UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO E DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

IAGO CASTRO CHAVES

DIFFERENTIALLY PRIVATE SELECTION USING SMOOTH SENSITIVITY

FORTALEZA
2024

IAGO CASTRO CHAVES

DIFFERENTIALLY PRIVATE SELECTION USING SMOOTH SENSITIVITY

Thesis submitted to the Programa de Pós-graduação em Ciência da Computação of the Centro de Ciências of the Universidade Federal do Ceará, as a partial requirement for obtaining the title of Doctor in Ciência da Computação. Concentration Area: Ciência da Computação

Advisor: Prof. Dr. Javam de Castro Machado

FORTALEZA
2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C438d Chaves, Iago Castro.
Differentially private selection using smooth sensitivity / Iago Castro Chaves. – 2024.
84 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2024.
Orientação: Prof. Dr. Javam de Castro Machado.

1. Seleção diferencialmente privada. 2. Privacidade diferencial. 3. Privacidade. I. Título.
CDD 005

IAGO CASTRO CHAVES

DIFFERENTIALLY PRIVATE SELECTION USING SMOOTH SENSITIVITY

Thesis submitted to the Programa de Pós-graduação em Ciência da Computação of the Centro de Ciências of the Universidade Federal do Ceará, as a partial requirement for obtaining the title of Doctor in Ciência da Computação. Concentration Area: Ciência da Computação

Approved on:

EXAMINATION BOARD

Prof. Dr. Javam de Castro Machado (Advisor)
Federal University of Ceará (UFC)

Prof. Dr. Victor Aguiar Evangelista
Federal University of Ceará (UFC)

Prof. Dr. César Lincoln Cavalcante Mattos
Federal University of Ceará (UFC)

Prof. Dr. Daniel Cardoso Moraes de Oliveira
Federal University Fluminense (UFF)

Prof. Dr. Diego Mesquita
Getúlio Vargas Foundation (FGV)

Dedico este trabalho primeiramente a Deus, por me guiar com sabedoria e paciência através desta jornada. À minha esposa, pelo amor incondicional e apoio constante que foram fundamentais em cada página escrita. E à minha família, pela força e inspiração diárias, compartilhando comigo a alegria e os desafios de cada passo dado.

ACKNOWLEDGEMENTS

Expresso minha profunda gratidão a Deus, cuja presença constante foi minha fonte de força e inspiração ao longo desta jornada. Seu amor inabalável e direção divina foram essenciais para superar cada desafio e alcançar este marco em minha vida. Sou sinceramente grato por cada momento de paz e clareza proporcionados, permitindo-me perseguir meus objetivos com fé e perseverança.

À minha esposa, Lívia Macêdo, cuja compreensão, paciência e apoio incansável nos momentos mais desafiadores foram essenciais. Sua presença amorosa, especialmente durante este período especial de nossa espera, foi meu maior conforto e motivação. Sou imensamente grato por cada gesto de encorajamento e por caminhar ao meu lado nesta jornada.

Agradeço profundamente aos meus pais, Ismael Chaves e Iracema Castro, pelo apoio incansável e pela confiança inabalável que depositaram em mim. Sua dedicação e amor foram fundamentais para que eu pudesse perseguir e alcançar meus objetivos.

Sou imensamente grato à minha irmã, ao seu marido Rodrigo e aos meus sobrinhos pelo amor, apoio e compreensão constantes durante minha jornada acadêmica. O carinho e a motivação que recebi de vocês foram essenciais nos momentos de desafio e incerteza. Agradecer é pouco para expressar o quanto a presença e o encorajamento de cada um de vocês foram importantes para mim. Vocês são uma parte vital do meu sucesso e crescimento pessoal, e sou eternamente grato por ter uma família tão maravilhosa ao meu lado.

Sou profundamente grato aos Professores Doutores Javam de Castro e João Paulo Gomes por sua orientação e valiosos conselhos ao longo de minha carreira acadêmica. A dedicação, sabedoria e paciência de ambos foram cruciais para o meu desenvolvimento profissional e pessoal, guiando-me através dos desafios e decisões importantes neste percurso.

Gostaria de expressar minha sincera gratidão a Victor Farias, Diego Mesquita e Amanda Peres pelas suas inestimáveis contribuições para este trabalho. O apoio, a expertise e as perspectivas únicas que cada um trouxe foram fundamentais para o enriquecimento e aprofundamento da pesquisa apresentada. Sua colaboração não apenas fortaleceu os resultados, mas também ampliou minha compreensão e apreço pelo campo de estudo. Estou profundamente grato por ter tido a oportunidade de trabalhar ao lado de profissionais tão dedicados e talentosos.

Sou imensamente grato aos meus amigos de laboratório e pesquisa: André Luís, Eduardo Rodrigues, Malu Maia, Daniel Praciano, Diogo Forte, Fernando Dione, Lucas Falcão, Felipe Timbó, Paulo Amora, Felipe Monteiro, Gabriel Magalhães, Serafim Costa, Ítalo Abreu,

Israel Vidal, Bruno Leal, Leonardo Linhares e Isabel ”Fernandes” Lima. Cada um de vocês desempenhou um papel crucial em minha jornada, oferecendo suporte, inspiração e momentos valiosos de troca de conhecimentos e camaradagem. A colaboração e o incentivo que recebi de todos vocês foram essenciais para superar os desafios e alcançar os objetivos desta pesquisa. A amizade e o apoio contínuo de vocês enriqueceram profundamente minha experiência acadêmica e minha vida.

Estendo meus sinceros agradecimentos a todos os colegas do Laboratório de Sistemas e Banco de Dados (LSBD) pela colaboração e pelo ambiente acadêmico estimulante que vocês proporcionaram. A interação diária, o compartilhamento de ideias e a assistência mútua foram fundamentais para o desenvolvimento do meu trabalho e crescimento pessoal. A atmosfera de companheirismo e suporte contínuo no LSBD desempenhou um papel vital em minha trajetória acadêmica, tornando esta experiência tanto produtiva quanto gratificante. Sou grato por ter tido a oportunidade de aprender e evoluir ao lado de profissionais tão dedicados e inspiradores.

Sou profundamente grato ao Laboratório de Sistemas e Banco de Dados (LSBD) por fornecer as instalações adequadas e o suporte essencial que possibilitaram o desenvolvimento da minha tese. Além disso, agradeço imensamente pelo apoio financeiro concedido para a participação em eventos científicos, o que foi crucial para a disseminação do meu trabalho e meu crescimento profissional na comunidade acadêmica. A contribuição do LSBD foi fundamental para alcançar os objetivos desta pesquisa.

Expresso minha sincera gratidão à CAPES/Brazil pelo suporte financeiro providenciado sob o número de concessão 88882.454584/2019-01. Este auxílio foi fundamental não apenas para o desenvolvimento e conclusão da minha pesquisa, mas também para minha formação acadêmica e profissional. O investimento em meu projeto e em minha capacitação por meio desta bolsa reflete o compromisso da CAPES com a promoção da ciência, tecnologia e educação no Brasil, e eu sou profundamente grato por ter sido um dos beneficiados por esta iniciativa.

Estendo minha profunda gratidão à Lenovo pelo suporte financeiro recebido através da Lei da Informática. Este apoio foi crucial para o desenvolvimento e sucesso do meu projeto de pesquisa, permitindo-me acesso a recursos e tecnologias essenciais para a realização do meu trabalho. A parceria da Lenovo, reconhecendo e investindo em pesquisa e desenvolvimento, é um exemplo brilhante de como a colaboração entre a indústria e a academia pode impulsionar a inovação e o avanço científico. Sou sinceramente grato por esta contribuição significativa ao meu percurso acadêmico e profissional.

“So the problem is not so much to see what nobody has yet seen, as to think what nobody has yet thought concerning that which everybody sees.” (Arthur Schopenhauer)

ABSTRACT

Differentially private selection mechanisms offer strong privacy guarantees for queries whose canonical outcome is the top-scoring element r within a finite set $\mathcal{R}$ according to a dataset-dependent utility function. While selection queries are pervasive throughout data science, there are few mechanisms to ensure their privacy. Additionally, the vast majority focus on achieving differential privacy (DP) through *global sensitivity*, possibly corrupting the query result with excessive noise and maiming downstream inferences. We propose the *Smooth Noisy Max* (SNM) algorithm to alleviate this issue. In particular, SNM algorithm leverages the notion of *smooth sensitivity* to provably provide smaller (upper bounds on) expected errors compared to methods based on global sensitivity under mild conditions. Empirical results show that our algorithm is more accurate than state-of-the-art differentially private selection methods in three applications: percentile selection, greedy decision trees, and random forest.

Keywords: differentially private selection; differential privacy.

RESUMO

Mecanismos de seleção diferencialmente privados oferecem garantias robustas de privacidade para consultas cujo resultado canônico é o elemento de maior pontuação r dentro de um conjunto finito $\mathcal{R}$ de acordo com uma função de utilidade dependente do conjunto de dados. Embora as consultas de seleção sejam bem difundidas em toda a ciência de dados, existem poucos mecanismos que proveem garantias de sua privacidade. Além disso, a grande maioria foca em alcançar privacidade diferencial (DP) por meio de *sensibilidade global*, possivelmente corrompendo o resultado da consulta com excessivo ruído e prejudicando inferências subsequentes. Para mitigar esse problema, propomos o algoritmo *Smooth Noisy Max* (SNM). Em particular, o algoritmo SNM aproveita o conceito de *sensibilidade suave* para fornecer erros esperados menores (limites superiores) quando comparados a métodos baseados em sensibilidade global sob leves condições. Resultados empíricos mostram que nosso algoritmo é mais preciso do que os métodos estado-da-arte de seleção diferencialmente privados em três diferentes aplicações: seleção de percentil, árvores de decisão gulosas e floresta aleatória.

Palavras-chave: seleção diferencialmente privada; privacidade diferencial.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CONTENTS

1 INTRODUCTION

In the twenty-first century, companies increasingly strive to collect as much data as possible from their customers. This trend has prompted organizations to assign the responsibility of managing this data to a specific role, known as the data *curator* (STONEBRAKER *et al.*, 2013). This curator functions as a trusted intermediary, gathering data from individuals and subsequently disseminating valuable information for public use or specialized analysis. The curator might release aggregated data, statistics, or analytical results derived from data mining or machine learning algorithms. Such dissemination is instrumental in enhancing service delivery, optimizing marketing strategies, and publishing demographic statistics.

Because of this behavior, numerous significant data breaches have prompted governments, organizations, and companies to reevaluate their approaches to privacy (ALTMAN *et al.*, 2015). Concurrently, many advancements in Machine Learning have stemmed from learning techniques that require substantial volumes of training data, such as deep learning. Additionally, research institutions frequently utilize and exchange data that includes *sensitive* or confidential information about individuals.

Data breaches extend beyond mere unauthorized disclosure of data and are susceptible to *linkage attacks*. A linkage attack involves an attacker integrating various data sources to re-identify individuals in anonymized databases by correlating shared information across these sources. A notable instance of such an attack occurred with Netflix, where their released database was cross-referenced with the IMDb database, leading to the re-identification of individuals (NARAYANAN; SHMATIKOV, 2008).

A different form of privacy breach involves *membership inference* (SHOKRI *et al.*, 2017). This type of attack aims to ascertain whether a specific data record was included in a model's training database. Such attacks pose a significant threat to contemporary state-of-the-art machine learning methods, which often operate as opaque black-boxes to practitioners. Even traditional machine learning techniques, such as random forests, are vulnerable to privacy breaches. For instance, *optimization-based reconstruction attacks* have demonstrated the capability to reconstruct entire training databases using only a trained random forest model (FERRY *et al.*, 2024).

Inappropriate disclosure of sensitive data can compromise the privacy of data subjects–as previously mentioned–potentially leading to adverse effects, civil liabilities, or even physical harm. Consequently, recent legislative measures, such as the *General Data Protection Regulation*

(GDPR) (EUROPE, 2018) and the *Lei Geral de Proteção de Dados Pessoais (LGPD)* (BRASIL, 2018), have been introduced to enforce data anonymity. These regulations mandate that personal information must be anonymized to prevent the individual from being identifiable from the released data. This process, commonly referred to in academia as *data re-identification*, aims to enhance privacy protections.

To overcome these privacy problems *differential privacy* was stated by Dwork and Roth (2014). The term *differential* means that the output of a task can not be "different" if a user opt-in or opt-out your data from the task input.

1.1 Differential Privacy

Differential privacy (DWORK; ROTH, 2014; DWORK *et al.*, 2006) represents the leading formal paradigm for data release, providing robust privacy assurances. This framework ensures that the information released is nearly indistinguishable, regardless of whether an individual's data is included in the sensitive database. It operates under the assumption that an attacker may have knowledge of all but one of the records in the database, the record they seek to uncover.

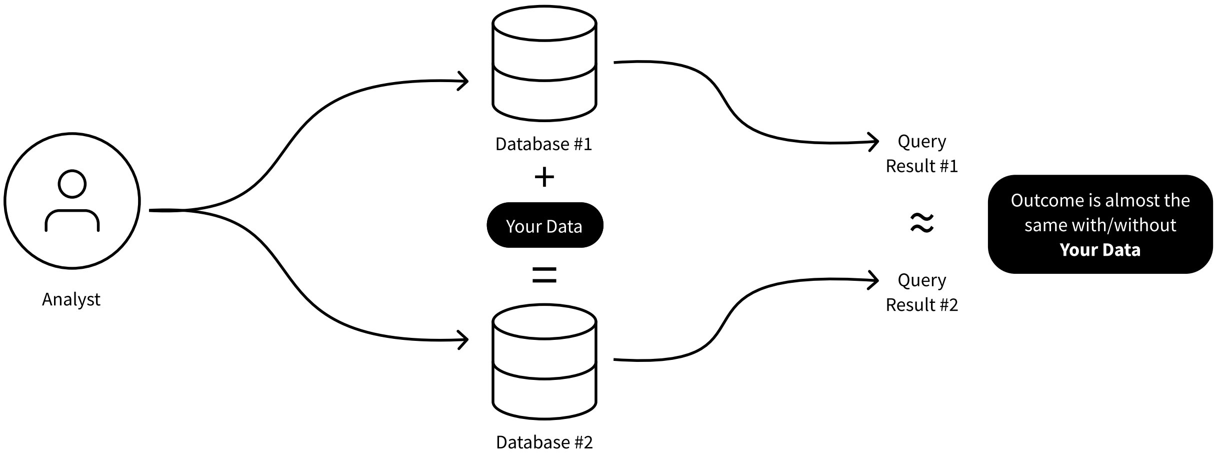
The core concept underlying differential privacy is that an analyst's query is addressed by a *randomized algorithm* that interacts with the private database and generates a randomized response from the *output distribution*. Such a randomized algorithm qualifies as a differentially private mechanism (also termed simply as "mechanism" in this context) if the output's probability distribution remains largely unchanged, irrespective of any individual's presence or absence in the database. This method provides statistical safeguards against deducing private information, even when auxiliary data is employed. An illustrative schema can be viewed in the Figure 1.

All mechanisms are designed to shape the output distribution so that both the true answer and other high-utility answers are sampled with a high probability. Such mechanisms furnish analysts with valuable information. The formal concept of utility is elaborated upon later in this section.

Algorithms can secure differential privacy through the technique of output perturbation, which involves disclosing the actual result of a non-private query f after the injection of noise. The intensity of this noise must be sufficient to obscure the identities of individuals within the input database x.

This work focuses on private selection (non-numeric queries), i.e., queries where the

Figure 1 – Diagram illustrating how a randomized differentially private algorithm works. The user's choice to opt-out does not affect the algorithm's output.



Source: elaborated by the author.

range is discrete. For instance, a query that returns the most frequent name from a database of people's names is non-numeric.

Private selection, also known as non-numerical selection, plays a crucial role in data analysis and selection tasks, such as classification (KOTSIANTIS *et al.*, 2007), synthetic data generation (CHEN *et al.*, 2015; ZHANG *et al.*, 2014), dimensionality reduction (CHAUDHURI *et al.*, 2013), and top-k queries (ILYAS *et al.*, 2008). The main idea is to select an outcome from a set of items that maximizes some objective function, also known as utility function (MCSHERRY; TALWAR, 2007), *i.e.*, an algorithm samples output from a set of outcomes following an adapted probability function.

To the best of our knowledge, only a limited number of mechanisms exist for ensuring private selection, including the exponential mechanism (MCSHERRY; TALWAR, 2007), the report-noisy-max algorithm (DWORK; ROTH, 2014), permute-and-flip (MCKENNA; SHELDON, 2020), and the local dampening mechanism (FARIAS *et al.*, 2023).

The exponential mechanism achieves differential privacy by sampling from all potential outcomes $\mathcal{R}$ according to an exponential distribution. The likelihood of selecting a particular outcome depends on its utility relative to the database $\mathbf{x} \in \mathcal{X}$. More specifically, the utility function accepts a database $\mathbf{x}$ and a potential outcome $r \in \mathcal{R}$, assigning it a score that reflects its appropriateness for the database. The higher the score, the more suitable the outcome.

The noise introduced by the exponential mechanism is characterized as sampling noise rather than numeric noise. This mechanism may select an outcome, such as a name, that is not the most frequent one – that is, one that does not possess the highest utility score. The quantity of noise injected is proportional to the measure known as *global sensitivity*. The concept of global sensitivity, denoted as Δu, quantifies the most significant change to the utility function u that could occur by adding or removing an individual across all potential databases $\mathcal{X}$ and for all potential outcomes $\mathcal{R}$.

Example 1 (Approval voting)**.** Consider an election that is conducted with an approval voting format. Instead of selecting just one candidate, voters can choose as many candidates as they support. The candidate with the most approvals wins the election. If there are m candidates, we can consider each voter's input as a vector of choices $x_i \subseteq [m]$ (the notation $[m]$ stands for $\{0, 1, \ldots, m\}$). The utility of the candidate c is the number of votes that included c in their input vector of choices, that is, $u(\mathbf{x}, c) = |\{i : c \in x_i\}|$. The global sensitivity Δu is the highest impact of adding or removing some voters from the election. Since one voter can only vote once on each candidate, its removal can impact at most by one vote. Thus, the global sensitivity is $\Delta u = 1$ for the above-mentioned utility function u.

In the same direction as the exponential mechanism, the report-noisy-max (RNM) algorithm is a differentially private selection algorithm. It adds independent noise on each utility value $u(\mathbf{x}, r)$ for all possible outcomes $r \in \mathcal{R}$ in $\mathbf{x}$, and the query result is the outcome of the larger noisy utility value. The RNM could be used along with several different distributions, such as exponential distribution, Laplace distribution, and Gumbel distribution. Each distribution yields distinct results, characterized by differences in the tail behavior of the distribution and expected error. The RNM works by adding tailored noise dependent on the notion of global sensitivity.

The global sensitivity, used for numerical and selection queries, is guided by the worst-case scenario, which usually adds high noise (ZHANG *et al.*, 2015; GONEM; GILAD-BACHRACH, 2018; SUN *et al.*, 2020; BUN; STEINKE, 2019). To mitigate that, Nissim *et al.* (2007) proposes an instance-based sensitivity for numerical queries, in other words, a sensitivity that depends locally on the input database $\mathbf{x}$, called smooth sensitivity. Note that the main difference between global and local sensitivity is that the first is not dependent on the input database $\mathbf{x}$ and the latter is.

We propose a novel differentially private selection algorithm termed *Smooth Noisy Max (SNM)*. Unlike traditional approaches that add noise based on global sensitivity, our algorithm employs local sensitivity for this purpose. Specifically, SNM leverages various probability distributions, each scaled according to a factor proportional to the instance-based sensitivity to introduce noise into the utility score of each potential outcome $r \in \mathcal{R}$. This method enhances the adaptability and accuracy of privacy preservation in data analysis; our experimental results will show that through data utility analysis.

1.2 Problem Statement

This thesis addresses the challenge of private data selection, ensuring that the selection remains relevant in producing meaningful outcomes even after the privacy process. Let $\mathbf{x} \in \mathcal{X}$ be a sensitive database, f a data selection function to be evaluated on $\mathbf{x}$. The function $f : \mathcal{X} \rightarrow \mathcal{R}$ takes a database as input, producing an outcome $r \in \mathcal{R}$.

The challenge is to output $f(\mathbf{x})$ without releasing much individual information. Thus, we design a randomized algorithm that outputs $r \in \mathcal{R}$ based on $f(\mathbf{x})$ such that it satisfies the differential privacy definition.

1.3 Hypothesis

Applying smooth sensitivity – an instance-based sensitivity measure – to the private selection mechanism guarantees (ε, δ)-differential privacy and yields superior outcomes compared to methodologies that apply global sensitivity, especially higher accuracy or reduced expected error. This assertion also extends to comparisons with competing approaches that employ instance-based sensitivity, such as local dampening.

1.4 Applications

We illustrate the advantages of the Smooth Noisy Max algorithm by applying it to three distinct problems:

Percentile Selection. Percentile selection is a common task to show the performance of a differentially private algorithm. The task is to return the *p-th* percentile value from a set of real numbers.

Greedy Decision Tree. We address the data mining challenge of constructing decision trees for classification purposes. Our approach involves a privacy-enhanced adaptation of the ID3 algorithm (QUINLAN, 1986), enabling the construction of decision trees from tabular datasets. For automated tree induction, we employ the *Max Operator* as the splitting criterion to select attributes for branching.

Random Forest. We developed a novel random forest algorithm, building upon the foundation set by prior state-of-the-art. Our version of the random forest algorithm integrates Smooth Noisy Max as a private selection mechanism within the majority selection process of each leaf node.

1.5 Contributions

The main contributions of this thesis are outlined as follows:

1. We extend the concept of smooth sensitivity, originally defined for numerical data, to the data selection setting;
2. We prove that smooth sensitivity cannot be utilized along with the exponential mechanism;
3. We propose the Smooth Noisy Max (SNM), a differentially private data selection algorithm that applies the notion of smooth sensitivity to reduce the amount of randomness in data selection;
4. We provide differential privacy guarantees for Smooth Noisy Max, along with theoretical strong utility guarantees showing that Smooth Noisy Max is never worse than the competitors under mild conditions;
5. We conducted an empirical comparison[1] of Smooth Noisy Max with competing methods across three applications: percentile selection, greedy decision trees, and random forests. Our findings indicate that SNM consistently outperforms state-of-the-art methods in terms of accuracy and expected error.

Most of the contributions of this thesis were previously submitted in our paper CHAVES, I. C.; FARIAS, V. A. E.; PEREZ, A.; MESQUITA, D.; MACHADO, J. **Differentially Private Selection using Smooth Sensitivity**. 2024. Submitted for publication to *SIGMOD International Conference on Management of Data (2025)*.

Additionally, we highlight side contributions to machine learning issues, which were developed and disseminated at various conferences throughout the course of this Ph.D. These contributions significantly broadened the scope and impact of our research endeavors.

- SENA, L. B.; PRACIANO, F. D. B. S.; CHAVES, I. C.; BRITO, F. T.; NETO, E. R. D.; MONTEIRO, J. M.; MACHADO, J. C. AUDIO-MC: A general framework for multi-context audio classification. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAM-

[1] The source code and other artifacts have been made available at <https://github.com/iagocc/smooth-noisy-max>

MOUDI, S. (Ed.). **Proceedings of the 24th International Conference on Enterprise Information Systems, ICEIS 2022, Online Streaming, April 25-27, 2022, Volume 1**. SCITEPRESS, 2022. p. 374–383. Available at: https://doi.org/10.5220/0011071500003179.

- CHAVES, I. C.; MARTINS, A. D. F.; PRACIANO, F. D. B. S.; BRITO, F. T.; MONTEIRO, J. M.; MACHADO, J. C. BPA: A multilingual sentiment analysis approach based on bilstm. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). **Proceedings of the 24th International Conference on Enterprise Information Systems, ICEIS 2022, Online Streaming, April 25-27, 2022, Volume 1**. SCITEPRESS, 2022. p. 553–560. Available at: https://doi.org/10.5220/0011071400003179.
- ALVES, D.; FARIAS, V. A. E. de; CHAVES, I. C.; CHAO, R.; MADEIRO, J. P.; GOMES, J. P. P.; MACHADO, J. C. Detecting customer induced damages in motherboards with deep neural networks. In: **International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022**. IEEE, 2022. p. 1–8. Available at: https://doi.org/10.1109/IJCNN55064.2022.9892047.
- SILVA, M. de L. M.; CHAVES, I. C.; MACHADO, J. C. Private reverse top-k algorithms applied on public data of COVID-19 in the state of ceará. **J. Inf. Data Manag.**, v. 12, n. 5, 2021. Available at: https://sol.sbc.org.br/journals/index.php/jidm/article/view/1941.
- LIMA, F. D. S.; PEREIRA, F. L. F.; CHAVES, I. C.; MACHADO, J. C.; GOMES, J. P. P. Predicting the health degree of hard disk drives with asymmetric and ordinal deep neural models. **IEEE Trans. Computers**, v. 70, n. 2, p. 188–198, 2021. Available at: https://doi.org/10.1109/TC.2020.2987018.
- PEREIRA, F. L. F.; CHAVES, I. C.; GOMES, J. P. P.; MACHADO, J. C. Using autoencoders for anomaly detection in hard disk drives. In: **2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020**. IEEE, 2020. p. 1–7. Available at: https://doi.org/10.1109/IJCNN48605.2020.9206689.
- SILVA, M. de L. M.; CHAVES, I. C.; MACHADO, J. de C. Aplicação de top-k reverso com privacidade sobre os dados públicos de COVID-19 no estado do ceará. In: **Proceedings of the 35th Brazilian Symposium on Databases, SBBD 2020, Online, September 28 - October 1, 2020**. SBC, 2020. p. 193–198. Available at: https://doi.org/10.5753/sbbd.2020.13640.
- CHAVES, I. C.; MACHADO, J. C. Differentially private group-by data releasing algorithm. In: **Proceedings of the 34th Brazilian Symposium on Databases, SBBD 2019,**

Fortaleza, CE, Brazil, October 7-10, 2019. SBC, 2019. p. 271–276. Available at: https://doi.org/10.5753/sbbd.2019.8835.

1.6 Organization

This thesis is organized as follows. In Chapter 2, we introduce the main concepts of differential privacy, *i.e.*, the definitions for the numerical setting.

Chapter 3 describes the concepts for the data selection setting and the related work. The chapter details the data selection competitors, such as exponential mechanism, permute-and-flip, report-noisy-max, and local dampening.

In Chapter 4, we show the core concepts of this work: Smooth Noisy Max algorithm and its differential privacy and accuracy statements.

Chapters 5,6 and 7 describe the novel percentile selection, greedy decision tree, and random forest algorithms, respectively, along with its experimental methodology and results.

Finally, in chapter 8, we conclude this thesis along with future work.

2 DIFFERENTIAL PRIVACY

In this chapter, we describe the main concepts on *Differential Privacy* that compose this thesis.

2.1 Probability and Random Variables

This section introduces fundamental statistical concepts essential for comprehending differential privacy definitions. We begin by defining *probability* from two perspectives: the frequentist and Bayesian approaches. From a frequentist standpoint, probability represents the long-term relative frequency of an event occurring under random conditions. Conversely, the Bayesian interpretation views probability as a measure of subjective belief in the likelihood of an event's occurrence.

The mathematical theory of probability, as formalized by Kolmogorov using set theory, is founded on the concept of a sample space Ω. This sample space is a set comprising all possible outcomes of the event or experiment under consideration. Within this framework, we denote a specific set of outcomes as $A \subseteq \Omega$, where A is a subset of Ω. The probability of this set of outcomes occurring is then expressed as $P[A]$.

A random variable is a function that assigns numerical values to the outcomes of a random experiment or process. More formally, it is a measurable function from a sample space Ω to a relevant space of numbers, *e.g.*, reals $\mathbb{R}$. Random variables can be discrete, taking on a countable set of values, or continuous, potentially assuming any value within a given range. They are fundamental to probability theory and statistics, providing a mathematical framework to describe and analyze uncertain events. To describe a random variable *X*, we need a probabilistic description, which turns out to be a function called the probability density function (PDF) of *X*.

The probability density function (PDF) characterizes the likelihood of a continuous random variable *X* taking values within an infinitesimal interval $[x, x + dx]$. This relationship is expressed as $P[x \leq X \leq x + dx] = \int_x^{x+dx} f_X(u)du$, where $f_X(u)$ is the PDF. We denote a random variable *X* distributed according to a PDF f_X as $X \sim f_X$. The cumulative distribution function (CDF), denoted by $F_X(x)$, is obtained by integrating the PDF: $F_X(x) = \int_{-\infty}^{x} f_X(u)du$. The CDF represents the probability that *X* takes on a value less than or equal to x.

The Laplace distribution is a widely used probability distribution in various fields, including differential privacy. It is characterized by its probability density function (PDF):

$f_{Lap}(x|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$ where $\mu \in \mathbb{R}$ is the location parameter and $b > 0$ is the scale parameter. In the context of differential privacy, we typically employ a centered Laplace distribution with $\mu = 0$ and a scale parameter b proportional to the global sensitivity of the query function. This specific parameterization allows for the calibration of noise addition to achieve the desired privacy guarantees.

In the context of differential privacy, consider a query with a true answer $q \in \mathbb{R}$. To protect privacy, we add noise drawn from a Laplace distribution to this query result: $\tilde{q} = q + X$ where $X \sim \text{Lap}(0, b)$ is a random variable following a Laplace distribution with location parameter 0 and scale parameter b. The probability density of obtaining any specific perturbed value $\tilde{q}$ is only determined by the probability density function of the Laplace noise.

Example 2. Consider a query with a true answer $q = 5$. We add Laplace noise to this value to obtain a perturbed answer $\tilde{q}$. The probability of obtaining a specific perturbed value, say $\tilde{q} = 5.1$, is equivalent to the probability of sampling a noise value of 0.1 from the Laplace distribution. Formally, we have $\tilde{q} = q + X$, where $X \sim \text{Lap}(0, b)$. For simplicity, let $b = 1$. The probability density of obtaining $\tilde{q} = 5.1$ is given by:

$$\begin{aligned} P(\tilde{q} = 5.1 | q = 5, b = 1) &= f_{\text{Lap}}(5.1 - 5|0, 1) \\ &= f_{\text{Lap}}(0.1|0, 1) \\ &= \frac{1}{2}\exp\left(-\frac{|0.1|}{1}\right) \\ &\approx 0.4524 \end{aligned}$$

Thus, the probability density of obtaining the perturbed value $\tilde{q} = 5.1$ is approximately 0.4524. Note that this is a probability density, not a probability, as we are dealing with a continuous distribution, *i.e.*, in our example, 0.4524 is not the probability of getting exactly 5.1, but it represents the relative likelihood of values around 5.1 compared to other possible values.

2.2 Database concepts

The data setting we consider is where a trusted curator holds a database $\mathbf{x}$ about n individuals, which we model as $\mathbf{x} \in \mathcal{X}$, for a *data universe* $\mathcal{X}$. We formalize the database as a *multiset* of records of $\mathcal{X}$. However, in the literature, it is common to represent the database as a histogram of the tuples in $\mathcal{X}$ or even an ordered n-tuple. Each specific definition is more convenient for different contexts, but the multiset representation will often be much more concise.

Definition 2.1 (Database)**.** A database $\mathbf{x}$ is a multiset of records so that $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$, where $x_i \in \mathcal{X}$ is a element of the universe of records $\mathcal{X}$.

Therefore, the distance between two databases can be determined by counting the records that differ between them. More specifically, this distance is quantified using the symmetric difference of two sets, denoted as $d(\mathbf{x}, \mathbf{y}) = |\mathbf{x} \oplus \mathbf{y}|$

Definition 2.2 (Distance between databases)**.** Consider two databases $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, the distance between them is defined as

$$d(\mathbf{x}, \mathbf{y}) = |\mathbf{x} \oplus \mathbf{y}|,$$

where $A \oplus B = (A \cup B) - (A \cap B)$.

A particular case of distance between databases is when they differ by at most one record, *i.e.*, $d(\mathbf{x}, \mathbf{y}) \leq 1$. We refer to these databases as *neighboring databases*. The concept of neighboring databases is crucial for the definition of differential privacy, as we will see in the next section.

In our definition of databases, two neighboring databases may arise from the addition, removal, or modification of a single tuple (KIFER; MACHANAVAJJHALA, 2011). In this thesis, we classify two databases as neighboring if they differ by at most one record as a result of the addition or deletion of a single tuple.

2.3 Formalizing differential privacy

A *privacy mechanism* is a randomized algorithm that takes the database as input and outputs a differentially private answer. A randomized algorithm with domain A and a discrete range $\mathcal{Y}$ is associated with a *probability simplex* over $\mathcal{Y}$, denoted by $\Delta(\mathcal{Y})$:

Definition 2.3 (Probability Simplex (DWORK; ROTH, 2014))**.** Given a discrete set $\mathcal{Y}$, the probability simplex over $\mathcal{Y}$, denoted by $\Delta(\mathcal{Y})$ is defined to be:

$$\Delta(\mathcal{Y}) = \left\{ x \in \mathbb{R}^{|\mathcal{Y}|} \mid x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{|\mathcal{Y}|} x_i = 1 \right\}$$

Definition 2.4 (Randomized Algorithm (DWORK; ROTH, 2014))**.** A randomized algorithm $\mathcal{M}$ with domain A and discrete range $\mathcal{Y}$ is associated with a mapping $\mathcal{M} : A \to \Delta(\mathcal{Y})$. On input $a \in A$, the algorithm $\mathcal{M}$ outputs $\mathcal{M}(a) = r$ with probability $(\mathcal{M}(a))_r$ for each $r \in \mathcal{Y}$.

As previously discussed, the output of the query f, denoted as $f(\mathbf{x})$, must be released without revealing significant information about individuals. To achieve this, a privacy-preserving mechanism should ensure that the output probability does not vary by more than a multiplicative factor of e^{ε} with the presence or absence of a single tuple. Consequently, we need to develop a randomized algorithm $\mathcal{M}(\mathbf{x})$ that introduces noise to $f(\mathbf{x})$ in a way that adheres to the differential privacy criteria outlined below.

Definition 2.5 ((ε, δ)-Differential privacy (DWORK; ROTH, 2014))**.** A randomized algorithm $\mathcal{M}$ satisfies (ε, δ)-differential privacy if, for any two neighboring databases $\mathbf{x}$ and $\mathbf{y}$, and for any possible output $\mathcal{Y}$ of the algorithm

$$Pr[\mathcal{M}(\mathbf{x}) \in \mathcal{Y}] \leq e^{\varepsilon} Pr[\mathcal{M}(\mathbf{y}) \in \mathcal{Y}] + \delta$$

where $Pr[\cdot]$ stands for probability of an event.

An alternative definition of differential privacy is provided in Remark 3.2 of Dwork and Roth (2014), using the concept of δ-approximate max divergence.

Definition 2.6 (δ-Approximate Max Divergence (DWORK; ROTH, 2014))**.**

$$D_{\infty}^{\delta}(X||Y) = \max_{S \subseteq \mathcal{Y} \,:\, Pr[X \in S] \geq \delta} \left[\log\left(\frac{Pr[X \in S] - \delta}{Pr[Y \in S]} \right) \right]$$

Definition 2.7 (Approx. Differential Privacy)**.** Note that a mechanism $\mathcal{M}$ is (ε, δ)-differentially private if and only if on every two neighboring databases $\mathbf{x}, \mathbf{y} \,:\, D_{\infty}^{\delta}(\mathcal{M}(\mathbf{x})||\mathcal{M}(\mathbf{y})) \leq \varepsilon$ and $D_{\infty}^{\delta}(\mathcal{M}(\mathbf{y})||\mathcal{M}(\mathbf{x})) \leq \varepsilon$.

When $\delta = 0$, the algorithm is ε-differentially private. We refer to ε-differential privacy as pure differential privacy. Conversely, (ε, δ)-differential privacy, where $\delta > 0$, is referred to as approximate differential privacy.

The parameter ε dictates how close the distribution of the outputs differs between databases $\mathbf{x}$ and $\mathbf{y}$. Lower values of ε ensure that these distributions are closely aligned, which enhances privacy at the expense of accuracy. Conversely, higher values of ε allow for greater variation between the distributions, improving accuracy but harming privacy. The data holder sets the value of ε, thereby defining the level of privacy offered.

Additionally, ε is referred to as the *privacy budget*. This concept is crucial when an analyst submits multiple queries to the database, as each query expends a portion of this budget.

The analyst must strategically allocate the privacy budget across different queries to balance the trade-off between data utility and privacy.

Along with the privacy budget ε, the parameter δ is used to control the probability of the algorithm deviating from the desired privacy guarantee. We can interpret δ as the "failure probability". With a certain probability, any outcome could occur, including the potential release of the entire sensitive dataset. For this reason, it's typically essential to ensure that this probability is kept very small. To help with understanding, we will introduce the concept of *privacy loss random variable.*

Definition 2.8 (Privacy loss random variable (DWORK; ROTH, 2014))**.** Let $\kappa \in \mathcal{Y}$, a randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$, $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ be a neighboring databases. Define the privacy loss random variable as

$$\mathcal{L}(\kappa) = \log\left(\frac{Pr[M(\mathbf{x}) = \kappa]}{Pr[M(\mathbf{y}) = \kappa]}\right)$$

Now, let us define the random variable $Z = \mathcal{L}(\mathcal{M}(\mathbf{x}))$, where Z implicitly depends on $\mathbf{x}$, $\mathbf{y}$, and $\mathcal{M}$. This random variable measures the extent to which observing the output of the algorithm $\mathcal{M}$ helps in distinguishing between the databases $\mathbf{x}$ and $\mathbf{y}$. Now, we can describe the δ parameter in terms of the privacy loss random variable. Consider $\mathcal{M}$ being (ε, δ)-differentially private, then $Pr[Z > \varepsilon] \leq \delta$, *i.e.*, the algorithm $\mathcal{M}$ will get the same privacy guarantees as ε-differential privacy with probability at least $1 - \delta$, and with probability δ the algorithm may fail to provide the desired privacy guarantees.

2.3.1 Composition

The analyst can pose several queries to the database to compose complex differentially private algorithms. There are two types of composition: sequential and parallel.

The sequential composition happens when a set of mechanisms is executed against a dataset. This implies that the privacy budget used on each computation sums up:

Theorem 2.9 (Sequential composition (DWORK; ROTH, 2014))**.** Let $\mathcal{M}_i : \mathcal{X} \to \mathcal{Y}$ be an $(\varepsilon_i, \delta_i)$-differentially private algorithm for $i \in [k]$. Then $\mathcal{M}(\mathbf{x}) = (\mathcal{M}_1(\mathbf{x}), \cdots, \mathcal{M}_k(\mathbf{x}))$ is $\left(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i\right)$-differentially private.

The Theorem 2.9 implies that if an analyst is given a privacy budget ε, she can execute any number of private queries as long as the sum of the budget used in each execution accumulates to ε.

On the other hand, if queries are applied to disjoint subsets of the database, then we can save privacy budget. This is the scenario for parallel composition. When the analyses are carried out with many $(\varepsilon_i, \delta_i)$-differentially private mechanisms operating on disjoint subsets, it composes a $(\max_i \varepsilon_i, \max_i \delta_i)$-differentially private mechanism which has a lower privacy cost.

Theorem 2.10 (Parallel composition (DWORK; ROTH, 2014))**.** Let $\mathcal{M}_i : \mathcal{X} \to \mathcal{Y}$ be an $(\varepsilon_i, \delta_i)$-differentially private algorithm for $i \in [k]$. Then $\mathcal{M}(\mathbf{x}) = (\mathcal{M}_1(\mathbf{x}), \cdots, \mathcal{M}_k(\mathbf{x}))$ is $(\max_{i=1}^{k} \varepsilon_i$, $\max_{i=1}^{k} \delta_i)$-differentially private.

2.3.2 Sensitivity

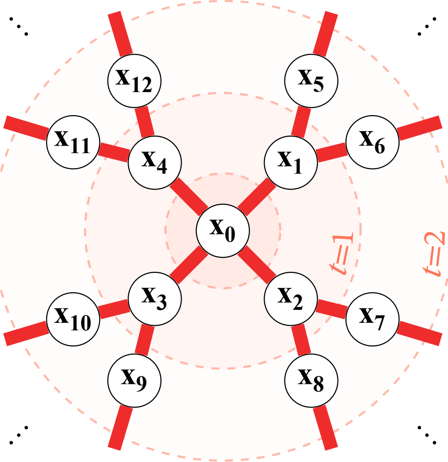
Several randomized algorithms may satisfy the definition of differential privacy. Nevertheless, the applicability of these algorithms may only be suitable across some problem domains (NISSIM *et al.*, 2007). The Laplace mechanism is one of the most well-known mechanisms for adding noise to a numerical query. It is a mechanism that draws noise from the Laplace distribution and adds to the query result. The Laplace distribution is a continuous probability distribution that, in this context, is centered at zero and has a single parameter, which is the scale parameter that is proportional to the global sensitivity of the query f. Usually, not only the Laplace mechanism, the quantity of noise introduced is proportional to the global sensitivity of the query. Global sensitivity quantifies the maximum change in a function's output when a single individual's data is modified, reflecting the largest difference between outputs for databases differing by one record. Figure 2 illustrates the global sensitivity concept, in the figure the edges represent the term $|f(\mathbf{x_i}) - f(\mathbf{x_j})|$ for two neighbors $\mathbf{x_i}$ and $\mathbf{x_j}$, so the global sensitivity is the maximum value among all red edges.

Definition 2.11 (Global sensitivity (DWORK; ROTH, 2014))**.** The global sensitivity of a function $f : \mathcal{X} \to \mathbb{R}$ is defined:

$$\Delta f = \max_{\substack{\mathbf{x},\mathbf{y} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq 1}} |f(\mathbf{x}) - f(\mathbf{y})|$$

However, the global sensitivity practical utility is often limited due to excessive noise generation, as the Laplace mechanism's scale parameter is $\Delta f/\varepsilon$ (DWORK; ROTH, 2014), leading to high noise levels for functions like k-clique counting (ZHANG *et al.*, 2015) and median queries (NISSIM *et al.*, 2007). Local sensitivity, which is database-specific, measures the

Figure 2 – Global Sensitivity. The edges represent the term $|f(\mathbf{x_i}) - f(\mathbf{x_j})|$ for two neighbors $\mathbf{x_i}$ and $\mathbf{x_j}$. The global sensitivity is the maximum value among all red edges. The value of t represents the distance from $\mathbf{x_0}$



Source: adapted from Farias (2021)

maximal output change for individual data modifications, and using instance-based sensitivity can reduce the introduced noise. Figure 3 illustrates the local sensitivity concept, where the edges represent the term $|f(\mathbf{x_i}) - f(\mathbf{x_j})|$ for two neighbors $\mathbf{x_i}$ and $\mathbf{x_j}$, so the local sensitivity $LS_f(\mathbf{x_0})$ is the maximum value among all red edges incident on $\mathbf{x_0}$.

Definition 2.12 (Local sensitivity (NISSIM *et al.*, 2007))**.** For a query $f : \mathcal{X} \to \mathbb{R}$ and a database $\mathbf{x} \in \mathcal{X}$, the local sensitivity of f at $\mathbf{x}$ is defined as:

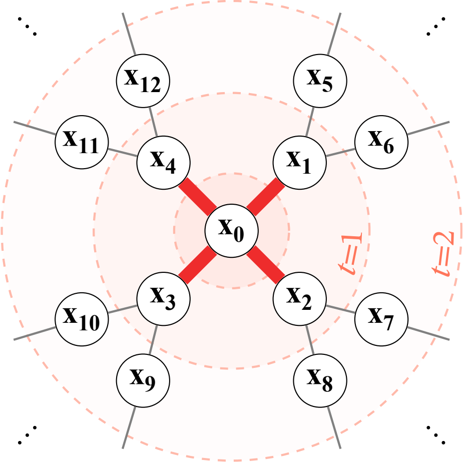
$$LS_f(\mathbf{x}) = \max_{\substack{\mathbf{y} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq 1}} |f(\mathbf{x}) - f(\mathbf{y})|$$

It is crucial to highlight that the global sensitivity is the maximum local sensitivity over all databases, $\Delta f = \max_{\mathbf{x} \in \mathcal{X}} LS_f(\mathbf{x})$. Nonetheless, using the local sensitivity, instead of global, would reduce the amount of noise produced by the random algorithm so much that it would not satisfy the differential privacy definition (NISSIM *et al.*, 2007).

To address the problem of achieving differential privacy for numerical queries with instance-based sensitivity, the work of Nissim *et al.* (2007) proposed the smooth sensitivity framework, which smooths the local sensitivity at a distance t.

The local sensitivity at a distance t measures the maximum local sensitivity LS_f over all databases up to the distance t from $\mathbf{x}$, *i.e.*, up to t modifications on the database $\mathbf{x}$. It is important to note that it is a generalization of the local sensitivity $LS_f(\mathbf{x}, 0) = LS_f(\mathbf{x})$, a particular case when the distance is set to 0. Figure 4 illustrates the local sensitivity at a distance $t = 1$, where the edges represent the term $|f(\mathbf{x_i}) - f(\mathbf{x_j})|$ for two neighbors $\mathbf{x_i}$ and $\mathbf{x_j}$, so the local

Figure 3 – Local Sensitivity. The edges represent the term $|f(\mathbf{x_i}) - f(\mathbf{x_j})|$ for two neighbors $\mathbf{x_i}$ and $\mathbf{x_j}$. The local sensitivity $LS_f(\mathbf{x_0})$ is the maximum value among all red edges incident on $\mathbf{x_0}$. The value of t represents the distance from $\mathbf{x_0}$.



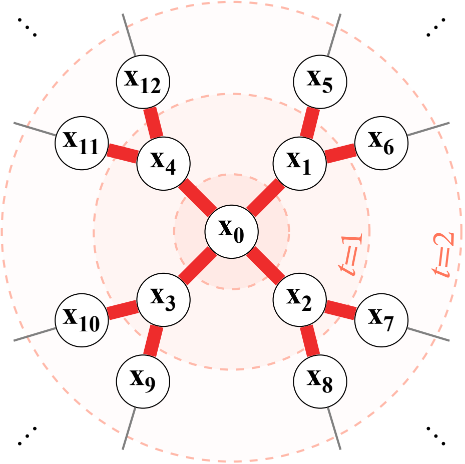
Source: adapted from Farias (2021)

sensitivity at distance $t = 1$ $LS_f(\mathbf{x_0}, 1)$ is the maximum value among all red edges incident on all the database at most distance 1 from $\mathbf{x_0}$ (which includes itself).

Definition 2.13 (Local sensitivity at distance t (NISSIM *et al.*, 2007))**.** For a query $f : \mathcal{X} \rightarrow \mathbb{R}^k$ and a database $\mathbf{x} \in \mathcal{X}$, the local sensitivity of f at $\mathbf{x}$ at distance t is defined as:

$$LS_f(\mathbf{x}, t) = \max_{\substack{\mathbf{y} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq t}} LS_f(\mathbf{y})$$

Figure 4 – Local sensitivity at distance $t = 1$. The edges represent the term $|f(\mathbf{x_i}) - f(\mathbf{x_j})|$ for two neighbors $\mathbf{x_i}$ and $\mathbf{x_j}$. The local sensitivity at distance $t = 1$ $LS_f(\mathbf{x_0}, 1)$ is the maximum value among all red edges incident on all the database at most distance 1 from $\mathbf{x_0}$ (which includes itself). The value of t represents the distance from $\mathbf{x_0}$.



Source: adapted from Farias (2021)

The sensitivity must itself be insensitive. To determine the appropriate noise magnitude, the work by NISSIM *et al.* (NISSIM *et al.*, 2007) utilizes a smooth upper bound on local sensitivity. Specifically, they define a function S that not only provides an upper limit on LS_f across all points but also ensures that $\ln(S(\cdot))$ maintains low sensitivity.

Definition 2.14 (Smooth bound (NISSIM *et al.*, 2007))**.** For a parameter $\beta > 0$, a database $\mathbf{x} \in \mathcal{X}$, a function $S : \mathcal{X} \to \mathbb{R}^+$ is a β-smooth upper bound on the local sensitivity of a function f if it satisfies the following requirements:

$$\forall \mathbf{x} \in \mathcal{X} : S(\mathbf{x}) \geq LS_f(\mathbf{x})$$

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}, d(\mathbf{x}, \mathbf{y}) \leq 1 : S(\mathbf{x}) \leq e^{\beta} S(\mathbf{y})$$

Definition 2.15 (Smooth sensitivity (NISSIM *et al.*, 2007))**.** For $\beta > 0$, a query f, a database $\mathbf{x} \in \mathcal{X}$, the β-smooth sensitivity of f is:

$$\mathcal{S}_{f,\beta}(\mathbf{x}) = \max_{t=0,1,\ldots,|\mathbf{x}|} \left(e^{-t\beta} \cdot LS_f(\mathbf{x}, t) \right)$$

The smooth sensitivity $\mathcal{S}_{f,\beta}$ is the smallest function to satisfy the smooth bound requirements (Def. 2.14).

Lemma 2.16 (Lemma 2.3 from Nissim *et al.* (2007))**.** $\mathcal{S}_{f,\beta}$ is a β-smooth upper bound on LS_f. In addition, $\mathcal{S}_{f,\beta}(\mathbf{x}) \leq S(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$ for every β-smooth upper bound S on LS_f.

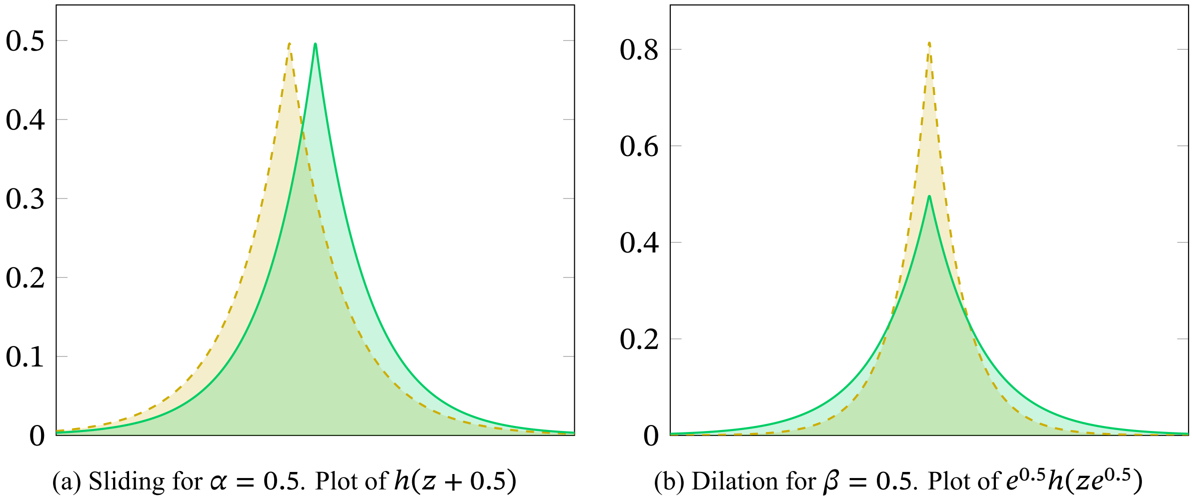
The smooth sensitivity decays the local sensitivity based on how far the neighboring database is from $\mathbf{x}$. The β parameter, which serves as a smoothing factor, is strategically chosen to mitigate inadvertent data disclosure risks that may arise when employing local sensitivity directly. The global sensitivity Δf is also a smooth upper bound of local sensitivity, *i.e.*, global sensitivity satisfies the Definition 2.14.

Corollary 2.17 (Smooth sensitivity upper bound)**.** For a query f, a database $\mathbf{x}$, the global sensitivity Δ_f is an upper bound of smooth sensitivity $\mathcal{S}_{f,\beta}$ *i.e.*, $\mathcal{S}_{f,\beta}(\mathbf{x}) \leq \Delta f$.

2.3.3 Calibrating noise according to a smooth upper bound

Mechanisms that the addition of noise is proportional to the smooth sensitivity are contingent upon whether the noise distribution meets the criteria necessary for achieving differential privacy, *i.e.*, (α, β)-admissibility.

Figure 5 – Example of sliding and dilation properties. The green curve represents the noise distribution $h(z) = (1/2) \cdot e^{-|z|}$, and the yellow dashed curve represent the transformed version.



(a) Sliding for $\alpha = 0.5$. Plot of $h(z + 0.5)$

(b) Dilation for $\beta = 0.5$. Plot of $e^{0.5}h(ze^{0.5})$

Source: elaborated by the author.

Definition 2.18 (Admissible Noise Distribution (NISSIM *et al.*, 2007))**.** A probability distribution on $\mathbb{R}^k$, given by a density function h, is (α, β)-admissible if, for $\alpha = \alpha(\varepsilon, \delta)$, $\beta = \beta(\varepsilon, \delta)$, the following two conditions hold for all $||\Delta||_1 \in \mathbb{R}^k$ and $\lambda \in \mathbb{R}$ satisfying $\Delta \leq \alpha$ and $|\lambda| \leq \beta$, and for all measurable subsets $S \subseteq \mathbb{R}^k$.

(i) *(Sliding)* $\quad \underset{Z\sim h}{Pr}[Z \in S] \leq e^{\frac{\varepsilon}{2}} \underset{Z\sim h}{Pr}[Z \in S + \Delta] + \frac{\delta}{2}$

(ii) *(Dilation)* $\quad \underset{Z\sim h}{Pr}[Z \in S] \leq e^{\frac{\varepsilon}{2}} \underset{Z\sim h}{Pr}[Z \in S \cdot e^{\lambda}] + \frac{\delta}{2}$

The Definition 2.18 stipulates that the noise distribution should remain relatively stable under translation (sliding) and scaling (dilation). Refer to Figure 5 for an illustration. A distribution that adheres to these properties is suitable for adding noise proportional to a smooth upper bound on local sensitivity.

Lemma 2.19 (Lemma 2.6 from Nissim *et al.* (2007))**.** Let h be an (α, β)-admissible noise distribution. Then, for a query $f : \mathcal{X} \to \mathcal{Y}$, $S : \mathcal{X} \to \mathbb{R}$ be a β-smooth upper bound on the local sensitivity of f, and a database $\mathbf{x}$, the mechanism $\mathcal{M}(\mathbf{x}) = f(\mathbf{x}) + \frac{S(\mathbf{x})}{\alpha}Z$ is (ε, δ)-differentially private, where $Z \sim h$.

We discuss three admissible distributions that are employed in this thesis. Our first choice is the Laplace distribution, the second is the Laplace Log-Normal distribution, and finally, the Student's t-distribution.

Lemma 2.20 (Lemma 2.9 from Nissim *et al.* (2007))**.** For $\varepsilon, \beta > 0$, the Laplace distribution $h(z) = \frac{1}{2}e^{-|z|}$ is (ε, β)-admissible with $\alpha = \frac{\varepsilon}{2}$, and $\beta = \frac{\varepsilon}{2\ln(2/\delta)}$.

Theorem 2.21 (Theorem 18 from Bun and Steinke (2019))**.** Let X and Y be independent random variables with X a standard Laplace and Y a standard Gaussian. Let $\sigma > 0$ and $Z = X \cdot e^{\sigma Y}$. The distribution of Z is denoted by $\mathtt{LLN}(\sigma)$. Z is (ε, β)-admissible with $\alpha = e^{-3/2\sigma^2}(\varepsilon - \beta/\sigma)$.

Theorem 2.22 (Theorem 31 from Bun and Steinke (2019))**.** Let Z be a random variable following the Student's t-distribution with ν degrees of freedom. Z is (ε, β)-admissible with $\alpha = \frac{2\sqrt{\nu}}{\nu+1}$.

Both the Laplace and Laplace Log-Normal distributions provide approximate differential privacy ($\delta > 0$). Nevertheless, the Student's t-distribution yield pure differential privacy ($\delta = 0$) under the setting of the Lemma 2.19.

2.4 Discussion

In this chapter, we have presented the fundamental concepts of differential privacy, including the definition, composition, sensitivity, and noise calibration under β-smooth upper bound setting. There are two points that are important to highlight:

Promises: Differential privacy ensures that individuals do not face additional risks from their data being in a private database $\mathbf{x}$ that they would not encounter if their data were absent from $\mathbf{x}$. While the release of results $\mathcal{M}(\mathbf{x})$ from a differentially private mechanism $\mathcal{M}$ may still lead to potential harm, differential privacy assures that the likelihood of harm has not substantially increased due to an individual's decision to contribute their data. This approach to privacy is fundamentally utilitarian; it hinges on the difference in harm probabilities that an individual evaluates when deciding whether to include their data in a database managed under differential privacy principles—specifically, weighing the risk of participation against the risk of non-participation.

Limitations: While differential privacy offers a robust guarantee, it does not provide absolute immunity from harm or create privacy where none existed before. More importantly, differential privacy does not ensure that personal secrets will remain undisclosed. Rather, it ensures that the act of participating in a survey does not reveal one's involvement or any specific information contributed. It is entirely possible for survey results to reflect statistical data about an individual. For instance, a health survey designed to identify early indicators of a disease might yield significant, or even definitive, results. The applicability of these conclusions to a specific individual, however, does not constitute a breach of privacy. This holds true even if the individual did not participate in the survey, as differential privacy guarantees that similar conclusions would likely be reached regardless of individual participation.

3 PRIVATE SELECTION – RELATED WORK

This chapter reviews the literature on private selection, a central theme of this thesis, which specifically tackles the private selection problem. It was challenging to separate this chapter from the theoretical background, as several detailed methods discussed herein form the cornerstone of differential privacy. Consequently, this chapter provides an in-depth exploration of various differentially private selection methods, which will serve as benchmarks for our proposed approach. These methods were identified through a snowballing process and a comprehensive review of the relevant literature.

Private selection refers to selecting the best item, or outcome, option from a set of possible outputs while ensuring the individual's data privacy. Formally, we want to build a private algorithm for a query $f : \mathcal{X} \rightarrow \mathcal{R}$ where all possible outcomes for f are discrete, *e.g.*, categorical values. In the private selection setting is necessary a utility function $u : \mathcal{X} \times \mathcal{R} \rightarrow \mathbb{R}$ that maps a database $\mathbf{x}$ and an output $r \in \mathcal{R}$ to a utility score $u(\mathbf{x}, r)$. This utility function is application-based, and the higher the utility values are, the better the outcome is for the database.

There are differentially private data selection algorithms directly related to this work, for instance, the well-established exponential mechanism (MCSHERRY; TALWAR, 2007) and the report-noisy-max algorithm (DWORK; ROTH, 2014). The recent ones include the permute-and-flip (MCKENNA; SHELDON, 2020) and the local dampening mechanism (FARIAS *et al.*, 2023).

3.1 Private selection setting

In the private selection setting, it is necessary to design a utility function that measures the quality of the database output. The utility function is application-based, and the higher the utility values are, the better the outcome. In the sense of private selection, we want the randomized algorithm to maximize the utility function approximately, *i.e.*, the probability of selecting an output with high utility is high.

Definition 3.1 (Utility function)**.** Consider a database $\mathbf{x} \in \mathcal{X}$, an output set $\mathcal{R}$. A utility function $u : \mathcal{X} \times \mathcal{R} \rightarrow \mathbb{R}$ maps a pair of a database and an outcome to a score.

In contrast to the standard differential privacy setup (numerical), we now want to understand the sensitivity of the utility u instead of the query f. But, notice that the utility function u is directly related to the query f. The global sensitivity of the utility function is the maximum change in the utility function when the database changes by one tuple for all possible outcomes $r \in \mathcal{R}$.

Definition 3.2 (Global Sensitivity (MCSHERRY; TALWAR, 2007))**.** Let $u : \mathcal{X} \times \mathcal{R} \to \mathbb{R}$ be a utility function that maps a pair of a database and an outcome to a score. The global sensitivity of a function u is defined as:

$$\Delta u = \max_{r \in \mathcal{R}} \max_{\substack{\mathbf{x},\mathbf{y} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq 1}} |u(\mathbf{x}, r) - u(\mathbf{y}, r)|$$

Observe that the key distinction between Definition 3.2 and the standard global sensitivity definition (Definition 2.11) lies in the use of the max operator across all possible outcomes. A similar distinction should also be noted in the definitions of local sensitivities.

Definition 3.3 (Local Sensitivity for private selection (FARIAS *et al.*, 2023))**.** Let $u : \mathcal{X} \times \mathcal{R} \to \mathbb{R}$ be a utility function that maps a pair of a database and an outcome to a score. The local sensitivity is defined as:

$$LS_u(\mathbf{x}) = \max_{r \in \mathcal{R}} \max_{\substack{\mathbf{y} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq 1}} |u(\mathbf{x}, r) - u(\mathbf{y}, r)|$$

Definition 3.4 (Local Sensitivity at distance t for private selection (FARIAS *et al.*, 2023))**.** Let $u : \mathcal{X} \times \mathcal{R} \to \mathbb{R}$ be a utility function that maps a pair of a database and an outcome to a score. The local sensitivity of a function u for the database $\mathbf{x}$ at distance t is defined as:

$$LS_u(\mathbf{x}, t) = \max_{\substack{\mathbf{y} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq t}} LS_u(\mathbf{y}) = \max_{\substack{\mathbf{y},\mathbf{z} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq t \\ d(\mathbf{y},\mathbf{z}) \leq 1}} \max_{r \in \mathcal{R}} |u(\mathbf{y}, r) - u(\mathbf{z}, r)|$$

The local sensitivity at distance t measures the maximum sensitivity of u across all elements r in $\mathcal{R}$ for a database $\mathbf{x}$ undergoing t modifications. This metric provides an overview of how u varies between neighboring databases. However, if even a single element in $\mathcal{R}$ presents high sensitivity (approaching Δu), $LS_u(\mathbf{x}, t)$ will be proportionately high. This can be problematic when most elements exhibit low sensitivity while only a few have high sensitivity, leading to an inflated $LS_u(\mathbf{x}, t)$ and compromising accuracy. To tackle that problem, the work of Farias *et al.* (2023) proposed a more specialized definition of local sensitivity named element local sensitivity, which measures the sensitivity of u for a given $r \in \mathcal{R}$ for an input database $\mathbf{x}$ at a distance t.

Definition 3.5 (Element Local Sensitivity at distance t)**.** Let $u : \mathcal{X} \times \mathcal{R} \rightarrow \mathbb{R}$ be a utility function that maps a pair of a database and an outcome to a score. The local sensitivity of a function u for the database $\mathbf{x}$ for specific outcome $r \in \mathcal{R}$ at distance t is defined as:

$$LS_u(\mathbf{x}, t, r) = \max_{\substack{\mathbf{y},\mathbf{z} \in \mathcal{X} \\ d(\mathbf{x},\mathbf{y}) \leq t \\ d(\mathbf{y},\mathbf{z}) \leq 1}} |u(\mathbf{y}, r) - u(\mathbf{z}, \mathbf{r})|$$

3.2 Exponential Mechanism

The exponential mechanism in the private selection setting is the *de facto* standard. It samples possible outputs from $\mathcal{R}$ with a probability that grows exponentially with their utility function u.

Definition 3.6 (Exponential Mechanism (MCSHERRY; TALWAR, 2007))**.** The exponential mechanism $\mathcal{M}_{u,\varepsilon}^{\text{exp}}(\mathbf{x}, r)$ selects an outcome from $r \in \mathcal{R}$ as follows:

$$\mathcal{M}_{u,\varepsilon}^{\text{exp}}(\mathbf{x}, r) \propto \exp\left(\frac{\varepsilon u(\mathbf{x}, r)}{2\Delta u}\right),$$

where Δu is the global sensitivity of the utility function u.

McSherry and Talwar (2007) showed that the exponential mechanism satisfies ε-differential privacy through global sensitivity. Therefore, the algorithm's utility is regulated by global sensitivity. We refer to algorithms' utility as inversely proportional to the error.

It is essential to understand how well the private selection approach can perform, *i.e.*, grasp how useful the exponential mechanism can be. However, the probability of choosing an output with near-maximum utility depends on the problem structure.

Theorem 3.7 (Utility of the exponential mechanism)**.** Fixing a database $\mathbf{x} \in \mathcal{X}$, for a given $t > 0$, and ξ representing the error, set of all possible outcome $\mathcal{R}$, with $\mathcal{M}_{u,\varepsilon}^{\text{exp}}$ exponential mechanism we have:

(i) $Pr\left[\xi(\mathcal{M}_{u,\varepsilon}^{\text{exp}}, \mathbf{x}) \geq \frac{2\Delta u\,(\ln(|\mathcal{R}|)+t)}{\varepsilon}\right] \leq e^{-t}$

(ii) $\mathbb{E}\left(\xi(\mathcal{M}_{u,\varepsilon}^{\text{exp}}, \mathbf{x})\right) \leq \frac{2\Delta u\,(\ln(|\mathcal{R}|)+1)}{\varepsilon}$

The Theorem 3.7 shows that the utility bounds of the exponential mechanism depend on the privacy budget, the global sensitivity, and the number of possible outcomes. For the sake of simplicity, sometimes this work refers to $\mathcal{M}_{u,\varepsilon}^{\text{exp}}(\cdot) = \mathcal{M}^{\text{exp}}(\cdot)$.

We conclude that the exponential mechanism is a powerful tool applicable across various problems, providing robust differential privacy guarantees. However, calculating the probability of a specific outcome involves evaluating the probabilities of all potential outcomes, resulting in high computational complexity. Another drawback is its dependence on global sensitivity, which can negatively impact the utility-to-noise ratio.

3.3 Permute-and-flip

Another private selection algorithm, called Permute-and-Flip, was proposed by McKenna and Sheldon (2020). The work proves that the expected error of permute-and-flip is never worse than that of the exponential mechanism. Moreover, it shows that the exponential mechanism can be viewed as a rejection sampling algorithm that samples uniformly from the outcome set $\mathcal{R}$ with replacement. On the other hand, the permute-and-flip works like an exponential mechanism but sampling without replacement from $\mathcal{R}$.

Algorithm 1: Permute-and-Flip Mechanism

```
$u_*(x) \leftarrow max_{r \in \mathcal{R}} u(\mathbf{x}, r)$;
repeat
    $r \sim$ Uniform[$\mathcal{R}$];
    $p_{\mathbf{x},r} = \exp\left(\frac{\varepsilon \cdot (u(\mathbf{x},r) - u_*(\mathbf{x}))}{2\Delta u}\right)$;
    $\mathcal{R} = \mathcal{R} \backslash r$ ;                    /* Without replacement */
until $Bernoulli(p_{\mathbf{x},r})$;
return $r$
```

The permute-and-flip algorithm (Algorithm 1) works by iterating over the set of outcomes $\mathcal{R}$ in a random order, and for each element r, it flips a biased coin with a certain probability. If the flipped coin lands tails, then r is removed from all possible outcomes. Otherwise (if it lands heads), r is the returned outcome for the mechanism. The likelihood of obtaining heads follows an exponential pattern concerning the quality score, thereby boosting the mechanism to produce results with superior quality scores. While the permute-and-flip algorithm achieves ε-differential privacy, this guarantee only applies under the global sensitivity Δu.

The paper by McKenna and Sheldon (2020) demonstrates that the permute-and-flip mechanism consistently maintains an expected error that does not exceed that of the exponential mechanism. Additionally, the likelihood of the error variable surpassing any given threshold t is always equal to or lower for permute-and-flip compared to the exponential mechanism. As a result, permute-and-flip inherits the robust theoretical guarantees associated with the exponential mechanism.

Definition 3.8 (Never worse). An algorithm $\mathcal{A}$ is said to be *never worse* than some other algorithm $\mathcal{B}$ when, given a dataset $\mathbf{x}$, and $t > 0$:

(i) $Pr\,[\xi\,(\mathcal{A}, \mathbf{x}) \geq t] \leq Pr\,[\xi\,(\mathcal{B}, \mathbf{x}) \geq t]$ for all $t \geq 0$;

(ii) $\mathbb{E}[\xi\,(\mathcal{A}, \mathbf{x})] \leq \mathbb{E}[\xi\,(\mathcal{B}, \mathbf{x})]$.

Theorem 3.9 (Theorem 2 of McKenna and Sheldon (2020)). $\mathcal{M}^{\text{pf}}_{u,\varepsilon}$ is never worse than $\mathcal{M}^{\text{exp}}$. That is, for any utility function u, and $t > 0$.

(i) $Pr[\xi(\mathcal{M}^{\text{pf}}_{u,\varepsilon}) \geq t] \leq Pr[\xi(\mathcal{M}^{\text{exp}}_{u,\varepsilon}) \geq t]$ for all $t \geq 0$;

(ii) $\mathbb{E}[\xi(\mathcal{M}^{\text{pf}}_{u,\varepsilon}, \mathbf{x})] \leq \mathbb{E}[\xi(\mathcal{M}^{\text{exp}}_{u,\varepsilon}, \mathbf{x})]$.

Theorem 3.10 (Utility of Permute-and-flip). Fixing a database $\mathbf{x} \in \mathcal{X}$, for a given $t > 0$, and ξ representing the error, set of all possible outcome $\mathcal{R}$, with $\mathcal{M}^{\text{pf}}_{u,\varepsilon}$ permute-and-flip have:

(i) $Pr\left[\xi(\mathcal{M}^{\text{pf}}_{u,\varepsilon}, \mathbf{x}) \geq \frac{2\Delta u\,(\ln(|\mathcal{R}|)+t)}{\varepsilon}\right] \leq e^{-t}$

(ii) $\mathbb{E}\left(\xi(\mathcal{M}^{\text{pf}}_{u,\varepsilon}, \mathbf{x})\right) \leq \frac{2\Delta u\,(\ln(|\mathcal{R}|)+1)}{\varepsilon}$

McKenna and Sheldon (2020) introduce a novel approach for differentially private selection that demonstrates improved error rates compared to the exponential mechanism. However, it falls short of establishing tighter utility bounds beyond those offered by the exponential mechanism. While Permute-and-flip addresses the computational complexity issues associated with the exponential mechanism, it remains dependent on global sensitivity.

3.4 Report-noisy-max

The report-noisy-max algorithm adds independent noise to each outcome utility score and returns the outcome with the highest noisy score. Dwork and Roth (2014) proposes the algorithm with noise sampled by the Laplace distribution. However, the algorithm can be generalized to other noise distributions, such as the Gumbel and Exponential distributions.

The algorithm, initially proposed by Dwork and Roth (2014), represents a comprehensive method for private selection that is adaptable across a wide range of probability distributions. It can be effectively compared to the permute-and-flip technique, which serves as a specific instance of this broader method Ding *et al.* (2021). Additionally, in certain scenarios, the report-noisy-max algorithm mirrors the behavior of the exponential mechanism, as outlined by Durfee and Rogers (2019). Algorithm 2 illustrates the application of the report-noisy-max using the exponential distribution.

Algorithm 2: Report-noisy-max algorithm - Exponential noise

```
for $r \in \mathcal{R}$ do
    $\tilde{u}(\mathbf{x}, r) \leftarrow u(\mathbf{x}, r) + \texttt{Expo}\left(\frac{\varepsilon}{2\Delta u}\right)$;
end
return $\arg\max_{r \in \mathcal{R}} \tilde{u}(\mathbf{x}, r)$
```

Specifically, the report-noisy-max with the Exponential distribution, denoted by $\mathcal{N}^{\texttt{exp}}$, samples noise from $\texttt{Expo}\,(\varepsilon/2\Delta u)$, and this version also has strong utility guarantees shown by the Theorem 3.11. It is identical to permute-and-flip (DING *et al.*, 2021).

Moreover, the report-noisy-max with the Gumbel distribution $\texttt{Gumbel}\,(2\Delta u/\varepsilon)$ is identical to the exponential mechanism (DURFEE; ROGERS, 2019). Nevertheless, the report-noisy-max only holds the differential privacy requirements under the global sensitivity of the utility function, which might lead to poor accuracy under certain scenarios (ZHANG *et al.*, 2015; GONEM; GILAD-BACHRACH, 2018; SUN *et al.*, 2020; BUN; STEINKE, 2019).

It is vital to notice that the report-noisy-max algorithm reports only the outcome with the highest noisy utility, nothing concerning the noisy utility value, nor something related to the other outcomes, *i.e.*, the method returns strictly what is wanted. The report-noisy-max inadvertently discards information (DING *et al.*, 2023). More precisely, without incurring any supplementary privacy costs, it can disclose an estimation of the difference between the largest and second-largest noisy utility values.

Besides all those strengths, it is also possible to grasp the utility bounds of the report-noisy-max.

Theorem 3.11. Consider the report-noisy-max with exponential distribution $\mathcal{N}^{\texttt{exp}}$ algorithm. Let $\mathbf{x} \in \mathcal{X}$ be a fixed database, ξ be the error, and $\mathcal{R}$ be the set of all possible outcomes. Then, for a given $t > 0$, the following inequalities hold:

(i) $Pr\left[\xi(\mathcal{N}^{\texttt{exp}}, \mathbf{x}) \geq \frac{2\Delta u\,(\ln(|\mathcal{R}|)+t)}{\varepsilon}\right] \leq e^{-t}$;

(ii) $\mathbb{E}\left(\xi(\mathcal{N}^{\texttt{exp}}, \mathbf{x})\right) \leq \frac{2\Delta u\,(\ln(|\mathcal{R}|)+1)}{\varepsilon}$.

The Report-Noisy-Max algorithm serves as a foundational component for differentially private selection methods, capable of being adapted to various noise distributions. Each type of noise uniquely affects the algorithm's performance, enabling it to achieve specific utility bounds. This adaptability makes it a powerful tool for tackling a diverse range of problems. Additionally, unlike the exponential mechanism, the Report-Noisy-Max algorithm does not encounter issues with time complexity, enhancing its practicality for real-world applications.

3.5 Local Dampening Mechanism

In specific scenarios, the global sensitivity may not be suitable because the global sensitivity may increase the signal-to-sensitivity ratio, implying inaccurate results. To address this issue, the local dampening mechanism (FARIAS *et al.*, 2023) designs an instance-based sensitivity to work along with a novel mechanism based on the exponential one. It also proposes new adapted versions of the local sensitivity at a distance t to the private selection setup.

3.5.1 Sensitivity functions

Whereas the local sensitivity at distance t provides an overview of the utility u variation in its neighborhood, it lacks in granting more information about the utility function u with a specific outcome r in its neighborhood. Therefore, Farias *et al.* (2023) proposes a novel generalization of local sensitivity called the element local sensitivity. It measures the sensitivity of a utility function u for a specific outcome r at a distance t, as defined in Definition 3.5.

The computation of the element's local sensitivity is only sometimes feasible because it could be NP-hard. Therefore, the paper proposes a heuristic to compute an upper bound to the element's local sensitivity, referred to as admissible function $\delta_u : \mathcal{X} \times \mathbb{N} \times \mathcal{R} \rightarrow \mathbb{R}$. As δ_u works like a upper bound on local sensitivity $\delta_u(\mathbf{x}, t, r) = \Delta u$, $\delta_u(\mathbf{x}, t, r) = LS_u(\mathbf{x})$, and $\delta_u(\mathbf{x}, t, r) = LS_u(\mathbf{x}, t, r)$ are possible sensitive functions δ_u.

The paper outlines four desired behaviors for the sensitivity functions: admissibility, boundedness, monotonicity, and stability.

Definition 3.12 (Admissibility)**.** Consider a $\mathbf{x} \in \mathcal{X}$, $r \in \mathcal{R}$. A sensitivity function $\delta_u(\mathbf{x}, t, r)$ is *admissible* if:

1. $\delta_u(\mathbf{x}, 0, r) \geq LS_u(\mathbf{x}, 0, r)$, for all $\mathbf{x} \in \mathcal{X}$ and all $r \in \mathcal{R}$
2. $\delta_u(\mathbf{x}, t + 1, r) \geq \delta_u(\mathbf{y}, t, r)$, for all $\mathbf{x}, \mathbf{y}$ such that $d(\mathbf{x}, \mathbf{y}) \leq 1$ and all $t \geq 0$

Lemma 3.13. The element local sensitivity $LS_u(\mathbf{x}, t, r)$ is admissible.

Definition 3.14 (Boundedness)**.** A sensitivity function $\delta_u(\mathbf{x}, t, r)$ is said to be bounded if $\delta_u(\mathbf{x}, t, r) = \Delta u$ for all $t \geq n$.

Lemma 3.15. If $\delta_u(\mathbf{x}, t, r)$ is admissable, then $min(\delta_u(\mathbf{x}, t, r), \Delta u)$ is admissable and bounded.

Definition 3.16 (Non-decreasing Monotonicity)**.** Let $u(\mathbf{x}, r)$ be an utility function and $\delta_u(\mathbf{x}, t, r)$ be a sensitivity function. $\delta_u(\mathbf{x}, t, r)$ is said to be monotonically non-decreasing if $\delta_u(\mathbf{x}, t, r) \geq \delta_u(\mathbf{x}, t, r')$ for all $\mathbf{x} \in \mathcal{X}$, $r, r' \in \mathcal{R}$, $t \geq 0$ such that $u(\mathbf{x}, r) \geq u(\mathbf{x}, r')$.

Definition 3.17 (Non-increasing Monotonicity)**.** Let $u(\mathbf{x}, r)$ be an utility function and $\delta_u(\mathbf{x}, t, r)$ be a sensitivity function. $\delta_u(\mathbf{x}, t, r)$ is said to be monotonically non-increasing if $\delta_u(\mathbf{x}, t, r) \geq \delta_u(\mathbf{x}, t, r')$ for all $\mathbf{x} \in \mathcal{X}$, $r, r' \in \mathcal{R}$, $t \geq 0$ such that $u(\mathbf{x}, r) \leq u(\mathbf{x}, r')$.

Definition 3.18 (Flat Monotonicity)**.** Let $u(\mathbf{x}, r)$ be an utility function and $\delta_u(\mathbf{x}, t, r)$ be a sensitivity function. $\delta_u(\mathbf{x}, t, r)$ is said to be flat if $\delta_u(\mathbf{x}, t, r) = \delta_u(\mathbf{x}, t, r')$ for all $\mathbf{x} \in \mathcal{X}$, $r, r' \in \mathcal{R}$, $t \geq 0$.

The paper refers to a *monotonic function* as a function that is either flat, monotonically non-decreasing or monotonically non-increasing.

Definition 3.19 (Stability)**.** A sensitivity function $\delta_u(\mathbf{x}, t, r)$ is stable if δ_u is admissible, bounded and monotonic.

The stability classification is employed in the accuracy analysis and it might seem quite restrictive at first glance.

3.5.2 Dampening

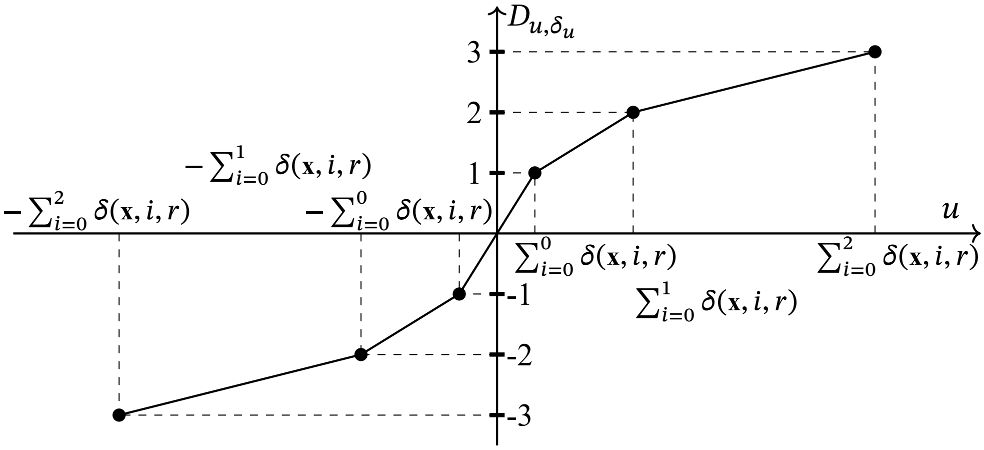
The local dampening attenuates the utility function in a specific way to make the signal-to-sensitivity ratio larger. This function is called D_{u,δ_u} and uses an admissible function δ_u that provides a dampened and scaled version of the original utility function. Figure 6 illustrates how the dampening function behaves.

Definition 3.20 (Dampening function)**.** Given a utility function $u(\mathbf{x}, r)$ and an admissible function $\delta_u(\mathbf{x}, t, r)$, the dampening function $D_{u,\delta_u}(\mathbf{x}, r)$ is defined as a piecewise linear interpolation over the points:

$$< \ldots, (b(\mathbf{x}, -1, r), -1), (b(\mathbf{x}, 0, r), 0), (b(\mathbf{x}, 1, r), 1), \ldots >$$

where $b(\mathbf{x}, i, r)$ is given by:

Figure 6 – Dampening function D_{u,δ_u}



Source: adapted from Farias (2021)

$$b(x,i,r) := \begin{cases} \sum_{j=0}^{i-1}\delta(\mathbf{x},j,r) & \text{if } i > 0 \\ 0 & \text{if } i = 0 \\ -b(x,-i,r) & \text{otherwise} \end{cases}$$

Therefore,

$$D_{u,\delta_u}(\mathbf{x},r) = \frac{u(\mathbf{x},r) - b(\mathbf{x},i,r)}{b(\mathbf{x},i+1,r) - b(\mathbf{x},i,r)} + i$$

where i is defined as the smallest integer such that $u(\mathbf{x},r) \in [b(\mathbf{x},i,r), b(\mathbf{x},i+1,r))$.

A crucial property of D_{u,δ_u} is that it scales u so that the sensitivity of D_{u,δ_u} is bounded to 1.

Lemma 3.21. $|D_{u,\delta_u}(\mathbf{x},r) - D_{u,\delta_u}(\mathbf{y},r)| \leq 1$ for all $\mathbf{x},\mathbf{y}$ such that $d(\mathbf{x},\mathbf{y}) \leq 1$ and all $r \in \mathcal{R}$ if δ_u is admissible.

And finally, the local dampening mechanism $\mathcal{M}^{\text{dam}}_{u,\varepsilon,\delta_u}$ selects an element $r \in \mathcal{R}$ with probability proportional to $\exp\left(\varepsilon \cdot D_{u,\delta_u}(\mathbf{x},r)/2\right)$.

$$\mathcal{M}^{\text{dam}}_{u,\varepsilon,\delta_u}(\mathbf{x},r) \propto \exp\left(\frac{\varepsilon \cdot D_{u,\delta_u}(\mathbf{x},r)}{2}\right)$$

The local dampening mechanism satisfies ε-differential privacy if δ_u is admissible.

Theorem 3.22. $\mathcal{M}^{\text{dam}}$ satisfies ε-differential privacy if δ_u is admissible.

The local dampening mechanism is especially effective when the sensitivity function is flat. To deal with non-flat sensitivity functions, the paper proposed the *shifted local dampening* mechanism. However, all the variants of local dampening suffer from the inversion problem. Consider a situation in which we apply dampening to the utility scores of elements $r \in \mathcal{R}$ using a non-monotonic sensitivity function, δ_u. This occurs, for example, when $\delta_u(\mathbf{x}, t, r)$ is used as the local sensitivity of an element, defined as $\delta_u(\mathbf{x}, t, r) = LS_u(\mathbf{x}, t, r)$.

In the inversion problem example, we have two elements, r_1 and r_2, with respective sensitivity and utility values. Initially, r_2 is more valuable than r_1 based on utility scores. However, after applying a dampening function D_{u,δ_u}, which adjusts utility scores based on their sensitivity, the utility ranking of the elements is reversed: r_1 becomes more favored than r_2. This dampening leads to a selection bias towards r_1, despite its originally lower utility, potentially compromising the accuracy of decisions based on these adjusted scores.

Corollary 3.23. Let δ_u be a stable function. The shifted local dampening mechanism $\mathcal{M}^{\text{dam}}(\mathbf{x}, \varepsilon, u, \delta_u, \mathcal{R})$ is never worse than the exponential mechanism $\mathcal{M}^{\text{exp}}$, that is:

1. $Pr[\xi(\mathcal{M}^{\text{dam}}, \mathbf{x}) \geq t] \leq Pr[\xi(\mathcal{M}^{\text{exp}}, \mathbf{x}) \geq t]$ for all $t \geq 0$,
2. $\mathbb{E}[\xi(\mathcal{M}^{\text{dam}}, \mathbf{x})] \leq \mathbb{E}[\xi(\mathcal{M}^{\text{exp}}, \mathbf{x})]$.

The paper does not provide explicit utility bounds or comparative analysis with the permute-and-flip and report-noisy-max methods. Although the inversion problem is clearly presented, it could present challenges for data analysts to detect when the inversion problem occur. The local dampening mechanism experiences the same algorithmic complexity as the exponential mechanism. Additionally, the work hinges on a very specific type of sensitivity (element local sensitivity) which could be challenging to accurately determine.

3.6 Discussion

In this chapter, we provided the background needed to understand the field of private selection and its key definitions. We also explored the private selection mechanisms, including the exponential mechanism, permute-and-flip, report-noisy-max, and local dampening.

We examined algorithms that employ global sensitivity, and we gave particular attention to the local dampening mechanism. This mechanism utilizes instance-based sensitivity in conjunction with a differentially private algorithm, a methodological approach also adopted in this thesis. However, the local dampening mechanism lacks extensibility across various noise distributions, *i.e.,* it is not possible to change the noise distribution for local dapening. Along with the lack of extensability, the local dapening has high time complexity. To elucidate this for the reader, we include Table 1, which compares all the private selection mechanisms discussed in this thesis in five different attributes. In the next chapter, we will present our proposed method that employs local sensitivity and has low algorithm complexity.

Table 1 – Comparative table of private selection methods.

	Sensitivity				
Algorithm	Global	Local	Element	Extensibility	Complexity
Exponential	✅	❌	❌	❌	High
Permute-and-flip	✅	❌	❌	❌	Low
Report-noisy-max	✅	❌	❌	✅	Low
Local dampening	✅	✅	✅	❌	High

Source: elaborated by the author.

4 SMOOTH NOISY MAX

This chapter introduces Smooth Noisy Max (SNM), an algorithm that tackles the differentially private selection problem. The report-noisy-max inspires the proposed algorithm. SNM offers significant advantages over the existing methods, such as simplicity, ease of implementation, low algorithm complexity, and explicit accuracy performance. In particular, our novel approach adopts an instance-based sensitivity rather than global sensitivity since the global one may increase the signal-to-sensitivity ratio, implying inaccurate results. More precisely, SNM applies the smooth sensitivity.

Definition 4.1 (Smooth sensitivity, adapted from Nissim *et al.* (2007))**.** For $\beta > 0$, the β-smooth sensitivity of the utility function u is:

$$\mathcal{S}_{u,\beta}(\mathbf{x}) = \max_{t=0,1,\ldots,|\mathbf{x}|} \left(e^{-t\beta} \cdot LS_u(\mathbf{x}, t) \right)$$

The smooth sensitivity attenuates the local sensitivity (Definition 3.4) based on the distance from **x**. Applying an instance-based sensitivity, such as smooth sensitivity, within a private selection algorithm is not always feasible for differential privacy. For instance, the exponential mechanism can not be used directly with the smooth sensitivity (see Theorem 7.2). On the other hand, the proposed Smooth Noisy Max algorithm can take advantage of the smooth framework and consequently decrease the signal-to-sensitivity ratio of the method. Additionally, it can keep the same differentially private guarantees of the standard report-noisy-max and ensures better accuracy.

SNM adds noise proportional to a smooth upper bound on the local sensitivity (*e.g.* smooth sensitivity $\mathcal{S}_{u,\beta}$) to its utility value for each possible outcome r for the query f at database **x**, *i.e.*, $u(\mathbf{x}, r)$. The noise, expressed by a random variable Z, is drawn from an (α, β)-admissible probability density function (Definition 2.18). For the sake of simplicity, we refer to $\mathcal{S}_{u,\beta}$ as $\mathcal{S}$. This procedure is explained in Algorithm 3.

Algorithm 3: Smooth Noisy Max Algorithm

```
for $r \in \mathcal{R}$ do
    $\tilde{u}(\mathbf{x}, r) \leftarrow u(\mathbf{x}, r) + \frac{2\mathcal{S}(\mathbf{x})}{\alpha} \cdot Z$;
end
return $\arg\max_{r \in \mathcal{R}} \tilde{u}(\mathbf{x}, r)$
```

4.1 Privacy Guarantees

In Theorem 4.2, we prove that the Smooth Noisy Max algorithm ensures (ε, δ)-differential privacy.

Theorem 4.2. The Smooth Noisy Max $\mathcal{A}_{u,\varepsilon}$ algorithm is (ε, δ)-differentially private if h is an (α, β)-admissible noise probability density function, and Z a random variable sampled according to h.

Proof. Consider two neighbor databases **x** and **y**. Fix any $i \in \mathcal{R}$ and let $\vec{z}_i = \{z_1, \dots, z_{|\mathcal{R}|}\} \backslash \{z_i\}$ be the fixed noises for all outputs except the ith output. We will argue for each $\vec{z}_i$ independently, similarly to what was done by Dwork and Roth (2014) (Claim 3.9). For simplicity of notation, denote $N(\mathbf{x}) = {}^{2S_{u,\beta}(\mathbf{x})}/_{\alpha}$, and the Smooth Noisy Max as $\mathcal{A}$. Then, the probability of $i \in \mathcal{R}$ being the output of the algorithm is given by

$$Pr[\mathcal{A}(\mathbf{x}) = i|\vec{z}] = Pr\left[u(\mathbf{x}, i) + N(\mathbf{x}) \cdot Z \geq \max_{j \in \mathcal{R}; j \neq i} \{u(\mathbf{x}, j) + z_j\}\right].$$

Let $\tilde{u}_* = \max\limits_{j \in \mathcal{R}; j \neq i} \{u(\mathbf{x}, j) + z_j\}$ and $\tilde{u}'_* = \max\limits_{j \in \mathcal{R}; j \neq i} \{u(\mathbf{y}, j) + z_j\}$. Then:

$$Pr[\mathcal{A}(\mathbf{x}) = i|\vec{z}_i] = Pr\left[Z \geq \frac{\tilde{u}_* - u(\mathbf{x}, i)}{N(\mathbf{x})}\right],$$

For the sake of simplicity let define $\mathrm{g}(i) = \frac{\tilde{u}_* - u(\mathbf{x}, i)}{N(\mathbf{x})}$ and $\mathrm{g}'(i) = \frac{\tilde{u}_* - u(\mathbf{y}, i)}{N(\mathbf{x})}$.

$$Pr[\mathcal{A}(\mathbf{x}) = i|\vec{z}_i] = Pr\left[Z \geq \mathrm{g}(i)\right],$$

Using the definition 2.6 for neighboring databases **x**, **y**, and $Z_X \sim \mathcal{A}(\mathbf{x})$, $Z_Y \sim \mathcal{A}(\mathbf{y})$:

$$D_\infty^\delta(Z_X||Z_Y) = \max_{\substack{S \subseteq \mathcal{R}: \\ Pr[Z_X \in S] \geq \delta}} \left[\log\left(\frac{Pr[Z_X \in S] - \delta}{Pr[Z_Y \in S]}\right)\right]$$

As our algorithms draws results from the discrete set of outputs, we can:

$$\begin{aligned} D_\infty^\delta(Z_X||Z_Y) &= \max_{\substack{S \subseteq \mathcal{R}: \\ Pr[Z_X \in S] \geq \delta}} \left[\log\left(\frac{\sum_{r \in S} Pr[Z_X = r] - \delta}{\sum_{r \in S} Pr[Z_Y = r]}\right)\right] \\ &= \max_{\substack{S \subseteq \mathcal{R}: \\ Pr[Z_X \in S] \geq \delta}} \left[\log\left(\frac{\sum_{r \in S} \int Pr[Z_X = r|\vec{z}_r] Pr[\vec{z}_r] d\vec{z}_r - \delta}{\sum_{r \in S} \int Pr[Z_Y = r|\vec{z}_r] Pr[\vec{z}_r] d\vec{z}_r}\right)\right] \end{aligned}$$

Since $Z \sim h$ and h is admissible, we can use the sliding property:

$$
\begin{aligned}
D_{\infty}^{\delta}(Z_X||Z_Y) = & \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum\limits_{r\in S}\int Pr[Z_X = r|\vec{z}_r]Pr[\vec{z}_r]d\vec{z}_r - \delta}{\sum\limits_{r\in S}\int Pr[Z_Y = r|\vec{z}_r]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
= & \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum\limits_{r\in S}\int Pr\left[Z_X \geq \frac{\tilde{u}_*-u(\mathrm{x},r)}{N(\mathrm{x})}\right]Pr[\vec{z}_r]d\vec{z}_r - \delta}{\sum\limits_{r\in S}\int Pr\left[Z_Y \geq \frac{\tilde{u}'_*-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
= & \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum\limits_{r\in S}\int Pr\left[Z_X \geq g(r)\right]Pr[\vec{z}_r]d\vec{z}_r - \delta}{\sum\limits_{r\in S}\int Pr\left[Z_Y \geq g'(r)\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
\leq & \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum\limits_{r\in S}\int Pr\left[Z_X \geq g(r) - g(r) + \frac{\tilde{u}'_*-u(\mathrm{y},r)}{N(\mathrm{x})}\right]\cdot e^{\frac{\varepsilon}{2}}Pr[\vec{z}_r]d\vec{z}_r + \frac{\delta}{2} - \delta}{\sum\limits_{r\in S}\int Pr\left[Z_Y \geq \frac{\tilde{u}'_*-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
\leq & \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum\limits_{r\in S}\int Pr\left[Z_X \geq \frac{\tilde{u}'_*-u(\mathrm{y},r)}{N(\mathrm{x})}\right]\cdot e^{\frac{\varepsilon}{2}}Pr[\vec{z}_r]d\vec{z}_r + \frac{\delta}{2} - \delta}{\sum\limits_{r\in S}\int Pr\left[Z_Y \geq \frac{\tilde{u}'_*-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right]
\end{aligned}
$$

The first inequality results from the sliding property since h is admissible. Notice that this property can be applied because, by the properties of smooth and local sensitivities, and since $\mathbf{x}$ and $\mathbf{y}$ are neighbors:

$$
\begin{aligned}
g(r) - \frac{\tilde{u}'_* - u(\mathbf{y},r)}{N(\mathbf{x})} = \frac{u(\mathbf{x},r) - u(\mathbf{y},r) - \tilde{u}_* + \tilde{u}'_*}{N(\mathbf{x})} &= \alpha\frac{u(\mathbf{x},r) - u(\mathbf{y},r) - \tilde{u}_* + \tilde{u}'_*}{2\mathcal{S}_{u,\beta}(\mathbf{x})}, \\
&\leq \frac{\alpha}{2LS(\mathbf{x})}\left(\underbrace{u(\mathbf{x},i) - u(\mathbf{y},i)}_{\leq LS(\mathbf{x})} + \underbrace{\tilde{u}'_* - \tilde{u}_*}_{\leq LS(\mathbf{x})}\right), \\
&\leq \alpha\frac{2LS(\mathbf{x})}{2LS(\mathbf{x})} = \alpha.
\end{aligned}
$$

Further we can apply the dilation property since h satisfies the dilation property and $\ln\frac{N(\mathrm{x})}{N(\mathrm{y})} =$

$\ln \frac{\mathcal{S}_{u,\beta}(\mathrm{x})}{\mathcal{S}_{u,\beta}(\mathrm{y})} \leq \beta$ (see Definition 2.14):

$$\begin{aligned}
D_{\infty}^{\delta}(Z_X||Z_Y) &\leq \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum_{r\in S}\int Pr\left[Z_X \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{x})}\right]\cdot e^{\frac{\varepsilon}{2}}Pr[\vec{z}_r]d\vec{z}_r + \frac{\delta}{2} - \delta}{\sum_{r\in S}\int Pr\left[Z_Y \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
&\leq \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum_{r\in S}\int Pr\left[Z_X \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{x})}\cdot\frac{N(\mathrm{x})}{N(\mathrm{y})}\right]\cdot e^{\varepsilon}Pr[\vec{z}_r]d\vec{z}_r + \delta - \delta}{\sum_{r\in S}\int Pr\left[Z_Y \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
&= \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{\sum_{r\in S}\int Pr\left[Z_X \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{y})}\right]\cdot e^{\varepsilon}Pr[\vec{z}_r]d\vec{z}_r + \delta - \delta}{\sum_{r\in S}\int Pr\left[Z_Y \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
&= \max_{\substack{S\subseteq\mathcal{R}:\\ Pr[Z_X\in S]\geq\delta}} \left[\log\left(\frac{e^{\varepsilon}\cdot\sum_{r\in S}\int Pr\left[Z_X \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}{\sum_{r\in S}\int Pr\left[Z_Y \geq \frac{\tilde{u}'_{*}-u(\mathrm{y},r)}{N(\mathrm{y})}\right]Pr[\vec{z}_r]d\vec{z}_r}\right)\right], \\
&= \varepsilon.
\end{aligned}$$

By symmetry, we can also prove that $D_{\infty}^{\delta}(Z_Y\|Z_X) \leq \varepsilon$. Then, by Definition 2.7, we conclude that SNM is (ε, δ)-differentially private. □

Corollary 4.3. The Smooth Noisy Max $\mathcal{A}_{u,\varepsilon}$ algorithm with sampled noise from the Student's T distribution is ε-differentially private. By scaling the Student's T distribution under the smooth sensitivity, pure differential privacy is assured (BUN; STEINKE, 2019).

Corollary 4.4. The Smooth Noisy Max $\mathcal{A}_{u,\varepsilon}$ algorithm with sampled noise from the Laplace distribution is (ε, δ)-differentially private, when β parameter is defined by $\varepsilon/2\log(2/\delta)$ (NISSIM *et al.*, 2007).

Corollary 4.5. The Smooth Noisy Max $\mathcal{A}_{u,\varepsilon}$ algorithm with sampled noise from the Laplace Log-Normal ($\mathtt{LLN}(\sigma)$) distribution is (ε, δ)-differentially private (BUN; STEINKE, 2019), when α parameter is defined as $e^{-3/2\sigma^2}\,(\varepsilon - \beta/\sigma)$.

We can also improve the noise addition under the monotonicity property. The utility function u is monotonic in the database if adding an element to the database cannot cause the value of the function to decrease, *e.g.*, counting queries.

Corollary 4.6. When the utility function u is monotonic in the database, then the Smooth Noisy Max $\mathcal{A}_{u,\varepsilon}$ scales the noise only by a factor of $\frac{\mathcal{S}(\mathrm{x})}{\alpha}$.

4.2 Utility Analysis

A significant characteristic of the Smooth Noisy Max algorithm is that it provides strong utility guarantees. Given a database $\mathbf{x}$, we bound the error of the private algorithm by a specific parameter t. The algorithm's accuracy is assessed based on the largest utility score $u^* = \max_{r \in \mathcal{R}} u(\mathbf{x}, r)$. It will be highly unlikely that the returned element r has a utility score significantly less than $O\left(u^* - (\mathcal{S}_{u,\beta}(\mathbf{x})/\varepsilon) \ln |\mathcal{R}|\right)$ when the noise distribution is Laplace. All subsequent proofs refer to the admissible Laplace distribution version of the Smooth Noisy Max algorithm.

Lemma 4.7. Given a fixed database $\mathbf{x} \in \mathcal{X}$, for the Smooth Noisy Max $\mathcal{A}$ algorithm with a standard Laplace distribution as noise function and any $t > 0$, the error $\xi(\mathcal{A}, \mathbf{x})$ satisfies

$$Pr[\xi(\mathcal{A}, \mathbf{x}) \geq t] \leq |\mathcal{R}| \exp\left(-\frac{\varepsilon t}{4\mathcal{S}_{u,\beta}(\mathbf{x})}\right).$$

Proof. Define $u^*(\mathbf{x}) = \max_{r \in \mathcal{R}} u(\mathbf{x}, r)$, so for each possible outcome $r \in \mathcal{R}$, the error can be written as $\xi(\mathcal{A}, \mathbf{x}) = u^*(\mathbf{x}) - u(\mathbf{x}, r)$. Thus, for $t > 0$:

$$Pr[\xi(\mathcal{A}, \mathbf{x}) \geq t] = Pr[u(\mathbf{x}, \mathcal{A}(\mathbf{x})) \leq u^*(\mathbf{x}) - t].$$

For simplicity of notation, define the following subsets of $\mathcal{R}$:

(i) $\mathcal{R}_t = \{r \in \mathcal{R} \,:\, u(\mathbf{x}, r) \leq u^*(\mathbf{x}) - t\}$;

(ii) $\mathcal{R}_* = \{r \in \mathcal{R} \,:\, u(\mathbf{x}, r) = u^*(\mathbf{x})\}$.

Also, consider the noisy utility $\tilde{u}(\mathbf{x}, r) = u(\mathbf{x}, r) + (2\mathcal{S}_{u,\beta}(\mathbf{x})/\alpha) \cdot z_r$, where $z_r \sim \mathsf{Lap}\,(0, 1)$, and its maximal value $\tilde{u}^*(\mathbf{x}) = \max_{r \in \mathcal{R}} \tilde{u}(\mathbf{x}, r)$. Notice that the probability of the output being in $\mathcal{R}_t$ is the same probability of existing some element in $\mathcal{R}_t$ with the greatest noisy utility. This way, $Pr[\xi(\mathcal{A}, \mathbf{x}) \geq t]$ is equivalent to the probability of existing some $r \in \mathcal{R}_t$ such that $\tilde{u}(\mathbf{x}, r) = \tilde{u}^*(\mathbf{x})$. In other words, $\exists r \in \mathcal{R}_t \,:\, \tilde{u}(\mathbf{x}, r) = \tilde{u}^*(\mathbf{x})$. Then:

$$\begin{aligned} Pr[\xi(\mathcal{A}, \mathbf{x}) \geq t] &= Pr[\exists r \in \mathcal{R}_t \,:\, \tilde{u}(\mathbf{x}, r) = \tilde{u}^*(\mathbf{x})], \\ &= Pr[\cup_{r \in \mathcal{R}_t} [\tilde{u}(\mathbf{x}, r) = \tilde{u}^*(\mathbf{x})]], \\ &\leq \sum_{r \in \mathcal{R}_t} Pr[\tilde{u}(\mathbf{x}, r) = \tilde{u}^*(\mathbf{x})]. \end{aligned}$$

Let r' be the most probable output in $\mathcal{R}_t$. In this case, we can write:

$$\begin{aligned} Pr[\xi(\mathcal{A},\mathbf{x}) \geq t] &\leq |\mathcal{R}_t|\, Pr[\tilde{u}(\mathbf{x},r') = \tilde{u}^*(\mathbf{x})], \\ &= |\mathcal{R}_t|\, Pr[\tilde{u}(\mathbf{x},r') \geq \tilde{u}^*(\mathbf{x})], \\ &= |\mathcal{R}_t|\, Pr[({}^{2\mathcal{S}_{u,\beta}(\mathbf{x})}/_{\alpha}) \cdot z_{r'} \geq \tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r')], \\ &\leq \frac{|\mathcal{R}_t|\, Pr[z_{r'} \geq (\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r')) \cdot \left({}^{\alpha}/_{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)]}{Pr[\mathcal{A}(\mathbf{x}) \in \mathcal{R}_*]}. \end{aligned}$$

Notice that $Pr[\tilde{u}(\mathbf{x},r') = \tilde{u}^*(\mathbf{x})] = Pr[\tilde{u}(\mathbf{x},r') \geq \tilde{u}^*(\mathbf{x})]$, since $\tilde{u}^*(\mathbf{x})$ is the maximal noisy utility. Now, consider $r^* = \arg\max_{r \in \mathcal{R}} u(\mathbf{x},r)$ and z_* as the error associated with r^*. Thus, we can write:

$$\begin{aligned} Pr[\mathcal{A}(\mathbf{x}) \in \mathcal{R}_*] &= Pr[\cup_{r \in \mathcal{R}_*}[\mathcal{A}(\mathbf{x}) = r]], \\ &= \sum_{r \in \mathcal{R}_*} Pr[\mathcal{A}(\mathbf{x}) = r], \\ &= |\mathcal{R}_*|\, Pr[\mathcal{A}(\mathbf{x}) = r^*], \\ &= |\mathcal{R}_*|\, Pr[z_* \geq (\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r^*)) \cdot \left({}^{\alpha}/_{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)]. \end{aligned}$$

The equality above is valid because $u(\mathbf{x},r) = u(\mathbf{x},r)\ \forall r \in \mathcal{R}_*$, so the chance of any of them being the output depends only on the noise, resulting in independent events with equal probability. As a consequence:

$$Pr[\xi(\mathcal{A},\mathbf{x}) \geq t] \leq \frac{|\mathcal{R}_t|\, Pr[z_{r'} \geq (\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r')) \cdot \left({}^{\alpha}/_{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)]}{|\mathcal{R}_*|\, Pr[z_* \geq (\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r^*)) \cdot \left({}^{\alpha}/_{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)]}.$$

However, as $z_{r'}, z_* \sim \mathtt{Lap}(0,1)$, $u(\mathbf{x},r^*) = u^*(\mathbf{x})$ and $u(\mathbf{x},r') \leq u^*(\mathbf{x}) - t$:

$$\begin{aligned} &\frac{|\mathcal{R}_t|\, Pr[z_{r'} \geq (\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r')) \cdot \left({}^{\alpha}/_{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)]}{|\mathcal{R}_*|\, Pr[z_* \geq (\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r_*)) \cdot \left({}^{\alpha}/_{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)]} \\ &= \frac{\frac{|\mathcal{R}_t|}{2} \exp\left(-\frac{\alpha(\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r'))}{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)}{\frac{|\mathcal{R}_*|}{2} \exp\left(-\frac{\alpha(\tilde{u}^*(\mathbf{x}) - u(\mathbf{x},r_*))}{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)}, \\ &\leq \frac{|\mathcal{R}_t|}{|\mathcal{R}_*|} \frac{\exp\left(-\frac{\alpha(\tilde{u}^*(\mathbf{x}) - u^*(\mathbf{x}) + t)}{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)}{\exp\left(-\frac{\alpha(\tilde{u}^*(\mathbf{x}) - u^*(\mathbf{x}))}{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)}, \\ &= \frac{|\mathcal{R}|}{|\mathcal{R}_*|} \exp\left(-\frac{\alpha t}{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right). \end{aligned}$$

We know that $\alpha = \frac{\varepsilon}{2}$ (see NISSIM *et al.*, Lemma 2.9 (NISSIM *et al.*, 2007)). Then, from the result above, we can finally conclude that:

$$Pr[\xi(\mathcal{A},\mathbf{x}) \geq t] \leq |\mathcal{R}| \exp\left(-\frac{\varepsilon t}{4\mathcal{S}_{u,\beta}(\mathbf{x})}\right).$$

□

Theorem 4.8. Let $\mathbf{x} \in \mathcal{X}$ be a fixed database. Then, for a given $t > 0$, the Smooth Noisy Max $\mathcal{A}$ algorithm with standard Laplace noise distribution will have the following properties:

(i) $Pr\left[\xi(\mathcal{A}, \mathbf{x}) \geq \frac{4\mathcal{S}_{u,\beta}(\mathbf{x})\,(\ln(|\mathcal{R}|)+t)}{\varepsilon}\right] \leq e^{-t}$;

(ii) $\mathbb{E}\,(\xi(\mathcal{A}, \mathbf{x})) \leq \frac{4\mathcal{S}_{u,\beta}(\mathbf{x})\,(\ln(|\mathcal{R}|)+1)}{\varepsilon}$.

The utility bounds presented by Theorem 4.8 provide tools to compare and show that the Smooth Noisy Max outperforms our related work, *i.e.*, report-noisy-max, exponential mechanism, and permute-and-flip. Firstly, we analyze the utility of the Smooth Noisy Max in contrast to the report-noisy-max with exponential noise, shown by Theorem 4.9.

Theorem 4.9. The Smooth Noisy Max $\mathcal{A}$ with Laplace noise distribution is *never worse* than $\mathcal{N}^{\exp}$ report-noisy-max algorithm with exponential noise when $\mathcal{S}_{u,\beta}(\mathbf{x}) \leq \frac{\Delta u}{2}$.

Proof. Using the lemma 4.7, we can obtain

$$Pr\,[\xi\,(\mathcal{A}, x) \geq t] \leq \frac{|\mathcal{R}|}{|\mathcal{R}_*|} \exp\left(-\frac{\varepsilon t}{4\mathcal{S}_{u,\beta}(\mathbf{x})}\right).$$

We observe that when $\mathcal{S}_{u,\beta}(\mathbf{x}) \leq \frac{\Delta u}{2}$, then

$$\frac{|\mathcal{R}|}{|\mathcal{R}_*|} \exp\left(-\frac{\varepsilon t}{4\mathcal{S}_{u,\beta}(\mathbf{x})}\right) \leq \frac{|\mathcal{R}|}{|\mathcal{R}_*|} \exp\left(-\frac{\varepsilon t}{2\Delta u}\right),$$

$$Pr\,[\xi\,(\mathcal{A}, x) \geq t] \leq Pr\,[\xi\,(\mathcal{N}^{\exp}, x) \geq t]\,.$$

Furthermore, by the Theorem 3.11, the first statement (*i*) holds. We want to prove the second statement (*ii*). The expected error can be expressed in terms of complementary cumulative distribution function:

$$\mathbb{E}(\xi(\mathcal{A}, x)) = \int_0^\infty Pr[\xi(\mathcal{A}, x) \geq t]dt.$$

We shown that $Pr\,[\xi\,(\mathcal{A}, x) \geq t] \leq Pr\,[\xi\,(\mathcal{N}^{\exp}, x) \geq t]$, thus:

$$\mathbb{E}(\xi(\mathcal{A}, x)) - \mathbb{E}(\xi(\mathcal{N}^{\exp}, x)) =$$

$$\int_0^\infty Pr\,[\xi\,(\mathcal{A}, x) \geq t] - Pr\,[\xi\,(\mathcal{N}^{\exp}, x) \geq t]\,dt \leq 0.$$

Thus, the Smooth Noisy Max with Laplace noise distribution is never worse than the report-noisy-max algorithm with exponential noise when $\mathcal{S}_{u,\beta}(\mathbf{x}) \leq \frac{\Delta u}{2}$. □

DING *et al.* (DING *et al.*, 2021) shows that the report-noisy-max with exponential noise is identical to the permute-and-flip, so as we know, by Theorem 4.9 the Smooth Noisy Max is never worse than report-noisy-max algorithm with exponential noise when $\mathcal{S}_{u,\beta}(\mathbf{x}) \leq \frac{\Delta u}{2}$, and consequently never worse than permute-and-flip mechanism under the same constraint.

The utility of our proposed method also outperforms the exponential mechanism and report-noisy-max with Gumbel noise when $\mathcal{S}_{u,\beta}(\mathbf{x}) \leq \frac{\Delta u}{2}$. Since, by transitivity, SNM surpass the permute-and-flip that exceeds the exponential mechanism (MCKENNA; SHELDON, 2020). Additionally, the exponential mechanism is identical to report-noisy-max with Gumbel noise (DURFEE; ROGERS, 2019); therefore, the Smooth Noisy Max is *never worse* than report-noisy-max with Gumbel noise. All these results are expressed by Corollary 4.10.

Corollary 4.10. When $\mathcal{S}_{u,\beta}(\mathbf{x}) \leq \frac{\Delta u}{2}$, SNM $\mathcal{A}$ algorithm is *never worse* than $\mathcal{M}^{\mathrm{pf}}$ permute-and-flip, $\mathcal{M}^{\mathrm{exp}}$ exponential mechanism, and $\mathcal{N}^{\mathrm{gum}}$ report-noisy-max algorithm with Gumbel noise.

The lack of utility bounds for the Local Dampening mechanism (FARIAS *et al.*, 2023) hampers a comparative assessment with our Smooth Noisy Max. Nevertheless, the paper conducts an exhaustive empirical analysis in the subsequent sections.

4.3 Discussion

In this chapter, we introduced the Smooth Noisy Max, a private selection algorithm that capitalizes on key strengths of the report-noisy-max, such as extensibility and low complexity. The SNM utilizes a type of local sensitivity known as smooth sensitivity, adapted specifically for private selection scenarios. We prove that SNM satisfies (ε, δ)-differential privacy and provided specific utility bounds. We prove that our proposed method is *nerver worse* than the exponential mechanism, permute-and-flip, and report-noisy-max when SNM applied the Laplace distribution.

However, this adapted version of smooth sensitivity does not leverage the element-specific sensitivity, such as the element local sensitivity used by Local Dampening. While there are preliminary indications that the SNM algorithm could potentially guarantee differential privacy with element sensitivity, fully exploring this possibility remains a subject for future research. The Table 2 summarizes all key points of the Smooth Noisy Max in comparison with the related work.

Table 2 – Comparative table of private selection methods.

	Sensitivity				
Algorithm	Global	Local	Element	Extensibility	Alg. Complexity
Exponential	✅	❌	❌	❌	High
Permute-and-flip	✅	❌	❌	❌	Low
Report-noisy-max	✅	❌	❌	✅	Low
Local dampening	✅	✅	✅	❌	High
Smooth Noisy Max	✅	✅	⚠️	✅	Low

Source: elaborated by the author.

Remark (Smooth Noisy Max with element local sensitivity)**.** We draft potential proof of the Smooth Noisy Max by incorporating element local sensitivity. In Theorem 4.2, we initially define $N(\mathbf{x}) = \frac{2\mathcal{S}(\mathbf{x})}{\alpha}$. However, to target element local sensitivity more effectively, we propose defining a smooth element local sensitivity and accordingly adjusting $N(\mathbf{x})$.

Definition 4.11 (Smooth Element Sensitivity)**.** For $\beta > 0$, the β-smooth element sensitivity of the utility function u is:

$$\mathcal{S}_{u,\beta}(\mathbf{x}, r) = \max_{t=0,1,\ldots,|\mathbf{x}|} \left(e^{-t\beta} \cdot LS_u(\mathbf{x}, t, r)\right)$$

Thus, the adapted $N(\mathbf{x}, r) = \frac{2\mathcal{S}_{u,\beta}(\mathbf{x},r)}{\alpha}$ implies that for $i \in \mathcal{R}$ be the output the *ith* noisy score should be the larger. Hence,

$$Pr[\mathcal{A}_{u,\varepsilon}(\mathbf{x}) = i|\vec{z}] = Pr\left[u(\mathbf{x}, i) + N(\mathbf{x}, i) \geq \max_{j\in\mathcal{R};j\neq i}\{u(\mathbf{x}, j) + z_j\}\right],$$

moving forward, to establish (ϵ, δ)-differential privacy, it is crucial to meticulously address the dilation property–the second inequality in Theorem 4.2. Fully analysis of this proof will be pursued in future research.

5 APPLICATION — PERCENTILE SELECTION

In this chapter, we address the percentile selection problem. The task is to return the *p-th* percentile value from a set of real numbers.

Nissim *et al.* (2007) and McKenna and Sheldon (2020) have dealt with similar problems. The Nissim *et al.* (2007) work addressed the challenge of privately releasing the numerical median of a dataset while preserving individual privacy. McKenna and Sheldon (2020) work attacks a similar problem, also for the data median, returning the bin value where it belongs.

5.1 Problem Statement

Given a dataset **x** represented as a vector $\{x_1, \ldots, x_n\}$. For simplicity's sake, assume that every database **x** is ordered such that $x_1 \leq \ldots \leq x_n$. Suppose that all the values lies in $[0, \Lambda]$, $0 \leq x_1 \leq \ldots \leq x_n \leq \Lambda$. The task is to return the percentile value where its element x_i is as close as possible to the p-th percentile element.

5.2 Private Mechanism and Sensitivity Analysis

Following the problem statement, various private selection algorithms are applicable, including the exponential mechanism, permute-and-flip, local dampening, and our proposed Smooth Noisy Max variants. The algorithms select any value from a discrete subset of $[0, \Lambda]$, *i.e.*, $\mathcal{R} \subseteq [0, \Lambda]$. We designed a utility function u_p that assigns a maximum score of 1 when element i matches the p-th element's value and a minimum score of 0 in all other cases, see Definition 5.1.

Definition 5.1 (Utility function for percentile selection problem)**.** Consider a database $\mathbf{x} \in \mathcal{X}$, $n = |\mathbf{x}|$, and $i \in \mathbb{Z}^{0+}$ a non-negative integer. The utility is defined as follows:

$$u_p(\mathbf{x}, i) = \begin{cases} 1 & \text{if } x_i = x_k, \text{where } k = \left\lfloor \frac{p \cdot n}{100} \right\rfloor \\ 0 & \text{otherwise} \end{cases}$$

Recall that the exponential mechanism and the permute-and-flip require the global sensitivity Δ_{u_p}, the local dampening requires the element local sensitivity and the Smooth Noisy Max expects the smooth sensitivity $\mathcal{S}_{u_p}$.

Global Sensitivity

The following example can show a worst-case scenario. For instance, let $p = 50$ implying that $k = \lfloor \frac{n}{2} \rfloor$. Let **x** be a dataset with $n > 2$ and even, where $x_{<k} = 0$ and $x_{\geq k} = \Lambda$. Let **y** be a neighboring dataset of **x**, where one element $x_{\geq k}$ has been removed. Thus we have $u(\mathbf{x}, k) = 1$, and $u(\mathbf{y}, k) = 0$ which implies that $u(\mathbf{x}, k) - u(\mathbf{y}, k) = 1$. Thus, $|u(\mathbf{x}, r) - u(\mathbf{y}, r)| \leq 1$ for all $r \in \mathcal{R}$, and any two neighboring datasets x, y.

Lemma 5.2 (Percentile selection global sensitivity)**.**

$$\Delta_{u_p} = 1$$

Local Sensitivity

One must first compute the local sensitivity at a distance t to compute the smooth sensitivity. Let $\mathbf{x} \in \mathcal{X}$ be a dataset, and $j = \min\left(\sum_{i=0}^{k-1} u(\mathbf{x}, x_i), \sum_{i=k+1}^{n} u(\mathbf{x}, x_i)\right)$ the smallest sequence of p-th value repetition length at left or right of position k. Thus, the p-th percentile value will remain the same until $2j + 1$ insertions and deletions from **x** because of the floor function in the k definition (Definition 5.1).

Lemma 5.3 (Percentile selection local sensitivity at distance t)**.**

$$LS_{u_p}(\mathbf{x}, t) = \begin{cases} 1 & \text{if } t \geq 2j + 1 \\ 0 & \text{otherwise} \end{cases}$$

Now, it is possible to calculate the smooth sensitivity of the percentile selection problem using the smooth sensitivity defined by Definition 4.1. The local sensitivity remains zero until $t < 2j + 1$ and changes to one when $t \geq 2j + 1$. Since the LS_{u_p} is constant when $t \geq 2j + 1$, the smooth sensitivity will be max when $t = 2j + 1$.

Lemma 5.4 (Percentile selection smooth sensitivity)**.**

$$\mathcal{S}_{u_p}(\mathbf{x}) = \exp(-(2j + 1) \cdot \varepsilon)$$

5.3 Experimental Evaluation

5.3.1 Datasets

We evaluated PATENT, HEPTH, and INCOME datasets from Hay *et al.* (2016). The PATENT dataset contains 32,558 tuples with a high percentage of zero-valued entries at 97.80%. In contrast, the HEPTH dataset comprises 347,414 tuples but only 21.17% zero-valued entries, indicating more varied data. Lastly, the INCOME dataset is the largest, with 20,787,122 tuples and 44.97% zero-valued entries, reflecting moderate homogeneity in its data. An essential attribute for those datasets is the amount of p-th value repetitions because of the utility function. Table 3 illustrates the datasets with their information.

Table 3 – Overview of the datasets

Dataset Name	**# Tuples**	**% Zero valued**
Patent	32,558	97.80%
Hepth	347,414	21.17%
Income	20,787,122	44.97%

Source: elaborated by the author.

5.3.2 Methods

We consider six approaches to the private percentile selection problem:

1. exponential mechanism (EM) using global sensitivity;
2. permute-and-flip (PF) using global sensitivity;
3. local dampening (LD) using the element local sensitivity $\hat{\delta}(\mathbf{x}, t, r) = LS_{u_p}(\mathbf{x}, t) = \max_{r' \in \mathcal{R}} LS_{u_p}(\mathbf{x}, t, r)$ from utility function (Definition 5.1);
4. local dampening (LD2) utilizing the utility presented in FARIAS *et al.*'s work;
5. Smooth Noisy Max via Laplace distribution (SNM-LAP) with the smooth sensitivity $\mathcal{S}_{u_p}$;
6. Smooth Noisy Max via Student's T distribution (SNM-T) with the smooth sensitivity $\mathcal{S}_{u_p}$.

5.3.3 Evaluation

We measured the absolute expected error (AEE) of each method for every specific scenario: $|\xi(\mathcal{A}, \mathbf{x})| = |x_k - \mathbb{E}(\mathcal{A}, \mathbf{x})|$. Understanding each outcome's associated probability is needed to find the expected value. Meanwhile, for the exponential mechanism, permute-and-flip, and local dampening, the probabilities of each outcome are straightforward to identify through the probability mass function of each mechanism. However, finding the probability of each outcome of the SNM algorithm is not straightforward. Reasoning about the output probability of other candidates is a condition for finding the probability of the output of a particular candidate. The specific candidate utility random variable should be greater than all others. This intricate probability function leads us to solve the following integral to find those probabilities numerically.

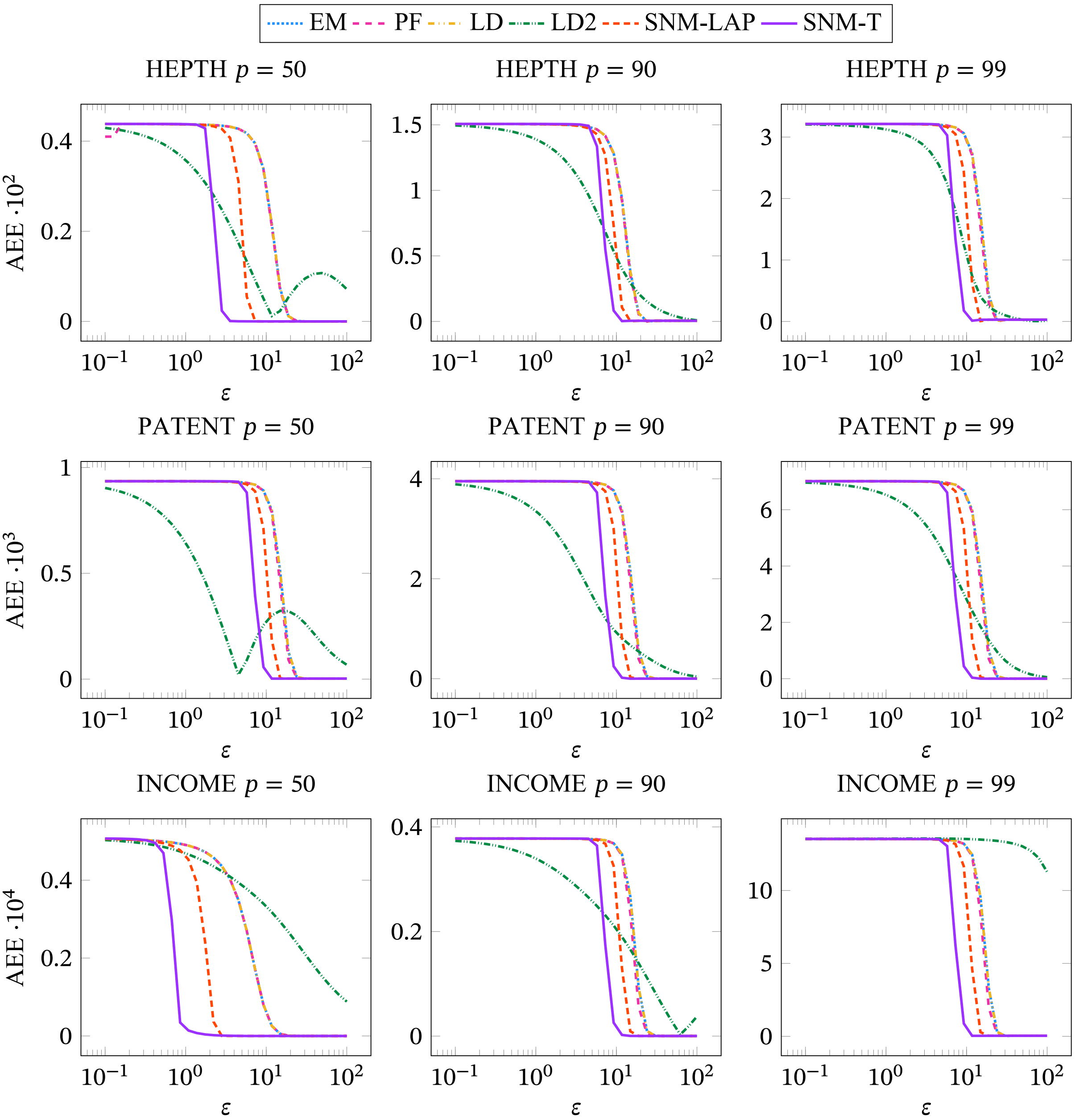
$$Pr[\mathcal{A}(\mathbf{x}) = r] = \int_{-\infty}^{\infty} f(i) \prod_{r \neq s} F\left(\frac{u(\mathbf{x}, r) - u(\mathbf{x}, s)}{N(\mathbf{x})} + i\right) di,$$

where $N(\mathbf{x}) = {}^{2S_{u,\beta}(\mathbf{x})}/_{\alpha}$, f and F represent the probability density function and the cumulative density function of the distribution used, respectively. Figure 7 shows the result varying the privacy budget $\varepsilon \in [10^{-1}, 10^{2}]$ and $p = 50, 90, 99$. For the Student's T distribution, the degree of freedom was set to 3. Each dataset has a ground-truth percentile value (GT) for each percentile. The desired behavior is that with a small privacy budget, the method outputs a value near the ground-truth value.

All versions of Smooth Noisy Max (SNM-T, SNM-LAP) have better accuracy when the dataset has several repetitions of the p-th value, *i.e.*, a significant j value. For instance, the median perspective ($p = 50$) in the PATENT dataset has $j = 0$, HEPTH has $j = 2$, and INCOME has $j = 10$. The datasets show different scenarios to assess the SNM algorithms compared to the competitors. The EM and PF methods have similar expected values in all scenarios. For the HEPTH dataset with $p = 50$, SNM-T method achieves a similar expected value, the difference in absolute values of a maximum of 5, with 69% and 85% less privacy budget than LD and LD2 methods, respectively. For $p = 90$, the SNM-T needs less than 51% and 76% privacy budget than the LD and LD2 methods, respectively. For $p = 99$, the behavior is similar when SNM-T requires less than 51% and 76% budget compared with LD and LD2. For the PATENT dataset, we observe up to 51%, 70%, and 70%, for $p \in \{50, 90, 99\}$ respectively, in privacy budget saving when compared to LD. In the PATENT dataset with $p = 50$, the LD2 method quickly reaches the desired value. And for $p \in \{90, 99\}$, we observe up to 70% of privacy budget saving when compared to LD2. For the INCOME dataset, when $p = 50$, the difference is more evident due to the high j value, but for $p \in \{90, 99\}$, the performance is quite the same in the other datasets.

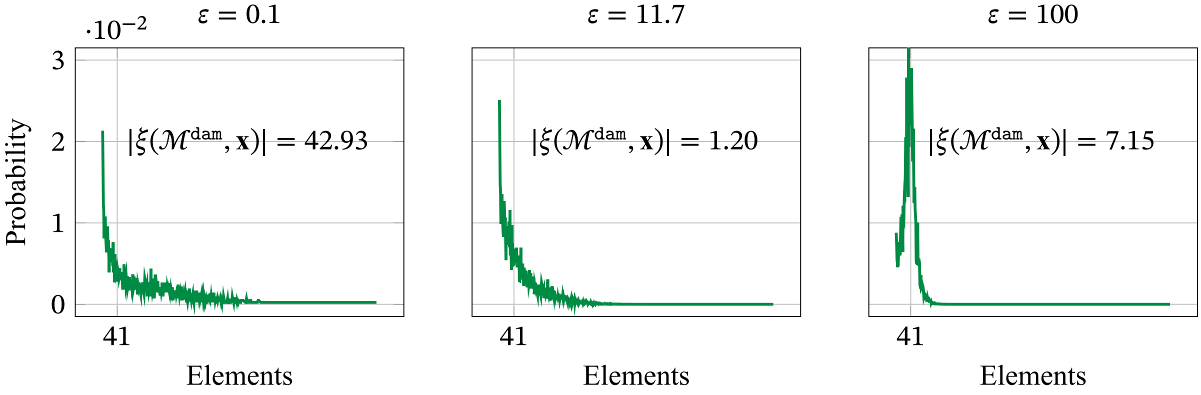
In the LD2 experiments on the HEPTH and PATENT datasets with $p = 50$, we observed a peculiar trend: the absolute expected error initially drops to low levels swiftly. However, as the privacy budget increases, the error, counterintuitively, increases. This rapid

Figure 7 – Comparison of private selection methods for percentile selection. Plots show the absolute expected error (AEE) as a function of the privacy budget of $\varepsilon \in [10^{-1}, 10^{2}]$. The x-axis uses a log scale. Overall, SNM-LAP and SNM-T achieve lower expected errors than other methods for any ε.



Source: elaborated by the author.

Figure 8 – Local Dampening (LD2) probabilities on the HEPTH dataset with $p = 50$. The first plot demonstrates that the probability of selecting element 41 (median) is low when the privacy budget is minimal. The second graph depicts a scenario with very low expected error, suggesting that the observed low expected error occurs by chance. The last plot illustrates that with an increased privacy budget, LD2 converges effectively.



Source: elaborated by the author.

convergence appears to be coincidental, with the algorithm still in the process of converging. Figure 8 visually captures this behavior.

6 APPLICATION — GREEDY DECISION TREE

Decision trees are compelling methods for classification and regression tasks (KOTSIANTIS *et al.*, 2007). A decision tree is a graphical representation of a set of rules, where each node represents a decision based on attributes from the training dataset.

The tree topology is settled by the training algorithm that receives, as input, a dataset and outputs a decision tree. The ID3 algorithm (QUINLAN, 1986) is one of the most known decision tree algorithms. It recursively selects the best attribute, according to some measure, to split the data until a stopping criterion is met. In this work, the split criterion is based on the Max Operator (FRIEDMAN; SCHUSTER, 2010), which is the summation of each attribute value of the class with the highest frequency.

6.1 Problem Statement

Table 4 – Notation table for private decision tree induction

Variable	**Definition**
$MaxOp$	Max Operator
$\mathcal{T}$	Dataset
A	Attribute set
A_i	i-th attribute
C	Class attribute
τ	Cardinality of a dataset $\mathcal{T}$: $\tau = \lvert\mathcal{T}\rvert$
r_A	Values of an attribute A in a record r
r_C	Values of the class attribute C in a record r
$\mathcal{T}_j^A$	Set of records $r \in \mathcal{T}$ where attribute A takes value j: $\mathcal{T}_j^A = \{r \in \mathcal{T} : r_A = j\}$
$\tau_c^{\mathcal{T}}$	Cardinality of $\mathcal{T}_j^A$: $\tau_c^{\mathcal{T}} = \lvert\mathcal{T}_j^A\rvert$
$\tau_c^{\mathcal{T}}$	Number of records $r \in \mathcal{T}$ where class attribute C takes value c: $\tau_c^{\mathcal{T}} = \lvert r \in \mathcal{T} : r_C = c\rvert$
$\tau_{j,c}^{\mathcal{T}}$	Number of records $r \in \mathcal{T}$ where attribute A takes value j and class attribute C takes value c: $\tau_{j,c}^{\mathcal{T}} = \lvert r \in \mathcal{T} : r_A = j \wedge r_c = c\rvert$

Source: elaborated by the author.

A decision tree induction algorithm takes as input a dataset $\mathcal{T}$ with attributes $A = \{A_1, \dots, A_d\}$ and a class attribute C and produces a decision tree. The task is to build a decision tree in a differentially private manner. Specifically, we base our approach on one of the most known tree induction algorithms, the ID3 algorithm. Table 4 shows all the notation used in this chapter.

6.2 Private Mechanism and Sensitivity Analysis

Blum *et al.* (2005) introduced the SuLQ framework, where they designed a differentially private version of ID3 as an application. The adapted application of the ID3 algorithm takes advantage of two SuLQ operators:

i) `NoisyCount`: a Laplace mechanism operator to provide a private estimate for a count query and

ii) `Partition`: an operator that splits the dataset into disjoint subsets.

The primary disadvantage of the ID3 algorithm proposed by Blum *et al.* (2005) is its inefficient use of the privacy budget when evaluating the information gain for each attribute separately. The work presented by Friedman and Schuster (2010), described by Algorithm 4, offers a more effective alternative using the exponential mechanism to evaluate each attribute independently, assessing all attributes simultaneously in a single query, resulting in the selection of an appropriate attribute for splitting. Line 13 is the exponential mechanism call that selects an attribute based on its information gain, which is the utility function. The function `BuildDiffID3`

Algorithm 4: Differentially Private ID3 (from Friedman and Schuster (2010))

```
1  Function GlobalDiffPID3( dataset $\mathcal{T}$, attribute set A, class attribute C, depth d,
     privacy budget ε ) do
2      $\varepsilon' \leftarrow \varepsilon/2\cdot(d+1)$;
3      return BuildDiffPID3($\mathcal{T}, A, C, d, \varepsilon'$)
4  end
5  Function BuildDiffPID3( dataset $\mathcal{T}$, attribute set A, class attribute C, depth d,
     privacy budget ε ) do
6      $t \leftarrow \max_{a \in A} |a|$;
7      $N_{\mathcal{T}} \leftarrow$ NoisyCount$_\varepsilon(\mathcal{T})$;
8      if $A = \emptyset$ or $d = 0$ or $N_{\mathcal{T}}/t|C| < \sqrt{2}/2$ then
9          $\mathcal{T}_c \leftarrow$ Partition($\mathcal{T}, \forall c \in C : r_c = c$);
10         $\forall c \in C : N_c \leftarrow$ NoisyCount$_\varepsilon(\mathcal{T}_c)$;
11         return a leaf labeled with $\arg\max_c N_c$
12     end
13     $\bar{A} \leftarrow \mathcal{M}^{\exp}(\mathcal{T}, \varepsilon, A)$ ;                /* Exp. mechanism call */
14     $\mathcal{T}_i \leftarrow$ Partition($\mathcal{T}, \forall i \in \bar{A} : r_{\bar{A}} = i$);
15     $\forall i \in \bar{A}$ : Subtree$_i \leftarrow$ BuildDiffPID3($\mathcal{T}_i, A \backslash \bar{A}, C, d-1, \varepsilon$);
16     return a tree with a root node labeled $\bar{A}$ and edges labeled 1 to $\bar{A}$ each going to
         Subtree$_i$
17 end
```

in algorithm 4 starts by checking properties like the number of attributes and the number of instances that are used as termination criteria to construct the leaves (lines 5-8). In lines 9-10, the algorithm partitions the dataset based on class labels and counts the instances for each class label. It also employs the Laplace mechanism for each class label count to select the class label for the leaf. Lines 13-16 build new decision rules recursively by privately choosing the attribute with the largest information gain value using the exponential mechanism. Moreover, it splits the dataset according to the selected attribute value and produces recursively new sub-trees for each dataset partition.

Several works address the private construction of decision trees and random forest (FLETCHER; ISLAM, 2015; FLETCHER; ISLAM, 2017; FLETCHER; ISLAM, 2019; JAGANNATHAN *et al.*, 2009; PATIL; SINGH, 2014; RANA *et al.*, 2015). However, only Farias *et al.* (2023) addresses the greedy decision tree construction algorithm applying local sensitivity. The approach proposed by Fletcher and Islam (2017) uses smooth sensitivity in the random forest algorithm through random decision trees. In this section, we focus on the greedy decision tree process. The following section will address the random forest application with random decision trees.

Our differentially private greedy decision tree application is similar to Algorithm 4. We simply replace the exponential mechanism on line 13 with our Smooth Noisy Max, applying a utility function based on the max operator (FRIEDMAN; SCHUSTER, 2010) that represents the summation of each attribute value of the class with the highest frequency.

Definition 6.1 (Max Operator)**.** Consider a dataset $\mathcal{T}$, and an attribute A_i, the Max operator is defined as follows: $MaxOp(\mathcal{T}, A_i) = \sum_{j \in A_i} \max_c \tau_{j,c}^{A_i}$, where $\tau_{j,c}^{A_i}$ counts the records in $\mathcal{T}$ with attribute $A_i = j$ and class $C = c$.

In our experiments, we observed that we should design a utility function representing a good split criterion and take advantage of smooth sensitivity definition to benefit from local sensitivity. Therefore, we define a utility function based on the max operator u_{mo}. That function outputs 1 only for the attribute $A_i \in A$, which is the highest value of $MaxOp$ among all others $A_k \in A$, and 0 otherwise.

Definition 6.2 (Greedy decision tree utility)**.** Consider a dataset $\mathcal{T}$, and an attribute A_j, the utility is defined as follows:

$$u_{mo}(\mathcal{T}, A_j) = \begin{cases} 1 & \text{if } A_j = \arg\max_{A_i \in A} MaxOp(\mathcal{T}, A_i) \\ 0 & \text{otherwise} \end{cases}$$

Global Sensitivity

The global sensitivity for u_{mo} is 1 (FRIEDMAN; SCHUSTER, 2010).

Local Sensitivity

To compute the smooth sensitivity, it is crucial to have a clear understanding of the local sensitivity at a distance of t. Additionally, it is worth noting that the utility value will remain unchanged until k additions or deletions occur in the training dataset $\mathcal{T}$. Here, k refers to the difference between the highest $MaxOp$ attribute and the second-highest attribute in the dataset.

Lemma 6.3 (Greedy decision tree local sensitivity at distance t)**.** Let $\mathcal{T}$ be a dataset, and A_j an attribute. The utility is defined as follows:

$$LS_{u_{mo}}(\mathcal{T}, t) = \begin{cases} 1 & \text{if } t \geq k \\ 0 & \text{otherwise} \end{cases}$$

The local sensitivity remains zero until $t < k$ and changes to one when $t \geq k$. Since the $LS_{u_{mo}}$ is constant when $t < k$, the smooth sensitivity will be max when $t = k$.

Lemma 6.4 (Greedy decision tree smooth sensitivity)**.**

$$\mathcal{S}_{u_{mo}}(\mathcal{T}) = \exp(-k \cdot \varepsilon).$$

6.3 Experimental Evaluation

6.3.1 Datasets

We make use of three tabular datasets:

i) The *National Long Term Care Survey (NLTCS)* (MANTON, 1999), comprising 16 binary attributes of $21,574$ surveyed individuals;

ii) the *American Community Surveys (ACS)* dataset (SERIES, 2015), which includes information from $47,461$ rows with 23 binary attributes, sourced from the 2013 and 2014 ACS sample sets in IPUMS-USA; and

iii) the *Adult* dataset (BLAKE; MERZ, 1998), containing $45,222$ records (excluding those with missing values), featuring 12 attributes, where 8 are discrete and 4 are continuous.

6.3.2 Methods

We experimented with several mechanisms, changing the default selection algorithm described in line 13 of the Algorithm 4.

i) Exponential mechanism (EM) with information gain using global sensitivity;

ii) Permute-and-flip (PF) with information gain using global sensitivity;

iii) Shifted Local dampening (SLD) with information gain using the element local sensitivity (FARIAS *et al.*, 2023);

iv) Smooth Noisy Max using the Laplace Log-Normal distribution (SNM-LLN);

v) Smooth Noisy Max using the Student's T distribution (SNM-T);

vi) Smooth Noisy Max using the Laplace distribution (SNM-LAP).

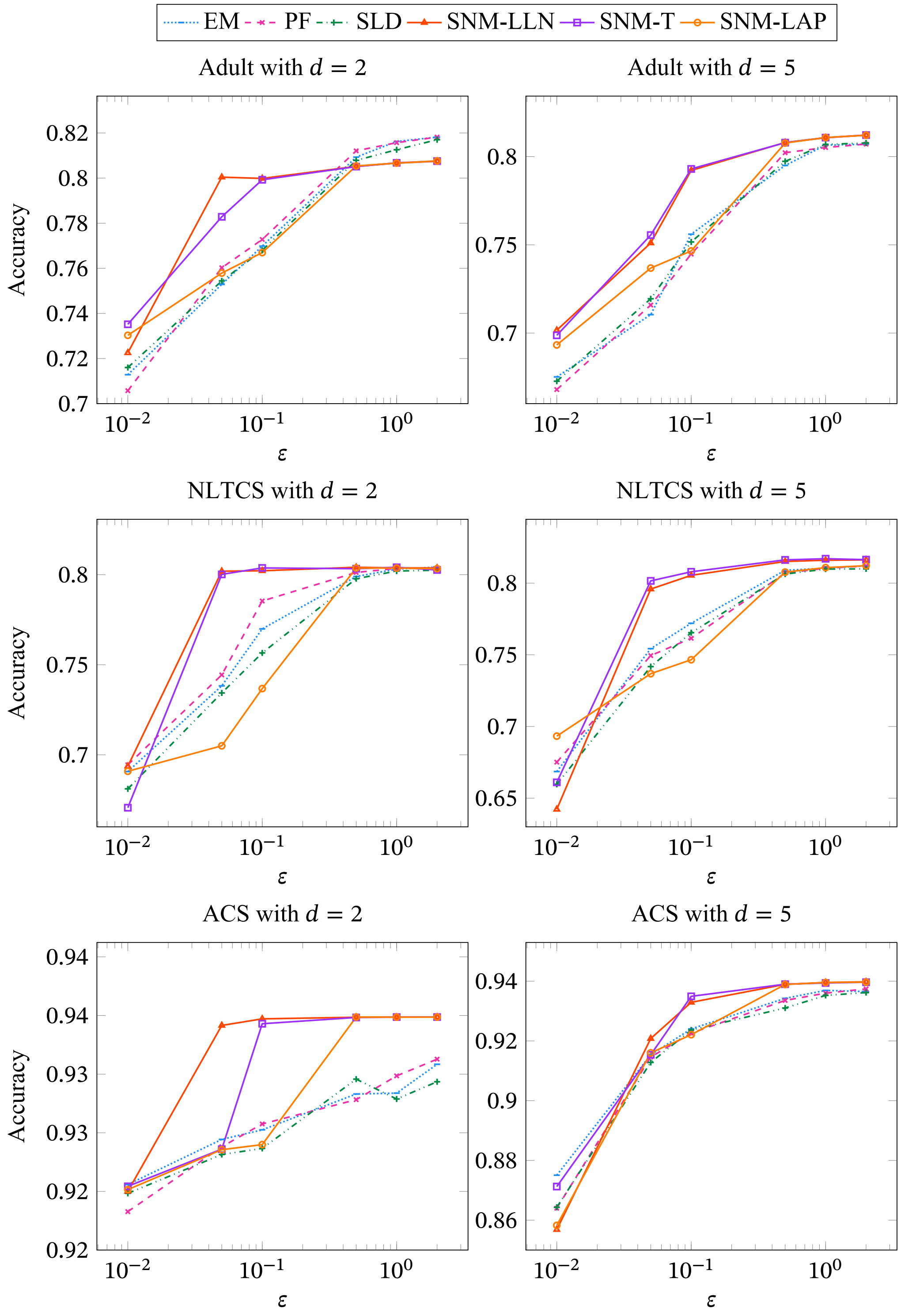
All variants of the Smooth Noisy Max algorithm use a utility function based on the Max Operator as the split criterion, leveraging smooth sensitivity $\mathcal{S}_{u_{mo}}$. Notably, SNM-LLN and SNM-LAP ensure approximate differential privacy ($\delta > 0$) rather than ε-differential privacy.

6.3.3 Evaluation

We measured the accuracy of each method by varying the privacy budget $\varepsilon \in \{0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$ and the max tree depth $d \in \{2, 5\}$. Each trial was measured using 10-fold validation, and each scenario ran 5 times. Figure 9 shows the average accuracy of those scenarios.

We observed that SNM algorithm with Student's T and Laplace Log-Normal outperformed the adversaries in almost all scenarios. In the Adult dataset with $d = 2$, SNM-LLN and SNM-T perform better with low ε values in comparison to the others. However, when the budget is higher, all the competitors have better accuracy, indicating that with the Adult dataset with shallow trees ($d = 2$), the information gain split criteria works better than the max operator even with a higher signal-to-sensitivity ratio when compared with the max operator. Nevertheless, with $d = 5$, SNM-LLN and SNM-T perform better for all ε values. When the dataset is NLTCS, SNM-T has up to 8.58% of improvement in accuracy when compared with the competitors. SNM-T improves up to 1.15% with the ACS dataset compared to the other methods.

Figure 9 – Comparison of private selection methods for the greedy decision tree application. The plots show the mean accuracy of greedy decision tree experiments - 5 runs of 10-fold cross-validation, where $d \in \{2, 5\}$ and $\varepsilon \in \{0.01, 0.05, 0.1, 0.5, 1, 2\}$. X axis is in log scale. All SNM variants consistently achieve superior accuracy compared to competing methods. Notably, the performance of SNM-T is especially significant, as it ensures ε-dp.



Source: elaborated by the author.

7 APPLICATION — RANDOM FOREST

Classification based on decision tree algorithms are remarkable tools for data mining (FRIEDMAN; SCHUSTER, 2010). They also serve as core building blocks for random forests (BREIMAN, 2001). The random forest algorithm is a supervised learning algorithm that combines the predictions of several decision trees, an ensemble of predictors. The algorithm starts by building a set of decision trees and then applies a majority voting to the outcomes of those trees.

The decision tree is a supervised learning algorithm based on a tree structure, where each intermediate node represents a decision based on a feature, and each leaf node represents a label. The algorithm starts from the root node and, based on comparing the feature value with a threshold on numerical features, it splits the tree. The algorithm goes to the right child node if the feature value exceeds the threshold. Otherwise, it goes to the left child node. When the feature selected is categorical, the node has one child for each possible categorical value, and the comparison is made by checking the equality of attribute value. The algorithm continues until it reaches a leaf node when the node's majority label is the tree's outcome.

This section presents an application of a differentially private random forest algorithm using the Smooth Noisy Max as a selection mechanism. The method is a random decision tree designed to save privacy budget in the splitting process. We describe and test the random forest algorithm with several selection mechanisms, including our Smooth Noisy Max, under different scenarios and datasets to compare its results against our competitors.

7.1 Problem Statement

A random forest algorithm takes as input a dataset $\mathbf{x}$ with attributes $F = \{F_1, \dots, F_d\}$, a max depth parameter h, and a parameter c that represents the forest size. The task is to build a forest with c trees $\mathcal{T} = \{\tau_1, \dots, \tau_c\}$ in a differential private manner.

7.2 Random Decision Trees

The most common approaches to building a decision tree are ID3 (QUINLAN, 1986), CART (BREIMAN *et al.*, 1984), and C4.5 (QUINLAN, 1993). They are based on some purity measures as splitting criteria. However, they have a lower generalization performance (BREIMAN, 2001). To overcome this problem, the random decision tree algorithm applies random splitting criteria. The generalization helps ensemble methods like the random forest to add diversity to the ensemble and, therefore, improve the performance (FLETCHER; ISLAM, 2019).

In a greedy decision tree algorithm, the splitting process of a node depends on the input data in the same way that the leaf node class counts dictated by the data, which may leak some information. Considering information leakage, we should prevent these privacy breaches using differential privacy. We must spend some privacy budget whenever data needs to be queried. So, seeking to save privacy budget, the random decision trees apply random split criteria to avoid the usage of privacy budget and save it for the leaf node class counts queries (FLETCHER; ISLAM, 2019).

Fletcher and Islam (2017) propose a random forest algorithm based on random decision trees that satisfies differential privacy. The algorithm applies the exponential mechanism in the leaves to select the majority label.

The work of Fletcher and Islam (2017) is shown by the Algorithm 5. The algorithm starts by splitting the dataset into c chunks. Then, each chunk x_i builds a random tree τ_i. Finally, the algorithm applies the exponential mechanism to select the majority label of the forest and adds the tree to the forest $\mathcal{T}$.

In lines 2 and 3, the dataset is partitioned into c chunks and iterated over it. The build tree function is called in line 4. The build tree function is a conventional recursive approach in that the features are randomly chosen for each node and the split point using only the data's domains, regardless of the data itself. The novel part of the proposed algorithm is the set majority function in line 5. The set majority function applies the exponential mechanism to select the majority label of the leaf node through a specifically designed utility function. The proposed utility function, shown by Definition 7.1, outputs 1 for the label with the highest count in the leaf node and 0 otherwise.

Definition 7.1 (Utility Function (FLETCHER; ISLAM, 2017))**.** The utility function u is defined as:

$$u(\mathbf{x}, r) = \begin{cases} 1 & \text{if } r = \arg\max_{i \in \mathcal{R}} n_i \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

where n_i is the number of samples in the leaf node with label i.

The global sensitivity of the utility function (Definition 7.1) is 1. The work of Fletcher and Islam (2017) applies the smooth sensitivity instead of global sensitivity to reach a

Algorithm 5: Random Forest Algorithm (FLETCHER; ISLAM, 2017)

```
Function buildForest(Dataset $\mathbf{x}$, Forest Size $c$, Features $F$, Depth $h$) do
    for $i \in split(\mathbf{x}, c)$ do
        $\tau \leftarrow$ setMajority(buildTree($x_i, F, h, 0$));
        $\mathcal{T} \leftarrow \mathcal{T} \cup \tau$ ;
    end
end
Function buildTree(Dataset $\mathbf{x}$, Features $F$, Max Depth $h$, Depth $d$) do
    $T \leftarrow \{\}$;
    if $d < h$ then
        Uniformly select attribute $f$ from $F$ to split current node;
        if $f$ is continuous then
            Uniformly select split point $p$ from the $f$'s domain;
            $\mathbf{x}_l, \mathbf{x}_r \leftarrow$ split($\mathbf{x}, f, p$) ;
            $T \cup$ buildTree($\mathbf{x}_l, F, h, d + 1$) $\cup$ buildTree($\mathbf{x}_r, F, h, d + 1$);
        end
    end
    else
        $F \leftarrow F \backslash f$ ;
        forall $a \in f$ do
            $\mathbf{x}_a \leftarrow$ getData($\mathbf{x}, f, a$) ;
            $T \cup$ buildTree($\mathbf{x}_a, F, h, d + 1$);
        end
    end
    return $T$
end
```

better signal-to-noise ratio. However, as proven in the Theorem 7.2 below, it does not satisfy differential privacy.

Theorem 7.2. The exponential mechanism setting $\mathcal{M}^{\exp}_{u,\varepsilon}(\mathbf{x}, r) \propto \exp\left(\frac{\varepsilon u(\mathbf{x},r)}{2\mathcal{S}_{u,\beta}(\mathbf{x})}\right)$ does not satisfy ε-differential privacy with smooth sensitivity instead of global sensitivity.

Proof. Assuming that the exponential mechanism with smooth sensitivity satisfies ε-differential privacy, consider an approval voting example. Here, voters can endorse multiple candidates instead of choosing just one. In this scenario, the utility function assigns a value of 1 to the candidate with the highest votes and 0 to all others.

The utility function exhibits a smooth sensitivity of $\mathcal{S}_{u,\beta}(\mathbf{x}) = \exp(-j\varepsilon)$, where j is the vote disparity between the top candidate and the runner-up in dataset $\mathbf{x}$ (Theorem 7.3). The local sensitivity of the utility function u remains zero until the vote gap j is large enough to affect the comparison, at which point it jumps to 1. The smooth sensitivity peaks when $t = j$, yielding $\mathcal{S}_{u,\beta}(\mathbf{x}) = \exp(-j\varepsilon)$.

For example, consider the output set $\mathcal{R} = [\text{C1}, \text{C2}, \text{C3}, \text{C4}, \text{C5}]$ with the vote count vector $\mathbf{v} = [22, 8, 17, 4, 0]$ from dataset $\mathbf{x}$. Here, candidate C1 leads with 22 votes, followed by others, with C2 receiving 8 votes, and so forth. The utility function (Definition 7.1) assigns a score of 1 solely to candidate C1. The vote difference between the leading candidate and the second-most voted, denoted as j, is 5.

To ensure the differential privacy definition is necessary to address all possible neighboring datasets from $\mathbf{x}$, for instance, the dataset $\mathbf{y}$ by adding one more vote for the second-most voted candidate (C3). Therefore, the j parameter reduces to the value for 4, implying a smooth sensitivity value of 0.135. Using the privacy budget as 0.5, we have $Pr[\mathcal{M}^{\text{exp}}_{u,0.5}(\mathbf{x}, \text{C3})] = 0.04$ and $Pr[\mathcal{M}^{\text{exp}}_{u,0.5}(\mathbf{y}, \text{C3})] = 0.10$, following the Definition 2.5:

$$Pr[\mathcal{M}^{\text{exp}}_{u,0.5}(\mathbf{x}, \text{C3})] \leq e^{0.5} Pr[\mathcal{M}^{\text{exp}}_{u,0.5}(\mathbf{y}, \text{C3})] \Rightarrow 0.04 \leq 0.16 \Rightarrow \top$$

$$Pr[\mathcal{M}^{\text{exp}}_{u,0.5}(\mathbf{y}, \text{C3})] \leq e^{0.5} Pr[\mathcal{M}^{\text{exp}}_{u,0.5}(\mathbf{x}, \text{C3})] \Rightarrow 0.10 \leq 0.07 \Rightarrow \bot$$

Therefore, by contradiction, the exponential mechanism setting does not hold ε-differential privacy with smooth sensitivity instead of global sensitivity. □

To address that issue, we replace the exponential mechanism with our SNM at the Fletcher and Islam (2017)'s random forest algorithm to provide a differentially private selection approach.

Algorithm 6: Set Majority Labels with Smooth Noisy Max

```
1 Function setMajority(Tree $\tau$) do
2     for $l \in \ell$ do
3         $labelCounts \leftarrow l.counts$ ;
4         $l.maj \leftarrow$ SNM($labelCounts$);
5     end
6 end
```

Algorithm 6 details our set majority function, which implements the Smooth Noisy Max algorithm. It begins by traversing all leaves of the tree τ (line 2). For each leaf l, the algorithm retrieves the label counts of the leaf node in line 3. Subsequently, the Smooth Noisy Max algorithm is applied to select the majority label of the leaf node in line 4. It is important to note that to execute the Smooth Noisy Max algorithm, the smooth sensitivity of the utility function is required, as demonstrated in Theorem 7.3.

Theorem 7.3 (Smooth sensitivity of Def. 7.1 (FLETCHER; ISLAM, 2019))**.** The smooth sensitivity of the utility function u (definition 7.1) is: $\mathcal{S}_{u,\beta}(\mathbf{x}) = \exp(-j\varepsilon)$, where j is the difference between the most frequent and the second-most frequent labels in $\mathbf{x}$.

7.3 Experimental Evaluation

This section presents the datasets, methods, and experimental evaluation results. We selected six datasets to evaluate the performance of our proposed method compared with other baselines.

7.3.1 Methods

Our evaluation employs the standard random forest algorithm (Algorithm 5). We term the non-private implementation of Algorithm 5 as WDP. The experiment employs various selection mechanisms, including the exponential mechanism (EM), permute-and-flip (PF), local dampening mechanism (LD), Smooth Noisy Max with Laplace Log-Normal distribution (SNM-LLN), Smooth Noisy Max with Student's T distribution (SNM-T), and Smooth Noisy Max with Laplace distribution (SNM-LAP). We configure all privacy-preserving mechanisms, excluding the non-private method, with the utility function defined in Definition 7.1. EM and PF utilize a global sensitivity of 1.0. We empirically determine the element local sensitivity across each dataset for local dampening. We follow Theorem 7.3 to find the smooth sensitivity within our Smooth Noisy Max. We aim to measure the accuracy impact of choosing the Smooth Noisy Max as a private selection method.

7.3.2 Evaluation

We measured the accuracy of those methods over ten executions using the accuracy metric. The process split the dataset in 80% for the training step and 20% for evaluation purposes. The privacy budget varies by $\{0.01, 0.05, 0.1, 1, 2\}$. We also set each random forest with 32 trees. The max tree depth was set for each dataset using Theorem 2 from Fletcher and Islam (2019)'s work.

7.3.3 Results

Our experimental procedure compares our method using the accuracy metric in 6 datasets. The datasets were selected based on their size, number of features, and number of classes. The Adult dataset comprises 48,842 instances with 6 continuous and 8 discrete features, a maximum tree depth of 9, and 2 classes. Compas contains 4,732 entries, 9 continuous and 4 discrete features, a depth of 5, and 11 classes. The Wine dataset involves 4,898 samples, all 11 continuous features, a depth of 10, and 7 classes. Mushroom includes 8,124 entries, 22 discrete features, a maximum depth of 11, and 2 classes. The Pen-digits dataset, one of the largest with 109,092 instances, features 17 continuous attributes, a depth of 12, and 10 classes. Finally, Wall-sensor offers 5,456 samples, 4 continuous features, a depth of 4, and 4 classes. Figure 10 shows the result of our proposed random forest algorithm using the Smooth Noisy Max with the other selection algorithms varying the budget parameter. Table 5 shows a characteristics comparison of the datasets.

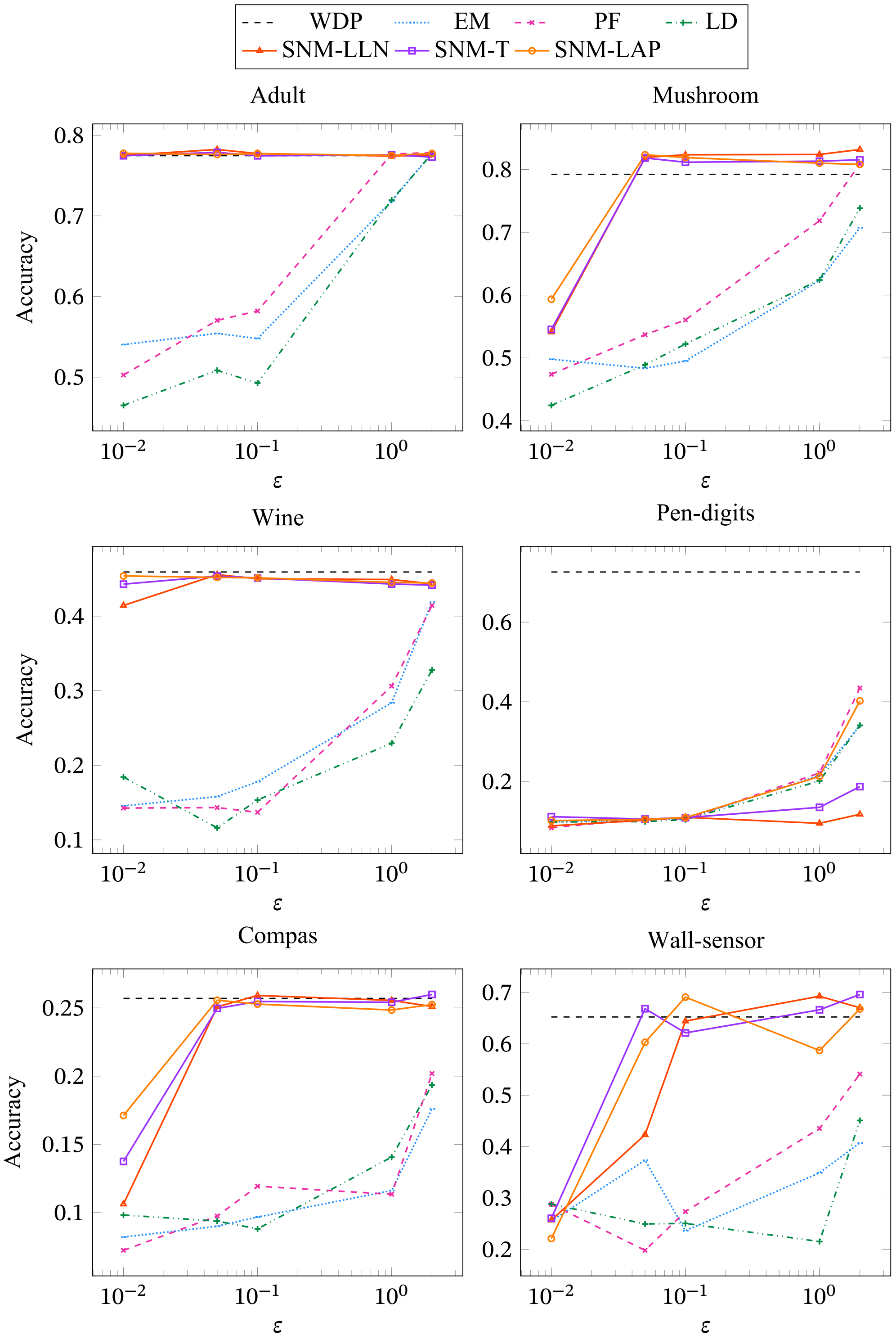
Table 5 – Dataset characteristics comparison.

	Adult	Compas	Wine	Mushroom	Pen-digits	Wall-sensor
Size	large	small	small	small	large	small
Mostly	discrete	continuous	continuous	discrete	continuous	continuous
Class amount	small	large	large	small	large	small

Source: elaborated by the author.

Firstly, we focus on the experiments using the Mushroom (REPOSITORY, 1987) and Adult (BECKER; KOHAVI, 1996) datasets. Both datasets have mostly discrete attributes and few classes but differ in size. The Adult dataset has more than 48 thousand tuples compared to almost 8 thousand in the Mushroom dataset. The max depth was set to 9 and 11 for Adult and Mushroom datasets. Looking at the results of our experiment in the Adult dataset, we can observe that even with a small privacy budget, we can deliver excellent accuracy results, achieving the version without private guarantees. Using the Mushroom dataset, when the budget is 0.1, all the versions of Smooth Noisy Max surpass the standard random forest, *i.e.*, our private method is better than the privateless version. By our method, the randomness input can improve the power of the tree generalization (BREIMAN, 2001), leading to better accuracy. Fixing with a privacy budget of 1, our method is similar to permute-and-flip's accuracy performance, but using the privacy budget of 0.01, we can deliver the same accuracy performance as the permute-and-flip

Figure 10 – Comparison of private selection methods for the random forest problem. The plots show mean accuracy for WDP, EM, PF, LD, and SNM random forest variants with 32 random trees varying $\varepsilon \in \{0.01, 0.05, 0.1, 1, 2\}$. X is in the log scale. The SNM flavors constantly reach the standard non-private random forest accuracy level. When compared with other private selection methods, the variants of SNM surpass in almost all ε values.



Source: elaborated by the author.

with 100 times more budget (showed by Figure 10).

The Wine Quality dataset (CORTEZ *et al.*, 2009) has almost 5 thousand records with 11 continuous features and zero discrete features. The Pen-Based Recognition of Handwritten Digits (Pen-digits) dataset (ALPAYDIN, 1996) has more than 109 thousand records with 17 continuous features and zero discrete features. The maximum depth was set to 10 and 12 for the wine and pen-digit datasets. The random forest results with all the versions of SNM reach almost the non-private version in the wine dataset, outperforming all the adversaries even with very low privacy budget values. All the private methods underperform in the experiments using the pen-digits dataset, mainly because of the dataset's j (Theorem 7.3) value. The difference between the highest class count and the second highest is narrow in the pen-digits dataset, implying a smooth sensitivity almost equal to global sensitivity.

The Compas and the Wall-sensor datasets have similar sizes but differ in the number of classes. The Compas (Correctional Offender Management Profiling for Alternative Sanctions) dataset (PROPUBLICA, 2016) has 11 classes, and the Wall-Following Robot Navigation Dataset (Wall-sensor) (FREIRE; BARRETO, 2009) has only four classes. Figure 10 shows that our proposed SNM versions outperform the private selection adversaries using the Compas and wall-sensor datasets. Even with many classes, the proposed random forest algorithm employing SNM with a small privacy budget reaches the standard random forest without any privacy concerns.

8 CONCLUSION

This chapter presents a comprehensive overview of the findings obtained through our research. We faced the problem of differentially private selection using smooth sensitivity —a instance-based sensitivity— implying lower error rates when compared with using the global sensitivity. We formally describe our approach, its privacy attributes, and its utility. Section 8.1 summarizes the achieved results compared with our hypothesis.

8.1 Summary of Results

We revisit the thesis hypothesis in this section:

Hypothesis: *"Applying smooth sensitivity – an instance-based sensitivity measure – to the private selection mechanism guarantees* (ε, δ)*-differential privacy and yields superior outcomes compared to methodologies that apply global sensitivity, especially higher accuracy or reduced expected error. This assertion also extends to comparisons with competing approaches that employ instance-based sensitivity, such as local dampening."*

We prove that our proposed method, Smooth Noisy Max, is (ε, δ)-differentially private. We also demonstrated that our algorithm's theoretical utility bounds, leveraging the Laplace distribution, consistently match or exceed that of competing methods under mild conditions. Additionally, we empirically compare Smooth Noisy Max with the competitors, e.g., local dampening, across various private selection scenarios:

i) **Percentile selection:** The percentile selection application utilized the Smooth Noisy Max algorithm to determine the p-th percentile in a dataset under the constraints of differential privacy. The Smooth Noisy Max algorithm demonstrated superior performance in preserving privacy while accurately estimating the p-th percentile. Compared to competitors, the use of smooth sensitivity provided a more nuanced approach, reducing the error rates and enhancing the reliability of percentile calculations under privacy constraints.

ii) **Greedy decision trees:** In the greedy decision tree application, the focus was on constructing decision trees using a privacy-preserving version of the ID3 algorithm. Decision trees are a fundamental component of many machine learning tasks, and ensuring their construction while adhering to privacy standards is essential for applications in sensitive domains. Implementing the Smooth Noisy Max within this context allowed for the effective selection of attributes while maintaining differential privacy. The trees generated using this approach were found to be nearly as accurate as non-private counterparts, showcasing the algorithm's efficacy in maintaining utility despite the privacy enhancements.

iii) **Random forest:** Extending the privacy-preserving techniques to ensemble methods, this application developed a novel approach for constructing random forests. Each decision tree within the forest was built using the Smooth Noisy Max algorithm to ensure the privacy of the underlying data. Our random forests achieved competitive accuracy with traditional random forests and beat the competitors while ensuring that each decision within the forest complied with differential privacy standards.

In conclusion, each application we experimented with in the research underscores the versatility and effectiveness of the Smooth Noisy Max algorithm in various data science and machine learning contexts. From basic statistical measures like percentiles to more complex models like decision trees and random forests, the algorithm preserves privacy and retains a high level of data utility.

In the experimental evaluation, a limitation of our proposed algorithm emerged, linked to the concept of local sensitivity at a distance t, which tends to converge rapidly to global sensitivity. This convergence is attributed to the max operator iterating over all possible outcomes, as specified in Definition 3.4. To mitigate this issue, it was imperative to devise specialized utility functions where, for instance, only the optimal response carries non-zero utility. Another challenge associated with smooth sensitivity is its computational complexity, which necessitates computationally intensive algorithms. To circumvent this, we employed simplified utility functions for which smooth sensitivity could be analytically determined, thus enhancing the computational efficiency of our approach.

8.2 Future Work

Future research emerging from this dissertation will focus on developing useful utility bounds for different noise distributions, particularly the Student's T and Laplace Log-Normal distributions. This direction is critical as it promises to broaden the applicability and effectiveness of Smooth Noisy Max under various data characteristics. By establishing rigorous utility bounds for these distributions, we aim to provide tools for the data analyst to enhance the adaptability and accuracy of privacy-preserving algorithms. Such advancements will provide deeper theoretical insights and facilitate more nuanced implementations that can be tailored to specific real-world datasets and privacy scenarios, thereby improving the trade-off between privacy protection and data utility in sensitive applications.

The incorporation of the concept of element local sensitivity presents a promising avenue for refining the Smooth Noisy Max. We believe that this approach could significantly enhance the method's ability to balance utility and privacy by providing more granular control over sensitivity adjustments based on each specific output. Future work will focus on extensively validating the application of element local sensitivity within the Smooth Noisy Max, proving differential privacy property, and exploring its potential to improve utility performance across various datasets and applications.

Additionally, formulating problems as single functions can be problematic due to the complexity and multifaceted nature of real-world scenarios. While grouping various objectives into a single function via weighting or ranking is theoretically feasible, this approach may not fully encapsulate the subtleties of certain issues. Therefore, investigating differentially private multi-objective selection represents a forthcoming area of research for us.

8.3 Broader Impact

The broader impacts of the research presented in this dissertation extend significantly beyond the technical contributions of enhancing privacy-preserving algorithms. Firstly, the advancement in differentially private selection mechanisms has profound implications for safeguarding individual privacy in an increasingly data-driven world. By refining and enhancing the effectiveness of algorithms using Smooth Noisy Max, this work not only improves the practical implementation of differential privacy but also broadens its applicability across various industries, including healthcare, finance, and social networking. These sectors often handle sensitive personal information where privacy is paramount, and improved algorithms help ensure that this data can be utilized for analytics without compromising individual privacy rights.

Moreover, the societal implications of these developments cannot be overstated. As organizations and governmental bodies increasingly rely on big data to make decisions that affect daily life, ensuring the privacy of individuals within these datasets becomes crucial. This research contributes to foundational technology and tools that can prevent misuse of personal data and mitigate risks associated with data breaches and unauthorized surveillance. The proposed algorithm allows for secure data analysis, providing insights that can drive policy-making, medical research, and personalized services, all while maintaining the confidentiality of the data subjects.

In an educational and policy-making context, the findings of this dissertation provide tools that can encourage a shift towards more responsible data practices. Demonstrating the viability of sophisticated privacy-preserving techniques provides a strong argument for regulators and policymakers to require or encourage the use of differential privacy in data analytics operations. This could lead to enhanced data protection regulations and standards that more effectively balance the utility of data analytics with the privacy expectations of individuals, fostering trust in digital services and technologies.

REFERENCES

ALPAYDIN, F. A. E. **Pen-Based Recognition of Handwritten Digits**. UCI Machine Learning Repository, 1996. Available at: https://archive.ics.uci.edu/dataset/81. Accessed on: 2024-09-23.

ALTMAN, M.; WOOD, A.; O'BRIEN, D. R.; VADHAN, S.; GASSER, U. Towards a modern approach to privacy-aware government data releases. **Berkeley Technology Law Journal**, JSTOR, v. 30, n. 3, p. 1967–2072, 2015.

ALVES, D.; FARIAS, V. A. E. de; CHAVES, I. C.; CHAO, R.; MADEIRO, J. P.; GOMES, J. P. P.; MACHADO, J. C. Detecting customer induced damages in motherboards with deep neural networks. In: **International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022**. IEEE, 2022. p. 1–8. Available at: https://doi.org/10.1109/IJCNN55064.2022.9892047.

BECKER, B. G.; KOHAVI, R. **Adult**. UCI Machine Learning Repository, 1996. Available at: https://doi.org/10.24432/C5XW20. Accessed on: 2024-09-23.

BLAKE, C. L.; MERZ, C. J. **UCI repository of machine learning databases**. 1998.

BLUM, A.; DWORK, C.; MCSHERRY, F.; NISSIM, K. Practical privacy: the SuLQ framework. In: LI, C. (Ed.). **Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA**. ACM, 2005. p. 128–138. Available at: https://doi.org/10.1145/1065167.1065184.

BRASIL. **Lei Geral de Proteção de Dados**. 2018. Planalto. Available at: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm. Accessed on: 2024-09-23.

BREIMAN, L. Random Forests. **Mach. Learn.**, v. 45, n. 1, p. 5–32, 2001. Available at: https://doi.org/10.1023/A:1010933404324.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. **Classification and Regression Trees**. United States of America: Wadsworth, 1984. ISBN 0-534-98053-8.

BUN, M.; STEINKE, T. Average-case averages: Private algorithms for smooth sensitivity and mean estimation. In: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; ALCHé-BUC, F. d'; FOX, E.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2019. v. 32. Available at: https://proceedings.neurips.cc/paper_files/paper/2019/file/3ef815416f775098fe977004015c6193-Paper.pdf.

CHAUDHURI, K.; SARWATE, A. D.; SINHA, K. A near-optimal algorithm for differentially-private principal components. **J. Mach. Learn. Res.**, v. 14, n. 1, p. 2905–2943, 2013. Available at: https://dl.acm.org/doi/10.5555/2567709.2567754.

CHAVES, I. C.; FARIAS, V. A. E.; PEREZ, A.; MESQUITA, D.; MACHADO, J. **Differentially Private Selection using Smooth Sensitivity**. 2024. Submitted for publication to *SIGMOD International Conference on Management of Data (2025)*.

CHAVES, I. C.; MACHADO, J. C. Differentially private group-by data releasing algorithm. In: **Proceedings of the 34th Brazilian Symposium on Databases, SBBD 2019, Fortaleza, CE, Brazil, October 7-10, 2019**. SBC, 2019. p. 271–276. Available at: https://doi.org/10.5753/sbbd.2019.8835.

CHAVES, I. C.; MARTINS, A. D. F.; PRACIANO, F. D. B. S.; BRITO, F. T.; MONTEIRO, J. M.; MACHADO, J. C. BPA: A multilingual sentiment analysis approach based on bilstm. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). **Proceedings of the 24th International Conference on Enterprise Information Systems, ICEIS 2022, Online Streaming, April 25-27, 2022, Volume 1**. SCITEPRESS, 2022. p. 553–560. Available at: https://doi.org/10.5220/0011071400003179.

CHEN, R.; XIAO, Q.; ZHANG, Y.; XU, J. Differentially Private High-dimensional Data Publication via Sampling-based Inference. In: CAO, L.; ZHANG, C.; JOACHIMS, T.; WEBB, G. I.; MARGINEANTU, D. D.; WILLIAMS, G. (Ed.). **Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015**. ACM, 2015. p. 129–138. Available at: https://doi.org/10.1145/2783258.2783379.

CORTEZ, P.; CERDEIRA, A.; ALMEIDA, F.; MATOS, T.; REIS, J. Modeling wine preferences by data mining from physicochemical properties. **Decis. Support Syst.**, v. 47, n. 4, p. 547–553, 2009. Available at: https://doi.org/10.1016/j.dss.2009.05.016.

DING, Z.; KIFER, D.; E., S. M. S. N.; STEINKE, T.; WANG, Y.; XIAO, Y.; ZHANG, D. The Permute-and-flip Mechanism is Identical to Report-N oisy-max with Exponential Noise. **CoRR**, abs/2105.07260, 2021. Available at: https://arxiv.org/abs/2105.07260.

DING, Z.; WANG, Y.; XIAO, Y.; WANG, G.; ZHANG, D.; KIFER, D. Free gap estimates from the exponential mechanism, sparse vector, noisy max and related algorithms. **VLDB J.**, v. 32, n. 1, p. 23–48, 2023. Available at: https://doi.org/10.1007/s00778-022-00728-2.

DURFEE, D.; ROGERS, R. M. Practical differentially private top-k selection with pay-what-you-get composition. In: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; ALCHé-BUC, F. d'; FOX, E.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2019. v. 32. Available at: https://proceedings.neurips.cc/paper_files/paper/2019/file/b139e104214a08ae3f2ebcce149cdf6e-Paper.pdf.

DWORK, C.; MCSHERRY, F.; NISSIM, K.; SMITH, A. D. Calibrating Noise to Sensitivity in Private Data Analysis. In: HALEVI, S.; RABIN, T. (Ed.). **Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings**. Springer, 2006. (Lecture Notes in Computer Science, v. 3876), p. 265–284. Available at: https://doi.org/10.1007/11681878_14.

DWORK, C.; ROTH, A. The Algorithmic Foundations of Differential Privacy. **Found. Trends Theor. Comput. Sci.**, v. 9, n. 3-4, p. 211–407, 2014. Available at: https://doi.org/10.1561/0400000042.

EUROPE. **General Data Protection Regulation (GDPR) – Official Legal Text**. 2018. Europe. Available at: https://gdpr-info.eu/. Accessed on: 2024-09-23.

FARIAS, V. A. E. d. **Local dampening: differential privacy for non-numeric queries via local sensitivity**. Phd Thesis (PhD thesis) — Universidade Federal do Ceará, 2021. Available at: http://repositorio.ufc.br/handle/riufc/59462.

FARIAS, V. A. E. de; BRITO, F. T.; FLYNN, C. J.; MACHADO, J. C.; MAJUMDAR, S.; SRIVASTAVA, D. Local dampening: differential privacy for non-numeric queries via local sensitivity. **VLDB J.**, v. 32, n. 6, p. 1191–1214, 2023. Available at: https://doi.org/10.1007/s00778-022-00774-w.

FERRY, J.; FUKASAWA, R.; PASCAL, T.; VIDAL, T. Trained random forests completely reveal your dataset. **CoRR**, abs/2402.19232, 2024. Available at: https://-doi.org/10.48550/arXiv.2402.19232.

FLETCHER, S.; ISLAM, M. Z. A Differentially Private Decision Forest. In: ONG, K.; ZHAO, Y.; STONE, M. G.; ISLAM, M. Z. (Ed.). **Thirteenth Australasian Data Mining Conference, AusDM 2015, Sydney, Australia, August 2015**. Australian Computer Society, 2015. (CRPIT, v. 168), p. 99–108. Available at: http://crpit.scem.westernsydney.edu.au/abstract-s/CRPITV168Fletcher.html.

FLETCHER, S.; ISLAM, M. Z. Differentially private random decision forests using smooth sensitivity. **Expert Syst. Appl.**, v. 78, p. 16–31, 2017. Available at: https://-doi.org/10.1016/j.eswa.2017.01.034.

FLETCHER, S.; ISLAM, M. Z. Decision Tree Classification with Differential Privacy. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 52, n. 4, p. 1–33, Aug. 2019.

FREIRE, M. V. A.; BARRETO, G. **Wall-Following Robot Navigation Data**. UCI Machine Learning Repository, 2009. Available at: https://archive.ics.uci.edu/dataset/194. Accessed on: 2024-09-23.

FRIEDMAN, A.; SCHUSTER, A. Data mining with differential privacy. In: RAO, B.; KRISHNAPURAM, B.; TOMKINS, A.; YANG, Q. (Ed.). **Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010**. ACM, 2010. p. 493–502. Available at: https://doi.org/10.1145/1835804.1835868.

GONEM, A.; GILAD-BACHRACH, R. Smooth Sensitivity Based Approach for Differentially P rivate PCA. In: JANOOS, F.; MOHRI, M.; SRIDHARAN, K. (Ed.). **Algorithmic Learning Theory, ALT 2018, 7-9 April 2018, Lanzarote, Canary Islands, Spain**. PMLR, 2018. (Proceedings of Machine Learning Research, v. 83), p. 438–450. Available at: http://proceedings.mlr.press/v83/gonem18a.html.

HAY, M.; MACHANAVAJJHALA, A.; MIKLAU, G.; CHEN, Y.; ZHANG, D. Principled Evaluation of Differentially Private Algorithms using DPBench. In: ÖZCAN, F.; KOUTRIKA, G.; MADDEN, S. (Ed.). **Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016**. ACM, 2016. p. 139–154. Available at: https://doi.org/10.1145/2882903.2882931.

ILYAS, I. F.; BESKALES, G.; SOLIMAN, M. A. A survey of top-k query processing techniques in relational database systems. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 40, n. 4, p. 1–58, 2008.

JAGANNATHAN, G.; PILLAIPAKKAMNATT, K.; WRIGHT, R. N. A Practical Differentially Private Random Decision Tree Classifier. In: SAYGIN, Y.; YU, J. X.; KARGUPTA, H.; WANG, W.; RANKA, S.; YU, P. S.; WU, X. (Ed.). **ICDM Workshops 2009, IEEE International Conference on Data Mining Workshops, Miami, Florida, USA, 6 December 2009**. IEEE Computer Society, 2009. p. 114–121. Available at: https://doi.org/10.1109/ICDMW.2009.93.

KIFER, D.; MACHANAVAJJHALA, A. No free lunch in data privacy. In: SELLIS, T. K.; MILLER, R. J.; KEMENTSIETSIDIS, A.; VELEGRAKIS, Y. (Ed.). **Proceedings**

of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011. ACM, 2011. p. 193–204. Available at: https://doi.org/10.1145/1989323.1989345.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. *et al.* Supervised machine learning: A review of classification techniques. **Emerging artificial intelligence applications in computer engineering**, Amsterdam, v. 160, n. 1, p. 3–24, 2007.

LIMA, F. D. S.; PEREIRA, F. L. F.; CHAVES, I. C.; MACHADO, J. C.; GOMES, J. P. P. Predicting the health degree of hard disk drives with asymmetric and ordinal deep neural models. **IEEE Trans. Computers**, v. 70, n. 2, p. 188–198, 2021. Available at: https://doi.org/10.1109/TC.2020.2987018.

MANTON, K. G. National Long Term Care Survey. **Encyclopedia of Aging, Second Edition. Springer, New York**, 1999.

MCKENNA, R.; SHELDON, D. R. Permute-and-flip: A new mechanism for differentially private selection. In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M.; LIN, H. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2020. v. 33, p. 193–203. Available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/01e00f2f4bfcbb7505cb641066f2859b-Paper.pdf.

MCSHERRY, F.; TALWAR, K. Mechanism Design via Differential Privacy. In: **48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings**. IEEE Computer Society, 2007. p. 94–103. Available at: https://doi.org/10.1109/FOCS.2007.41.

NARAYANAN, A.; SHMATIKOV, V. Robust de-anonymization of large sparse datasets. In: **2008 IEEE Symposium on Security and Privacy (SP 2008), 18-21 May 2008, Oakland, California, USA**. IEEE Computer Society, 2008. p. 111–125. Available at: https://doi.org/10.1109/SP.2008.33.

NISSIM, K.; RASKHODNIKOVA, S.; SMITH, A. D. Smooth sensitivity and sampling in private data analysis. In: JOHNSON, D. S.; FEIGE, U. (Ed.). **Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007**. ACM, 2007. p. 75–84. Available at: https://doi.org/10.1145/1250790.1250803.

PATIL, A.; SINGH, S. Differential private random forest. In: **2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014, Delhi, India, September 24-27, 2014**. IEEE, 2014. p. 2623–2630. Available at: https://doi.org/10.1109/ICACCI.2014.6968348.

PEREIRA, F. L. F.; CHAVES, I. C.; GOMES, J. P. P.; MACHADO, J. C. Using autoencoders for anomaly detection in hard disk drives. In: **2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020**. IEEE, 2020. p. 1–7. Available at: https://doi.org/10.1109/IJCNN48605.2020.9206689.

PROPUBLICA. **GitHub - propublica/compas-analysis: Data and analysis for “M achine Bias” — github.com**. 2016. Github. Accessed on: 2024-09-23.

QUINLAN, J. R. Induction of Decision Trees. **Mach. Learn.**, v. 1, n. 1, p. 81–106, 1986. Available at: https://doi.org/10.1023/A:1022643204877.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. Burlington, Massachusetts, USA: Morgan Kaufmann, 1993. ISBN 1-55860-238-0.

RANA, S.; GUPTA, S. K.; VENKATESH, S. Differentially Private Random Forest with High Utility. In: AGGARWAL, C. C.; ZHOU, Z.; TUZHILIN, A.; XIONG, H.; WU, X. (Ed.). **2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015**. IEEE Computer Society, 2015. p. 955–960. Available at: https://doi.org/10.1109/ICDM.2015.76.

REPOSITORY, U. M. L. **Mushroom**. 1987. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5959T. Accessed on: 2024-09-23.

SENA, L. B.; PRACIANO, F. D. B. S.; CHAVES, I. C.; BRITO, F. T.; NETO, E. R. D.; MONTEIRO, J. M.; MACHADO, J. C. AUDIO-MC: A general framework for multi-context audio classification. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). **Proceedings of the 24th International Conference on Enterprise Information Systems, ICEIS 2022, Online Streaming, April 25-27, 2022, Volume 1**. SCITEPRESS, 2022. p. 374–383. Available at: https://doi.org/10.5220/0011071500003179.

SERIES, I. P. U. M. Version 6.0. **Minneapolis: University of**, 2015.

SHOKRI, R.; STRONATI, M.; SONG, C.; SHMATIKOV, V. Membership inference attacks against machine learning models. In: **2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017**. IEEE Computer Society, 2017. p. 3–18. Available at: https://doi.org/10.1109/SP.2017.41.

SILVA, M. de L. M.; CHAVES, I. C.; MACHADO, J. C. Private reverse top-k algorithms applied on public data of COVID-19 in the state of ceará. **J. Inf. Data Manag.**, v. 12, n. 5, 2021. Available at: https://sol.sbc.org.br/journals/index.php/jidm/article/view/1941.

SILVA, M. de L. M.; CHAVES, I. C.; MACHADO, J. de C. Aplicação de top-k reverso com privacidade sobre os dados públicos de COVID-19 no estado do ceará. In: **Proceedings of the 35th Brazilian Symposium on Databases, SBBD 2020, Online, September 28 - October 1, 2020**. SBC, 2020. p. 193–198. Available at: https://doi.org/10.5753/sbbd.2020.13640.

STONEBRAKER, M.; BRUCKNER, D.; ILYAS, I. F.; BESKALES, G.; CHERNIACK, M.; ZDONIK, S. B.; PAGAN, A.; XU, S. Data curation at scale: The data tamer system. In: **Sixth Biennial Conference on Innovative Data Systems Research, CIDR 2013, Asilomar, CA, USA, January 6-9, 2013, Online Proceedings**. www.cidrdb.org, 2013. Available at: http://cidrdb.org/cidr2013/Papers/CIDR13_Paper28.pdf.

SUN, L.; ZHOU, Y.; YU, P. S.; XIONG, C. Differentially Private Deep Learning with Smooth S ensitivity. **CoRR**, abs/2003.00505, 2020. Available at: https://arxiv.org/abs/2003.00505.

ZHANG, J.; CORMODE, G.; PROCOPIUC, C. M.; SRIVASTAVA, D.; XIAO, X. PrivBayes: private data release via bayesian networks. In: DYRESON, C. E.; LI, F.; ÖZSU, M. T. (Ed.). **International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014**. ACM, 2014. p. 1423–1434. Available at: https://doi.org/10.1145/2588555.2588573.

ZHANG, J.; CORMODE, G.; PROCOPIUC, C. M.; SRIVASTAVA, D.; XIAO, X. Private Release of Graph Statistics using Ladder F unctions. In: SELLIS, T. K.; DAVIDSON, S. B.;

IVES, Z. G. (Ed.). **Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015**. ACM, 2015. p. 731–745. Available at: https://doi.org/10.1145/2723372.2737785.