in the data-generation distribution respectively, on the privacy at risk. We do not assume the knowledge of the sensitivity of the query. We study privacy at risk at a given confidence level γ_3 on the joint support of the noise distribution and the data-generation distribution.

We estimate sensitivity using the empirical cumulative distribution of sensitivity. We construct the empirical distribution over the sensitivities using the sampling technique presented in Section 8.3.2. Since we use the sampled sensitivity (Definition 8) to calibrate the Laplace mechanism, we evaluate the confidence level for *empirical privacy at risk* $\hat{\gamma_3}$.

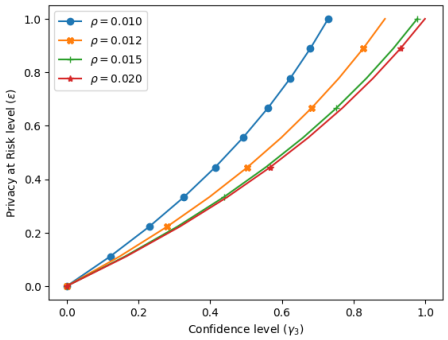
For the Laplace mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}$ calibrated with sampled sensitivity Δ_{S_f} and privacy level ϵ_0, we present the analytical bound on the confidence level $\hat{\gamma_3}$ in Theorem 5.

Theorem 5. *Analytical bound on the confidence level for empirical privacy at risk* $\hat{\gamma_3} \in [0, 1]$ *to achieve a Privacy at Risk level* $\epsilon > 0$ *for Laplace mechanism* $\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}$ *with sampled sensitivity* Δ_{S_f} *of a query* $f : \mathcal{D} \to \mathbb{R}^k$ *is*

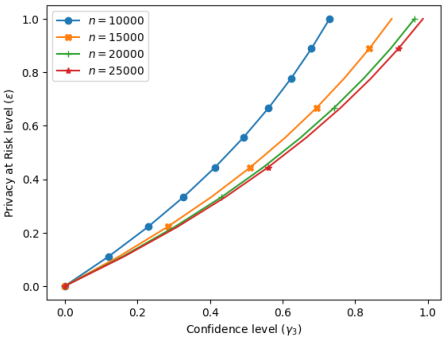
$$\hat{\gamma_3} \geq \gamma_3(1 - 2e^{-2\rho^2 n}) \tag{8.13}$$

where n *is the number of samples used for estimating the sensitivity,* ρ *is the accuracy parameter.* γ_3 *denotes the confidence level for the privacy at risk.*

The accuracy parameter ρ controls the closeness between the empirical cumulative distribution of the sensitivity to the true cumulative distribution of the sensitivity. Figure 8.4 shows the dependence of the accuracy parameter on the number of samples on the privacy at risk. In Figure 8.4a, we observe that the for a fixed number of samples and a privacy at risk level, the confidence level decreases with the value of accuracy parameter. For a fixed number of samples, smaller values of the accuracy parameter reduce the probability of similarity between the empirical cumulative distribution of sensitivity and the true cumulative distribution. Therefore, we observe the reduction in the confidence level for a fixed privacy at risk level. In Figure 8.4b, we observe that for a fixed value of accuracy parameter and a fixed level of privacy at risk, the confidence level increases with the number of samples. For a fixed value of the accuracy parameter, larger values of the sample size increase the probability of similarity between the empirical cumulative distribution of sensitivity and the true cumulative



(a) Privacy at risk level ϵ for varying confidence levels γ_3 for different accuracy parameters ρ.



(b) Privacy at risk level ϵ for varying confidence levels γ_3 for different sample sizes n.

Figure 8.4: Dependence of accuracy and the number of samples on privacy at risk for Laplace mechanism $\mathcal{L}_{1.0}^{\Delta_{S_f}}$. In the figure on the left hand side, we fix the number of samples to 10000. In the figure on the right hand side, we fix the accuracy parameter to 0.01.

distribution. Therefore, we observe the increase in the confidence level for a fixed privacy at risk level.

Now, we present the details of the derivation of the analytical bound on the confidence level for empirical privacy risk in Theorem 5.

In addition to the events defined in Equation 8.11, we define an additional event $A_{\epsilon_0}^{\Delta_{S_f}}$, defined in Equation 8.14, as a set of outputs of Laplace mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}$ that satisfy the constraint of ϵ-differential privacy for a specified privacy at risk level ϵ.

$$A_{\epsilon_0}^{\Delta_{S_f}} \triangleq \left\{ z \sim \mathcal{L}_{\epsilon_0}^{\Delta_{S_f}} \ : \ \log \left| \frac{\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}(x)}{\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}(y)} \right| \leq \epsilon, x, y \sim \mathcal{G} \right\} \tag{8.14}$$

Corollary 3.

$$\mathbb{P}(A_{\epsilon_0}^{\Delta_{S_f}} | B^{\Delta_{S_f}}) = \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \eta\epsilon_0)}$$

where T *follows* $\text{BesselK}(k, \Delta_{S_f}/\epsilon_0)$ *and* $\eta = \frac{\Delta_f}{\Delta_{S_f}}$.

Proof. We provide the sketch of the proof. Proof follows from the proof of Lemma 9. For a Laplace mechanism calibrated with the sampled sensitivity Δ_{S_f}

and privacy level ϵ_0, Equation 8.6 translates to,

$$\left(\frac{\epsilon_0}{\Delta_{S_f}}\sum_{i=1}^{k}|(|f(y_i)-z|-|f(x_i)-z|)|\right) \sim \text{Truncated BesselK}(k, \Delta_{S_f}/\epsilon_0, \Delta_f).$$

since, $\sum_{i=1}^{k}|(|f(y_i)-z|-|f(x_i)-z|)| \leq \Delta_f$. Using Lemma 8 and Equation 8.7,

$$\mathbb{P}(A_{\epsilon_0}^{\Delta_{S_f}}) = \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \eta\epsilon_0)}$$

where T follows BesselK$(k, \Delta_{S_f}/\epsilon_0)$ and $\eta = \frac{\Delta_f}{\Delta_{S_f}}$. □

In this section, we do not assume the knowledge of the sensitivity of the query. Using the empirical estimation presented in Section 8.3.2, if we choose the sampled sensitivity for confidence level $\gamma_2 = 1$, we obtain an approximation for η.

Lemma 11. *For a given value of accuracy parameter ρ,*

$$\frac{\Delta_f}{\Delta^*_{S_f}} = 1 + \mathcal{O}\left(\frac{\rho}{\Delta^*_{S_f}}\right)$$

where $\Delta^*_{S_f} = F_n^{-1}(1)$. $\mathcal{O}\left(\frac{\rho}{\Delta^*_{S_f}}\right)$ denotes order of $\frac{\rho}{\Delta^*_{S_f}}$, i.e., $\mathcal{O}\left(\frac{\rho}{\Delta^*_{S_f}}\right) = k\frac{\rho}{\Delta^*_{S_f}}$ for some $k \geq 1$.

Proof. For a given value of accuracy parameter ρ and any $\Delta > 0$,

$$F_n(\Delta) - F(\Delta) \leq \rho$$

Since above inequality is true for any value of Δ, let $\Delta = F^{-1}(1)$. Therefore,

$$\begin{aligned} F_n(F^{-1}(1)) - F(F^{-1}(1)) &\leq \rho \\ F_n(F^{-1}(1)) &\leq 1 + \rho \end{aligned} \tag{8.15}$$

Since a cumulative distribution function is 1-Lipschitz [PP02],

$$
\begin{aligned}
|F_n(F_n^{-1}(1)) - F_n(F^{-1}(1))| &\leq |F_n^{-1}(1) - F^{-1}(1)| \\
|F_n(F_n^{-1}(1)) - F_n(F^{-1}(1))| &\leq |\Delta^*_{S_f} - \Delta_f| \\
\rho &\leq \Delta_f - \Delta^*_{S_f} \\
1 + \frac{\rho}{\Delta^*_{S_f}} &\leq \frac{\Delta_f}{\Delta^*_{S_f}}
\end{aligned}
$$

where we used result from Equation 8.15 in step 3. Introducing $\mathcal{O}\left(\frac{\rho}{\Delta^*_{S_f}}\right)$ completes the proof. □

Lemma 12. *For Laplace Mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}$ with sampled sensitivity Δ_{S_f} of a query $f : \mathcal{D} \to \mathbb{R}^k$ and for any $Z \subseteq Range(\mathcal{L}_{\epsilon}^{\Delta_{S_f}})$,*

$$
\mathbb{P}\left[\log\left|\frac{\mathbb{P}(\mathcal{L}_{\epsilon_0}(x) \in Z)}{\mathbb{P}(\mathcal{L}_{\epsilon_0}(y) \in Z)}\right| \leq \epsilon\right] \geq \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \eta\epsilon_0)}\gamma_2(1 - 2e^{-2\rho^2 n})
$$

where n is the number of samples used to find sampled sensitivity, $\rho \in [0, 1]$ is a accuracy parameter and $\eta = \frac{\Delta_f}{\Delta_{S_f}}$. The outer probability is calculated with respect to support of the data-generation distribution $\mathcal{G}$.

Proof. The proof follows from the proof of Lemma 9 and Lemma 12. Consider,

$$
\begin{aligned}
\mathbb{P}(A_{\epsilon_0}^{\Delta_{S_f}}) &\geq \mathbb{P}(A_{\epsilon_0}^{\Delta_{S_f}}|B^{\Delta_{S_f}})\mathbb{P}(B^{\Delta_{S_f}}|C_\rho^{\Delta_{S_f}})\mathbb{P}(C_\rho^{\Delta_{S_f}}) \\
&\geq \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \eta\epsilon_0)} \cdot \gamma_2 \cdot (1 - 2e^{-2\rho^2 n}) \qquad (8.16)
\end{aligned}
$$

The first term in the final step of Equation 8.16 follows from the result in Corollary 3 where T follows BesselK$(k, \frac{\Delta_{S_f}}{\epsilon_0})$. It is the probability with which the Laplace mechanism $\mathcal{L}_{\epsilon_0}^{\Delta_{S_f}}$ satisfies ϵ-differential privacy for a given value of sampled sensitivity. Last two terms in the final step of Equation 8.16 follow from the result in Lemma 10. □

Probability of occurrence of event $A_{\epsilon_0}^{\Delta_{S_f}}$ calculated by accounting for both explicit and implicit sources of randomness gives the confidence level for privacy at risk

level ϵ. Thus, the proof of Lemma 12 completes the proof for Theorem 5.

Comparing the equations in Theorem 5 and Lemma 12, we observe that

$$\gamma_3 \triangleq \frac{\mathbb{P}(T \leq \epsilon)}{\mathbb{P}(T \leq \eta\epsilon_0)} \cdot \gamma_2 \tag{8.17}$$

The confidence level for privacy at risk, as defined in Equation 8.17, is free from the term that accounts for the accuracy of sampled estimate. If we know cumulative distribution of the sensitivity, we do not suffer from the uncertainty of introduced by sampling from the empirical distribution.

8.4 Applications of privacy at risk

In this section, we demonstrate the application of privacy at risk in solving two real-world decision-making problems.

8.4.1 Balancing utility and privacy

In this section, we empirically illustrate and discuss the steps that a data steward needs to take and the issues that she needs to consider in order to realise a required privacy at risk level ϵ for a confidence level γ when seeking to disclose the result of a query.

We consider a query that returns the parameter of a ridge regression [Mur12b] for an input dataset. It is a basic and widely used statistical analysis tool. We use the privacy-preserving mechanism presented by Ligett et al. [LNR$^+$17] for ridge regression. It is a Laplace mechanism that induces noise in the output parameters of the ridge regression. The authors provide a theoretical upper bound on the sensitivity of the ridge regression, which we refer as *sensitivity*, in the experiments.

Dataset. We conduct experiments on a subset of the 2000 US census dataset provided by Minnesota Population Center in its Integrated Public Use Microdata

Series [IPU09]. The census dataset consists of 1% sample of the original census data. It spans over 1.23 million households with records of 2.8 million people. The value of several attributes is not necessarily available for every household. We have therefore selected $212,605$ records, corresponding to the household heads, and 6 attributes, namely, *Age, Gender, Race, Marital Status, Education, Income*, whose values are available for the $212,605$ records.

In order to satisfy the constraint in the derivation of the sensitivity of ridge regression [LNR$^+$17], we, without loss of generality, normalise the dataset in the following way. We normalise *Income* attribute such that the values lie in $[0, 1]$. We normalise other attributes such that l_2 norm of each data point is unity.

Experimental Setup. All experiments are run on Linux machine with 12-core 3.60GHz Intel® Core i7™processor with 64GB memory. Python® 2.7.6 is used as the scripting language.

Result Analysis

We train ridge regression model to predict *Income* using other attributes as predictors. We split the dataset into the training dataset (80%) and testing dataset (20%). We compute the *root mean squared error (RMSE)* of ridge regression, trained on the training data with regularisation parameter set to 0.01, on the testing dataset. We use it as the metric of *utility loss*. Smaller the value of RMSE, smaller the loss in utility. For a given value of privacy at risk level, we compute 50 runs of an experiment of a differentially private ridge regression and report the means over the 50 runs of the experiment.

Let us now provide illustrative experiments under the three different cases. In every scenario, the data steward is given a privacy at risk level ϵ and the confidence level γ and wants to disclose the parameters of a ridge regression model that she trains on the census dataset. She needs to calibrate the Laplace mechanism to achieve the privacy at risk required the ridge regression query.

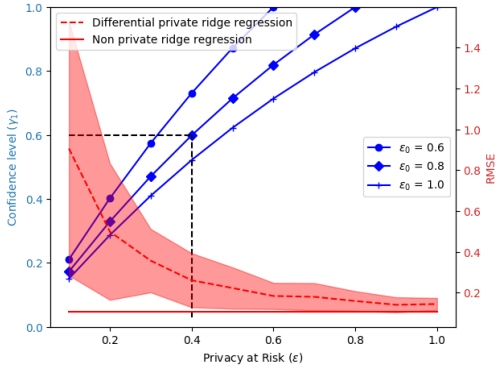


Figure 8.5: Utility, measured by RMSE (right y-axis), and privacy at risk level ϵ for Laplace mechanism (left y-axis) for varying confidence levels γ_1.

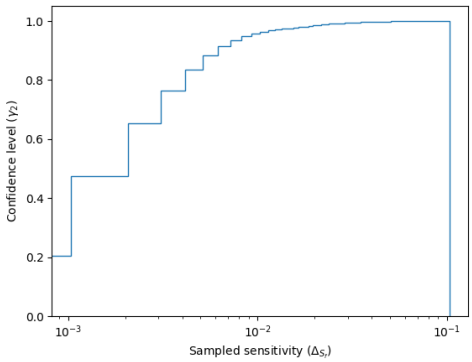


Figure 8.6: Empirical cumulative distribution of the sensitivities of ridge regression queries constructed using 15000 samples of neighboring datasets.

The Case of Explicit Randomness (cf. Section 8.3.1) In this scenario, the data steward knows the sensitivity for the ridge regression. She needs to compute the privacy level, ϵ_0, to calibrate the Laplace mechanism. She uses Equation 1 that links the desired privacy at risk level ϵ, the confidence level γ_1 and the privacy level of noise ϵ_0. Specifically, for given ϵ and γ_1, she computes ϵ_0 by solving the equation:

$$\gamma_1 \mathbb{P}(T \leq \epsilon_0) - \mathbb{P}(T \leq \epsilon) = 0.$$

Since the equation does not give an analytical formula for ϵ_0, the data steward uses a root finding algorithm such as Newton-Raphson method [Pre07] to solve the above equation. For instance, if she needs to achieve a privacy at risk level $\epsilon = 0.4$ with confidence level $\gamma_1 = 0.6$, she can substitute these values in the above equation and solve the equation to get the privacy level of noise $\epsilon_0 = 0.8$.

Figure 8.5 shows the variation of privacy at risk level ϵ and confidence level γ_1. It also depicts the variation of utility loss for different privacy at risk levels in Figure 8.5.

In accordance to the data steward's problem, if she needs to achieve a privacy at risk level $\epsilon = 0.4$ with confidence level $\gamma_1 = 0.6$, she obtains the privacy level of

noise to be $\epsilon_0 = 0.8$. Additionally, we observe that the choice of privacy level 0.8 instead of 0.4 to calibrate the Laplace mechanism gives lower utility loss for the data steward. This is the benefit drawn from the risk taken under the control of privacy at risk.

Thus, she uses privacy level ϵ_0 and the sensitivity of the function to calibrate Laplace mechanism.

The Case of Implicit Randomness (cf. Section 8.3.2) In this scenario, the data steward does not know the sensitivity of ridge regression. She assesses that she can afford to sample at most n times from the population dataset. She understands the effect of the uncertainty introduced by the statistical estimation of the sensitivity. Therefore, she uses the confidence level for empirical privacy at risk $\hat{\gamma_2}$.

Given the value of n, she chooses the value of the accuracy parameter using Figure 8.3. For instance, if the number of samples that she can draw is 10^4, she chooses the value of the accuracy parameter $\rho = 0.01$. Next, she uses Equation 8.18 to determine the value of probabilistic tolerance, α, for the sample size n. For instance, if the data steward is not allowed to access more than $15,000$ samples, for the accuracy of 0.01 the probabilistic tolerance is 0.9.

$$\alpha = 1 - 2e^{(-2\rho^2 n)} \tag{8.18}$$

She constructs an empirical cumulative distribution over the sensitivities as described in Section 8.3.2. Such an empirical cumulative distribution is shown in Figure 8.6. Using the computed probabilistic tolerance and desired confidence level $\hat{\gamma_2}$, she uses equation in Theorem 4 to determine γ_2. She computes the sampled sensitivity using the empirical distribution function and the confidence level for privacy Δ_{S_f} at risk γ_2. For instance, using the empirical cumulative distribution in Figure 8.6 she calculates the value of the sampled sensitivity to be approximately 0.001 for $\gamma_2 = 0.4$ and approximately 0.01 for $\gamma_2 = 0.85$

Thus, she uses privacy level ϵ, sets the number of samples to be n and computes the sampled sensitivity Δ_{S_f} to calibrate the Laplace mechanism.

The Case of Explicit and Implicit Randomness (cf. Section 8.3.3) In this scenario, the data steward does not know the sensitivity of ridge regression. She is not allowed to sample more than n times from a population dataset. For a given confidence level γ_2 and the privacy at risk ϵ, she calibrates the Laplace mechanism using illustration for Section 8.3.3. The privacy level in this calibration yields utility loss that is more than her requirement. Therefore, she wants to re-calibrate the Laplace mechanism in order to reduce utility loss.

For the re-calibration, the data steward uses privacy level of the pre-calibrated Laplace mechanism, i.e. ϵ, as the privacy at risk level and she provides a new confidence level for empirical privacy at risk $\hat{\gamma_3}$. Using Equation 8.17 and Equation 8.15, she calculates:

$$\hat{\gamma_3}\mathbb{P}(T \leq \eta\epsilon_0) - \alpha\gamma_2\ \mathbb{P}(T \leq \epsilon) = 0$$

She solves such an equation for ϵ_0 using the root finding technique such as Newton-Raphson method [Pre07]. For instance, if she needs to achieve a privacy at risk level $\epsilon = 0.4$ with confidence levels $\hat{\gamma_3} = 0.9$ and $\gamma_2 = 0.9$, she can substitute these values and the values of tolerance parameter and sampled sensitivity, as used in the previous experiments, in the above equation. Then, solving the equation leads to the privacy level of noise $\epsilon_0 = 0.8$.

Thus, she re-calibrates the Laplace mechanism with privacy level ϵ_0, sets the number of samples to be n and sampled sensitivity Δ_{S_f}.

8.4.2 Minimising compensation budget

We use multiples services such as messaging, mailing, media streaming on our mobiles and computers on daily basis. Many service providers collect users' data

to enhance user experience. In order to avoid misuse of this data, we require a legal framework that not only limits the use of the collected data but also proposes reparative measures in case of a data leak. General Data Protection Regulation (GDPR) [gdp16] is such a legal framework. Starting from May 2018, every business entity that holds or processes data of EU citizens, irrespective of the geographical location of the entity itself, must comply with GDPR.

Section 82 in GDPR states that any person who suffers from material or non-material damage as a result of a personal data breach has the right to demand compensation from the data processor. Therefore, every GDPR compliant business entity that either holds or processes personal data needs to secure a certain budget in the worst case scenario of the personal data breach. In order to reduce the risk of such an unfortunate event, the business entity may use privacy-preserving mechanisms that provide provable privacy guarantees while publishing their results. In order to calculate the compensation budget for a business entity, we devise cost models that map the privacy guarantees provided by differential privacy and privacy at risk to monetary costs. The discussions demonstrate the usefulness of privacy at risk over differential privacy to reduce expenditure.

Cost model for differential privacy. Let E be the compensation budget that a business entity has to pay to every stakeholder in case of a personal data breach when the data is processed without any provable privacy guarantees. Let E_ϵ^{dp} be the compensation budget that a business entity has to pay to every stakeholder in case of a personal data breach when the data is processed with privacy guarantees in terms of ϵ-differential privacy.

Privacy level, ϵ, in ϵ-differential privacy is the quantifier of indistinguishability of the outputs of a privacy-preserving mechanism when two neighbouring datasets are provided as inputs. When the privacy level is zero, the privacy-preserving mechanism outputs all results with equal probability. The indistinguishability reduces with increase in the privacy level. Thus, privacy level of zero bears the

lowest risk of personal data breach and the risk increases with the privacy level. E_ϵ^{dp} needs to be commensurate to such a risk and, therefore, it needs to satisfy the following constraints.

- For all $\epsilon \in \mathbb{R}^{\geq 0}$, $E_\epsilon^{dp} \leq E$.
- E_ϵ^{dp} is a monotonically increasing function of ϵ.
- As $\epsilon \to 0$, $E_\epsilon^{dp} \to E_{min}$ where E_{min} is the unavoidable cost that business entity might need to pay in case of personal data breach even after the privacy measures are employed.
- As $\epsilon \to \infty$, $E_\epsilon^{dp} \to E$.

There are various functions that satisfy these constraints. In absence of any further constraints, we model E_ϵ^{dp} as defined in Equation 8.19.

$$E_\epsilon^{dp} \triangleq E_{min} + Ee^{-\frac{c}{\epsilon}} \tag{8.19}$$

E_ϵ^{dp} has two parameters, namely $c > 0$ and $E_{min} \geq 0$. c controls the rate of change in the cost as the privacy level changes and E_{min} is a privacy level independent bias. For this study, we use a simplified model with $c = 1$ and $E_{min} = 0$.

Cost model for privacy at risk. Let, $E_{\epsilon_0}^{par}(\epsilon, \gamma)$ be the compensation that a business entity has to pay to every stakeholder in case of a personal data breach when the data is processed with an ϵ_0-differentially private privacy-preserving mechanism but a (ϵ, γ)-privacy at risk guarantee is provided.

In Section 8.2, we define privacy at risk as a refinement over the existing privacy definition of differential privacy. Privacy at risk provides a quantifiable probabilistic privacy guarantees for any differentially private privacy-preserving mechanism.

Any ϵ_0-differentially private privacy-preserving mechanism that satisfies (ϵ, γ)-privacy at risk is ϵ-differentially private with probability at most γ and ϵ_0-differentially private with probability $(1 - \gamma)$. Thus, we calculate $E_{\epsilon_0}^{par}$ using

Equation 8.20.

$$E^{par}_{\epsilon_0}(\epsilon, \gamma) \triangleq \gamma E^{dp}_{\epsilon} + (1 - \gamma) E^{dp}_{\epsilon_0} \tag{8.20}$$

Existence of minimum compensation budget. We want to find the privacy at risk level, say ϵ_{min}, that yields the lowest compensation budget. We do that by minimising Equation 8.20 with respect to ϵ.

Lemma 13. *$E^{par}_{\epsilon_0}(\epsilon, \gamma)$ is a convex function of ϵ.*

By Lemma 13, there exists a unique ϵ_{min} that minimises the compensation budget for a given privacy level ϵ_0. Since the confidence level γ in Equation 8.20 is a function of privacy at risk level ϵ, analytical calculation of ϵ_{min} is not possible in the most general case. When the output of the query is a real number, we derive the analytic form (Equation 8.8) to compute the confidence level under the consideration of explicit randomness. In such a case, ϵ_{min} is calculated by differentiating Eqaution 8.20 with respect to ϵ and equating it to zero. It gives us Equation 8.21 that we solve using any root finding technique such as Newton-Raphson method [Pre07] to compute ϵ_{min}.

$$\frac{1}{\epsilon} - \ln\left(1 - \frac{1 - e^{\epsilon}}{\epsilon^2}\right) = \frac{1}{\epsilon_0} \tag{8.21}$$

Illustration. Suppose that the health centre in a university that complies to GDPR publishes statistics of its staff health checkup, such as obesity statistics, twice in a year. In January 2018, the health centre publishes that 34 out of 99 faculty members suffer from obesity. In July 2018, the health centre publishes that 35 out of 100 faculty members suffer from obesity. An intruder, perhaps an analyst working for an insurance company, checks the staff listings in January 2018 and July 2018, which are publicly available on website of the university. The intruder does not find any change other than the recruitment of John Doe in April 2018. Thus, with high probability, the intruder deduces that John Doe suffers from obesity. In order to avoid such a privacy breach, the health centre decides to publish the results using the Laplace mechanism. In this case, the

Laplace mechanism operates on the count query.

In order to control the amount of noise, the health centre needs to appropriately set the privacy level. Suppose that the health centre decides to use the expected mean absolute error, defined in Equation 8.22, as the measure of *effectiveness* for the Laplace mechanism.

$$\mathbb{E}\left[|\mathcal{L}_\epsilon^1(x) - f(x)|\right] = \frac{1}{\epsilon} \tag{8.22}$$

Equation 8.22 makes use of the fact that the sensitivity of the count query is one. Suppose that the health centre requires the expected mean absolute error of at most two in order to maintain the quality of the published statistics. In this case, the privacy level has to be at least 0.5.

In order to compute the budget, the health centre requires an estimate of E. Research in the health-care domain shows that the incremental cost of premiums for the health insurance with morbid obesity ranges between \$5467 to \$5530 [MBO+12]. A staff member may expect up to \$5500 rise in the premiums if the data gets leaked. Therefore, with reference to the research in [MBO+12], the health centre takes \$5500 as an estimate of E. For the staff size of 100 and the privacy level 0.5, the health centre uses Equation 8.19 in its simplified setting to compute the total budget of \$74434.40.

Is it possible to reduce this budget without degrading the effectiveness of the Laplace mechanism? In Section 8.2, we study privacy at risk of an ϵ_0-differentially private Laplace mechanism. Under the consideration of the explicit randomness introduced by the Laplace noise distribution, we show that ϵ_0-differentially private Laplace mechanism satisfies (ϵ, γ)-privacy at risk. γ is computed using the formula in Theorem 3. In Figure 8.7, we plot the budget for various privacy at risk levels. We observe that the privacy at risk level 0.274, which is same as ϵ_{min} computed by solving Equation 8.21, yields the lowest compensation budget of \$37805.86. Thus, by using privacy at risk, the health centre is able to save \$36628.532 without

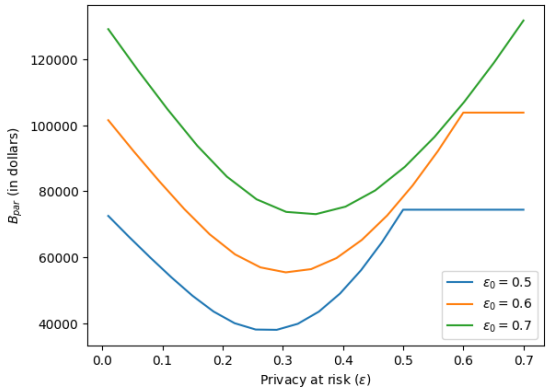


Figure 8.7: Variation in the budget for Laplace mechanism $\mathcal{L}^1_{\epsilon_0}$ under privacy at risk considering explicit randomness in the Laplace mechanism for the illustration in Section 8.4.2.

sacrificing the quality of the published results.

Bounds on the privacy at risk level. For a fixed budget, say B, rearrangement of Equation 8.20 gives us an upper bound on the privacy at risk level ϵ. We use the cost model with $c = 1$ and $E_{min} = 0$ to derive the upper bound. If we have a maximum permissible expected mean absolute error T, we use Equation 8.22 to obtain a lower bound on the privacy at risk level. Equation 8.23 illustrates the upper and lower bounds that dictate the permissible range of ϵ that a data publisher can promise depending on the budget and the permissible error constraints.

$$\frac{1}{T} \leq \epsilon \leq \left[\ln \left(\frac{\gamma E}{B - (1-\gamma) E^{dp}_{\epsilon_0}} \right) \right]^{-1} \tag{8.23}$$

Thus, the privacy at risk is constrained by the effectiveness requirement from below and by the monetary budget from above. Hsu et al. [HGH+14] calculate upper and lower bound on the privacy level in the differential privacy. They use a different cost model owing to the scenario of research study that compensates its participants for their data and releases the results in a differentially private manner. Their cost model is different than our GDPR inspired modelling.

8.5 Related Work

Researchers have proposed different privacy-preserving mechanisms [DR+14] to make different queries differentially private. These mechanisms can be broadly classified into two categories. In one category, the mechanisms explicitly add calibrated noise, such as Laplace noise [DMNS06b] or Gaussian noise [DR+14], to the outputs of the query. In the other category, the mechanisms [CMS11, ZZX+12, ACC12, HRW13b] alter the query function so that the modified function becomes differentially private. The latter category is referred to as the *functional mechanism*. Privacy-preserving mechanisms in both of these categories perturb the original output of the query and make it difficult for a malicious data analyst to recover the original output of the query. These mechanisms induce randomness using the explicit noise distribution. Calibration of these mechanisms require the knowledge of the sensitivity of the query. Nissim et al. [NRS07] consider the implicit randomness in the data-generation distribution to compute an estimate of the sensitivity. The authors propose the smooth sensitivity function that is an envelope over the local sensitivities for all individual datasets. Local sensitivity of a dataset is the maximum change in the value of the query over all of its neighboring datasets. In general, it is not easy to analytically estimate the smooth sensitivity function for a general query. Rubinstein et al. [RA17] also study the inherent randomness in the data-generation algorithm. They do not use the local sensitivity. We adopt their approach of sampling the sensitivity from the empirical distribution of the sensitivity. They use order statistics to choose a particular value of the sensitivity. We use the confidence level, which provides a mediation tool for business entities to assess the actual business risks, on the sensitivity distribution to estimate the sensitivity.

In order to account for both sources of randomness, refinements of ϵ-differential privacy are proposed in order to bound the probability of occurrence of worst case scenarios.

Probabilistic differential privacy [MKA+08] considers upper bounds of the worst

case privacy loss for corresponding confidence levels on the noise distribution. Definition of probabilistic differential privacy incorporates the explicit randomness induced by the noise distribution and bounds the probability over the space of noisy outputs to satisfy the ϵ-differential privacy definition. Concentrated differential privacy [DR16] considers the expected values of the privacy loss for the corresponding confidence levels on the noise distribution. Definition of concentrated differential privacy incorporates the explicit randomness induced by the noise distribution but considering only the expected value of privacy loss satisfying ϵ-differential privacy definition instead of using the confidence levels limits its scope. Random differential privacy [HRW13b] considers the privacy loss for corresponding confidence levels on the implicit randomness in the data-generation distribution. Definition of random differential privacy incorporates the implicit randomness induced by the data-generation distribution and bounds the probability over the space of datasets generated from the given distribution to satisfy the ϵ-differential privacy definition. Approximate differential privacy [DKM$^+$06] adds a constant bias to the privacy guarantee provided by the differential privacy. It is not a probabilistic refinement of the differential privacy.

In this work, we consider the widely used Laplace mechanism [DMNS06b]. The Laplace mechanism adds Laplacian noise [Jor00] to the query output. The noise is calibrated by sensitivity of query and the desired privacy level [XWG11, ZZX$^+$12, ACC12].

We find works in the domain of game theoretic methods that propose methods for rational agents to evaluate the cost of differential privacy [GR15, CCK$^+$16]. Our approach is inspired by the approach by Hsu et al. [HGH$^+$14]. They model the cost under a scenario of a research study wherein the participants are reimbursed for their participation. Our cost modelling is driven by the scenario of securing a compensation budget in compliance with GDPR. Our requirement differs from the requirements for the scenario in [HGH$^+$14]. In our case, there is no monetary incentive for participants to share their data.

8.6 Discussion

We propose privacy at risk that quantifies the probability with which a privacy-preserving mechanism satisfies differential privacy for a specified privacy level. The probabilistic quantification depends on two sources of randomness, the explicit randomness induced by the noise distribution and the implicit randomness induced by the data-generation distribution, and the coupling between the two. We instantiate privacy at risk for the Laplace mechanism.

We demonstrate the applicability of privacy at risk in two decision-making problems. Firstly, we illustrate the use of privacy at risk by a data steward to balance the privacy-utility trade-off. Secondly, we propose a cost model that bridges the gap between the privacy level and the compensation budget estimated by a GDPR compliant business entity. We show the existence of a privacy level that yields the minimum compensation budget under the cost model of privacy at risk. Thus, privacy at risk not only quantifies privacy level for a given privacy-preserving mechanism but also facilitates minimisation of monetary risk under the proposed cost model.

Privacy at risk may be fully analytically computed in cases where the data-generation, or the sensitivity distribution, the noise distribution and the query are analytically known and take convenient forms. We are now looking at such convenient but realistic cases.

Publication

Work in this chapter is part of the following publication.

- Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Differential privacy at risk. Submitted in *Journal of Confidentiality and Privacy (Under review)*

Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd.

Part IV

Conclusion

Chapter 9

Conclusion and future works

In this work, we address the concerns of the risk of breach of privacy while releasing datasets as well as machine learning models trained on the datasets.

In the first part of this thesis, we discuss the privacy risk of releasing datasets. We use generative models to synthesise the datasets catered to the desired requirements. Synthetic datasets do not contain any point that has correspondence to any data-point in the real-world dataset. Therefore, synthetic datasets provide immunity from the privacy risk due to identity disclosure. We propose two generative models to synthesise two different kinds of datasets. We show their effectiveness on the real-world datasets.

In the second part of this thesis, we discuss the privacy risk of releasing machine learning models. We study two classes of machine learning models namely parametric and non-parametric models under their release. We use the functional mechanism to provide differential privacy guarantees for the direct release of the parametric models. We use the functional perturbation to provide privacy guarantee for the release of non-parametric models as a service. We studied the challenges of using differential privacy as a privacy definition in a real-world setting. We propose privacy at risk as a probabilistic refinement of differential privacy. Privacy at risk provides confidence levels on the differential privacy levels. We instantiate privacy at risk for Laplace mechanism and provide. We also propose a cost model that helps in bridging the abstract privacy level to the compensation budget that a GDPR compliant company assigns in case of

breach of privacy. We apply the proposed cost model to privacy at risk and show how an organisation is able to save money on compensation budget without compromising the privacy guarantees provided to its customers.

Future works

Generative Adversarial Networks (GANs) have been shown to be very effective at reconstructing as well as generating data [SWL18]. In future, we would like to use GANs to generate synthetic datasets. Synthetic data generation provides a pragmatic way for organisations to share sensitive data among them. But *vanilla* synthetic dataset generation does not guarantee that the generated dataset does not contain any real data-points. Use of overfitted machine learning models to generate synthetic dataset may lead to leakage of information in the training datasets. Thus, synthetic dataset generation without any privacy guarantee may lead to attribute disclosure. We are looking at possibilities to provide differential privacy guarantees for the data generating machine learning algorithms.

Differential privacy for machine learning models has become an active research topic in the last decade. There has been recent work [WLF16] on the connections between differential privacy and generalisability of machine learning models. There is a possibility to tune regularisation constant to achieve desired level of differential privacy. Such a privacy-preserving mechanism would lead to stronger privacy guarantees without sacrificing on the utility. Distributed machine learning has recently gathered attention in the industry. Organisations want to build machine learning models without actually sharing data with each other. Providing privacy guarantees for distributed machine learning opens an interesting research problem. It may involve the consideration of security protocols that are needed while sharing model parameters among the different parties.

We proposed privacy at risk as a probabilistic privacy definition that extends ϵ-differential privacy. In order to instantiate privacy at risk for privacy-preserving mechanisms that satisfy (ϵ, δ)-differential privacy, we need to further incorporate

slack parameter δ into the definition of privacy at risk. We want to study the relationship between privacy at risk and Rényi differential privacy [Mir17] and concentrated differential privacy [DR16].

In the end, we understand that the problem of data privacy is multifaceted. On the one hand, it is a research topic for computer scientist to design effective privacy-preserving mechanisms that give provable privacy guarantees. On the other hand, it is equally a research problem for public policy makers to amend the existing laws to incorporate new privacy definitions [BDR18]. Such an interaction necessitates privacy guarantees that are not only quantifiable but also interpretable. Cost model for the privacy is one of the many solutions to improve interpretability. There is a need to design the experiments that empirically validate abstract privacy guarantees.

References

[Aal93] Odd O Aalen. Further results on the non-parametric linear regression model in survival analysis. *Statistics in medicine*, 12(17):1569–1588, 1993.

[Abo18] John M Abowd. The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867. ACM, 2018.

[ACC12] Gergely Acs, Claude Castelluccia, and Rui Chen. Differentially private histogram publishing through lossy compression. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1–10. IEEE, 2012.

[ACG+16] 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*, pages 308–318. ACM, 2016.

[AD10] RA Askey and AB Olde Daalhuis. Generalized hypergeometric functions and meijer g-function. *NIST handbook of mathematical functions*, pages 403–418, 2010.

[AMS+15] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.

References

[AR17] Francesco AldÃă and Benjamin Rubinstein. The bernstein mechanism: Function release under differential privacy. 2017.

[BDR18] Steven M Bellovin, Preetam K Dutta, and Nathan Reitinger. Privacy and synthetic datasets. *Stanford Technology Law Review, Forthcoming*, 2018.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[BGV92] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[BSF94] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[BTS17] Matej Balog, Ilya Tolstikhin, and Bernhard Schölkopf. Differentially private database release via kernel mean embeddings. *arXiv preprint arXiv:1710.01641*, 2017.

[CCA+09] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[CCK+16] Yiling Chen, Stephen Chong, Ian A Kash, Tal Moran, and Salil Vadhan. Truthful mechanisms for agents that value privacy. *ACM*

Transactions on Economics and Computation (TEAC), 4(3):13, 2016.

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Cho15] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[CM09] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, pages 289–296, 2009.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[CR10] Gregory Caiola and Jerome P Reiter. Random forests for generating partially synthetic, categorical data. *Trans. Data Privacy*, 3(1):27–42, 2010.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[CV05] Guangqing Chi and Paul Voss. Migration decision-making: a hierarchical regression approach. *Journal of Regional Analysis and Policy*, 35(2), 2005.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[CVSG13] Yoon-Sik Cho, Greg Ver Steeg, and Aram Galstyan. Socially relevant venue clustering from check-in data. In *11th Workshop on Mining*

and Learning with Graphs, MLG–2013, 2013.

[Dal77] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15(429-444):2–1, 1977.

[DB16] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[DBR08] Jörg Drechsler, Stefan Bender, and Susanne Rässler. Comparing fully and partially synthetic datasets for statistical disclosure control in the german iab establishment panel. *Trans. Data Privacy*, 1(3):105–130, 2008.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Eurocrypt*, volume 4004, pages 486–503. Springer, 2006.

[DKT17] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[DMCC13] Andrea De Montis, Simone Caschili, and Alessandro Chessa. Commuter networks and community detection: a method for planning sub regional areas. *The European Physical Journal Special Topics*, 215(1):75–91, 2013.

[DMNS06a] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[DMNS06b] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. *Calibrating Noise to Sensitivity in Private Data Analysis*, pages 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

References

[DN12] Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572. Association for Computational Linguistics, 2012.

[Dod02] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc., 2002.

[DR08] Jörg Drechsler and Jerome P Reiter. Accounting for intruder uncertainty due to sampling when estimating identification disclosure risks in partially synthetic data. In *International Conference on Privacy in Statistical Databases*, pages 227–238. Springer, 2008.

[DR11] Jörg Drechsler and Jerome P Reiter. An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. *Computational Statistics & Data Analysis*, 55(12):3232–3243, 2011.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[DR16] Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

[Dre10] Jörg Drechsler. Using support vector machines for generating synthetic datasets. In *Privacy in Statistical Databases*, number 6344, pages 148–161. Springer, 2010.

[Dre11] Jörg Drechsler. *Synthetic datasets for statistical disclosure control: theory and implementation*, volume 201. Springer Science & Business Media, 2011.

References

[DSSU17] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

[Dwo06] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12. Springer Berlin Heidelberg, 2006.

[DY+14] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM, 2015.

[FLJ+14] M Fredrikson, E Lantz, S Jha, S Lin, D Page, and T Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *Proceedings of the USENIX Security Symposium. UNIX Security Symposium*, volume 2014, pages 17–32. NIH Public Access, 2014.

[FWB10] Brian Ferris, Kari Watkins, and Alan Borning. Onebusaway: results from providing real-time arrival information for public transit. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1807–1816. ACM, 2010.

[GAP18] Simson L Garfinkel, John M Abowd, and Sarah Powazek. Issues encountered deploying differential privacy. *arXiv preprint arXiv:1809.02201*, 2018.

[gdp16] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)(text with eea relevance). *Official Journal of the European Union*, L(119):1–88, 2016.

[GJM13] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.

[GL15] Huiji Gao and Huan Liu. Mining human mobility in location-based social networks. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 7(2):1–115, 2015.

[GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[GP17] Ian Goodfeloow and Nicolas Papernot. Is attacking machine learning easier than defending it?, 2017.

[GR15] Arpita Ghosh and Aaron Roth. Selling privacy at auction. *Games and Economic Behavior*, 91:334–346, 2015.

[Gra13] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[GS04] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[GVS+16] Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. Contextual lstm (clstm) models for large scale nlp

tasks. *arXiv preprint arXiv:1602.06291*, 2016.

[HGH+14] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*, pages 398–410. IEEE, 2014.

[HJE13] Bo Hu, Mohsin Jamali, and Martin Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1073–1078. IEEE, 2013.

[HK70] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[How67] Nancy Howell. Data from a partial census of the !kung san, dobe., 1967.

[HRRS11] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 196. John Wiley & Sons, 2011.

[HRW12] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Random differential privacy. *Journal of Privacy and Confidentiality*, 4(2):43–59, 2012.

[HRW13a] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research (JMLR)*, 14(Feb):703–727, 2013.

[HRW13b] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(Feb):703–727, 2013.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HS06] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[HSR+08] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the forty-second ACM Symposium on Theory of Computing (STOC)*. ACM, 2010.

[Int78] Michael D Intriligator. Econometric models, techniques, and applications. Technical report, Prentice-Hall Englewood Cliffs, NJ, 1978.

[IPU09] Minnesota population center. integrated public use microdata series âĂŞ international: Version 5.0. `https://international.ipums.org.`, 2009.

[ISVB05] Ganesh Iyer, David Soberman, and J Miguel Villas-Boas. The targeting of advertising. *Marketing Science*, 24(3):461–476, 2005.

[ITA+16] Kazuki Irie, Zoltán Tüske, Tamer Alkhouli, Ralf Schlüter, and Hermann Ney. Lstm, gru, highway and a bit of attention: an empirical overview for language modeling in speech recognition. *in INTERSPEECH*, 2016.

[Iye02] Vijay S Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the eighth ACM SIGKDD international conference*

on Knowledge discovery and data mining, pages 279–288. ACM, 2002.

[Jor00] Philippe Jorion. Value at risk: The new benchmark for managing financial risk. 01 2000.

[JT13] Prateek Jain and Abhradeep Thakurta. Differentially private learning with kernels. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 118–126. PMLR, 2013.

[JTC12] Kenneth Joseph, Chun How Tan, and Kathleen M Carley. Beyond local, categories and friends: clustering foursquare users with latent topics. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 919–926. ACM, 2012.

[JZS15] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *in the Proceedings of The 32nd ICML*, pages 2342–2350, 2015.

[KB14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KG06] Daniel Kifer and Johannes Gehrke. Injecting utility into anonymized datasets. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 217–228. ACM, 2006.

[KKO+06] Alan F Karr, Christine N Kohnen, Anna Oganian, Jerome P Reiter, and Ashish P Sanil. A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician*, 60(3):224–232, 2006.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KST12] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1, 2012.

[LBE15] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[LC11a] Neal Lathia and Licia Capra. How smart is your smartcard?: measuring travel behaviours, perceptions, and incentives. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 291–300. ACM, 2011.

[LC11b] Neal Lathia and Licia Capra. Mining mobility data to minimise travellers' spending on public transport. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1181–1189. ACM, 2011.

[LC11c] Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*, pages 325–340. Springer, 2011.

[LDR06] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 25–25. IEEE, 2006.

[LEHC15] Yu Liu, Martin Ester, Bo Hu, and David W Cheung. Spatio-temporal topic models for check-in data. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 889–894. IEEE, 2015.

[Lei11] Jing Lei. Differentially private m-estimators. In *Advances in Neural Information Processing Systems*, pages 361–369, 2011.

[Lic13] M. Lichman. UCI machine learning repository, 2013.

[Lin04] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL-04 Workshop*, volume 8, 2004.

[Lit93] Roderick JA Little. Statistical analysis of masked data. *Journal of Official statistics*, 9(2):407, 1993.

[LJJ12] Xuelian Long, Lei Jin, and James Joshi. Exploring trajectory-driven local geographic topics in foursquare. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 927–934. ACM, 2012.

[LLJ15] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

[LLV07] 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*, pages 106–115. IEEE, 2007.

[LNR+17] Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Steven Z Wu. Accuracy first: Selecting a differential privacy level for accuracy constrained erm. In *Advances in Neural Information Processing Systems*, pages 2563–2573, 2017.

[LQC12] Neal Lathia, Daniele Quercia, and Jon Crowcroft. The hidden image of the city: sensing community well-being from urban mobility. In *Pervasive computing*, pages 91–98. Springer, 2012.

[M+90] Pascal Massart et al. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The annals of Probability*, 18(3):1269–1283, 1990.

[MBO+12] James P Moriarty, Megan E Branda, Kerry D Olsen, Nilay D Shah, Bijan J Borah, Amy E Wagie, Jason S Egginton, and James M

Naessens. The effects of incremental costs of smoking and obesity on health care costs among adults: a 7-year longitudinal study. *Journal of Occupational and Environmental Medicine*, 54(3):286–291, 2012.

[MGKV06] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. \ell-diversity: Privacy beyond\kappa-anonymity. In *null*, page 24. IEEE, 2006.

[Mir17] Ilya Mironov. Renyi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE, 2017.

[MKA+08] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. Privacy: Theory meets practice on the map. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 277–286. IEEE, 2008.

[MKB+10] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[MKB+11] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE, 2011.

[MMS93] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[Mur12a] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[Mur12b] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

References

[MZ12] Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. *SLT*, 12:234–239, 2012.

[Ngu15] Dat Quoc Nguyen. jLDADMM: A Java package for the LDA and DMM topic models. `http://jldadmm.sourceforge.net/`, 2015.

[NRD16] Beata Nowok, Gillian Raab, and Chris Dibben. synthpop: Bespoke creation of synthetic data in r. *Journal of Statistical Software, Articles*, 74(11):1–26, 2016.

[NRS07] 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*, pages 75–84. ACM, 2007.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[NTC18] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés. Differentially private distributed convex optimization via functional perturbation. *IEEE Transactions on Control of Network Systems*, 5(1):395–408, 2018.

[OCPB16] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169, June 2016.

[Par62] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[Pop16] Daniel Pop. Machine learning and cloud computing: Survey of distributed and saas solutions. *arXiv preprint arXiv:1603.08767*, 2016.

References

[PP02] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[Pre07] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[RA17] Benjamin IP Rubinstein and Francesco Aldà. Pain-free random differential privacy with sensitivity sampling. In *International Conference on Machine Learning*, pages 2950–2959, 2017.

[Ras04] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[RBHT12] Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1), 2012.

[Rei03] Jerome P Reiter. Inference for partially synthetic, public use microdata sets. *Survey Methodology*, 29(2):181–188, 2003.

[Rei05a] Jerome P Reiter. Estimating risks of identification disclosure in microdata. *Journal of the American Statistical Association*, 100(472):1103–1112, 2005.

[Rei05b] Jerome P Reiter. Using cart to generate partially synthetic public use microdata. *Journal of Official Statistics*, 21(3):441, 2005.

[RGC15] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, pages 896–902. IEEE, 2015.

[RRR03] Trivellore E Raghunathan, Jerome P Reiter, and Donald B Rubin. Multiple imputation for statistical disclosure limitation. *Journal of official statistics*, 19(1):1, 2003.

[Rub86] Donald B Rubin. Basic ideas of multiple imputation for nonresponse. *Survey Methodology*, 12(1):37–47, 1986.

[Rub93] Donald B Rubin. Discussion statistical disclosure limitation. *Journal of official Statistics*, 9(2):461, 1993.

[Sam01] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[Siz10] Sergej Sizov. Geofolk: latent spatial semantics in web 2.0 social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 281–290. ACM, 2010.

[SMH11] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[SMT09] Carolin Strobl, James Malley, and Gerhard Tutz. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4):323, 2009.

[SRS17] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the*

2017 ACM SIGSAC Conference on Computer and Communications Security, pages 587–601. ACM, 2017.

[SS98] Alex J Smola and Bernhard Schölkopf. *Learning with kernels*, volume 4. Citeseer, 1998.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 3–18. IEEE, 2017.

[Sut13] Ilya Sutskever. *Training recurrent neural networks*. PhD thesis, University of Toronto, 2013.

[Swe97] Latanya Sweeney. Computational disclosure control for medical microdata: the datafly system. In *Record Linkage Techniques 1997: Proceedings of an International Workshop and Exposition*, pages 442–453, 1997.

[Swe01] Latanya Sweeney. *Computational disclosure control: a primer on data privacy protection*. PhD thesis, Massachusetts Institute of Technology, 2001.

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[SWL18] Leon Sixt, Benjamin Wild, and Tim Landgraf. Rendergan: Generating realistic labeled data. *Frontiers in Robotics and AI*, 5:66, 2018.

[SZL16] Michael Thomas Smith, Max Zwiessele, and Neil D Lawrence. Differentially private gaussian processes. *arXiv preprint arXiv:1606.00720*, 2016.

[The16] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[TTZ15] Kunal Talwar, Abhradeep Guha Thakurta, and Li Zhang. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3025–3033, 2015.

[TZJ+16] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security Symposium*, pages 601–618, 2016.

[U.S04] U.S. Dept. of Labor, Employee Benefits Security Administration. The Health Insurance Portability and Accountability Act of 2004 (HIPAA), 2004.

[WB90] Samuel D. Warren and Louis D. Brandeis. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[WDW12] Leon Willenborg and Ton De Waal. *Elements of statistical disclosure control*, volume 155. Springer Science & Business Media, 2012.

[WLF16] Yu-Xiang Wang, Jing Lei, and Stephen E. Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of erm principle. *J. Mach. Learn. Res.*, 17(1), January 2016.

[WROK09] Mi-Ja Woo, Jerome P Reiter, Anna Oganian, and Alan F Karr. Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality*, 1(1):7, 2009.

[WWY15] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244. ACM, 2015.

[XWC+14] Mingqiang Xue, Huayu Wu, Wei Chen, Wee Siong Ng, and Gin Howe Goh. Identifying tourists from public transport commuters. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1779–1788. ACM, 2014.

[XWG11] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1200–1214, 2011.

[YCH+15] Hongzhi Yin, Bin Cui, Zi Huang, Weiqing Wang, Xian Wu, and Xiaofang Zhou. Joint modeling of users' interests and mobility patterns for point-of-interest recommendation. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 819–822. ACM, 2015.

[YJV06] Hwanjo Yu, Xiaoqian Jiang, and Jaideep Vaidya. Privacy-preserving svm using nonlinear kernels on horizontally partitioned data. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 603–610. ACM, 2006.

[YRUF14] Fei Yu, Michal Rybar, Caroline Uhler, and Stephen E Fienberg. Differentially-private logistic regression for detecting multiple-snp association in gwas databases. In *International Conference on Privacy in Statistical Databases*, pages 170–184. Springer, 2014.

[YZX12] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings*

of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 186–194. ACM, 2012.

[ZCWY14] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38, 2014.

[ZH05] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[ZVW] Ying Zhang, Stephan Vogel, and Alex Waibel. Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 2051–2054.

[ZZX+12] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.

Part V

Appendix

Appendix A

Experimental evaluation of spatiotemporal LDA

In this section, we firstly present the qualitative evaluation of the spatiotemporal models on the real world dataset. Later, we present comparative performance evaluation. We illustrate the conduct experiments on the synthetic datasets and show that the proposed models are more effective than a graph based community detection algorithm - Louvain algorithm [BGLL08]. We further reason why one observes the fall in the effectiveness of any graph based algorithm compared to the generative models.

A.1 Qualitative evaluation

Figure A.1 shows the topics for weekdays and Figure A.2 for weekends when one ignores the temporal dimension. In the map, top 10 places for each topic are shown. The lines in the map correspond to the MRT lines in Singapore. Although we start without any assumption related to geographic proximity, the observed clusters (topics) correspond to some segments of the MRT lines.

A closer observation of Figure A.1 shows that the topics are concentrated towards the southern part of Singapore, known as the Central Business District (CBD), which hosts numerous workplaces. For instance, topics 0, 1, 5, 7, which appear at the periphery of Singapore, correspond to the segments of the MRT lines passing through residential areas whereas topics 2 and 4 show the transition from

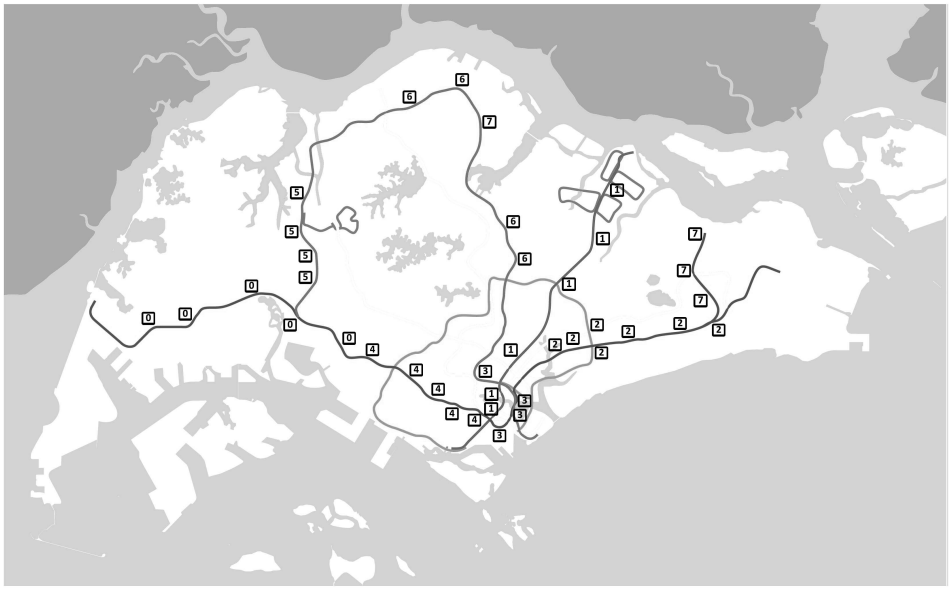


Figure A.1: Weekday topics for SLDA

residential areas to business district. Typically, if we look at topic 4 it links the area with high-tech companies and dissolves in the business district. Topic 3 exclusively captures the CBD area in Singapore. So, it asserts the hypothesis that on weekdays, people commute between the residential areas to the work places.

Compared to Figure A.1, topics in Figure A.2 represent segments of the MRT line in the peripheral parts of Singapore, mostly near the shopping malls close to residential areas. For instance, for people living in western residential areas, topic 3 and 6, Jurong East, which appears at the intersection of two topics, is a popular weekend destination. Orchard, which is situated in CBD, is one of the most popular shopping place in Singapore and it appears in Topic 0, 2, 4, 5 in Figure A.2. Closer observation tells that all the topics which appear at Orchard are spatially closer to CBD. So, it asserts the hypothesis that on weekends people travel from residential areas to the shopping or leisure centers in their vicinity.

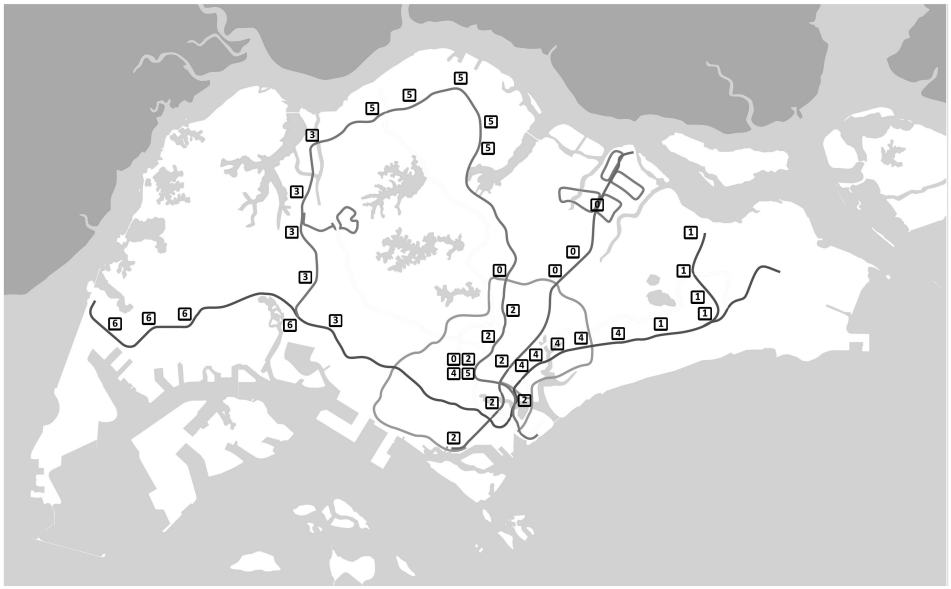


Figure A.2: Weekend topics for SLDA

Figure A.3 shows the topics found by TLDA model. We observe topics in both the weekend and the weekdays dataset capture similar temporal trends. In Figure A.3, we show a selection of them: Topics 4, 5 and 7 are from weekdays dataset and topic 2 is from weekend dataset. We observe that such model captures temporal patterns of travel. We clearly see that in topic 7 which is peaked around 9 a.m. and 6 p.m. captures the community of the working people. By observing topic 2, we see that there is a community of people who work on weekends. Although both of them denote the communities of working class, there is a subtle difference between them. On weekdays, the distribution is sharply peaked around 9 a.m. whereas in topic 2, which is from weekend dataset, we see the distribution to be flattened. On the weekdays, the working professionals, going to business district, need to follow strict working hours. On the weekends, shops open at different times and so the workers travel at different times. Aside from these, we have topics which capture the people travelling trends in the night (topic 4) as well as in the afternoon (topic 5).

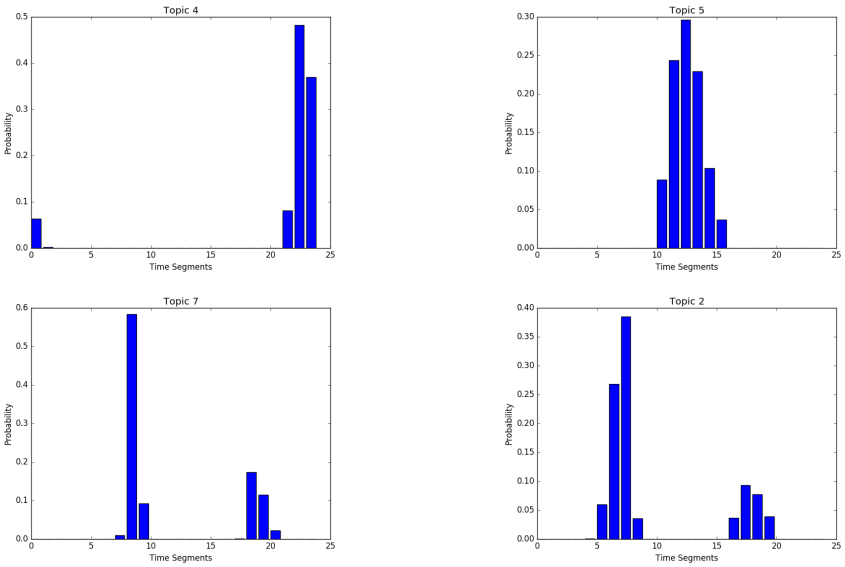


Figure A.3: Topics for TLDA

Figure A.4 and Figure A.5 show the topics for weekday when we consider both spatial and temporal dimension independently for each other. All the topics found on the weekday follow the typical bimodal pattern shown in Figure A.4. Similarly, all of the topics found on the weekend follow the pattern in which we do not observe high traffic in the morning before 9 a.m. as shown in Figure A.5. We observe that when one considers location and time simultaneously, the temporal topics not only follow the pattern of global statistics shown in Figure 5.2 but also do not significantly differ among themselves except a few fluctuations within a window of an hour or two.

The reasons are attributed to the area of Singapore and connectivity among different regions in it. Singapore is a city nation where majority of the population lives in the suburbs and travel to downtown for working. Singapore is highly connected with an efficient public transport system[1]. So, a commuter in Singapore does not take more than 90 minutes to reach one place from other place.

[1] `http://www.worldcitiessummit.com.sg/sites/sites2.globalsignin.com.2.wcs-2014/files/Smart_Mobility_Innovative_Solutions.pdf`

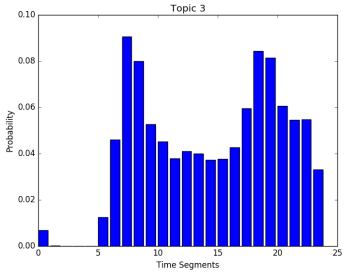


Figure A.4: Weekday temporal topic (STLDA)

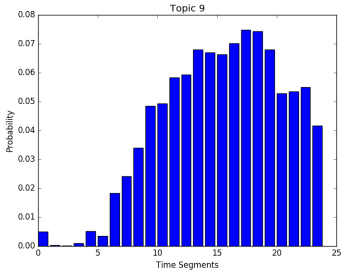


Figure A.5: Weekend temporal topic (STLDA)

A.2 Comparative performance evaluation

A.2.1 Dataset

We generate a synthetic yet realistic dataset. Following our findings with the real dataset, we consider 10 communities with 1000 different commuters in each one. We consider 10,000 different locations. From a statistical analysis of the real data, we observe that every community has a skewed distribution over the places. Very few places have a high probability to be visited by commuters in the community while the majority of remaining places have negligible probability to be visited. Agreeing to these observations, we distribute places over communities. Each community is characterised by a probability distribution of the places visited by commuters of the community. Additionally, we observe, in the real dataset, that a commuter generally visits up to 8 places. For every commuter, we sample the number of her visits from a Gamma distribution[2]. We then sample her actual visits from the distribution of places in her community. This distribution is given by a Zipf distribution for an order of places following the probability distribution that characterises her community.

[2]shape parameter is set to 6 and scale parameter is set to 1.2

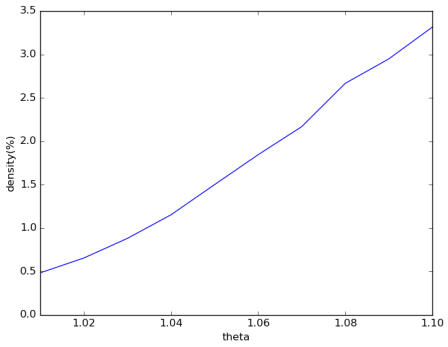


Figure A.6: Density variation with parameter theta

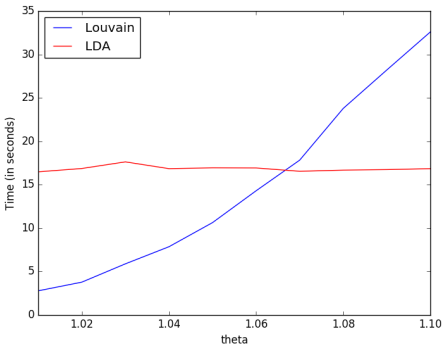
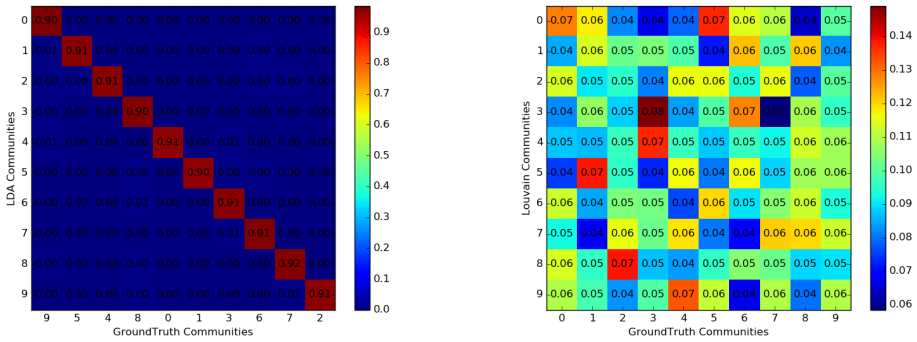


Figure A.7: Comparative Efficiency Evaluation

A.2.2 Comparison with a graph based technique

We compare our method with a state of the art graph community detection method: Louvain algorithm [BGLL08]. It is a graph partitioning technique that provides a computationally efficient approach to detect communities through greedy optimisation of *modularity*. In order to compare the proposed method with a graph clustering method, we construct corresponding *commuter graph* in which commuters are the vertices of the graph. We add an edge between two vertices of the graph whenever the corresponding two commuters share a common visit. The graph has no self-loops. Edges are weighted by the number of common visits between the two commuters that the two vertices represent.

We have verified that the parameter θ of the Zipf distribution controls the density of the graph. The higher the value of θ, more skewed the distribution and denser the corresponding graph. As the value of θ increases, fewer places have high probability to be visited while the probability of the remaining places to be visited drops off more quickly. With high values of θ, commuters in a community are more likely to visit the same few places with high probability. Hence, the density of the graph increases. Figure A.6 shows the almost linear increase of density of the graph with the increase of the value of parameter θ.



(a) LDA with $K = 10$ **(b)** Louvain with top 10 communities

Figure A.8: Comparison of LDA and Louvain with ground truth

Results Analysis

We evaluate the **effectiveness** of the proposed method comparatively to the Louvain algorithm against the groundtruth which comprises of 10 communities as described above. In all ten cases, Louvain algorithm finds communities whose count far exceeds than the number of communities in the groundtruth. However, the top-10 found communities contain more than 97% of the commuters. We consider these top-10 communities for evaluation. We run LDA for K equals 10 and 2000 Gibbs sampling iterations. LDA is able to find a quasi one-to-one mapping to the ground truth. Figure A.8 shows the confusion matrices. Each cell in the confusion matrix is annotated by Jaccard similarity between corresponding communities. Due to lack of space, we show the confusion matrix for only one of the datasets ($\theta = 1.1$). The confusion matrices clearly show that LDA is more effective than Louvain. From a closer inspection, Jaccard similarity values in Louvain comparison are very small and suggest a very poor effectiveness for the problem at hand. Results for other values of density which are not presented here in details due to space limitation, confirm that the observation remains consistent.

We observe a drastic drop in the effectiveness in case of Louvain algorithm in Figure A.8b. Reasons for such an observation are rooted at the inner details

of the two techniques. LDA uses iterative Gibbs sampling for inference which changes the topics *viz.* the distribution over visits in every iteration. Accordingly, the allottment of commuters, corresponding topic also changes in every iteration. The cohesion of topics goes on improving over the iteration until convergence. In contrast, when we weigh the edges in the graph by co-occurrences, we lose this dimension of topics defined over visits. We do no more have freedom to probe whether the co-occurring places on the edge (which we have taken into account by assigning weight) belong to a same topic or not. Once the graph is constructed, the network structure becomes agnostic to the qualitative notion of topics as clusters of visits. This loss of degree of freedom hampers the effectiveness of graph based community detection techniques.

We comparatively evaluate the **efficiency** of the proposed algorithm and that of the Louvain algorithm on ten synthetic datasets for varying values of the parameter θ. Figure A.7 shows the running time for the two algorithms. We see that Louvain algorithm is more efficient for sparser graphs. We get this efficiency at the cost of effectiveness since for sparse graphs we get large number of small communities.

Thus, the use of generative model that we propose is qualitatively effective and its algorithm with Gibbs sampling practically efficient.

Appendix B

Generating fake but realistic headlines using deep neural networks

Social media platforms such as Twitter and Facebook implement filters to detect fake news as they foresee their transition from social media platform to primary sources of news. The robustness of such filters lies in the variety and the quality of the data used to train them. There is, therefore, a need for a tool that automatically generates fake but realistic news.

In this chapter, we propose a deep learning model that automatically generates news headlines. The model is trained with a corpus of existing headlines from different topics. Once trained, the model generates a fake but realistic headline given a seed and a topic. For example, given the seed "Kim Jong Un" and the topic "Business", the model generates the headline "kim jong un says climate change is already making money".

In order to better capture and leverage the syntactic structure of the headlines for the task of synthetic headline generation, we extend the architecture - Contextual Long Short Term Memory, proposed by Ghosh et al. - to also learn a part-of-speech model. We empirically and comparatively evaluate the performance of the proposed model on a real corpora of headlines. We compare our proposed approach and its variants using Long Short Term Memory and Gated Recurrent Units as the building blocks. We evaluate and compare the topical coherence of the generated headlines using a state-of-the-art classifier. We, also, evaluate the quality of the generated headline using a machine translation quality metric

and its novelty using a metric we propose for this purpose. We show that the proposed model is practical and competitively efficient and effective.

B.1 Introduction

In the Digital News Report 2016[1], Reuters Institute for the Study of Journalism claims that 51% of the people in their study indicate the use of social media platforms as their primary source of news. This transition of social media platforms to news sources further accentuates the issue of the trustworthiness of the news which is published on the social media platforms. In order to address this, social media platform like Facebook has already started working with five fact-checking organisations to implement a filter which can flag fake news on the platform[2].

Starting from the traditional problem of spam filtering to a more sophisticated problem of anomaly detection, machine learning techniques provide a toolbox to solve such a spectrum of problems. Machine learning techniques require a good quality training data for the filters to be robust and effective. To train fake news filters, they need a large amount of fake but realistic news. Fake news, which are generated by a juxtaposition of a couple of news without any context, do not lead to robust filtering. Therefore, there is a need of a tool which automatically generates a large amount of good quality fake but realistic news.

In this chapter, we propose a deep learning model that automatically generates news headlines given a seed and the context. For instance, for a seed "obama says that", typical news headlines generated under different contexts are given as below:

- **Technology**: obama says that google is having new surface pro with retina

[1] http://www.digitalnewsreport.org/survey/2016/overview-key-findings-2016/

[2] https://www.theguardian.com/technology/2016/dec/15/facebook-flag-fake-news-fact-check

display design

- **Business**: obama says that mark zuckerberg has been told to limit carbon emissions
- **Medicine**: obama says that study says west africa ebola outbreak has killed million
- **Entertainment**: obama says that he was called out of kim kardashian kanye west wedding

We expect that the news headlines generated by the model should not only adhere to the provided context but also to conform to the structure of the sentence. In order to catch the attention of the readers, news headlines follow the structure which deviates from the conventional grammar to a certain extent. We extend the architecture of Contextual Long Short Term Memory (CLSTM), proposed by Ghosh et al. [GVS+16], to learn the part-of-speech model for news headlines. We compare Recurrent Neural Networks (RNNs) variants towards the effectiveness of generating news headlines. We qualitatively and quantitatively compare the topical coherence and the syntactic quality of the generated headlines and show that the proposed model is competitively efficient and effective.

B.2 Related work

In the last four-five years, with the advancement in the computing powers, neural networks have taken a rebirth. Neural networks with multiple hidden layers, dubbed as "Deep Neural Networks", have been applied in many fields starting from classical fields like multimedia and text analysis [GJM13, Sut13, KSH12, SMH11] to more applied fields [WWY15, DY+14]. Different categories of neural networks have been shown to be effective and specific to different kinds of tasks. For instance, Restricted Boltzmann Machines are widely used for unsupervised learning as well as for dimensionality reduction [HS06] whereas Convolutional Neural Networks are widely used for image classification task [KSH12].

Recurrent Neural Networks [Sut13] (RNNs) are used learn the patterns in the sequence data due to their ability to capture interdependence among the observations [GMH13, Gra13]. In [CGCB14], Chung et al. show that the extensions of RNN, namely Long Short Term Memory (LSTM) [HS97] and Gated Recurrent Unit (GRU) [CVMG+14], are more effective than simple RNNs at capturing longer trends in the sequence data. However, they do not conclude which of these gated recurrent model is better than the other. Readers are advised to refer to [LBE15] for an extensive survey of RNNs and their successors.

Recurrent neural networks and their extensions are widely used by researchers in the domain of text analysis and language modelling. Sutskever et al. [SMH11] have used multiplicative RNN to generate text. In [Gra13], Graves has used LSTM to generate text data as well as images with cursive script corresponding to the input text. Autoencoder [HS06] is a class of neural networks which researchers have widely used for finding latent patterns in the data. Li et al. [LLJ15] have used LSTM-autoencoder to generates text preserving the multi-sentence structure in the paragraphs. They give entire paragraph as the input to the system that outputs the text which is both semantically and syntactically closer to the input paragraph. Tomas et al. [MKB+10, MKB+11] have proposed RNN based language models which have shown to outperform classical probabilistic language models. In [MZ12], Tomas et al. provide a context along with the text as an input to RNN and later predict the next word given the context of preceding text. They use LDA [BNJ03] to find topics in the text and propose a technique to compute topical features of the input which are fed to RNN along with the input. Ghosh et al. [GVS+16] have extended idea in [MZ12] by using LSTM instead of RNN. They use the language model at the level of a a word as well as at the level of a sentence and perform experiments to predict next word as well as next sentence given the input concatenated with the topic. There have been evidences of LSTM outperforming GRU for the task of language modelling [JZS15, ITA+16]. Nevertheless, we compare our proposed model using both of these gated recurrent building blocks. We use the simple RNN as our baseline for the comparison.

Despite these applications of deep neural networks on the textual data, there are few caveats in these applications. For instance, although in [GVS+16] authors develop CLSTM which is able to generate text, they evaluate its predictive properties purely using objective metric like perplexity. The model is not truly evaluated to see how effective it is towards generating the data. In this chapter, our aim is to use deep neural networks to generate the text and hence evaluate the quality of synthetically generated text against its topical coherence as well as grammatical coherence.

B.3 Methodology

B.3.1 Background: Recurrent neural network

Recurrent Neural Network (RNN) is an adaptation of the standard feed-forward neural network wherein connections between hidden layers form a loop. Simple RNN architecture consists of an input layer (x), a hidden layer (h), and an output layer (y). Unlike the standard feed-forward networks, the hidden layer of RNN receives an additional input from the previous hidden layer. These recurrent connections give RNN the power to learn sequential patterns in the input. We use the many-to-many variant of RNN architecture which outputs n-gram given the previous n-gram as the input. For instance, given {*(hello, how, are)*} trigram as the input, RNN outputs {*(how, are, you)*} as the preceding trigram.

Bengio et al. [BSF94] show that learning the long-term dependencies using gradient descent becomes difficult because the gradients eventually either vanish or explode. The gated recurrent models, LSTM [HS97] and GRU [CVMG+14], alleviate these problems by adding gates and memory cells (in the case of LSTM) in the hidden layer to control the information flow. LSTM introduces three gates namely forget gate (f), input gate (i), and output gate (o). Forget gate filters the amount of information to retain from the previous step, whereas input and output gate defines the amount of information to store in the memory cell and

the amount of information to transfer to the next step, respectively. Equation B.1 shows the formula to calculate the forget gate activations at a certain step t. For given layers or gates m and n, W_{mn} denotes the weight matrix and b_m is the bias vector for the respective gate. h is the activation vector for the hidden state and $\sigma(\cdot)$ denotes the sigmoid function. Readers are advised to refer to [HS97] for the complete formulae of each gate and layer in LSTM.

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{B.1}$$

GRU simplifies LSTM by merging the memory cell and the hidden state, so there is only one output in GRU. It uses two gates which are update and reset gate. Update gate unifies the input gate and the forget gate in LSTM to control the amount of information from the previous hidden state. The reset gate combines the input with the previous hidden state to generate the current hidden state.

B.3.2 Proposed syntacto-contextual architecture

Simple RNNs predict the next word solely based on the word dependencies which are learnt during the training phase. Given a certain text as a seed, the seed may give rise to different texts depending on the context. Refer to the Section B.1 for an illustration. [GVS+16] extends the standard LSTM to Contextual Long Short Term Memory (CLSTM) model which accepts the context as an input along with the text. For example, an input pair {*(where, is, your), (technology)*} generates an output like {*(is, your, phone)*}. CLSTM is a special case of the architecture shown in Figure B.1 using LSTM as the gated recurrent model.

In order to use the model for the purpose of text generation, contextual information is not sufficient to obtain a good quality output. A good quality text is coherent not only in terms of its semantics but also in terms of its syntax. By providing the syntactic information along with the text, we extend the contextual model to Syntacto-Contextual (SC) models. Figure B.2 shows the general archi-

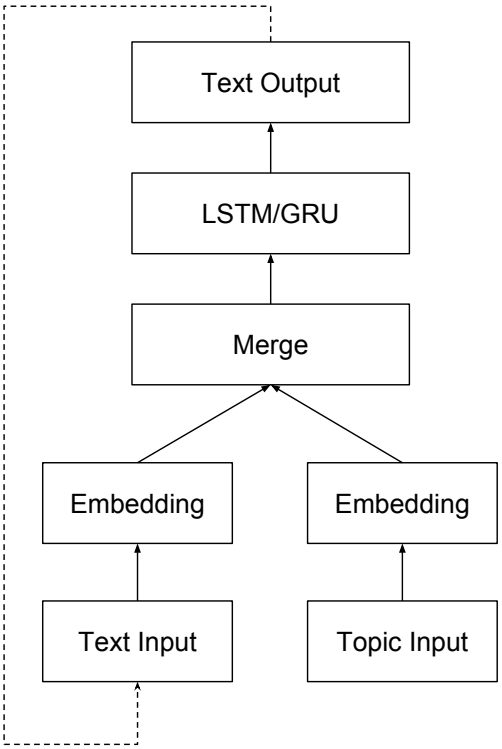


Figure B.1: Contextual Architecture [GVS+16]

tecture of the proposed model. We encode the patterns in the syntactic meta information and input text using the gated recurrent units and, later, merge them with the context. The proposed model not only outputs text but also corresponding syntactic information. For instance, an input {(where, is, your), (adverb, verb, pronoun), (technology)} generates output like {(is, your, phone), (verb, pronoun, noun)}.

Mathematically, the addition of context and syntactic information amounts to learning a few extra weight parameters. Specifically, in case of LSTM , Equation B.1 will be modified to Equation B.2 and Equation B.3, for CLSTM and SCLSTM respectively. In Equation B.2 and Equation B.3, p represents topic embedding and s represents embedding of the syntactic information.

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f + \mathbf{W_{pf}p_t}) \tag{B.2}$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f + \mathbf{W_{pf}p_t} + \mathbf{W_{sf}s_t}) \tag{B.3}$$

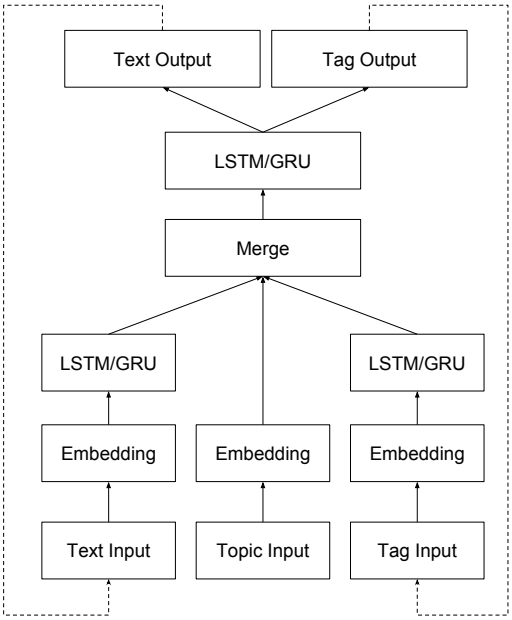


Figure B.2: Syntacto-Contextual Architecture

For the current study, we annotate the text input with the part-of-speech tags using Penn Treebank tagset [MMS93]. We learn the parameters of the model using stochastic gradient descent by minimising the loss for both output text and output tags. We, also, work on a variation of the contextual architecture which does not accept topic as an input and uses conventional RNN instead of LSTM. This model is treated as the baseline against which all of the models will be compared.

For each of the model, we embed the input in a vector space. We merge the inputs by column wise concatenation of the vectors. We perform experiments using both LSTM and GRU as the gated recurrent units. The output layer is a softmax layer that represents the probability of each word or tag. We sample from that probability to get the next word and tag output.

B.4 Experimental evaluation

We conduct a comparative study on five different models using a real-world News Aggregator Dataset. In the beginning of this section, we present the details of the datset and the experimental setup for the study. We, further, describe various quality metrics which we use to evaluate the effectiveness of the models. We perform quantitative analysis using these metric and present our results. We complete the evaluation by presenting micro-analysis for a sample of generated news headlines to show the qualitative improvement observed in the task of news headline generation.

B.4.1 Dataset and experimental setup

Dataset. We use the News Aggregator dataset[3] consisting of the news headlines collected by the news aggregator from 11,242 online news hostnames, such as *time.com*, *forbes.com*, *reuters.com*, etc. between 10 March 2014 to 10 August 2014. The dataset contains 422,937 news articles divided into four categories, namely business, technology, entertainment, and health. We randomly select 45000 news headlines, which contain more than three words, from each category because we give trigram as the input to the models. We pre-process the data in two steps. Firstly, we remove all non alpha-numeric characters from the news titles. Secondly, we convert all the text into lower case. After the pre-processing, the data contains 4,274,380 unique trigrams and 39,461 unique words.

Experimental Setup. All programs are run on Linux machine with quad core 2.40GHz Intel® Core i7™processor with 64GB memory. The machine is equipped with two Nvidia GTX 1080 GPUs. Python® 2.7.6 is used as the scripting language. We use a high-level neural network Python library, *Keras* [Cho15] which runs on top of *Theano* [The16]. We use categorical cross entropy as our loss function and use ADAM [KB14] as an optimiser to automatically adjust the

[3] `https://archive.ics.uci.edu/ml/datasets/News+Aggregator`

learning rate.

We conduct experiments and comparatively evaluate five models. We refer to those models as, **baseline** - a simple RNN model, **CLSTM** - contextual architecture with LSTM as the gated recurrent model, **CGRU** - contextual architecture with GRU as the gated recurrent model, **SCLSTM** - syntacto-contextual architecture with LSTM as the gated recurrent model, **SCGRU** - syntacto-contextual architecture with GRU as the gated recurrent model, in the rest of the evaluation. All inputs are embedded into a 200-dimensional vector space. We use recurrent layers each with 512 hidden units with 0.5 dropout rate to prevent overfitting. To control the randomness of the prediction, we set the temperature parameter in our output softmax layer to 0.4. We use the batch size of 32 to train the model until the validation error stops decreasing.

B.4.2 Evaluation metrics

In this section, we present different evaluation metrics that we use for the quantitative analysis. Along with purely objective quantitative metrics such as perplexity, machine translation quality metric, and topical precision, we use metrics like grammatical correctness, n-gram repetition for a finer effectiveness analysis. Additionally, we devise a novelty metric to qualitatively analyse the current use case of news headline generation.

Perplexity is commonly used as the performance measure [GVS$^+$16, Gra13, JZS15, ITA$^+$16] to evaluate the predictive power of a language model. Given N test data with w_t as the target outputs, the perplexity is calculated by using Equation B.4, where $p^i_{w_t}$ is the probability of the target output of sample i. A good language model assigns a higher probability to the word that actually occurs in the test data. Thus, a *language model with lower perplexity is a better model.*

$$\text{Perplexity} = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log p^i_{w_t}} \tag{B.4}$$

As it happens, the exponent in the Equation B.4 is the approximation of cross-entropy[4], which is the loss function we minimise to train the model, given a sequence of fixed length.

Although the task under consideration of the presented work is not of a word or a topic prediction, we simply use perplexity as a purely objective baseline metric. We complement it by using various application specific measures in order to evaluate the effectiveness of the quality of the generated text.

Topical coherence refers to the extent to which the generated text adheres to the desired topic. In order to evaluate the topical coherence, one requires a faithful classifier which predicts the topic of generated text. We treat the topics predicted by the classifier as the ground truth to quantitatively evaluate the topical coherence. The proposed method generates a news headline given a seed and a topic of the news. People have widely used Multinomial naive Bayes classifier to deal with text data due to independence among the words given a certain class[5]. We train a Multinomial naive Bayes classifier with Laplace smoothing on the news dataset consisting of 45000 news from each of the four categories. We hold out 20% of the data for validation. By proper tuning of the smoothing parameter, we achieve 89% validation accuracy on the news dataset. We do not use this metric for the baseline model.

Taking existing text as a reference, a **quality** metric evaluates the effectiveness of the generated text in correspondence to the reference. Such a metric measures the *closeness* of the generated text to the reference text. Metrics such as BLEU [PRWZ02], Rouge [Dod02], NIST [Lin04] are widely used to evaluate the quality of machine translation. All of these metrics use "gold standard", which is either the original text or the text written by the domain experts, to check the quality of the generated text. We use BLEU as the metric to evaluate the quality of generated text. For a generated news headline, we calculate its BLEU

[4] `http://cs224d.stanford.edu/lecture_notes/notes4.pdf`

[5] `https://www.kaggle.com/uciml/news-aggregator-dataset`

score by taking all the sentences in the respective topic from the dataset as the reference. Interested readers should refer to [ZVW] for a detailed qualitative and quantitative interpretation of BLEU scores.

With the motivation of the current work presented in the Section B.1, we want the generated text from our model to be as *novel* as possible. So as to have a robust fake news filter, the fake news, which is used to train the model, should not be a juxtaposition of few existing news headlines. More the patterns it learns from the training data to generate a single headline, more novel is the generated headline. We define **novelty** of the generated output as the number of unique patterns the model learns from the training data in order to generate that output. We realise this metric by calculating longest common sentence common to the generated headline and each of the headline in the dataset. Each of these sentences stands as a pattern that the model has learned to generate the text. *Novelty* of a generated headline is taken as the number of unique longest common sentences.

The good quality generated text should be both *novel* and grammatically correct. **Grammatical correctness** refers the judgement on whether the generated text adheres to the set of grammatical rules defined by a certain language. Researchers either employ experts for evaluation or use advanced grammatical evaluation tools which require the gold standard reference for the evaluation [DN12]. We use an open-source grammar and spell checker software called LanguageTool[6] to check the grammatical correctness of our generated headlines. LanguageTool uses NLP based 1516 English grammar rules to detect syntactical errors. Aside from NLP based rules, it used English specific spelling rules to detect spelling errors in the text. To evaluate grammatical correctness, we calculate the percentage of grammatically correct sentences as predicted by the LanguageTool.

We find that LanguageTool only recognises word repetition as an error. Consider a generated headline *beverly hills hotel for the first in the first in the world* as an

[6]API and Java package available at `https://languagetool.org`

example. In this headline, there is a trigram repetition - *the first in* - that passes LanguageTool grammatical test. Such headlines are not said to be good quality headlines. We add new rules with a regular expression to detect such repetitions. We count **n-gram repetitions** within a sentence for values of n greater than two.

B.4.3 Results

To generate the output, we need an initial trigram as a seed. We randomly pick the initial seed from the set of news headlines from the specified topic. We use windowing technique to generate the next output. We remove the first word and append the output to the back of the seed to generate the next output. The process stop when specified sentence length is generated. We generate 100 sentences for each topic in which each sentence contains 3 seed words and 10 generated words.

Quantitative evaluation

Table B.1 summarises the quantitative evaluation of all the models using metrics described in Section B.4.2. Scores in bold numbers denote the best value for each metric. We can see that for Contextual architecture, GRU is a better gated recurrent model. Conversely, LSTM is better for Syntacto-Contextual architecture.

For Syntacto-Contextual architecture, we only consider the perplexity of the text output to make a fair comparison with the Contextual architecture. We analyze that our Syntacto-Contextual architecture has a higher perplexity score because the model jointly minimises both text and syntactical output losses. On the other hand, the baseline model has a low perplexity score because it simply predicts the next trigram with control on neither the context nor the syntax.

A high score on classification precision substantiates that all of these models

Table B.1: Quantitative and comparative evaluation of baseline, CLSTM [GVS+16], CGRU, SCLSTM and SCGRU.

	Baseline	**CLSTM**	**CGRU**	**SCLSTM**	**SCGRU**
Perplexity	108.383	119.10	**92.22**	146.93	175.83
Topical coherence (%)	-	84.25	77.25	**94.75**	87.50
Quality (BLEU)	0.613	0.637	**0.655**	0.633	0.625
Novelty	21.605	24.67	25.21	**26.57**	25.65
Grammatical correctness (%)	28.25	49.75	50.75	**75.25**	69.00
n-gram repetitions	11	30	12	**5**	8

generate headlines which are coherent with the topic label with which they are generated. We observe that all of the models achieve a competitive BLEU score. Although Contextual architecture performs slightly better in terms of BLEU score, Syntacto-Contextual architecture achieves a higher novelty score. In the qualitative evaluation, we present a more detailed comparative analysis of BLEU scores and novelty scores.

We observe that the news headlines generated by Syntacto-Contextual architecture are more grammatically correct than other models. Figure B.3 shows the histogram of n-gram repetitions in the generated news headline. We see that the Syntacto-Contextual architecture gives rise to news headlines with less number of n-gram repetitions.

Lastly, we have empirically evaluated, but not presented here, the time taken by different models for one epoch. CLSTM takes 2000 seconds for one epoch whereas SCLSTM takes 2131 seconds for one epoch. Despite the Syntacto-Contextual architecture being a more complex architecture than Contextual architecture, it shows that it is competitively efficient.

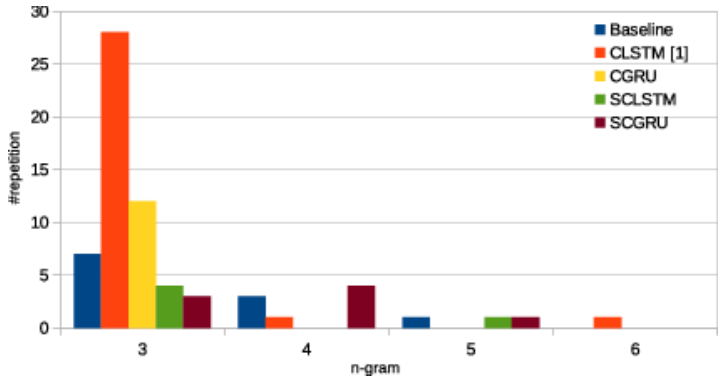


Figure B.3: n-gram repetition analysis

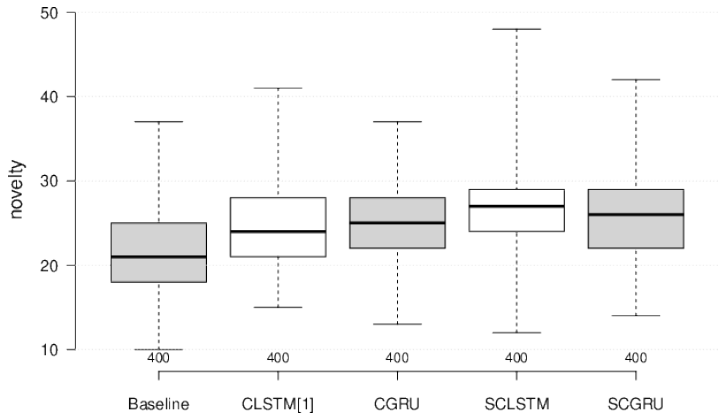


Figure B.4: Boxplot for novelty metric

Qualitative evaluation.

Table B.2 and Table B.3 presents the samples of generated news from CLSTM proposed by [GVS+16] and SCLSTM, which outweighs the rest of the models in the quantitative analysis.

In Table B.1, we see that the Contextual architecture models receive a higher BLEU score than the proposed architecture models. BLEU score is calculated using n-gram precisions with the news headlines as the reference. It is not always necessary that the higher BLEU score leads towards a good quality text generation. Qualitative analysis of generated headlines shows that the higher BLEU score, in the most cases, is the result of the juxtaposition of the existing news headlines. For instance, consider a headline generated by CLSTM model

Table B.2: Generated news headlines for topics medicine and business

Category	**CLSTM** [GVS+16]	**SCLSTM**
Medicine	first case chikungunya virus in saudi arabia reports new mers cov in the	first case chikungunya virus found in florida state health care system draws attention
	mosquitoes test positive for west africa in guinea in june to be the	mosquitoes test positive for west nile virus found in storage room at home
	ticks and lyme disease in the us in the us are the best	ticks and lyme disease rates double in autism risk in china in west
Business	us accuses china and russia to pay billion in billion in us	us accuses china of using internet explorer bug leaves users to change passwords
	wake of massive data breach of possible to buy buy stake in us	wake of massive recall of million vehicles for ignition switch problem in china
	japan fukushima nuclear plant in kansas for first time in the first time	japan fukushima nuclear plant linked to higher growth risk of climate change ipcc

as an example - "justin bieber apologises for racist joke in new york city to take on" - which receives a BLEU score of 0.92. When we search for the same news in the dataset, we find that this generated news is a combination of two patterns from the following two headlines, "justin bieber apologizes for racist joke" and "uber temporarily cuts fares in new york city to take on city cabs". Whereas the headline generated by SCLSTM with the same seed is quite a novel headline. In the training dataset there is mention of neither Justin Bieber joking on Twitter nor joke for gay fans. Similar observation can be made with the news related to Fukushima. In the training data set there is no news headline which links

Table B.3: Generated news headlines for topics entertainment and technology

Category	**CLSTM** [GVS+16]	**SCLSTM**
Entertainment	justin bieber apologizes for racist joke in new york city to take on	justin bieber apologizes for racist joke joke on twitter for gay fans have
	the fault in our stars trailer for the hunger games mockingjay part teaser	the fault in our stars star chris hemsworth and elsa pataky reveal his
	giant practical spaceship interiors for joint venture in mexico production of star wars	giant practical spaceship hits the hollywood walk of fame induction ceremony ceremony in
Technology	first android wear watches and google be to be available in the uk	first android wear watch google play edition is now available on xbox one
	samsung sues newspaper for anti vaccine and other devices may be the best	samsung sues newspaper over facebook experiment on users with new profile feature is
	obama warns google glass to be forgotten on the us government issues recall	obama warns google apple to make android support support for mobile for mobile

Fukushima with climate change. Additionally, there is no training data which links higher growth risk to climate change as well. Thus, we observe that the headlines generated using SCLSTM are qualitatively better than CLSTM.

All of the models presented in the work are probabilistic models. Text generation being a probabilistic event, on the one hand it is possible that contextual architecture generates a good quality headline at a certain occasion. For instance, we see that CLSTM also generates some good quality news headlines such as “the fault

in our stars trailer for the hunger games mockingjay part teaser". On the other hand, it is possible that Syntacto-Contextual architecture generates some news headline with poor quality or repetitions, such as "obama warns google apple to make android support support for mobile for mobile". In order to qualitatively analyse the novelty of generated sentence, we need to observe how likely such events occur. Figure B.4 shows the boxplot of novelty numbers we calculate for each of 400 generated news headlines using different models. As discussed earlier, we want our model to generate novel news headlines. So, we prefer higher novelty scores. Although the mean novelty of all of the models lie around 24, we see that SCLSTM is more likely to generate the novel headlines. Additionally, we observe that contextual and Syntacto-Contextual architectures performs better than the baseline model.

As mentioned in the quantitative evaluation, Contextual architecture gives rise to news headlines with a large number of n-gram repetitions. In an extreme case, CLSTM model generates the following headline, "lorillard *inc nyse wmt wal mart stores* inc nyse wmt wal mart stores", that contains 6-gram repetition. The news headline generated by CLSTM - "samsung sues newspaper for anti vaccine and other devices may be the best"- exemplifies the smaller topical coherence observed for the Contextual architecture models.

In order to garner the opinion of real-world users, we use CrowdFlower[7] to conduct a crowdsource based study. In this study, we generate two news headlines using CLSTM and SCLSTM using the same seed and ask the workers to choose a more realistic headline between two. We generate such a pair of headlines for 200 different seeds. Each pair is evaluated by three workers and majority vote is used to choose the right answer. At the end of the study, 66% workers agree that SCLSTM generates more realistic headlines than CLSTM.

[7] `https://www.crowdflower.com/`

B.5 Discussion

In [GVS+16], Ghosh et al. proposed a deep learning model to predict the next word or sentence given the context of the input text. In this work, we adapted and extended their model towards automatic generation of news headlines. The contribution of the proposed work is two-fold. Firstly, in order to generate news headlines which are not only topically coherent but also syntactically sensible, we proposed an architecture that learns part-of-speech model along with the context of the textual input. Secondly, we performed thorough qualitative and quantitative analysis to assess the quality of the generated news headlines using existing metrics as well as a novelty metric proposed for the current application. We comparatively evaluated the proposed models with [GVS+16] and a baseline. To this end, we show that the proposed approach is competitively better and generates good quality news headlines given a seed and the topic of the interest.

Publication

Work in this chapter is part of the following publication.

- Ashish Dandekar, Remmy A. M. Zen, and Stéphane Bressan. Generating fake but realistic headlines using deep neural networks. In *Database and Expert Systems Applications- 28th International Conference, DEXA 2017, Lyon, France, Proceedings, Part II, pages 427-440*

Acknowledgement

This work is partially supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme and by the National University of Singapore under a grant from Singapore Ministry of Education for research project number T1 251RES1607. We kindly acknowledged UCI for maintaining repository of publicly available Machine Learning dataset [Lic13].