UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

VICTOR AGUIAR EVANGELISTA DE FARIAS

LOCAL DAMPENING: DIFFERENTIAL PRIVACY FOR NON-NUMERIC QUERIES VIA LOCAL SENSITIVITY

FORTALEZA

2021

VICTOR AGUIAR EVANGELISTA DE FARIAS

LOCAL DAMPENING: DIFFERENTIAL PRIVACY FOR NON-NUMERIC QUERIES VIA LOCAL SENSITIVITY

Tese apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Bancos de Dados

Orientador: Prof. Dr. Javam de Castro Machado

FORTALEZA

2021

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

F238l Farias, Victor Aguiar Evangelista de.
Local Dampening : Differential Privacy for Non-numeric Queries via Local Sensitivity / Victor Aguiar Evangelista de Farias. – 2021.
100 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2021.
Orientação: Prof. Dr. Javam de Castro Machado.

1. Privacidade Diferencial. 2. Anonimização de dados. 3. Análise em grafos. 4. Árvores de decisão. I. Título.

CDD 005

VICTOR AGUIAR EVANGELISTA DE FARIAS

LOCAL DAMPENING: DIFFERENTIAL PRIVACY FOR NON-NUMERIC QUERIES VIA LOCAL SENSITIVITY

Tese apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Bancos de Dados

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Javam de Castro Machado (Orientador)
Universidade Federal do Ceará (UFC)

Dr. Divesh Srivastava
AT&T Labs Research - USA

Profa. Dr. Agma Juci Machado Traina
Universidade de São Paulo – São Carlos (USP)

Prof. Dr. Altigran Soares da Silva
Universidade Federal do Amazonas (UFAM)

Prof. Dr. José Soares Andrade Junior
Universidade Federal do Ceará (UFC)

Prof. Dr. João Paulo Pordeus Gomes
Universidade Federal do Ceará (UFC)

Aos meu pais, por sempre terem acreditado em mim.

ACKNOWLEDGEMENTS

This acknowledgement is to express my deep gratitude to all who were by my side on this journey.

To my parents, Clara Maria Nantua Evangelista de Farias e Francisco Aguiar de Farias Junior, for the care, patience, trust and education during my whole life. Everything I achieved in this life, I own to them.

To my advisor and mentor, Prof. Dr. Javam de Castro Machado, for all the contributions and opportunities given during this journey of ten years now, since when I was an undergraduate student. He was crucial for my journey to become a researcher.

To Dr. Divesh Srivastava for welcoming me in AT&T Research Labs - NYC - USA for ten months to work on this thesis. I'm really glad that we have been collaborating for this time and this thesis is a product of this collaboration.

To Prof. Dr. Agma Juci Machado Traina, Prof. Dr. Altigran Soares da Silva, Prof. Dr. José Soares Andrade Junior and Prof. Dr. João Paulo Pordeus Gomes, for accepting to be in my thesis defense.

To my colleagues at AT&T labs research, Cheryl Flynn Brooks and Subho Majumdar, for all the contributions to this work and my friend, Lauro Lins, at AT&T labs research for the teachings about science and our good time in NYC.

To my girlfriend, Bruna Prudêncio de Mendonça, who faced my dream of pursuing a Phd as her dream and for all the love, support, patience on the most hard and stressful times during this journey.

To my friends, Felipe Timbó, and his wife, Isabelle Timbó, for companionship. I appreciate all the help with this work and with the arrangements of my stay in NYC.

To my friends, Antônio, Camila, Gustavo and Lucas, that were always there for me even in different countries and cities. To all my friends in NYC. To all my childhood friends from the neighborhood where I grew up. To my friends and colleagues at LSBD, specially the ones of laboratory 4.

To my colleague professors of UFC Quixadá campus where I have been working for five years so far. To the professors of the computer science department of UFC for the motivation over the years.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 88887.364998/2019-00.

> “No book can ever be finished. While working on it we learn just enough to find it immature the moment we turn away from it.”
>
> (Karl Popper)

RESUMO

Privacidade diferencial é a definição formal do estado da arte para publicação de dados sob fortes garantias de privacidade. Uma variedade de mecanismos foram propostos na literatura para publicar as saídas de consultas numéricas (e.g., mecanismo de Laplace e o mecanismo smooth sensitivity). Esses mecanismos garantem a privacidade diferencial adicionando ruído na saída verdadeira da consulta. A quantidade de ruído adicionada é calibrada usando as noções de sensibilidade global e sensibilidade local da consulta que medem o impacto da adição ou remoção de um indivíduo na saída da consulta. Mecanismos numéricos que usam sensibilidade local adicionam menos ruído e, consequentemente, tem uma resposta mais acurada. Contudo, mesmo que também haja trabalhos para consultas não-numéricas usando sensibilidade global (e.g., mecanismo exponencial), a literatura carece de mecanismos genéricos para publicação de saídas não-numéricas que usem sensibilidade local para reduzir o ruído. Nesse trabalho, remediamos essa deficiência apresentando o *mecanismo local dampening*. Nós adaptamos a noção de sensibilidade local da configuração numérica para a configuração não-numérica e a usamos para criar um mecanismo não-numérico genérico. Nós provemos uma comparação teórica com o mecanismo exponencial e mostramos sob quais condições o mecanismo local dampening é mais acurado que o mecanismo exponencial. Nós ilustramos a efetividade do mecanismo local dampening aplicando-o em três problemas diversos: (i) Seleção de mediana. Nós reportamos o elemento mediano de um banco de dados; (ii) Análise de nós influentes. Dado uma métrica de influência, nós publicamos os top-k nós mais influentes da rede; (iii) Indução de árvores de decisão. Nós provemos uma adaptação privada para o algoritmo ID3 para construir árvores de decisão a partir de um dado tabular. Nossa avaliação experimental mostra que nós reduzimos o erro para a aplicação de seleção de mediana em até 18%, reduzimos o uso de orçamento de privacidade em 2 a 4 ordens de magnitude para a aplicação de análise de nós influentes e aumentamos a acurácia em até 8% para árvores a aplicação em indução de árvores de decisão quando comparado a abordagens que usam sensibilidade global.

Palavras-chave: Privacidade Diferencial. Anonimização de dados. Análise em grafos. Árvores de decisão.

ABSTRACT

Differential privacy is the state-of-the-art formal definition for data release under strong privacy guarantees. A variety of mechanisms has been proposed in the literature for releasing the output of numeric queries (e.g., the Laplace mechanism and smooth sensitivity mechanism). Those mechanisms guarantee different privacy by adding noise to the true query's output. The amount of noise added is calibrated by the notions of global sensitivity and local sensitivity of the query that measure the impact of the addition or removal of an individual on the query's output. Mechanisms that use local sensitivity add less noise and, consequently, have a more accurate answer. However, although there has been some work on generic mechanisms for releasing the output of non-numeric queries using global sensitivity (e.g., the Exponential mechanism), the literature lacks generic mechanisms for releasing the output of non-numeric queries using local sensitivity to reduce the noise in the query's output. In this work, we remedy this shortcoming and present the *local dampening mechanism*. We adapt the notion of local sensitivity for the non-numeric setting and leverage it to design a generic non-numeric mechanism. We provide theoretical comparisons to the exponential mechanism and show under which conditions the local dampening mechanism is more accurate than the exponential mechanism. We illustrate the effectiveness of the local dampening mechanism by applying it to three diverse problems: (i) median selection. We report the median element in the database; (ii) Influential node analysis. Given an influence metric, we release the top-k most influential nodes while preserving the privacy of the relationship between nodes in the network; (iii) Decision tree induction. We provide a private adaptation to the ID3 algorithm to build decision trees from a given tabular dataset. Experimental evaluation shows that we can reduce the error for median selection application up to 18%, reduce the use of privacy budget by 2 to 4 orders of magnitude for influential node analysis application and increase accuracy up to 8% for decision tree induction when compared to global sensitivity based approaches.

Keywords: Differential Privacy. Data anonymization. Graph analysis. Decision Trees.

LIST OF FIGURES

LIST OF TABLES

LISTA DE ALGORITMOS

CONTENTS

1 INTRODUCTION

Many corporations, organizations, and government offices have been gathering data over the last decades. The entity that manages this data is called *curator*. The curator is a trusted tier that collects the data from individuals and then publishes valuable information for public use or specialized *analysts*. It may publish aggregated information, statistics, or some analysis using data mining algorithms. That is very useful to provide better services, more effective marketing, or a population statistics publication.

However, most datasets contain private or sensitive information from individuals. Recent regulations on data privacy, such as *General Data Protection Regulation (GPDR)* (EUROPEAN COMMISSION, 2018) and *Lei Geral de Proteção de Dados Pessoais (LGPD)* (BRASIL, 2018), pose the requirement of anonymity. Specifically, they require that an individual's information be rendered anonymous so that the individual is no longer identifiable from the published information. In the literature, this process is known as *data re-identification*.

Many examples show that naive anonymization, i.e., removing all explicit identifiers, does not prevent the re-identification of an individual from its data. One notable demonstration of re-identification was carried out by Narayanan and Shmatikov (2008). In this work, the authors proposed a de-anonymization technique and provided a practical analysis of the Netflix Prize dataset. With little auxiliary information, the privacy of the Netflix users was broken. Given 8 movie ratings of a user where 2 of them can be completely wrong, and dates can have a 14-day error, 99% of the records can be uniquely identified.

The fragility of naive anonymization was also shown in a case study involving crawled data from Flickr and Twitter in 2007/2008 (NARAYANAN; SHMATIKOV, 2009). One-third of Flickr and Twitter's verifiable members could be recognized in the anonymous Twitter network with a 12% error rate. To overcome that problem, many privacy models were designed with their own privacy requirements. The privacy model is placed between individuals and public users.

One of the most well-known privacy models is *k-anonymity* (SAMARATI; SWEENEY, 1998; SWEENEY, 2002). It requires that a published information from individual should not be distinguishable from at least $k-1$ individuals in the release. Based on k-anonymity, other privacy models have been proposed as *l-diversity* (MACHANAVAJJHALA *et al.*, 2007), *t-closeness* (LI *et al.*, 2007; LI *et al.*, 2009) and *δ-presence* (NERGIZ *et al.*, 2007).

Those models have a common problem; they assume that the attacker has limited background knowledge. The attacker can acquire background information on many parts of the process: the algorithm used for anonymization, domain knowledge, or public records.

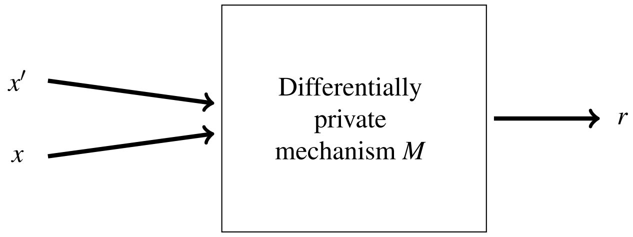
An example is a class of attack based on the principle of minimality which many algorithms satisfying k-anonymity had their privacy broken (CORMODE *et al.*, 2010; JIN *et al.*, 2010; XIAO *et al.*, 2010). Then, other attacks have been developed for those traditional privacy models as the *composition attack* (GANTA *et al.*, 2008) and foreground knowledge attack (WONG *et al.*, 2011).

1.1 Differential Privacy

Differential privacy (DWORK, 2011; DWORK *et al.*, 2006b) is the state-of-the-art formal definition for data release under strong privacy guarantees. It imposes near-indistinguishability on the released information whether an individual belongs to a sensitive database or not. It assumes that the attacker knows about $n-1$ records of the sensitive dataset except for the record that he/she is trying to learn about.

The fundamental intuition is that an analyst's query is answered by a *randomized algorithm* that queries the private database and returns a randomized answer sampled from *output distribution*. A randomized algorithm is a differentially private mechanism (also referred to as a mechanism in this work) if the probability distribution of the outputs does not change significantly based on the presence or absence of an individual. It ensures statistical guarantees against the inference of private information through the use of auxiliary information.

Figure 1 – Differentially private mechanism. A differentially private mechanism M should produce any output r with almost the same probability whether any single user is in the database (x) or not (x').



$$Pr[M(x) = r] \approx Pr[M(x') = r]$$

Source: elaborated by the author.

All mechanisms strive to shape the output distribution such that the true answer and answers with high utility have a high probability of being sampled. Mechanisms that achieve that provide useful information to the analyst. The formal notion of utility is discussed later in this section.

Algorithms can achieve differential privacy by employing output perturbation, which releases the true output of a given non-private query f with noise injected. The magnitude of the noise should be large enough to cover the identity of the individuals in the input database x.

In this work, we focus on non-numeric queries, i.e., queries where the range is non-numeric. For instance, a query that returns the most frequent name from a database of people's names is non-numeric. In contrast, numeric queries return numeric answers, e.g., a query that returns the mean salary of the company's employees.

Given a non-numeric query $f : \mathscr{D}^n \to \mathscr{R}$, where $\mathscr{R}$ is its non-numeric range, the *exponential mechanism* (MCSHERRY; TALWAR, 2007) achieves differential privacy by sampling elements from $\mathscr{R}$ based on the exponential distribution. This requires an *utility function* $u(x,r)$ that takes as input a database x and an element $r \in \mathscr{R}$ and outputs a numeric *utility score* that measures the utility of r. The larger $u(x,r)$, the higher the probability of the exponential mechanism outputting r. For instance, in the *most commom name* query, a reasonable utility function $u(x,r)$ returns the frequency of the name r in the database x as the utility score for r.

The noise added by the exponential mechanism is not numeric, so it is sampling noise. The exponential mechanism can sample a name that is not the most frequent, i.e., a name that does not have the best utility score. The amount of noise injected is proportional to the concept of *global sensitivity*. The global sensitivity measures the worst-case impact on the utility function $u(x,r)$ of the addition or removal of an individual from x, for all databases x and all $r \in \mathscr{R}$. Note that the global sensitivity does not depend on the input database. Given that, our goal is to produce private algorithms with low sensitivity to inject less noise and, consequently, have better accuracy.

Example 1.1.1 introduces the running example of this thesis. Example 1.1.2 describes the global sensitivity concept we use in our running example, which is based on a graph analysis centrality metrics, called *Egocentric Betweenness Centrality* (EBC).

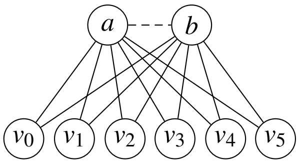
Example 1.1.1. *(Running Example) Here we introduce the running example of this thesis. Consider an application where, given a graph $G = (V,E)$, the analyst's non-numeric query should report the node with the largest EBC (FREEMAN, 1978; MARSDEN, 2002; EVERETT; BORGATTI, 2005). The EBC metric measures the degree to which nodes stand between each other, defined as*

$$EBC(c) = \sum_{u,v \in N_c | u \neq v} \frac{p_{uv}(c)}{q_{uv}(c)},$$

where $N_c = \{v \in V | \{c,v\} \in E\}$ is the set of neighbors of a given node c, $q_{uv}(c)$ is the number of shortest paths connecting u and v on the induced subgraph $G[N_c \cup \{c\}]$ and $p_{uv}(c)$ is the number of those paths that include c.

For instance, let G be the graph illustrated in Figure 2. Node a has EBC score equal to 7.5 since there are $\binom{6}{2} = 15$ pairs of neighbors of the form $\{v_i, v_j\}$, for $0 \leq i < j \leq 5$, that each contributes with 0.5 as they have two geodesic paths of length 2 from v_i to v_j, where only one contains a. Pairs of the form $\{b, v_i\}$, for $0 \leq i \leq 5$ do not contribute to the score of a since there is only one geodesic path (length 1) from b to v_i and it does not contain a.

Figure 2 – Egocentric betweenness sensitivity



Source: elaborated by the author.

Example 1.1.2. *(Global Sensitivity) To compute global sensitivity, one should calculate the difference of utility scores $|u(x,r) - u(x',r)|$ for all pairs of databases x and x' that differ in one individual and for elements r in the range of the numeric query. In our running example, the utility function is EBC and x and x' are graph database is a graph. We use edge differential privacy where the information to be protect is the edges. Thus when computing $|u(x,r) - u(x',r)|$ for x and x' that differ in one edge.*

The global sensitivity is usually given in closed form formula. We show the complete development and proofs for that on Chapter 6. However, to illustrate this example we show the pair x and x' that maximizes $|u(x,r) - u(x',r)|$. Let x be the graph in Figure 2 and x' be the graph obtained from x by removing the edge (a,b). The new EBC score of a is 15, 1 point for each one of the 15 pairs $\{v_i, v_j\}$, $0 \leq i \leq 5$, as now there is only one geodesic path from v_i to v_j which includes a (path $< v_i, a, v_j >$). The paths of the form $< v_i, b, v_j >$ are not counted since b no longer belongs to N_a. Thus, the global sensitivity is equal to 7.5.

In this application (Example 1.1.2), the global sensitivity has a significant concern: the gadget found this example, formed by two nodes with a high degree that share all neighbors, and those neighbors do not have an edge to each other, is unlikely to be found in real-world graphs. Therefore, real-world graphs are far from the worst-case scenario, and mechanisms calibrated by the global sensitivity may be unreasonably large, which implies adding overwhelming amounts of noise.

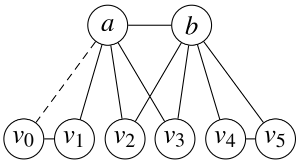
For numeric queries, Nissim *et al.* (2007) proposed the smooth sensitivity framework that adds instance-based noise calibrated as a function of x. They introduced the notion of *local sensitivity*, which quantifies the impact of addition or removal of an individual for the input database instance x, resulting in a lower bound to the global sensitivity. Note that the main difference between the global sensitivity and local sensitivity is that the first is not dependent on the input database x and the latter is. Many works use this notion to reduce the amount of noise added to release more useful statistical information (BLOCKI *et al.*, 2013; KARWA *et al.*, 2011; KASIVISWANATHAN *et al.*, 2013; LU; MIKLAU, 2014; ZHANG *et al.*, 2015).

Example 1.1.3 shows how local sensitivity would be measured in the non-numeric graph application and that it is significantly smaller for more usual graphs.

Example 1.1.3. *(Local Sensitivity Example) In this instance (Figure 3), verify that the node a has EBC score equal to* 6.5*: 1 for each pair* $\{v_0, v_2\}$, $\{v_0, v_3\}$, $\{v_0, b\}$, $\{v_1, v_2\}$, $\{v_1, v_3\}$ *and* $\{v_1, b\}$ *and 0.5 for* $\{v_2, v_3\}$.

The worst measurement of the sensitivity (difference of EBC when adding/removing an edge) the utility function for a node is given by removing the edge (a, v_0)*. That reduces the EBC score of a by only* 3 *(*1 *for each pair* $\{v_0, b\}$, $\{v_0, v_2\}$ *and* $\{v_0, v_3\}$ *since* v_0 *is no longer a neighbor of a). This means that local sensitivity for this instance is 3, which is smaller than its global sensitivity.*

Figure 3 – Egocentric betweenness sensitivity - Local Sensitivity



Source: elaborated by the author.

Furthermore, we identified that we could explore a more specific notion of local sensitivity which we call *element local sensitivity*. Traditional local sensitivity measures the largest impact of the addition or deletion of an individual to the input database over all outputs $r \in \mathscr{R}$. Element local sensitivity computes this impact, but only for some given element $r \in \mathscr{R}$.

That allows us to explore local measurements of the sensitivity of f even if traditional local sensitivity is near the global sensitivity, but, for most elements in $\mathscr{R}$, the element local sensitivity is low. Example 1.1.4 shows a case where element local sensitivity is significantly smaller than global and traditional local sensitivity for some elements $r \in \mathscr{R}$.

Example 1.1.4. *(Element Local Sensitivity) Consider the graph in Figure 2. The removal of edge (a,b) sets the traditional local sensitivity to 7.5 which is also the case for global sensitivity. But measurements of sensitivity per node (element) are much smaller. For instance, the sensitivity for a node v_i ($0 \leq i \leq 5$) is 1 which is set by the removal of edge (a,b) where $EBC(v_i)$ increases from 0 to 1 (path $< a, v_i, b >$).*

A non-numeric mechanism applying local sensitivity could add less noise to the output than a global sensitivity-based approach. To the best of our knowledge, the literature lacks generic mechanisms that apply local sensitivity to non-numeric settings, which arises as a great research opportunity.

This thesis introduces the local dampening mechanism, a novel framework to provide differential privacy for non-numeric queries using local sensitivity. Also, we extend the local dampening mechanism to provide better accuracy for element local sensitivity. We develop a theoretical accuracy analysis and a guide to construct accurate local dampening instances.

1.2 Problem Statement

In this thesis, we address the problem of releasing the output of a non-numeric function using differential privacy. Let x be a sensitive database and f a non-numeric function to be evaluated on x. The database is represented as vector $x \in \mathscr{D}^n$ where each entry represents an individual tuple, and $\mathscr{D}$ is the set of all possible tuple values. The function $f : \mathscr{D}^n \rightarrow \mathscr{R}$ receives the dataset $x \in \mathscr{D}^n$ to be evaluated and outputs an element r in its non-numeric range $\mathscr{R}$.

The task is to release the output $f(x)$ without leaking much information about the individuals using differential privacy. For that, we need to design a randomized algorithm $(A)(x)$ that adds noise to $f(x)$ such that it satisfies the formal definition of differential privacy (Definition 2.2.4).

The exponential mechanism is an example of a mechanism that uses this setup. A mechanism that addresses this problem is a natural building block to compose other complex private algorithms and can potentially be used in any work in the literature that uses the exponential mechanism as in (ZHANG *et al.*, 2017; FRIEDMAN; SCHUSTER, 2010, 2010; MCSHERRY, 2009; MOHAMMED *et al.*, 2011; HARDT *et al.*, 2012).

1.3 Applications

We illustrate the effectiveness of our approach by applying it to three very different problems: median selection, influential node analysis, and decision tree induction.

Median selection. Median selection is a commonly addressed problem to show the accuracy of differentially private mechanisms. The task is to report the label of the median element in a given database.

Influential node analysis. Identifying influential nodes in a network is an important task for social network marketing (MA *et al.*, 2008). The goal is to search for central nodes in a graph database. Given a centrality/influence metric, we release the label of the top-k most central nodes while preserving the privacy of the relationships between nodes in the graph. In this work, we use EBC as an influence metric. EBC metric identifies influential nodes that are important in different loosely connected parties. This is the graph application shown in Examples 1.1.2, 1.1.3 and 1.1.4.

Decision tree induction. We tackle a data mining problem which is constructing decision trees for classification. We provide a private adaptation to the ID3 algorithm to build a decision tree from a given tabular dataset. For the automatic tree induction, we use *Information Gain* (IG) as the split criterion for choosing an attribute to branch.

1.4 Thesis Contribution

Most of the contributions of this thesis were previously published in our paper (FARIAS *et al.*, 2020):

- FARIAS, V. A. E. de; BRITO, F. T.; FLYNN, C.; MACHADO, J. C.; MAJUMDAR, S.;SRIVASTAVA, D. Local dampening: Differential privacy for non-numeric queries via local sensitivity. Proc. VLDB Endow., v. 14, n. 4, p. 521–533, 2020. Available in: http://www.vldb.org/pvldb/vol14/p521-farias.pdf.

The main contributions of this thesis are the following:

- We adapt the concept of local sensitivity originally defined for the numeric setting to the non-numeric setting.

- We introduce the notion of element local sensitivity to the non-numeric setting, which is a specialized definition of local sensitivity where the sensitivity is measured for a single element in the range of the function to be evaluated.
- We propose the *local dampening mechanism*, an output perturbing non-numeric differentially private mechanism that applies the notion of local sensitivity for the non-numeric setting to attenuate the utility function in order to reduce the amount of noise injected compared to traditional global sensitivity based approaches. Local dampening also employs the exponential distribution as the exponential mechanism.
- We present the second version of our approach named the shifted local dampening mechanism, which can effectively use the element local sensitivity to improve accuracy.
- We develop a theoretical and empirical accuracy analysis where we enumerate some conditions in which the local dampening mechanism benefits from the local sensitivity notions. Under those conditions, we show that the exponential mechanism is an instance of the local dampening mechanism, and it is the worst instance of the local dampening mechanism in terms of accuracy. Also, we discuss the scenario where those conditions are not met and how we can still have good accuracy.
- We tackle the median selection problem where a private mechanism should report the label of the median element. Empirical results show that the local dampening mechanism can improve up to 29% about global sensitivity approaches.
- We apply the local dampening mechanism to construct differentially private algorithms for a graph problem called Influential Node Analysis using Egocentric Betweenness Centrality as the influence metric, and we show how to compute local sensitivity for this application. Experimental results show that our approach could be as accurate as global sensitivity-based mechanisms using 2 to 4 orders of magnitude less privacy budget than global sensitivity-based approaches.
- We address the application of building private algorithms for decision tree induction as an example data-mining application for tabular data. We present a differentially private adaptation of the entropy-based ID3 algorithm using the local dampening mechanism, and we provide a way to compute the local sensitivity efficiently. We can improve accuracy up to 5% compared to previous works.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we introduce the main concepts of differential privacy. Particularly, we present the definition of differential privacy, non-numeric differential privacy, the notions of global and local sensitivity, and the composition theorems.

Chapter 3 describes the essential elements that compose this thesis: the element local sensitivity, admissible functions, and the dampening function of the utility function. Given that, we propose the local dampening mechanism and prove it to be differentially private. Also, we present the related work, the well-known exponential mechanism, and permute-and-flip.

In Chapter 4, we introduce the shifted local dampening mechanism. We analyze some problems of the original local dampening mechanism and how the shifted local dampening aims to solve them. Also, we provide an accuracy analysis where we show theoretical analysis and a guide to build a good shifted local dampening instance.

The median selection problem is tackled in Chapter 5. In Chapter 6, we apply the local dampening mechanism to the influential node analysis problem using EBC and present the related work to this application. Chapter 7 address the decision tree induction problem using IG as the split criterion and also present the related work for this problem.

Finally, Chapter 8 we draw conclusions and propose future works based on this thesis.

2 DIFFERENTIAL PRIVACY

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

2.1 Privacy Model

Differential privacy is the state-of-the-art framework for private data analysis and publishing. It provides rigorous privacy guarantees for releasing information from sensitive databases.

The differential privacy model assumes that the attacker may have knowledge about arbitrary background information except the record that he/she wants to learn from. It is assumed that the attacker may have knowledge of $n-1$ tuples of the sensitive database as background information. Then he/she wants to learn about the n-th tuple. Without privacy protection, the attacker could just submit a query to get aggregate information of the n tuples and compare to the background information of the $n-1$ tuples. This simple comparison could leak the existence of the n-th tuple in the database and any kind of sensitive information.

This way, the intuition is that the released information can not reveal the existence of a tuple in a database. It imposes near-indistinguishability on the released information whether an individual belongs to a sensitive database or not. A differently private algorithm is a randomized algorithm for which the output distribution does not change significantly based on the presence or absence of an individual. This way, an attacker cannot draw conclusion about the n-th tuple from the background information.

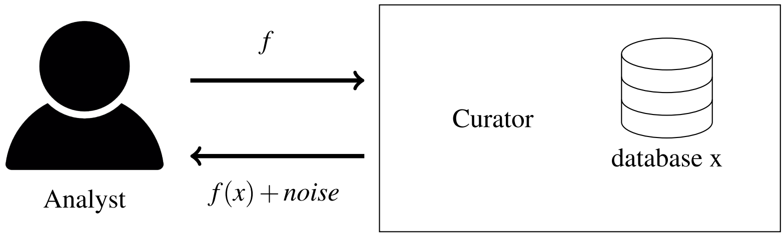
In this model, the analyst does not submit his/her queries directly to the database. There is the role of the curator. The curator is in charge of collecting the data from individuals to constitute the database. Also, the curator is responsible for answering the queries of the analyst by consulting the sensitive database and releasing a differentially privately answer.

Curators can achieve differential privacy by employing output perturbation, which releases the true output of a given non-private query f with noise injected (Figure 4).

2.2 Basic Definitions

A *privacy mechanism* is a randomized algorithm that takes the database as input and output a differentially private answer. A randomized algorithm with domain A and range $\mathscr{R}$ is

Figure 4 – Differential Privacy Model - Output Perturbation



Source: elaborated by the author.

Figure 5 – Two neighboring databases, i.e., $d(x,y) = 1$

$$x = \begin{array}{|c|} \hline t_1 \\ \hline t_2 \\ \hline \vdots \\ \hline t_n \\ \hline \end{array} \qquad y = \begin{array}{|c|} \hline t_1 \\ \hline t'_2 \\ \hline \vdots \\ \hline t_n \\ \hline \end{array}$$

Source: elaborated by the author.

associated with a *probability simplex* over $\mathscr{R}$, denoted by $\Delta(\mathscr{R})$:

Definition 2.2.1. *(Probability Simplex (DWORK et al., 2014)) Given a discrete set $\mathscr{R}$, the probability simplex over $\mathscr{R}$, denoted by $\Delta(\mathscr{R})$ is defined to be:*

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

Definition 2.2.2. *(Randomized Algorithm (DWORK et al., 2014)) A randomized algorithm M with domain A and discrete range $\mathscr{R}$ is associated with a mapping $\mathscr{M} : A \to \Delta(\mathscr{R})$. On input $a \in A$, the algorithm M outputs $M(a) = r$ with probability $(\mathscr{M}(a))_r$ for each $r \in \mathscr{R}$.*

A database x is represented as a vector, $x \in \mathscr{D}^n$, where $\mathscr{D}$ is the tuple domain. The notion of distance between two databases measures how many tuples two given databases differ:

Definition 2.2.3. *(Distance Between two Databases) The distance between $d(x,y)$ two databases x and y is the hamming distance $H(x,y)$:*

$$H(x,y) = |\{i \mid x_i \neq y_i, \ i = 1...n\}|$$

Figure 5 illustrates two databases which are at distance 1. Two databases are said to be *neighbors* if the distance between them is 1.

As we mentioned before, the output $f(x)$ must be released without leaking much information about the individuals. A private mechanism needs not to change its output probability by a multiplicative factor $\exp(\varepsilon)$ under the presence or absence of single tuple. For that, we need to design a randomized algorithm $M(x)$ that adds noise to $f(x)$ such that it satisfies the definition of differential privacy stated below.

Definition 2.2.4. *(ε-Differential Privacy (DWORK et al., 2006a; DWORK et al., 2006b)). A randomized algorithm $M : \mathscr{D}^n \to \mathscr{R}$ satisfies ε-differential privacy, if for any two databases x and y satisfying $d(x,y) \leq 1$ and for any possible output $r \in \mathscr{R}$, we have*

$$Pr[M(x) = r] \leq \exp(\varepsilon)\, Pr[M(y) = r]$$

where $Pr[\cdot]$ denotes the probability of an event.

The parameter ε dictates how close the distribution of the outputs differs between the databases x and y. Small values of ε means that those two distributions must be really close which hurts accuracy but provides a better indistinguishability, i.e., a better privacy level. For large ε, the opposite happens, the two distributions can differ more which means a better accuracy and a lower level of privacy. The value of ε is given by the data holder to the analyst. Thus, this parameter controls how much privacy the data holder wants to provide.

The analyst may want to submit multiple queries to the database. The parameter ε is also called as the *privacy budget* since each query submitted to the database consumes a part of the budget. The analyst decides how much budget he/she spends in each query. In section 2.5, we describe how the budget can be used by the analyst to execute his/her queries.

2.3 The Non-numeric Setting

We consider two settings when building private mechanism: the numeric setting and the non-numeric setting. The numeric setting is when one needs to construct a private mechanism for a function $f : \mathscr{D}^n \to \mathscr{R}$ where f outputs a vector of numeric values, i.e., $\mathscr{R} = \mathbb{R}^d$. In this case, the Laplace Mechanism applies (DWORK *et al.*, 2006b).

In this work, we address the non-numeric setting. We aim to build a private mechanism for a function $f : \mathscr{D}^n \to \mathscr{R}$ where f outputs non-numeric values, i.e., we refer $\mathscr{R}$ as a discrete set of outputs $\mathscr{R} = \{r_1, r_2, r_3, \dots\}$.

In the non-numeric setting, the analyst provides an utility function $u : \mathscr{D}^n \times \mathscr{R} \to \mathbb{R}$ that takes a database x and an output $r \in \mathscr{R}$ and produces an *utility score* $u(x,r)$. The utility function is application-specific and each application requires its own utility function.

The utility score represents how good an output r is for the dataset x. This means that, for a given input database x, the analyst prefers that the mechanism outputs the elements with high utility score. Thus a mechanism answering to f needs to output a high utility output with higher probability.

Recalling the "most common name" query presented in Chapter 1. The task is to return the most frequent name from a database of people's names is a non-numeric query. A suitable utility function produces a utility score $u(x, name)$ as the frequency of the output *name* in the database x.

2.4 Sensitivity on the non-numeric setting

Differentially private mechanisms usually perturbs the true output with noise. The amount of noise added to the true output of a non-numeric function $f : \mathscr{D}^n \to \mathscr{R}$ is proportional to the *sensitivity* of the utility function $u : \mathscr{D}^n \times \mathscr{R} \to \mathbb{R}$. The various notions of sensitivity used in this work are presented in this section.

2.4.1 Global Sensitivity

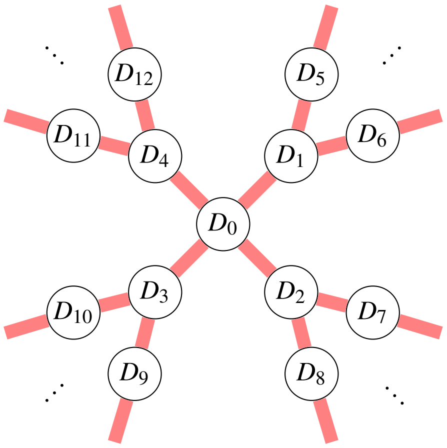
The global sensitivity of u is defined as the maximum possible difference of utility scores at all possible pairs of database entries x, y and all possible elements $r \in \mathscr{R}$:

Definition 2.4.1. *(Global Sensitivity Δu (MCSHERRY; TALWAR, 2007)). Given a utility function $u : \mathscr{D}^n \times \mathscr{R} \to \mathbb{R}$ that takes as input a database $x \in \mathscr{D}^n$ and an element $r \in \mathscr{R}$ and outputs a numeric score for r in x. The global sensitivity of u is defined as:*

$$\Delta u = \max_{r \in \mathscr{R}} \max_{x,y | d(x,y) \leq 1} |u(x,r) - u(y,r)|.$$

Intuitively, the global sensitivity measures the maximum change on the utility score over all $r \in \mathscr{R}$ between any two neighboring databases in the universe of all databases. Figure 6 illustrates the global sensitivity. The edges represent the term $\max_{r \in \mathscr{R}} |u(x, D_i) - u(y, D_j)|$ for two neighbors D_i and D_j. The global sensitivity is the maximum value among all red edges. Refer to Example 1.1.2 for an illustration in our running example.

Figure 6 – Global Sensitivity. The edges represent the term $\max_{r\in\mathscr{R}} |u(x,D_i) - u(y,D_j)|$ for two neighbors D_i and D_j. The global sensitivity is the maximum value among all red edges.



Source: elaborated by the author.

2.4.2 *Local Sensitivity*

The local sensitivity was originally proposed to the numeric setting in (NISSIM *et al.*, 2007). In this work, we provide an adaptation of the local sensitivity to the non-numeric setting.

The concept of local sensitivity captures the sensitivity locally on the input database x instead of searching for the sensitivity in the universe of databases $\mathscr{D}^n$. Figure 7 depicts the local sensitivity. The edges represent the term $\max_{r\in\mathscr{R}} |u(x,D_i) - u(y,D_j)|$ for two neighbors D_i and D_j. The local sensitivity $LS^u(D_0)$ is the maximum value among all red edges incident on D_0. Refer to Example 1.1.3 for an illustration in our running example.

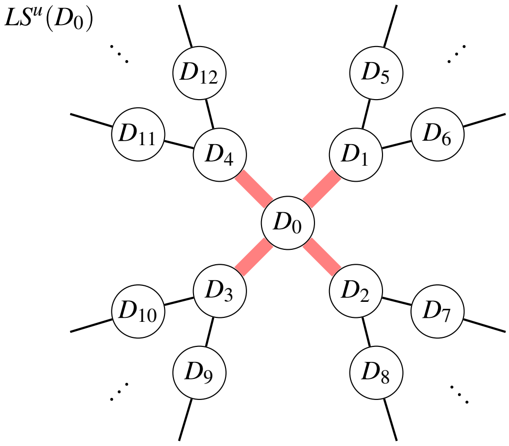
The local sensitivity for the non-numeric setting is given as:

Definition 2.4.2. *(Local Sensitivity, adapted from (NISSIM et al., 2007)). Given a utility function $u : \mathscr{D}^n \times \mathscr{R} \to \mathbb{R}$ that takes as input a database $x \in \mathscr{D}^n$ and an element $r \in \mathscr{R}$ and outputs a numeric score for r in x, the local sensitivity of u is defined as*

$$LS^u(x) = \max_{r\in\mathscr{R}} \max_{y|d(x,y)\leq 1} |u(x,r) - u(y,r)|$$

Observe that the global sensitivity is given as the maximum local sensitivity over the set of all databases, $\Delta u = \max_x LS^u(x)$.

Figure 7 – Local Sensitivity. The edges represent the term $\max_{r\in\mathscr{R}}|u(x,D_i)-u(y,D_j)|$ for two neighbors D_i and D_j. The local sensitivity $LS^u(D_0)$ is the maximum value among all red edges incident on D_0.



Source: elaborated by the author.

However, using solely the local sensitivity to build a mechanism is not enough to satisfy differential privacy. Thus, as the smooth sensitivity framework (NISSIM *et al.*, 2007), a part of our solution is based on local sensitivity at distance t. We adapt the notion of local sensitivity at distance t (NISSIM *et al.*, 2007) to the non-numeric setting for the use on this work:

Definition 2.4.3. *(Local Sensitivity at distance t, adapted from (NISSIM et al., 2007)). Given a utility function $u : \mathscr{D}^n \times \mathscr{R} \to \mathbb{R}$ that takes as input a database $x \in \mathscr{D}^n$ and an element $r \in \mathscr{R}$ and outputs a numeric score for r in x. The local sensitivity at distance t of u is defined as*

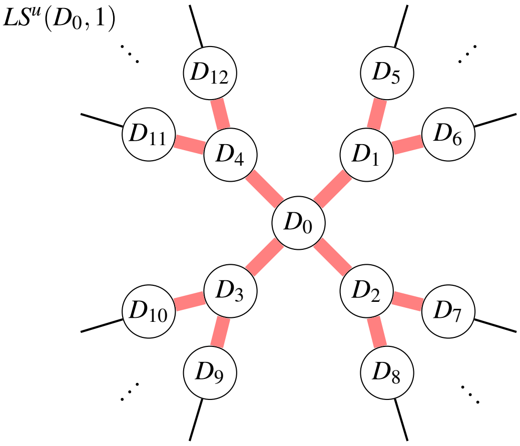
$$LS^u(x,t) = \max_{y|d(x,y)\leq t} LS^u(y)$$

Local sensitivity at distance t, $LS^u(x,t)$, measures the maximum local sensitivity $LS^u(y)$ over all databases y at maximum distance t, i.e., we allow t modifications on the database before computing its local sensitivity. Note that $LS^u(x,0) = LS^u(x)$ which is shown in Figure 7. In Figure 8, we illustrate the local sensitivity at distance 1. The edges represent the term $\max_{r\in\mathscr{R}}|u(x,D_i)-u(y,D_j)|$ for two neighbors D_i and D_j. The local sensitivity at distance 1 $LS^u(D_0,1)$ is the maximum value among all red edges incident on all the database at most distance 1 from D_0 (which includes itself).

A concrete example of the local sensitivity at distance 1 is given in Example 2.4.1.

Example 2.4.1. *(Local sensitivity at distance* 1*) Consider the graph G of Figure 9c. The local sensitivity at distance t allows t extra modifications before measuring local sensitivity. As discussed in Example 1.1.3, the local sensitivity of G is* 3 *(at distance* 0*): $LS^{EBC}(G,0) = 3$.*

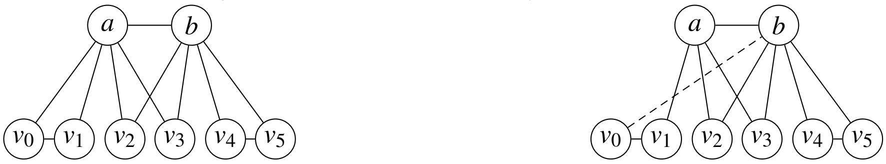
Figure 8 – Local sensitivity at distance 1. The edges represent the term $\max_{r \in \mathscr{R}} |u(x, D_i) - u(y, D_j)|$ for two neighbors D_i and D_j. The local sensitivity at distance 1 $LS^u(D_0, 1)$ is the maximum value among all red edges incident on all the database at most distance 1 from D_0 (which includes itself).



Source: elaborated by the author.

Now to compute local sensitivity at distance 1*, we need to find which edge to add or remove in order to compute the maximum local sensitivity at distance* 1*. This case is found by removing edge* (a, v_0) *as shown in Figure 9d obtaining* G'*. Then the local sensitivity of* G' *is* 5 *where node* b *increases by* 5 *units when adding edge* (b, v_0) *(*1 *for each pair* $\{v_0, v_2\}$*,* $\{v_0, v_3\}$*,* $\{v_0, v_4\}$*,* $\{v_0, v_5\}$ *and* $\{v_0, a\}$*). This means that* $LS^{EBC}(G, 1) = 5$

Figure 9 – Local Sensitivity at distance 1



(c) Original Graph G

(d) G' at distance 1 from G

Source: elaborated by the author.

Local sensitivity tends to be smaller than global sensitivity for a variety of problems (BLOCKI *et al.*, 2013; KARWA *et al.*, 2011; KASIVISWANATHAN *et al.*, 2013; LU; MIKLAU, 2014; NISSIM *et al.*, 2007; ZHANG *et al.*, 2015). In those problems, the real-world databases are very different from the worst case scenario of the global sensitivity and they have a low observed local sensitivity. Recall that the noise injected by mechanisms are proportional to the sensitivity. Thus, in those problems, the local sensitivity is a way to reduce noise.

2.5 Composition

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

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 2.5.1. *(Sequential composition (MCSHERRY; TALWAR, 2007; MCSHERRY, 2009)) Let $M_i : \mathscr{D}^n \rightarrow \mathscr{R}_i$ be an ε_i-differentially private algorithm for $i \in [k]$. Then $M(x) = (M_1(x), \cdots, M_k(x))$ is ($\sum_{i=1}^{k}$)-differentially private.*

The Theorem 2.5.1 implies that if an analyst is given a privacy budget ε, he/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 is carried with many ε_i-differentially private mechanisms operating on disjoint subsets, it composes a $\max_i \varepsilon_i$-differentially private mechanism which has a lower privacy cost.

Theorem 2.5.2. *(Parallel composition (MCSHERRY, 2009)) Let $M_i : \mathscr{D}^n \rightarrow \mathscr{R}_i$ be a ε_i-differentially private algorithm for $i \in [k]$ and $x_1, ..., x_k$ be disjoint subsets of $\mathscr{D}^n$. Then $M(x) = (M_1(x_1), \cdots, M_k(x_k))$ is ($\max_i \varepsilon_i$)-differentially private.*

2.6 Discussion

In this chapter, we provided some background concepts. We formally defined the differential privacy model for non-numeric queries which is the model that our approach satisfies, Chapter 3 contains the formal proof.

We introduced various notions of sensitivity. Specifically, we presented the notion of global sensitivity and provided the adaption of the two definitions of local sensitivity that were originally proposed to the numeric setting. Applying local sensitivity to the non-numeric setting to reduce noise is the novelty of this work.

Lastly, we discussed some theorems that allows us to compose complex algorithms from the execution of simpler queries and, still, have an algorithm that satisfies differential privacy. This is specially useful in our application chapters.

3 LOCAL DAMPENING MECHANISM

This chapter presents the local dampening mechanism for answering queries with non-numeric output under differential privacy. Our approach uses the non-numeric setting for differential privacy (Section 2.3).

Our approach requires the computation of any of the sensitivity notions described in the Section 2.4. Additionally, we introduce a new notion of sensitivity called element local sensitivity. It measures the worst impact on the sensitivity for a given element $r \in \mathscr{R}$ when adding or removing an individual from the input database x, i.e., the largest difference $|u(x,r) - u(y,r)|$ for all neighbors y of x.

The local dampening mechanism applies a *sensitivity function* to dampen the utility function u and construct its dampened version, referred to D_{u,δ^u}. Specifically, we attenuate u such that the signal-to-sensitivity ratio (i.e. u/sensitivity) is larger which results in higher accuracy. A sensitivity function is a function that computes one of the notions of sensitivity or an upper bound on the sensitivity. This concept is specially useful when computing the sensitivity is not possible or efficient but computing an upper bound is simpler, as it can be NP-hard (NISSIM *et al.*, 2007; ZHANG *et al.*, 2015).

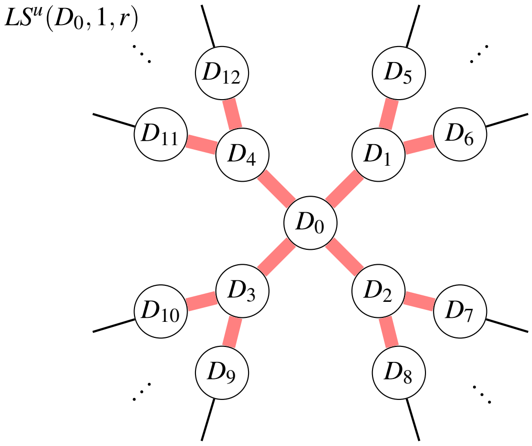
We lay the groundwork of our analysis with the definition of element local sensitivity in Section 3.1. We then define local dampening in Section 3.3, and provide a privacy guarantee for our mechanism in Section 3.5.

3.1 Element Local Sensitivity

The local sensitivity $LS^u(x,t)$ quantifies the maximum sensitivity of u over all elements $r \in \mathscr{R}$ for an input database x with t modifications (Definition 2.4.3). That gives a high-level description of the variation of u in neighboring databases. However, if just one element in $\mathscr{R}$ has a high value of sensitivity (close to Δu), $LS^u(x,t)$ will be equally large. That is ineffective in a scenario where most of the elements have low sensitivity and just few have high sensitivity, which makes $LS^u(x,t)$ large and consequently hurts accuracy.

We introduce a more specialized definition of local sensitivity named element local sensitivity, denoted as $LS^u(x,t,r)$, which measures the sensitivity of u for a given $r \in \mathscr{R}$ for an input database x at distance t (definition 3.1.1). This allows us to grasp the sensitivity of u for a single element.

Figure 10 – Element local sensitivity at distance 1, $LS^u(D_0,1,r)$. The edges represent the term $|u(x,D_i)-u(y,D_j)|$ for two neighbors D_i and D_j. $LS^u(D_0,1,r)$ is the maximum value among all red edges.



Source: elaborated by the author.

$$LS^u(x,t,r) = \max_{y|d(x,y)\leq t, d(y,z)\leq 1} |u(x,r)-u(y,r)|$$

Definition 3.1.1. *(Element Local Sensitivity at distance t). Given a utility function $u(x,r)$ that takes as input a database x and an element r and outputs a numeric score for x, the element local sensitivity at distance t of u is defined as*

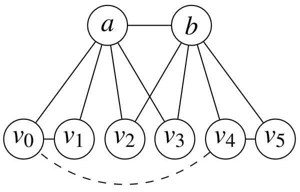
$$LS^u(x,t,r) = \max_{y\in\mathscr{D}^n|d(x,y)\leq t, z\in\mathscr{D}^n|d(y,z)\leq 1} |u(y,r)-u(z,r)|,$$

where $d(x,y)$ denotes the distance between two databases.

Intuitively, to compute element local sensitivity, one needs to identify which addition or removal of an individual on the input database x causes the most impact on the utility score of a given element r, i.e., the largest difference $|u(x,r)-u(y,r)|$ for all neighbors y of x. Note that we can obtain $LS^u(x,t)$ from this definition: $LS^u(x,t) = \max_{r\in\mathscr{R}} LS^u(y,t,r)$ as $LS^u(x,t,r) = \max_{y|d(x,y)\leq t} LS^u(y,0,r)$.

Example 3.1.1. *(Element local sensitivity) We illustrate this definition with the same setup from previous examples. Let G be the graph from Figure 9c. Suppose we want to compute the element local sensitivity for v_4, $LS^u(G,0,v_4)$. We measure only the worst impact of the addition or removal of an edge on the value of the EBC score for v_4. This is obtained by adding the edge (v_0,v_4) (Figure 11). The EBC score increases by 2 (1 for each pair $\{b,v_0\}$ and $\{v_0,v_5\}$). Thus $LS^u(G,0,v_4)=2$ which is smaller than local sensitivity $LS^u(G,0)=3$ (Example 1.1.3) and $\Delta u = 7.5$ (Example 1.1.2).*

Figure 11 – Element Local Sensitivity for v_4



Source: elaborated by the author.

3.2 Sensitivity Functions

Computing local sensitivity $LS^u(x,t)$ or element local sensitivity $LS^u(x,t,r)$ is not always feasible, as it can be NP-hard (NISSIM *et al.*, 2007; ZHANG *et al.*, 2015). To navigate this problem, we can relax the need for the computation of $LS^u(x,t)$ or $LS^u(x,t,r)$ and build a computationally efficient function $\delta^u(x,t,r)$ that computes an upper bound for $LS^u(x,t)$ or $LS^u(x,t,r)$ that is still smaller than Δu. We refer to δ^u as a sensitivity function that has the following signature $\delta^u : \mathscr{D}^n \times \mathbb{N}^0 \times \mathscr{R} \to \mathbb{R}$. Note that $\delta^u(x,t,r) = \Delta u$, $\delta^u(x,t,r) = LS^u(x,t)$ or $\delta^u(x,t,r) = LS^u(x,t,r)$ are sensitivity functions.

We define a classification for sensitivity functions according to four aspects: admissibility, boundedness, monotonicity and stability.

3.2.1 Admissibility

The sensitivity function δ^u needs to have some properties to be admissible in the local dampening mechanism to guarantee differential privacy:

Definition 3.2.1. *(Admissibility). A sensitivity function $\delta^u(x,t,r)$ is admissible if:*

1. $\delta^u(x,0,r) \geq LS^u(x,0,r)$*, for all* $x \in \mathscr{D}^n$ *and all* $r \in \mathscr{R}$
2. $\delta^u(x,t+1,r) \geq \delta^u(y,t,r)$*, for all* x,y *such that* $d(x,y) \leq 1$ *and all* $t \geq 0$

The global sensitivity Δu is admissable as any constant value would trivially satisfy Definition 3.2.1. We also show that the function $LS^u(x,t,r)$ itself is admissible (Lemma 3.2.1).

Lemma 3.2.1. *The element local sensitivity $LS^u(x,t,r)$ is admissible.*

Proof. We need to satisfy the two conditions of the admissibility of functions.

1. $LS^u(x,0,r) \geq LS^u(x,0,r)$
2. Since $\{y|d(x,y) \leq t\} \subset \{y|d(x',y) \leq t+1\}$ for any neighboring databases x,x', we have that

$$
\begin{aligned}
LS^u(x,t,r) &= \max_{y|d(x,y)\leq t, z|d(y,z)\leq 1} |u(y,r)-u(z,r)| \\
&\leq \max_{y|d(x',y)\leq t+1, z|d(y,z)\leq 1} |u(y,r)-u(z,r)| \\
&= LS^u(x',t+1,r)
\end{aligned}
$$

Thus $LS^u(x,t,r)$ is an admissible function.

□

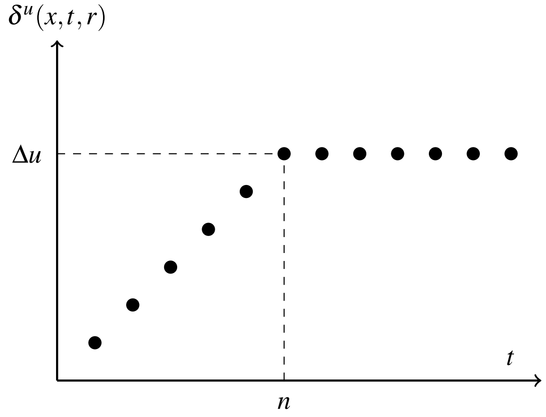
In Section 3.2.3, we discuss that $LS^u(x,t)$ and $LS^u(x,t,r)$ are also admissible functions.

3.2.2 Boundedness

Some sensitivity functions, such as $LS^u(x,t)$ and $LS^u(x,t,r)$, converge to Δu, by design, as t grows. This follows from the fact that the maximum distance of two databases is at most n by the hamming distance definition. Thus when $t = n$, $LS^u(x,t)$ and $LS^u(x,t,r)$ measure sensitivity in all possible databases. We refer to those functions as *bounded functions*.

Definition 3.2.2. *(Boundedness) A sensitivity function $\delta^u(x,t,r)$ is said to be bounded if $\delta^u(x,t,r) = \Delta u$ for all $t \geq n$.*

Figure 12 – Boundedness - $\delta^u(x,t,r)$ converges to Δu when $t \geq n$ where n is the size of the database.



Source: elaborated by the author.

Note that, one can easily force a given function $\delta^u(x,t,r)$ to be bounded by replacing it by its bounded version $\min(\delta^u(x,t,r),\Delta u)$. We now show that $\min(\delta^u(x,t,r),\Delta u)$ is admissible and bounded.

Lemma 3.2.2. *If $\delta^u(x,t,r)$ is admissable, then min$(\delta^u(x,t,r),\Delta u)$ is admissable and bounded.*

Proof. We show that $\min(\delta^u(x,t,r),\Delta u)$ is admissible. First we show that $\min(\delta^u(x,0,r),\Delta u) \geq LS^u(x,0,r)$. Thus, as $\delta^u(x,0,r) \geq LS^u(x,0,r)$ and $\Delta u \geq LS^u(x,0,r)$ we have that $\min(\delta^u(x,0,r),\Delta u) \geq LS^u(x,0,r)$.

Now, Suppose that $t > 0$, let y be a neighboring database of x. We have that $\delta^u(x,t+1,r),\Delta u) \geq \delta^u(y,t,r),\Delta u)$ as δ^u is admissible. This, $\min(\delta^u(x,t+1,r),\Delta u) \geq \min(\delta^u(y,t,r),\Delta u)$ holds. Thus $\min(\delta^u(x,t,r),\Delta u)$ is admissible.

We move to show that $\min(\delta^u(x,t+1,r),\Delta u)$ is bounded. Suppose that $t \geq n$ The maximum hamming distance between two datasets is at most n. Thus $\{y|d(x,y) \leq n\} = D^n$. So we have:

$$LS^u(x,t,r) = \max_{y|d(x,y)\leq t} LS^u(y,0,r) = \max_{y\in D^n} LS^u(y,0,r) = \Delta u$$

Therefore, we have that $\delta^u(x,t,r) \geq LS^u(x,t,r)$ since δ^u is admissible. Thus it implies that $\delta^u(x,t,r) = \Delta u$. Finally, $\min(\delta^u(x,t,r),\Delta u) = \Delta u$ for any $t > n$. □

Thus, we can replace $\delta^u(x,t,r)$ with $\min(\delta^u(x,t,r),\Delta u)$ since its admissable. In terms of accuracy, this replacement is beneficial. We have that $\delta^u(x,t,r) \geq \min(\delta^u(x,t,r),\Delta u)$ for all database x, $t \geq 0$ and $r \in \mathscr{R}$. Thus $\delta^u(x,t,r)$ is always larger than $\min(\delta^u(x,t,r),\Delta u)$ meaning that local dampening injects less noise as sensitivity is proportional to the noise. This means that we can impose boundedness for any function and, beyond that, we have gains in accuracy as it injects less noise.

3.2.3 Monotonicity

We introduce the notion of monotonicity in our context. When the utility score $u(x,r)$ is a monotonic function of $\delta^u(x,t,r)$ over $r \in \mathscr{R}$, we say that $\delta^u(x,t,r)$ is monotonic. We have three classifications for monotonicity.

Definition 3.2.3. *(Non-decreasing Monotonicity) Let $u(x,r)$ be an utility function and $\delta^u(x,t,r)$ be a sensitivity function. $\delta^u(x,t,r)$ is said to be monotonically non-decreasing if $\delta^u(x,t,r) \geq \delta^u(x,t,r')$ for all $x \in \mathscr{D}^n$, $r, r' \in \mathscr{R}$, $t \geq 0$ such that $u(x,r) \geq u(x,r')$.*

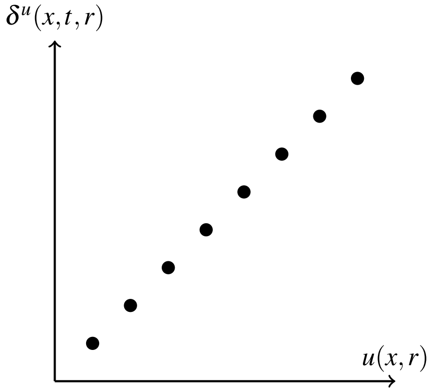
And its symmetric definition is:

Definition 3.2.4. *(Non-increasing Monotonicity) Let $u(x,r)$ be an utility function and $\delta^u(x,t,r)$ be a sensitivity function. $\delta^u(x,t,r)$ is said to be monotonically non-increasing if $\delta^u(x,t,r) \geq \delta^u(x,t,r')$ for all $x \in \mathscr{D}^n$, $r, r' \in \mathscr{R}$, $t \geq 0$ such that $u(x,r) \leq u(x,r')$.*

Also, a sensitivity can be *flat*:

Definition 3.2.5. *(Flat Monotonicity) Let $u(x,r)$ be an utility function and $\delta^u(x,t,r)$ be a sensitivity function. $\delta^u(x,t,r)$ is said to be flat if $\delta^u(x,t,r) = \delta^u(x,t,r')$ for all $x \in \mathscr{D}^n$, $r, r' \in \mathscr{R}$, $t \geq 0$.*

Figure 13 – Non-decreasing monotonicity - The larger $\delta^u(x,t,r)$, larger the $u(x,r)$.



Source: elaborated by the author.

We refer to a *monotonic function* as a function that is either flat, monotonically non-decreasing or monotonically non-increasing.

Note that flat sensitivity functions are independent on r and they are both monotonic non-increasing and monotonic non-decreasing. The global sensitivity Δu and the local sensitivity $LS^u(x,t)$ are flat sensitivity functions since they do not depend on r.

Additionally, given an utility function u and an sensitivity function $\delta^u(x,t,r)$, one can build a function $\hat{\delta}^u(x,t,r)$ from $\delta^u(x,t,r)$ such that $\hat{\delta}^u(x,t,r)$ is flat.

$$\hat{\delta}^u(x,t,r) = \max_{r' \in \mathscr{R}} \delta^u(x,t,r').$$

Basically, $\hat{\delta}^u(x,t,r)$ increases the value for a given $r \in \mathscr{R}$ and $t \geq 0$ to the maximum value for $\delta(x,t,r')$ among all $r' \in \mathscr{R}$. This results in the same value $\hat{\delta}^u(x,t,r)$ for any given r. A drawback of using $\hat{\delta}^u(x,t,r)$ is that $\hat{\delta}^u(x,t,r) \geq \delta^u(x,t,r)$, for all x, $t \geq 0$ and $r \in \mathscr{R}$ meaning that $\hat{\delta}^u(x,t,r)$ returns a large upper bound for sensitivity and, consequently, hurts accuracy.

An intermediate result shows that $\hat{\delta}^u(x,t,r)$ is admissible:

Lemma 3.2.3. *Let $\delta_1(x,t,r),\ldots,\delta_p(x,t,r)$ be admissible functions. Then $\delta(x,t,r)$ defined as $\delta(x,t,r) = \max(\delta_1(x,t,r),\ldots,\delta_p(x,t,r))$ is an admissible function.*

The proof of Lemma 3.2.3 is immediate given by the admissibility of $\delta_1(x,t,r),\ldots,\delta_p(x,t,r)$. Lemma 3.2.3 entails in some important results: (i) $\hat{\delta}^u(x,t,r)$ is admissible if δ^u is admissible and (ii) $LS^u(x,t)$ is an admissible function once $LS^u(x,t) = max_{r\in\mathscr{R}}LS^u(x,t,r)$ and $LS^u(x,t,r)$ is an admissible function (Theorem 3.2.1).

3.2.4 Stability

An important classification that will be used in our accuracy analysis is the stability.

Definition 3.2.6. *(Stability) A sensitivity function $\delta^u(x,t,r)$ is stable if δ^u is admissible, bounded and monotonic.*

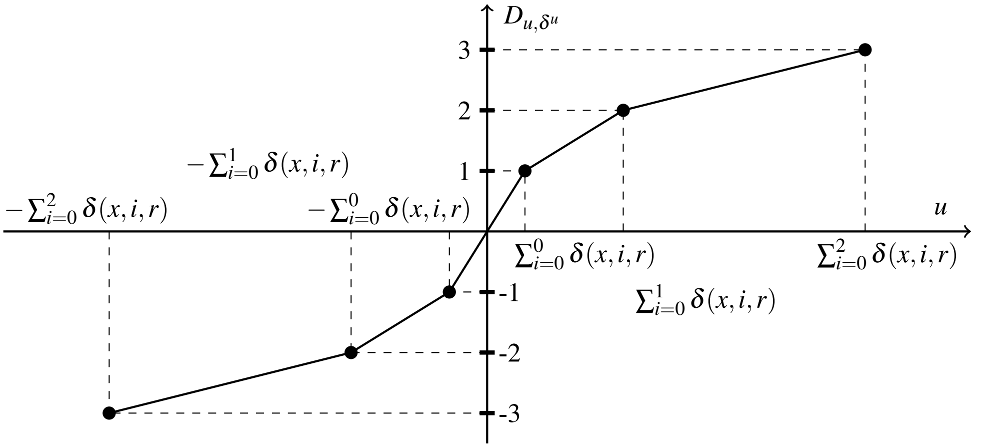
Meeting all three requirements (admissibility, boundedness and monotonicity) for designing a stable function may sound very restrictive. However, for all definitions of sensitivity, two of them are naturally stable: global sensitivity Δu and local sensitivity $LS^u(x,t)$. Only the element local sensitivity $LS^u(x,t,r)$ can be non-monotonic and, consequently, non-stable. Nevertheless, in Section 4.4.3, we argue that the requirement of strict monotonicity can be relaxed and an admissable bounded function with "weak" monotonicity can perform well in the local dampening mechanism.

Besides, for any function, the requirement of boundedness can be easily imposed as shown in Section 3.2.2 while still providing lower sensitivity.

3.3 Dampening Function

A crucial part of our mechanism is the *dampening function*. We now define the dampening function $D_{u,\delta^u}(x,r)$, which uses an admissible sensitivity function $\delta^u(x,t,r)$ to return a dampened and scaled version of the original utility function.

Figure 14 – Dampening function D_{u,δ^u}



Source: elaborated by the author.

Definition 3.3.1. *(Dampening function). Given a utility function $u(x,r)$ and an admissible function $\delta^u(x,t,r)$, the dampening function $D_{u,\delta^u}(x,r)$ is defined as a piecewise linear interpolation over the points:*

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

where $b(x,i,r)$ is given by:

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

Therefore,

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

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

Figure 14 shows the general scheme of D_{u,δ^u}. 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 3.3.1).

Lemma 3.3.1. $|D_{u,\delta^u}(x,r) - D_{u,\delta^u}(y,r)| \leq 1$ *for all x,y such that $d(x,y) \leq 1$ and all $r \in R$ if δ^u is admissible.*

Proof. Fix a database $x \in D^n$ and let $y \in D^n$ be a neighbor of x such that $d(x,y) \leq 1$. Assume $u(x,r)$ lies in $[b(x,i,r), b(x,i+1,r))$ for some $i \in \mathbb{Z}$. We first show that $D_{u,\delta^u}(x,r) - D_{u,\delta^u}(y,r) \leq 1$. We analyse it in two cases: (1) $u(x,r) \geq 0$ and (2) $u(x,r) < 0$.

Case (1). Assume $u(x,r) \geq 0$. By construction of the dampening function $D_{u,\delta}$, $i \geq 0$ holds. Thus, one can find bounds for $u(y,r)$ using the definition of LS^u and the admissibility of δ^u.

$$\begin{aligned}
u(y,r) &\geq u(x,r) - LS^u(x,0,r) \\
&\geq b(x,i,r) - \delta^u(x,0,r) \\
&= \sum_{j=1}^{i-1} \delta^u(x,j,r) \geq \sum_{j=0}^{i-2} \delta^u(y,j,r) \\
&= b(y,i-1,r)
\end{aligned}$$

and

$$\begin{aligned}
u(y,r) &\leq u(x,r) + LS^u(x,0,r) \\
&\leq b(x,i+1,r) + \delta^u(x,0,r) \\
&= \sum_{j=0}^{i} \delta^u(x,j,r) + \delta^u(x,0,r) \\
&\leq \sum_{j=1}^{i+1} \delta^u(y,j,r) + \delta^u(x,0,r) = b(y,i+2,r)
\end{aligned}$$

Thus $u(y,r) \in [b(y,i-1,r), b(y,i+2,r))$. We split the argument in three subcases: (1.1) $u(y,r) \in [b(y,i-1,r), b(y,i,r))$. (1.2) $u(y,r) \in [b(y,i,r), b(y,i+1,r))$ and (1.3) $u(y,r) \in [b(y,i+1,r), b(y,i+2,r))$.

Case (1.1). Assume that $u(y,r) \in [b(y,i-1,r), b(y,i,r))$. Thus we get the following:

$$\begin{aligned}
&D_{u,\delta^u}(x,r) - D_{u,\delta^u}(y,r) && (3.1)\\
&= \frac{u(x,r) - b(x,i,r)}{b(x,i+1,r) - b(x,i,r)} + i - \\
&\quad - \frac{u(y,r) - b(y,i-1,r)}{b(y,i,r) - b(y,i-1,r)} - i + 1 && (3.2)\\
&\leq \frac{u(x,r) - b(x,i,r) - u(y,r) + b(y,i-1,r)}{b(y,i,r) - b(y,i-1,r)} + 1 && (3.3)\\
&\leq \frac{u(x,r) - b(x,i,r) - u(x,r) + \delta(x,0,r) + b(y,i-1,r)}{b(y,i,r) - b(y,i-1,r)} + 1 && (3.4)\\
&\leq \frac{-b(x,i,r) + b(x,i,r)}{b(y,i,r) - b(y,i-1,r)} + 1 \leq 1 && (3.5)
\end{aligned}$$

The rationale for the equations above is the following: Equation (3.2) follows from the application of the definition of D_{u,δ^u}; Equation (3.3), as $b(x,i+1,r) - b(x,i,r) = \delta_u(x,i,r) \geq \delta_u(y,i-1,r) = b(y,i,r) - b(y,i-1,r)$, we have that:

$$\frac{u(x,r) - b(x,i,r)}{b(x,i+1,r) - b(x,i,r)} \leq \frac{u(x,r) - b(x,i,r)}{b(y,i,r) - b(y,i-1,r)}$$

Equation (3.4) holds since $u(y,r) \geq u(x,r) - LS^u(x,0,r) \geq u(x,r) - \delta^u(x,0,r)$. That $D_{u,\delta^u}(y,r) - D_{u,\delta^u}(x,r) \leq 1$ follows by symmetry.

Case (1.2). Assume that $u(y,r) \in [b(y,i,r), b(y,i+1,r))$ which entails that $D_{u,\delta^u}(y,r) \in [i,i+1)$. Likewise, as $u(x,r) \in [b(x,i,r), b(x,i+1,r))$ by assumption, it holds that $D_{u,\delta^u}(x,r) \in [i,i+1)$. In what follows, we get that:

$$|D_{u,\delta}(x,r) - D_{u,\delta}(y,r)| \leq 1$$

Case (1.3). Assume that $u(y,r) \in [b(y,i+1,r), b(y,i+2,r))$. This case follows similar reasoning to the case (1.1), so we omit this part of the proof.

Case (2). Assume $u(x,r) < 0$. This case is symmetric to the case (1) as $D_{u,\delta}$ is symmetric.

Given that, $|D_{u,\delta}(x,r) - D_{u,\delta}(y,r)| \leq 1$ holds for all pairs of neighboring databases $x, y \in D^n$ where $d(x,y) \leq 1$ and for all $r \in R$. □

3.4 Local Dampening Mechanism

We now state the local dampening mechanism a generic non-numeric differentially private mechanism. It takes a database x, the privacy budget ε, the utility function u, an admissable sensitivity function δ^u and the range $\mathscr{R}$ of the function to be sanitized.

It samples an element from $r \in \mathscr{R}$ based on its dampened utility score $D_{u,\delta^u}(x,r)$. The larger the score, the higher the probability of sampling it.

Definition 3.4.1. *(Local dampening mechanism). The local dampening mechanism* $M_{LD}(x, \varepsilon, u, \delta^u, \mathscr{R})$ *selects and outputs an element* $r \in \mathscr{R}$ *with probability proportional to* $\exp\left(\frac{\varepsilon\, D_{u,\delta^u}(x,r)}{2}\right)$.

This version of the local dampening mechanism is specially effective when the sensitivity function is flat. In the following example, we demonstrate the process operation of the local dampening mechanism.

Example 3.4.1. *(Local dampening mechanism) This example explores the local dampening mechanism using the local sensitivity definition while the element local sensitivity is addressed in Chapter 4. Let G be the graph of Figure 9c. As we have discussed in Example 2.4.1, we have that* $LS^{EBC}(G,0) = 3$ *and* $LS^{EBC}(G,1) = 5$*. The EBC scores for the vertices are* $EBC(a) = EBC(b) = 6.5$ *and* $EBC(v_i) = 0$*, for* $0 \leq i \leq 5$*. Their dampened EBC scores are:*

$$D_{EBC,LS^{EBC}}(G,a) = D_{EBC,LS^{EBC}}(G,b) = 1.7$$
$$D_{EBC,LS^{EBC}}(G,v_i) = 0, \textit{for } 0 \leq i \leq 5$$

For instance, assuming $\varepsilon = 2.0$*, the probability for each node to be selected is:*

$$Pr[a \textit{ is selected}] = Pr[b \textit{ is selected}] \propto \exp(1.7) = 5.47$$
$$Pr[v_i \textit{ is selected}] \propto \exp(0) = 1.0, \textit{for } 0 \leq i \leq 5$$

Normalizing, we have that $Pr[a \textit{ is selected}] = Pr[b \textit{ is selected}] = 0.32$ *and* $Pr[v_i \textit{ is selected}] = 0.06$*. Thus the local dampening mechanism samples a element with those probabilities.*

3.5 Privacy Guarantee

We now prove that the local dampening mechanism M_{LD} ensures ε-differential privacy (Theorem 4.3.1). The privacy correctness proof follows from the exponential mechanism correctness (MCSHERRY; TALWAR, 2007) and Lemma 3.3.1.

Theorem 3.5.1. M_{LD} *satisfies* ε*-Differential Privacy if* δ *is admissible.*

Proof. Given two neighboring databases $x, y \in D^n$ (i.e., $d(x,y) \leq 1$) and an output $r \in \mathscr{R}$. We show that the ratio of the probability of r being produced by local dampening mechanism on database x and y is bounded by $\exp(\varepsilon)$.

$$\begin{aligned}
\frac{P_x(r)}{P_y(r)} &= \frac{P[M_{LD}(x,u,\mathscr{R}) = r]}{P[M_{LD}(y,u,\mathscr{R}) = r]} \\
&= \frac{\left(\frac{\exp(\frac{\varepsilon D_{u,\delta}(x,r)}{2})}{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon D_{u,\delta}(x,r')}{2})}\right)}{\left(\frac{\exp(\frac{\varepsilon D_{u,\delta}(y,r)}{2})}{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon D_{u,\delta}(y,r')}{2})}\right)} \\
&= \left(\frac{\exp(\frac{\varepsilon D_{u,\delta}(x,r)}{2})}{\exp(\frac{\varepsilon D_{u,\delta}(y,r)}{2})}\right) \cdot \left(\frac{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon D_{u,\delta}(y,r')}{2})}{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon D_{u,\delta}(x,r')}{2})}\right)
\end{aligned}$$

$$
\begin{aligned}
&\leq \exp\left(\frac{\varepsilon(D_{u,\delta}(x,r') - D_{u,\delta}(y,r'))}{2}\right) \\
&\cdot \left(\frac{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon(D_{u,\delta}(x,r')+1)}{2})}{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon D_{u,\delta}(x,r')}{2})}\right) \\
&\leq \exp\left(\frac{\varepsilon}{2}\right) \cdot \exp\left(\frac{\varepsilon}{2}\right) \cdot \left(\frac{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon D_{u,\delta}(x,r')}{2})}{\sum_{r' \in \mathscr{R}} \exp(\frac{\varepsilon D_{u,\delta}(x,r')}{2})}\right) \\
&= \exp(\varepsilon)
\end{aligned}
$$

The two inequalities follow from lemma 3.3.1. By symmetry, $\frac{P[M_{LD}(x,u,\mathscr{R})=r]}{P[M_{LD}(y,u,\mathscr{R})=r]} \geq \exp(-\varepsilon)$ holds. □

3.6 Related Work

There is a vast literature on differential privacy for numeric queries, and we refer the interested reader to (MACHANAVAJJHALA *et al.*, 2017) for a recent survey. In this section, we discuss differential privacy approaches for the non-numeric setting.

3.6.1 Exponential Mechanism

The exponential mechanism M_{EM} (MCSHERRY; TALWAR, 2007) is the most used approach for providing differential privacy to the non-numeric setting. It uses a notion of global sensitivity Δu (Definition 2.4.1), adapted from Dwork *et al.* (2006b).

The exponential mechanism privately answers a function $f : \mathscr{D}^n \to \mathscr{R}$ applied to database x by sampling an element $r \in \mathscr{R}$ with probability proportional to its utility score $u(x,r)$. It uses the exponential distribution to assign probabilities for each $r \in \mathscr{R}$. The exponential mechanism is stated as follows:

Definition 3.6.1. *(Exponential Mechanism (MCSHERRY; TALWAR, 2007)). The exponential mechanism $M_{EM}(x,\varepsilon,u,\mathscr{R})$ selects and outputs an element $r \in \mathscr{R}$ with probability proportional to* $\exp\left(\frac{\varepsilon\, u(x,r)}{2\Delta u}\right)$.

McSherry and Talwar (2007) showed that the exponential mechanism satisfies ε-differential privacy.

In Chapter 4, we show that, under some conditions, the exponential mechanism is never worse than the local dampening in terms of accuracy. Additionally, we carry out an experimental evaluation with the two applications that we tackle in this work: influential node analysis (Chapter 6) and decision tree induction (Chapter 7).

Example 3.6.1. *(Comparison local dampening mechanism with exponential mechanism) We make a simple comparison of the probabilities of the local dampening mechanism with the exponential mechanism in Example 3.4.1.*

In example Example 3.4.1, we have that $Pr[a \text{ is selected}] = Pr[b \text{ is selected}] = 0.32$ and $Pr[v_i \text{ is selected}] = 0.06$. While, according to Definition 3.6.1, the exponential mechanism obtained that $Pr[a \text{ is selected}] = Pr[b \text{ is selected}] = 0.22$ and $Pr[v_i \text{ is selected}] = 0.09$. Thus local dampening yields a higher probability of choosing the node with highest score.

3.6.2 Permute-and-Flip

The *permute-and-flip M_{PF}* (MCKENNA; SHELDON, 2020) mechanism is recent work that also address differential privacy for the non-numeric setting. It is defined as an iterative algorithm that employs the exponential distribution to assign probabilities for each element r.

Algorithm 1 formally defines permute-and-flip.

Algorithm 1: Permute-and-Flip

```
Procedure $M_{PF}$(Database $x$, Privacy Budget $B$, utility function $u$,
  Range set $\mathscr{R}$)
    $u^* = \max_{r \in \mathscr{R}} u(x,r)$
    for $r \in RandomPermutation(\mathscr{R})$ do
        $p_r = \exp\left(\frac{\varepsilon}{2\Delta u}(u(x,r) - u^*)\right)$
        if $Bernoulli(p_r)$ then
            return $r$
        end
    end
```

Basically, the algorithm iterates over a random permutation of the elements $r \in \mathscr{R}$ and then flip a biased coin with probability $\frac{\varepsilon}{2\Delta u}(u(x,r) - u^*)$. u^* is the maximum utility observed over all elements in the range set $\mathscr{R}$ given the input database x. Thus, the closer $u(x,r) - u^*$, more likely is r to be outputted. The mechanism is guaranteed to terminate with a result because if $u(x,r) = u^*$, then the probability of heads is 1.

McKenna and Sheldon (2020) show that their approach is also never worse than the exponential mechanism in terms of accuracy. We conduct an empirical comparison of permute-and-flip mechanism to local dampening in Chapters 6 and 7.

3.7 Discussion

The concept of local sensitivity was introduced in (NISSIM *et al.*, 2007) for numeric queries. The authors proposed the smooth sensitivity framework, which is a generic approach for numeric queries. They applied it to compute the median, the cost of a minimum spanning tree, the count of triangles in a graph and k-means. Also local sensitivity was used in many other works (ZHANG *et al.*, 2015; KARWA *et al.*, 2011; CORMODE *et al.*, 2012; KASIVISWANATHAN *et al.*, 2013; RASTOGI *et al.*, 2009).

On the other hand, many differential privacy works have tackled non-numeric problems using non-numeric queries as part of their approaches (ZHANG *et al.*, 2017; HARDT *et al.*, 2012; FRIEDMAN; SCHUSTER, 2010; MOHAMMED *et al.*, 2011; HARDT *et al.*, 2010; CORMODE *et al.*, 2012). However, to the best of our knowledge, the literature lacks a generic framework for providing differential privacy for non-numeric queries using local sensitivity. Our work fills this gap.

We adapted and defined notion of local sensitivity for non-numeric queries. Also we defined the family of sensitivity function which include the definitions of local sensitivity and provided a classification for them. Given that, we proposed the local dampening mechanism that uses the local sensitivity to attenuate the utility function and reduce the noise injected to the output.

In next chapter, we provide a new version of the local dampening mechanism and theoretical accuracy guarantees for it.

4 SHIFTED LOCAL DAMPENING MECHANISM

In this chapter, we present a second version of the local dampening mechanism name *shifted local dampening* mechanism M_{SLD}. This version is designed for non-flat monotonic sensitivity functions which is the most usual case in our experiments.

We develop an insightful discussion on accuracy of the shifted local dampening mechanism. We provide tools to compare two instances of the shifted local mechanism in terms of accuracy. Also, these tools guide on the design of good sensitivity functions that provide accurate M_{SLD} instances. We show that, with a stable sensitivity function, the local dampening mechanism is never worse than the exponential mechanism. Additionally, even if the stability condition is not met, we discuss how to construct good sensitivity functions.

4.1 Inversion problem

First, we exemplify an anomaly that happens when the sensitivity function is not monotonic.

Consider the scenario where we dampen the utility scores of the elements $r \in \mathscr{R}$ with the sensitivity function δ^u that is not monotonic. This might be the case when we use $\delta^u(x,t,r)$ as the element local sensitivity, $\delta^u(x,t,r) = LS^u(x,t,r)$.

In this scenario, Example 4.1.1 illustrates a case where the local dampening change the relative order of the dampened utility scores compared to the original utility scores. We refer to this problem as the *inversion problem.*

Example 4.1.1. *(Inversion problem) Consider the following setup:* $\mathscr{R} = \{r_1, r_2\}$, $\delta^u(x,0,r_1) = 1$, $\delta^u(x,1,r_1) = 2$, $\delta^u(x,0,r_2) = 4$, $u(x,r_1) = 3$ *and* $u(x,r_2) = 4$. *When applying* D_{u,δ^u} *to* r_1 *and* r_2, *we obtain* $D_{u,\delta^u}(x,r_1) = 2$ *and* $D_{u,\delta^u}(x,r_2) = 1$. *Originally,* r_2 *is more useful than* r_1 *but after dampening it inverts. This hurts accuracy since the local dampening mechanism will choose* r_1 *with higher probability.*

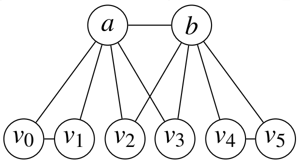
4.2 Shifted Local Dampening

The key idea for this extension is the use of shifting in the utility score to take advantage of non-flat monotonic sensitivity functions δ^u. The discussion in this section is focused on non-flat monotonic sensitivity functions. However, we show later that the shifted local dampening also performs well for non strictly monotonic functions.

Example 4.2.1 shows a case where shifting increases the probability of high utility elements to be chosen (i.e. improves accuracy) when δ^u is monotonically non-decreasing.

Example 4.2.1. *(Utility function shifting) Consider the graph G from figure 15. For nodes a and b, their measured element local sensitivities are:* $LS^{EBC}(G,0,a) = LS^{EBC}(G,0,b) = 3$ *and* $LS^{EBC}(G,1,a) = LS^{EBC}(G,1,b) = 5$. *For a node* v_i, *for* $0 \leq i \leq 5$, *its measured sensitivity is* $LS^{EBC}(G,0,v_i) = 2$. *We observe the non-decreasing monotonicity of* LS^{EBC}, *since the EBC scores are* $EBC(a) = EBC(b) = 6.5$ *and* $EBC(v_i) = 0$, *for* $0 \leq i \leq 5$.

Figure 15 – Original Graph G



Source: elaborated by the author.

For instance, shifting the EBC scores by -7, *we get that* $EBC'(a) = EBC'(b) = -0.5$ *and* $EBC'(v_i) = -7$, *for* $0 \leq i \leq 5$. *Then we compute their dampened* EBC' *scores:*

$$D_{EBC',LS^{EBC}}(G,a) = D_{EBC',LS^{EBC}}(G,b) = 0.1$$

$$D_{EBC',LS^{EBC}}(G,v_i) = -2, \textit{for } 0 \leq i \leq 5$$

Let $\varepsilon = 2.0$. *The probability for each node to be selected is:*

$$Pr[a \textit{ is selected}] = Pr[b \textit{ is selected}] \propto \exp(0.1) = 0.44$$

$$Pr[v_i \textit{ is selected}] \propto \exp(-2) = 0.13, \textit{for } 0 \leq i \leq 5$$

Normalizing, we have that $Pr[a \textit{ is selected}] = Pr[b \textit{ is selected}] = 0.472$ *and* $Pr[v_i \textit{ is selected}] = 0.0046$. *Recall that, the exponential mechanism obtained that* $Pr[a \textit{ is selected}] = Pr[b \textit{ is selected}] = 0.22$ *and* $Pr[v_i \textit{ is selected}] = 0.09$ *(Example 3.6.1) and, for the unshifted local dampening mechanism, (Example 3.4.1), we have that* $Pr[a \textit{ is selected}] = Pr[b \textit{ is selected}] = 0.32$ *and* $Pr[v_i \textit{ is selected}] = 0.06$. *The nodes with highest score increase probability compared to the unshifted local dampening and the exponential mechanism.*

For the sake of argument, suppose that $\delta^u(x,t,r)$ is monotonically non-decreasing. We design the shifting in a way that it rearranges the utilities scores such that the distribution of the utility scores is more spread.

The idea is the following: we shift left enough so that all utility scores are negative. The elements with larger utility score are the elements with smallest absolute value after shifting. Thus, these shifted scores are dampened with large $\delta^u(x,t,r)$ (by assumption of non-decreasing monotonicity). This implies that large utility scores are dampened closer to 0 and the opposite happens to elements with small utility scores. The elements with small utility scores are dampened with small $\delta^u(x,t,r)$ and, consequently, the scores are less attenuated and far from 0. This implicates in more spread distribution of utility scores.

Hereby we propose to replace the original utility function u with its shifted version u^s where s is the utility score shift and

$$u^s(x,r) = u(x,r) - s$$

One could design a private query, consuming part of the privacy budget, to choose s such that it minimizes some loss function to optimize accuracy. In this work, we set s to a value that does not depend on private data, $s \to \infty$. In what follows, the shifted local dampening mechanism is stated as follows:

Definition 4.2.1. *(Shifted Local Dampening Mechanism - non-decreasing sensitivity function). The shifted local dampening mechanism $M_{SLD}(x,\varepsilon,u,\delta^u,\mathscr{R})$ outputs an element $r \in \mathscr{R}$ with probability equals to*

$$\lim_{s \to \infty} \left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r)}{2}\right)}{\sum_{r' \in \mathscr{R}} \exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2}\right)} \right).$$

When δ^u is monotonically non-increasing the following definition of the shifted local dampening mechanism applies:

Definition 4.2.2. *(Shifted Local Dampening Mechanism - non-increasing sensitivity function). The shifted local dampening mechanism $M_{SLD}(x,\varepsilon,u,\delta^u,\mathscr{R})$ outputs an element $r \in \mathscr{R}$ with probability equals to*

$$\lim_{s \to -\infty} \left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r)}{2}\right)}{\sum_{r' \in \mathscr{R}} \exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2}\right)} \right).$$

For the case of functions that do not depend on r, both versions of the shifted local dampening mechanism are applicable.

4.3 Privacy Guarantee

We now prove that the shifted local dampening mechanism M_{SLD} ensures ε-differential privacy. For the privacy guarantee, the sensitivity function δ^u just needs to be admissable and bounded but not necessarily monotonic. Recall that boundedness can be easily achieved (Section 3.2.2).

We first show an intermediate result:

Lemma 4.3.1. *If δ^u is admissible and bounded sensitivity function then $\frac{\exp(\varepsilon D_{u^s,\delta^u}(x,r)/2)}{\sum_{r'\in\mathscr{R}}\exp(\varepsilon D_{u^s,\delta^u}(x,r')/2)} = \frac{\exp(\varepsilon D_{u^{s_0},\delta^u}(x,r)/2)}{\sum_{r'\in\mathscr{R}}\exp(\varepsilon D_{u^{s_0},\delta^u}(x,r')/2)}$ for $s \geq s_0$ where $s_0 = n\Delta u + \max_{r'\in R} u(x,r')$ and n is the size of the input database.*

Proof. Let $r \in \mathscr{R}$ be an output element. By definition of u^s, observe that

$$u^s(x,r) = u(x,r) - s \leq u(x,r) - n\Delta u - \max_{r'\in R} u(x,r') = u^{s_0}(x,r)$$

since $s \geq s_0$. And also, as $n \geq 0$, we get that

$$u^{s_0} = u(x,r) - n\Delta u - \max_{r'\in R} u(x,r') \leq -n\Delta u \leq 0$$

This means that $u^{s_0}(x,r)$ is non-positive and, by consequence, $u^s(x,r)$ is also non-positive for all $s > s_0$ and all $r \in R$. Therefore, by the construction of $D_{\delta^u,u}$, we have that $u^{s_0} \in [b(x,i,r), b(x,i+1,r))$ for some $i \leq 0$ since $u^{s_0}(x,r) \leq 0$. As δ^u is bounded:

$$\begin{aligned}
& b(x,i,r) \leq u^{s_0}(x,r) \\
& \Rightarrow -\sum_{j=0}^{-i-1} \delta^u(x,j,r) \leq u(x,r) - n\Delta u - \max_{r'\in R} u(x,r') \\
& \Rightarrow (i+1)\Delta u \leq -n\Delta u \\
& \Rightarrow i+1 \leq -n \\
& \Rightarrow n \leq -i-1
\end{aligned}$$

This last fact, the admissibility and convergence (δ^u is bounded) of δ^u lead us to show that the difference $b(x,k,r) - b(x,i,r)$ is equal to $(k-i)\Delta u$ for all $k \leq i < 0$. We will use this fact posteriorly.

$$b(x,k,i) - b(x,i,r) = \tag{4.1}$$

$$= -\sum_{j=0}^{-k-1} \delta^u(x,-j,r) + \sum_{j=0}^{-i-1} \delta^u(x,-j,r) \tag{4.2}$$

$$= -\sum_{j=-i}^{-k-1} \delta^u(x,-j,r) \tag{4.3}$$

$$= -\sum_{j=-i}^{-k-1} \delta^u(x,n,r) \tag{4.4}$$

$$= (k-i)\Delta u \tag{4.5}$$

Note that $u^s \in [b(x,k,r), b(x,k+1,r))$ for some $k \leq i \leq 0$ since $u^s(x,r) \leq u^{s_0}(x,r) \leq 0$. Given that, we calculate the difference:

$$D_{u^{s_0},\delta^u}(x,r) - D_{u^s,\delta^u}(x,r) \tag{4.6}$$

$$= \frac{u^{s_0}(x,r) - b(x,i,r)}{b(x,i+1,r) - b(x,i,r)} + i - \frac{u^s(x,r) - b(x,k,r)}{b(x,k+1,r) - b(x,k,r)} - k \tag{4.7}$$

$$= \frac{u^{s_0}(x,r) - b(x,i,r) - u^s(x,r) + b(x,k,r)}{\Delta u} \quad -k+i \tag{4.8}$$

$$= \frac{u^{s_0}(x,r) - b(x,i,r) - u^{s_0}(x,r) - s_0 + s + b(x,k,r)}{\Delta u} - k + i \tag{4.9}$$

$$= \frac{s-s_0}{\Delta u} + \frac{b(x,k,r) - b(x,i,r)}{\Delta u} - k + i \tag{4.10}$$

$$= \frac{s-s_0}{\Delta u} + \frac{(k-i)\Delta u}{\Delta u} - k + i = \frac{s-s_0}{\Delta u} \tag{4.11}$$

Equation (4.9) holds since $u^s(x,r) = u(x,r) - s = u^{s_0}(x,r) + s_o - s$ and equation 4.10 follows from equation 4.5.

Finally,

$$\frac{\exp(\varepsilon\, D_{u^s,\delta^u}(x,r)/2)}{\sum_{r' \in R} \exp(\varepsilon\, D_{u^s,\delta^u}(x,r')/2)}$$

$$
\begin{aligned}
&= \frac{\exp(\varepsilon\ (D_{u^{s_0},\delta^u}(x,r) - (s-s_0)/\Delta u)/2)}{\sum_{r'\in R}\exp(\varepsilon\ (D_{u^{s_0},\delta^u}(x,r') - (s-s_0)/\Delta u)/2)} \\
&= \frac{\exp(-\varepsilon(s-s_0)/2\Delta u).\exp(\varepsilon\ D_{u^{s_0},\delta^u}(x,r)/2)}{\exp(-\varepsilon(s-s_0)/2\Delta u).\sum_{r'\in R}\exp(\varepsilon\ D_{u^{s_0},\delta^u}(x,r')/2)} \\
&= \frac{\exp(\varepsilon\ D_{u^{s_0},\delta^u}(x,r)/2)}{\sum_{r'\in R}\exp(\varepsilon\ D_{u^{s_0},\delta^u}(x,r')/2)}
\end{aligned}
$$

□

Lemma also 4.3.1 gives hint about the implementation. It suffices to shift by $n\Delta u + \max_{r'\in R} u(x,r')$ to meet the definition of the shifted local dampening. Also, from Lemma 4.3.1, it follows directly (Corollary 4.3.1).

Corollary 4.3.1. $\lim_{s\to\infty}\left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r)}{2}\right)}{\sum_{r'\in R}\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2}\right)}\right)$ *exists and is equal to* $\frac{\exp(\varepsilon\ D_{u^s,\delta^u}(x,r)/2)}{\sum_{r'\in\mathscr{R}}\exp(\varepsilon\ D_{u^s,\delta^u}(x,r')/2)}$ *for* $s \geq s_0$ *where* $s_0 = n\Delta u + \max_{r'\in R} u(x,r')$ *and* n *is the size of the input database.*

The privacy correctness proof follows from the exponential mechanism correctness (MCSHERRY; TALWAR, 2007), Lemma 3.3.1 and Corollary 4.3.1. In this proof we use the non-decreasing admissable function version of the local dampening (Definition 4.2.1). The non-increasing version (Definition 4.2.2) privacy guarantee proof is symmetric.

Theorem 4.3.1. *M_{SLD} satisfies ε-Differential Privacy if δ^u is admissible and bounded.*

Proof. Given two neighboring databases $x, y \in D^n$ (i.e., $d(x,y) \leq 1$) and an output $r \in R$. We need show that the ratio of the probability of r being produced by shifted local dampening mechanism on database x and y is bounded by $\exp(\varepsilon)$.

$$
\begin{aligned}
\frac{P_x(r)}{P_y(r)} &= \frac{P[M_{SLD}(x,\varepsilon,u,\delta^u,\mathscr{R}) = r]}{P[M_{SLD}(y,\varepsilon,u,\delta^u,\mathscr{R}) = r]} \\
&= \frac{\lim_{s\to\infty}\left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r)}{2}\right)}{\sum_{r'\in R}\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2}\right)}\right)}{\lim_{s\to\infty}\left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(y,r)}{2}\right)}{\sum_{r'\in R}\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(y,r')}{2}\right)}\right)} \\
&= \lim_{s\to\infty}\left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r)}{2}\right)}{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(y,r)}{2}\right)}\cdot\frac{\sum_{r'\in\mathscr{R}}\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(y,r')}{2}\right)}{\sum_{r'\in\mathscr{R}}\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2}\right)}\right) \\
&\leq \lim_{s\to\infty}\left(\exp\left(\frac{\varepsilon(D_{u^s,\delta}(x,r') - D_{u^s,\delta}(y,r'))}{2}\right)\cdot\right.
\end{aligned}
$$

$$\left(\frac{\sum_{r'\in\mathscr{R}}\exp(\frac{\varepsilon(D_{u^s,\delta^u}(x,r')+1)}{2})}{\sum_{r'\in\mathscr{R}}\exp(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2})}\right)\Bigg)$$

$$\leq \lim_{s\to\infty}\left(\exp\left(\frac{\varepsilon}{2}\right)^2\cdot\left(\frac{\sum_{r'\in\mathscr{R}}\exp(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2})}{\sum_{r'\in\mathscr{R}}\exp(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2})}\right)\right)$$

$$= \lim_{s\to\infty}(\exp(\varepsilon)) = \exp(\varepsilon)$$

The two inequalities follow from Lemma 3.3.1 and the limit operations are allowed since $\lim_{s\to\infty}\left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r)}{2}\right)}{\sum_{r'\in R}\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(x,r')}{2}\right)}\right)$ and $\lim_{s\to\infty}\left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(y,r)}{2}\right)}{\sum_{r'\in R}\exp\left(\frac{\varepsilon D_{u^s,\delta^u}(y,r')}{2}\right)}\right)$ exist. By symmetry, $\frac{P[M_{SLD}(x,\varepsilon,u,\delta^u,\mathscr{R})=r]}{P[M_{SLD}(y,\varepsilon,u,\delta^u,\mathscr{R})=r]} \geq \exp(-\varepsilon)$ holds. □

4.4 Accuracy Analysis

In this section, we provide theoretical analysis on the accuracy. We aim to answer to the following questions: i) How to compare two instances of the local dampening with two different admissible functions?; ii) Under which conditions does the local dampening performs more accurately than the exponential mechanism?; iii) If those conditions are not met, how to build good admissible functions? and iv) How does local dampening compare to the exponential mechanism in terms of accuracy?.

We evaluate the accuracy of a given mechanism M by studying the error random variable $\mathscr{E}$. $\mathscr{E}$ gives how much the element sampled by M differ from the optimal element in terms of utility.

$$\mathscr{E}(M,x) = u^* - u(x,M(x))$$

where u^* is the optimal utility score, $u^* = \max_{r\in\mathscr{R}} u(x,r)$.

To compare two instances of the local dampening for the same problem, we need to analyse the features of the function δ^u. We develop a discussion on accuracy guarantees for stable functions where we show how to compare two stable functions and show that, using a stable function, the local dampening mechanism is never worse than the exponential mechanism in terms of accuracy.

4.4.1 Accuracy Analysis for Stable Sensitivity Functions

Two instances of the local dampening mechanism can be compared by their stable sensitivity functions. As lower sensitivity means higher accuracy, a stable sensitivity function that produces lower values implies in higher accuracy. For that analysis we establish a relation of dominance between two stable sensitivity functions:

Definition 4.4.1. *(Dominance) Let $\delta^u(x,t,r)$ and $\bar{\delta}^u(x,t,r)$ be two stable sensitivity functions and x be a database. Let $\alpha(x,t,r)$ be referred to the gap between $\delta^u(x,t,r)$ and $\bar{\delta}^u(x,t,r)$: $\delta^u(x,t,r) = \bar{\delta}^u(x,t,r) + \alpha(x,t,r)$. Assume that $\mathscr{R} = \{r_1,...,r_q\}$ is ordered such that $u(x,r_1) \geq \cdots \geq u(x,r_q)$. If $\alpha(x,t,r_1) \geq \alpha(x,t,r_2) \geq \cdots \geq \alpha(x,t,r_q) \geq 0$ for all $t \geq 0$ then $\delta^u(x,t,r)$ dominates $\bar{\delta}^u(x,t,r)$.*

Given that, we can affirm that an instance of the local dampening mechanism using $\delta^u(x,t,r)$ is never worse than an instance using the dominated $\bar{\delta}^u(x,t,r)$:

Lemma 4.4.1. *(Local Dampening Accuracy) Let $\delta^u(x,t,r)$ and $\bar{\delta}^u(x,t,r)$ be two stable functions and x be a database. If $\delta^u(x,t,r)$ dominates $\bar{\delta}^u(x,t,r)$ then:*

1. *$Pr[\mathscr{E}(M_{SLD},x) \geq \theta] \leq Pr[\mathscr{E}(\overline{M}_{SLD},x) \geq t]$ for all $\theta \geq 0$,*
2. *$\mathbb{E}[\mathscr{E}(M_{SLD},x)] \leq \mathbb{E}[\mathscr{E}(\overline{M}_{SLD},x)]$,*

where M_{SLD} represents an instance of the shifted local dampening mechanism using δ^u as sensitivity function while $\overline{M_{SLD}}$ is an instance using $\bar{\delta}^u$.

The proof of Lemma 4.4.1 is deferred to the Appendix A. We can use Lemma 4.4.1 as tool understand the accuracy of the local dampening mechanism. It suggests that a sensitivity function should be as inclined as possible, i.e., a higher difference between two gaps $\alpha(x,t,r_i)$ and $\alpha(x,t,r_{i+1})$ implies in higher accuracy. Also, the gaps $\alpha(x,t,r_i)$ should be as large as possible.

4.4.2 Comparison to the Exponential Mechanism

A very useful property of both versions of the local dampening mechanism is that the exponential mechanism is an instance of the local dampening mechanism. The exponential mechanism is obtained by setting $\delta^u(x,t,r) = \Delta u$ in an instance of the shifted local dampening.

Thus we can use Lemma 4.4.1 to compare any instance of the exponential mechanism using a given stable function $\delta^u(x,t,r)$ against the exponential mechanism. Note by the assumption of boundedness of the stable sensitivity function $\delta^u(x,t,r)$ we have that $\delta^u(x,t,r) \leq \Delta u$, for all x, $t \geq 0$ and $r \in \mathscr{R}$. It implies that $\delta^u(x,t,r)$ dominates Δu. Thus the following corollary holds:

Corollary 4.4.1. *Let δ^u be a stable function. The shifted local dampening mechanism $M_{SLD}(x,\varepsilon,u,\delta^u,\mathscr{R})$ is never worse than the exponential mechanism $M_{EM}(x,\varepsilon,u,\mathscr{R})$, that is:*

1. *$Pr[\mathscr{E}(M_{SLD},x) \geq t] \leq Pr[\mathscr{E}(M_{EM},x) \geq t]$ for all $t \geq 0$,*
2. *$\mathbb{E}[\mathscr{E}(M_{SLD},x)] \leq \mathbb{E}[\mathscr{E}(M_{EM},x)]$.*

Note that this results implies that we can use local sensitivity $LS^u(x,t)$ safely since $LS^u(x,t)$ is a stable sensitivity function, i.e., using the shifted local dampening mechanism with $LS^u(x,t)$ as sensitivity function is never worse than the exponential mechanism. Yet, it suggests that the larger the difference between $LS^u(x,t)$ and Δu, the more accurate it is in relation to the exponential mechanism.

This result also suggests that using the Δu as a sensitivity function is the worst case stable function. Given that, what would be the best stable function? The element local sensitivity $LS^u(x,t,r)$ function is a good candidate. As shown before, $LS^u(x,t,r)$ is admissable and bounded. However, $LS^u(x,t,r)$ is not necessarily monotonic. We demonstrate that $LS^u(x,t,r)$ is minimum admissable, i.e. it dominates all admissable functions:

Lemma 4.4.2. *$LS^u(x,t,r)$ is minimum admissable, i.e. $LS^u(x,t,r)$ dominates any admissible sensitivity function $\delta^u(x,t,r)$.*

Proof. We show that $LS^u(x,t,r)$ is less than or equal any admissible sensitivity function $\delta^u(x,t,r)$ by induction on t.

Basis: for $t = 0$, $LS^u(x,0,r) \leq \delta^u(x,0,r)$ holds since δ^u is admissible for all $x \in D^n, r \in R$

Inductive step: suppose that $LS^u(x,t,r) \leq \delta^u(x,t,r)$ is true for all $x \in D^n, r \in R$. We must show that $LS^u(x,t+1,r) \leq \delta^u(x,t+1,r)$ is true for all $x \in D^n, r \in R$. By the definition of element local sensitivity:

$$
\begin{aligned}
LS^u(x,t+1,r) &= \max_{y|d(x,y)\leq t+1} LS^u(y,0,r) \\
&= \max_{y|d(x,y)\leq 1} \max_{y|d(y,z)\leq t} LS^u(z,0,r)
\end{aligned}
$$

$$\leq \max_{y|d(x,y)\leq 1} \delta^u(y,t,r) \leq \delta^u(x,t+1,r)$$

First inequality holds by hypothesis and the second inequality follows by the admissibility of δ^u. Thus $LS^u(x,t+1,r) \leq \delta^u(x,t+1,r)$ for all $x \in D^n, r \in \mathscr{R}, t \geq 0$. □

Even if $LS^u(x,t,r)$ happens to be non monotonic, we devote next subsection to discuss that only a "weak" monotonicity is enough for our mechanism. Also, we discuss other cases where the local dampening mechanism also performs well.

4.4.3 Relaxing Monotonicity

We have shown theoretical guarantees for the accuracy of the shifted local dampening mechanism using stable functions. For sensitivity function like the global sensitivity Δu and the local sensitivity $LS^u(x,t)$ we have strong accuracy guarantees.

Strict monotonicity may be a complex goal to achieve. In the applications and datasets analysed in our experimental section, none of them satisfy the strict monotonicity requirement. Yet, the shifted local dampening mechanism outperforms the exponential mechanism in our experiments.

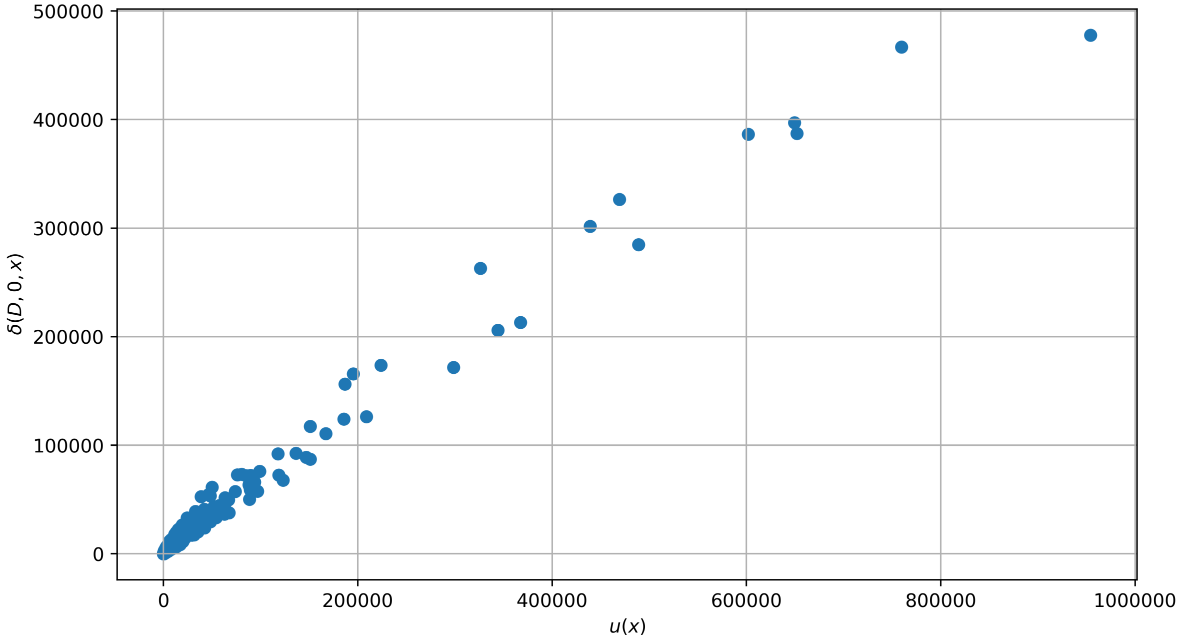
For those sensitivity functions that violates monotonicity, we use the results on Section 4.4.1 as guide to construct a good sensitivity function. The same analysis also work here, Lemma 4.4.1 suggests that a sensitivity function should be as inclined as possible, i.e., a higher difference between two gaps $\alpha(x,t,r_i)$ and $\alpha(x,t,r_{i+1})$ implies in higher accuracy. Also, the gaps $\alpha(x,t,r_i)$ should be as large as possible.

For the running example of this thesis (Example 1.1.2), we designed an admissible function δ^{EBC} stated in Definition 6.3.1 for the use of the shifted local dampening mechanism. Figure 16 displays the value of $u(x,r)$ on the x-axis against $\delta^u(x,0,r)$ for the Enron graph database (LESKOVEC; KREVL, 2014). This example clearly violates strict monotonicity. But it shows a weak monotonicity for the sensitivity function $\delta^E BC$ in the sense that $EBC(x,r)$ is still positively correlated with $\delta^{EBC}(x,t,r)$ with respect to r.

We argue that this kind of behavior is enough for a good performance of the shifted local dampening. Our empirical results corroborates with this argument.

Additionally, Lemma 4.4.1 also suggests that functions that do not exhibit correlation but have lower value than Δu also perform well which is the case for local sensitivity $LS^u(x,t)$.

Figure 16 – Correlation between $EBC(x,r)$ and $\delta^{EBC}(x,0,r)$ for EBC metric for Enron Dataset.



Source: elaborated by the author.

4.5 Conclusion

In this chapter, we introduced an extension to the local dampening mechanism. This version fixes the inversion problem that happens to the standard local dampening mechanism when the sensitivity function is monotonic. The ideia of the shifted local dampening mechanism is to shift the utility function such that the utility scores of the elements become more spread which translates in a higher accuracy. We prove it to satisfy ε-differential privacy.

We provided a theoretical analysis of the accuracy of the shifted local dampening mechanism. We show that when the sensitivity function is stable, the shifted local dampening is never worse than the exponential mechanism in terms of accuracy. Also, we discussed how to build sensitivity functions that produces instances of the shifted local dampening mechanism with high accuracy. Lastly, building a stable sensitivity function can be a hard task for some problems. Thus we argued that even with the sensitivity function violates the hardest condition to achieve stability, the monotonicity, it suffices to have a correlation between the utility function and the sensitivity. In the next chapters, we provide an empirical evaluation to demonstrate that.

5 APPLICATION 1: MEDIAN SELECTION

Median selection is a known basic problem for testing the accuracy of private mechanisms. A median selection algorithm should output the label of the element with the closest value to the median.

Nissim *et al.* (2007) McKenna and Sheldon (2020) tackled a different version of this problem. Nissim *et al.* (2007) tackled the numeric version of the median selection problem where the task is to return the median value itself and not the label of the median element. McKenna and Sheldon (2020) addressed binned version of the median selection problem where the data is binned in k buckets and the goal is to return the median bin. The latter version has a low global sensitivity. We provide an experimental comparison against the exponential mechanism (MCSHERRY; TALWAR, 2007) and permute-and-flip (MCKENNA; SHELDON, 2020).

5.1 Problem Statement

Given a database $x \in \mathbb{R}^n$ represented as vector of real numbers $< x_1, \cdots, x_n >$. Suppose that x is ordered such that $x_1 \leq \cdots \leq x_n$. Also suppose that all the values lies in $[0, \Lambda]$, $0 \leq x_1 \leq \cdots \leq x_n \leq \Lambda$.

The task is to return the index i where its element x_i is as close as possible to the median element x_m where $m = \lceil \frac{n}{2} \rceil$. Note that $\mathscr{R} = \{1, \cdots, n\}$. The utility function for a given index i is the distance from x_i to x_m multiplied by -1 so that closer elements have higher utility score:

Definition 5.1.1. *(Utility function for median selection problem).* $u_{med}(x, i) = -|x_m - x_i|$.

5.2 Private Mechanism and Sensitivity Analysis

This problem is solved by a single call to a non-numeric mechanism. Here we use the exponential mechanism and the permute-and-flip mechanism to compare to the local dampening mechanism. The exponential mechanism and the permute-and-flip mechanism require the computation of the global sensitivity Δu_{med} while the local dampening mechanism requires the computation of the element local sensitivity u_{med}.

5.2.1 Global Sensitivity

The global sensitivity Δu_{med} is set by the following example: let $x \in \mathbb{R}^n$ be a database where $x_1 = x_2 = \cdots = x_{n-1} = 0$ and $x_n = \Lambda$. Let $y =< y_1, \cdots, y_n >\in \mathbb{R}^n$ be a neighboring database of x obtained from x by changing the value of x_n to 0, $y_n = 0$. Thus we have that $y_1, ..., y_n = 0$. In what follows, $u(x,n) = \Lambda$ and $u(y,n) = 0$ as $x_m = y_m = 0$ which implies that $u(x,n) - u(y,n) = \Lambda$.

$$\Delta u_{med} = \max_{i \in \{1,...,n\}} \max_{x,y | d(x,y) \leq 1} |u(x,i) - u(y,i)| \tag{5.1}$$

This happens to be the largest possible $|u(x,i) - u(y,i)|$ since $|u(x,i) - u(y,i)| \leq \Lambda$ for any $x, y \in real^n$ and $i \in [1,n]$. The latter follows from the fact that the distance from x_i to x_m is positive and smaller than Λ, $0 \leq u(x,i) \leq \Lambda$ and $0 \leq u(y,i) \leq \Lambda$.

Lemma 5.2.1. *(Median Selection Global Sensitivity.)* $\Delta u_{med} = \Lambda$.

5.2.2 Element local sensitivity

Element local sensitivity at distance 0. Before calculating the element local sensitivity of u_{med} at distance t, we discuss how to compute the element local sensitivity at distance 0 $LS^{u_{med}}(x,0,i)$.

Observe that a naive computation of $LS^{u_{med}}(x,0,i)$ is infeasible. It needs to iterate over each neighboring database y of x and take $|u(x,r) - u(y,r)|$. The number of neighboring databases is infinite because we can set a given x_i to any real value in $[0,\Lambda]$.

$$LS^u(x,0,r) = \max_{y | d(x,y) \leq 1} |u(x,r) - u(y,r)|$$

Thus we provide a way to efficiently compute $LS^{u_{med}}(x,0,i)$ in $O(1)$ time complexity.

Lemma 5.2.2. *(Median Selection Element Local Sensitivity at distance 0)*

$$LS^{u_{med}}(x,0,i) = \max(|x_m - x_i|, x_{m+1} - x_m, x_m - x_{m-1}, p(x,i), q(x,i)),$$

where

$$p(x,i) = \max \begin{cases} \Lambda - x_i & \textit{if } i > m \\ \Lambda - x_{m+1} & \textit{if } i = m \\ \Lambda + x_i - 3x_m + x_{m+1} & i < m \end{cases},$$

$$q(x,i) = \max \begin{cases} x_i & \textit{if } i > m \\ x_{m-1} & \textit{if } i = m \\ 3x_m - x_i - x_{m-1} & i < m \end{cases},$$

and $0 \leq x_1 \leq \cdots \leq x_n \leq \Lambda$.

The proof of Lemma 5.2.2 is deferred to the Appendix A.

Element local sensitivity at distance t. Now we proceed to compute $LS^{u_{med}}(x,t,r)$.

$$LS^{u_{med}}(x,t,r) = \max_{y|d(x,y)\leq t} LS^{u_{med}}(y,0,r)$$

Given a distance t, our task is to compute $LS^u(y,0,r)$ over all y such that $d(x,y) \leq t$. A naive brute force approach would be infeasible since there are infinite databases at distance t from x as discussed previously for the computation of element local sensitivity at distance 0.

However, there is a small subset, referred to $candidates(x,t,r)$, of $\{y|d(x,y) \leq t\}$ where we can evaluate $LS^u(y,0,r)$ only on the databases of $candidates(x,t,r)$ to obtain $LS^u(x,t,r)$. We show in our proofs that the databases $\{y|d(x,y) \leq t\} - candidates(x,t,r)$ are safe to discard, i.e., it exists a database $y \in candidates(x,t,r)$ where $LS^u(x,t,r) = LS^u(y,0,r)$. So we rewrite the element local sensitivity of u_{med} as:

Lemma 5.2.3. *(Element local sensitivity at distance t for median selection)*

$$LS^{u_{med}}(x,t,r) = \max_{candidates(x,t,r)} LS^{u_{med}}(y,0,r).$$

The algorithm 2 depicts how to compute $candidates(x,t,r)$. The algorithm $candidates(x,t,r)$ returns a subset of only 6 elements of $\{y|d(x,y) \leq t\}$ that maximizes $\max_{y|d(x,y)\leq t} LS^{u_{med}}(y,0,r)$. The proof of Lemma 5.2.3 is deferred to the Appendix A.

5.3 Experimental Evaluation

Datasets. We tested most of the datasets from Hay *et al.* (2016). Many of them have the local sensitivity near the global sensitivity and some have the local sensitivity far from the global sensitivity, then our approach is beneficial. We report the results for two datasets

Algorithm 2: Candidates Algorithm

```
Procedure Candidates(Dataset $x$, distance $t$, range element $r$)
    if $t = 0$ then
        return $(x)$
    end
    if $t = 1$ then
        Obtain $x'_1$ from $x$ by moving $x_r$ to $\Lambda$
        Obtain $x'_2$ as a copy of $x'_1$
        Obtain $x'_3$ from $x$ by moving $x_r$ to 0
        Obtain $x'_4$ as copy of $x'_4$
        Obtain $x'_5$ from $x$ by moving the median element $x_m$ to $\Lambda$
        Obtain $x'_6$ from $x$ by moving the median element $x_m$ to 0
        return $(x'_1, x'_2, x'_3, x'_4, x'_5, x'_6)$
    end
    $x_1, x_2, x_3, x_4, x_5, x_6 = Candidates(x, t-1, r)$
    Obtain $x'_1$ from $x_1$ by moving the median element $x_m$ to to 0
    Obtain $x'_2$ from $x_2$ by moving the median element $x_m$ to to $\Lambda$
    Obtain $x'_3$ from $x_3$ by moving the median element $x_m$ to to 0
    Obtain $x'_4$ from $x_4$ by moving the median element $x_m$ to to $\Lambda$
    Obtain $x'_5$ from $x_5$ by moving the median element $x_m$ to to 0
    Obtain $x'_6$ from $x_6$ by moving the median element $x_m$ to to $\Lambda$
    return $(x'_1, x'_2, x'_3, x'_4, x'_5, x'_6)$
```

where the local sensitivity is reasonably smaller from the global sensitivity: PATENT dataset and HEPTH dataset. Also we show one dataset where the local sensitivity is very close to the global sensitivity to show our approach behaves on this scenario: INCOME dataset.

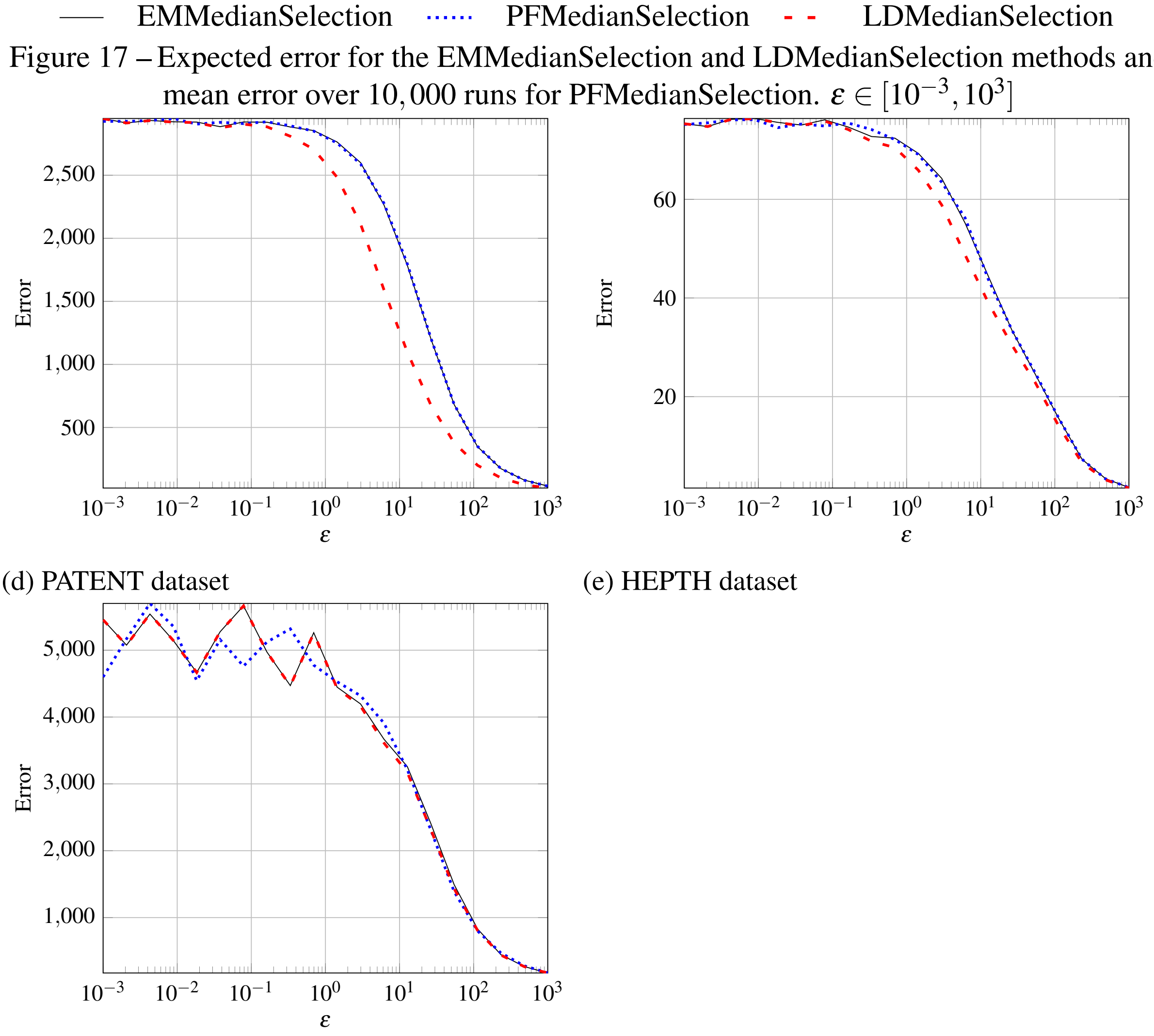
Methods. We test three approaches for the median selection problem: i) EMMedianSelection, the exponential mechanism using global sensitivity; ii) PFMedianSelection, the permute-and-flip mechanism using the global sensitivity and iii) LDMedianSelection, the local dampening mechanism using local sensitivity.

Evaluation. We measure the error: $|retrieved\_median\_value - true\_median\_value|$. For the EMMedianSelection and LDMedianSelection methods, we report the expected error and for the PFMedianSelection we report the mean error over $10,000$ runs. We vary $\varepsilon \in [10^{-3}, 10^{3}]$.

Figure 17 displays the results. For PATENT dataset, we observe a mean reduction of 18% in the error by the LDMedianSelection in relation to both the EMMedianSelection and the PFMedianSelection over all tested values for ε. This reduction is specially noticeable for higher values of ε. For HEPTH dataset, the reduction is at most 12% and the mean 4% for all tested values.

For the INCOME dataset, we show that the LDMedianSelection has the same

Figure 17 – Expected error for the EMMedianSelection and LDMedianSelection methods and mean error over 10,000 runs for PFMedianSelection. $\varepsilon \in [10^{-3}, 10^{3}]$



(d) PATENT dataset

(e) HEPTH dataset

(f) INCOME dataset

Source: elaborated by the author.

accuracy as the EMMedianSelection and the PFMedianSelection in the scenario where the local sensitivity is near to the global sensitivity. This behavior is similar to the other datasets from Hay *et al.* (2016) not presented here.

5.4 Conclusion

In this chapter, we presented the non-numeric version of the median selection problem. This application is a commonly addressed in other works as a example of application to differently private mechanisms since it can be solved by a single call to a non-numeric mechanism.

Our goal is to assess the accuracy of the local dampening mechanism compared to global sensitivity based approaches: the exponential mechanism and permute-and-flip mechanism.

For that, we also calculated the global sensitivity for the use of the exponential mechanism and permute-and-flip mechanism and also the element local sensitivity for the local dampening mechanism.

Experimental results show that the local dampening mechanism outperforms the global sensitivity based approaches on the datasets where the local sensitivity is lower than the global sensitivity.

6 APPLICATION 2: INFLUENTIAL NODE ANALYSIS

Identifying influential nodes in a network is an important task for social network analysis for marketing purposes (MA *et al.*, 2008). This analysis has great value for making a more effective marketing campaign since influential nodes have great capacity to diffuse a message through the network.

This chapter addresses the Influential Node Analysis problem. We present i) the formal problem statement ii) a private mechanism that tackles this problem iii) the sensitivity analysis of the private queries of the mechanism and iv) an experimental comparison to global sensitivity based approaches and also with a related approaches.

6.1 Problem statement

Formally, influential node analysis is a query over an input graph database $G = (V, E)$ that releases the labels of the top k nodes that maximize a given influence metric. The task is to design a private mechanism that answers to the influential node analysis query.

Specifically, we use Egocentric Betweenness Centrality (EBC) (Definition 6.1.1) as an influence measure. EBC allows to identify influential nodes that are important in different loosely connected parties.

Definition 6.1.1. *(Egocentric Betweenness Centrality (EBC) (EVERETT; BORGATTI, 2005; FREEMAN, 1978))*

$$EBC(c) = \sum_{u,v \in N_c | u \neq v} \frac{p_{uv}(c)}{q_{uv}(c)},$$

where $N_c = \{v \in V | \{c, v\} \in E\}$ is the set of neighbors of the central node c, $q_{uv}(c)$ is the number of geodesic paths connecting u and v on the induced subgraph $G[N_c \cup \{c\}]$ and $p_{uv}(c)$ is the number of those paths that include c.

We use edge differential privacy for graph databases where the goal is to protect sensitive information about the edges in G (KASIVISWANATHAN *et al.*, 2013). The graph G is denoted as a vector belonging to $\{0,1\}^{\binom{n}{2}}$ where n is the number of nodes in the input graph and each entry on this vector represents the existence of an edge in G, 1 means that it exists and 0 means otherwise. By Definition 2.2.3 neighboring graphs differ in exactly one edge.

6.2 Private Mechanism

We propose *PrivTopk*, a top-k algorithm template which chooses iteratively k nodes that maximize EBC (Algorithm 3). In each iteration, the algorithm makes a call to a non-numeric mechanism (line 5) that returns a node which maximizes EBC that was not previously chosen.

We experiment with four instances of this algorithm template:

1. *EMPrivTopk*, where we replace line 5 with an exponential mechanism call
2. *PFPrivTopk*, where we replace line 5 with a permute-and-flip mechanism call
3. *LDPrivTopk* where we replace line 5 with a local dampening call
4. *SLDPrivTopk* where we replace line 5 by a shifted local dampening mechanism.

Algorithm 3: PrivTopk

```
1 Procedure PrivTopk(Graph $G = (V, E)$, Privacy Budget $\varepsilon$, Integer $k$)
2     $\varepsilon' = \varepsilon / k$
3     $\Omega = \emptyset$
4     for $j \leftarrow 1$ to $k$ do
5         $v = MEC(G, \varepsilon', EBC, V)$ // Non-numeric mechanism call
6         $\Omega = \Omega \cup \{v\}$
7     end
8     return $\Omega$
```

The privacy correctness of the algorithm follows from the sequential composition property of differential privacy (MCSHERRY; TALWAR, 2007). Our algorithm issues k calls to a private mechanism with privacy budget $\varepsilon' = \varepsilon/k$. By the sequential composition theorem (Theorem 2.5.1) the total privacy budget consumed in the entire algorithm is $\varepsilon' \times k = (\varepsilon/k) \times k = \varepsilon$. Thus Algorithm 3 satisfies ε-differential privacy.

6.3 Sensitivity Analysis

First we need to show a useful lemma that is an intermediate result that helps on the proof of the sensitivity. The proof of lemma 6.3.1 is deferred to the appendix.

Lemma 6.3.1. *Let G and G' be two neighboring graphs and v a node belonging to $V(G)$ and $V(G')$, we have that:*

$$\max_{G,G' | d(G,G') \leq 1} |EBC^{G}(v) - EBC^{G'}(v)| = \max\Big(d^{G}(v)(d^{G}(v)-1)/4, d^{G}(v)\Big),$$

where $d^{G}(v)$ denotes the degree of v in G, i.e., $d^{G}(v) = |N_v^{G}|$.

Global Sensitivity. We provide the global sensitivity for EBC to the exponential mechanism and to the permute-and-flip mechanism:

Lemma 6.3.2. *(EBC global sensitivity) The global sensitivity ΔEBC for EBC is given by*

$$\Delta EBC = \max\left(\frac{\Delta(G)(\Delta(G)-1)}{4}, \Delta(G)\right),$$

where $\Delta(G)$ is the maximum degree of the input graph G. In this work, we assume the maximum degree is public information or that we have an upper bound for it.

Lemma 6.3.2 is a direct consequence of Lemma 6.3.1 given that $\Delta(G)$ is the degree of the node with largest degree.

Element local sensitivity. For the local dampening call, we provide an upper bound to the element local sensitivity using the admissible sensitivity function δ^{EBC}:

Definition 6.3.1. *(Sensitivity function $\delta^{EBC}(G,t,v)$). The sensitivity function δ^{EBC} for EBC is defined as*

$$\delta^{EBC}(G,t,v) = \max\left(\frac{(d^G(v)+t)(d^G(v)+t-1)}{4}, d^G(v)+t\right).$$

We also show that $\delta^{EBC}(G,t,v)$ is admissible (Lemma 6.3.3). Note that δ^{EBC} is not naturally bounded, however, we use the method described on Section 3.2.2 to transform it in a bounded function.

Lemma 6.3.3. *The sensitivity function $\delta^{EBC}(G,t,v)$ is an admissible sensitivity function.*

Proof. We show that the sensitivity function $\delta^{EBC}(G,t,v)$ is admissible. First, we show that $\delta^{EBC}(G,0,v) = (d(d-1)/4, d) \geq LS^{EBC}(G,0,v)$ where d is the degree of v. Lemma 6.3.1 proves that for every pair of neighboring graphs G', G^*, $|EBC^{G'}(v) - EBC^{G^*}(v)| = \max(d(d-1)/4, d)$. By fixing G, we obtain that

$$\begin{aligned} LS^{EBC}(G,0,v) &= \max_{G', d(G,G') \leq 1} |EBC^{G}(v) - EBC^{G'}(v)| \\ &\leq \max\left(d^G(v)(d^G(v)-1)/4, d^G(v)\right) = \delta^{EBC}(G,0,v) \end{aligned}$$

It remains to demonstrate that $\delta^{EBC}(G,t,v) \leq \delta^{EBC}(G',t+1,v)$ for all neighboring graphs G' of G. We first show that $|d^G - d^{G'}| \leq 1$ where d^G and $d^{G'}$ are the degree of v in G and G' respectively G and G' differ in just one edge, say e. Suppose e belongs to G and not to G', if e is not incident on v in G then $|d^G - d^{G'}| = 0$. Otherwise, if e is incident on v in G then $d^G - d^{G'} = 1$. So $d^G - d^{G'} \leq 1$. By symmetry, $d^{G'} - d^G \leq 0$ holds. Then $|d^G - d^{G'}| \leq 1$

Applying this last fact to $\delta^{EBC}(G,t,v)$ we have:

$$\begin{aligned}
\delta^{EBC}(G,t,v) &= \max\left(\frac{(d^G+t)(d^G+t-1)}{4}, d^G+t\right) \\
&\leq \max\left(\frac{(d^{G'}+t+1)(d^G+t)}{4}, d^G+t+1\right) \\
&= \delta^{EBC}(G',t+1,v)
\end{aligned}$$

□

In terms of correlation between $EBC(v)$ and $deg^G(v)+t$, note that for a given node v with degree $d^G(v)$, there are $\binom{d^G(v)}{2} = (deg^G(v)\cdot(deg^G(v)-1))/2$ terms in the EBC equation for v (Definition 6.1.1), i.e., pairs $(u,z) \in N_v^G$. As each term contributes at most 1 to EBC, it suggests that there is a correlation between $EBC(v)$ and $deg^G(v)+t$ and consequently, between $EBC(v)$ and $\delta^{EBC}(G,t,v)$. Empirical observation of the datasets confirmed that correlation. For this reason, the shifted local dampening mechanism call in SLDPrivTopk is suitable.

6.4 Related Work

The literature provides some work to release statistics on graphs which are presented in this section. Also, we present some work on releasing linear statistics over relational databases that can be used to release EBC and compare experimentally to our approach.

6.4.1 Differentially Private Graph Analysis

Kasiviswanathan *et al.* (2013) observed that the sensitivity of many graph problems is a function of the maximum degree of the input graph G, so they proposed a generic projection that truncates the maximum degree of G. This projection is built upon smooth sensitivity framework (NISSIM *et al.*, 2007) but the target query is answered using the global sensitivity on the truncated graphs which may still be high. Moreover, it satisfies a weaker definition of privacy: (ε, δ)-differential privacy.

There is a number of works that aims to publish subgraph counting as k-triangles, k-stars and k-cliques. Kasiviswanathan *et al.* (2013) also addresses this problem applying a linear programming-based approach that release those counts for graphs that satisfies α-decay.

A new notion of sensitivity called restricted sensitivity was introduced by Blocki *et al.* (2013) to answer subgraph counts. In this setting, the querier may have some belief about the structure of the input graph, so the restricted sensitivity measures sensitivity only on the subset of graphs which are believed to be inputs to the algorithm. However, this work satisfies only (ε, δ)-differential privacy.

Blocki *et al.* (2013) introduced the *ladder functions* to answer to subgraph counts. A latter function is a structure built upon the local sensitivity of the subgraph count. It rank the possible outputs of the subgraph count query and sample a given output using the exponential mechanism with low sensitivity.

The work presented in (KARWA *et al.*, 2011) is directed application of the smooth sensitivity framework (NISSIM *et al.*, 2007) also for answering to subgraph count queries. The authors provide the bounds of the local sensitivity of k-triangles and k-stars and show that they are more accurate than related work.

About centrality metrics. A recent work Laeuchli *et al.* (2021) analyzes three centrality measures on graphs: eigenvector, laplacian and closeness centrality. The result is that releasing those metrics using either the laplace mechanism (based on global sensitivity) or the smooth sensitivity framework (based on local sensitivity) is infeasible. To show that, they demonstrate that the local sensitivity is unbounded or, even it is bounded, it is too large and it results in overwhelming addition of noise.

To the best of out knowledge, in the literature, none of the works on graph analysis tackles top-k or EBC release.

6.4.2 Releasing Linear Statistics over Relational Databases

A body of work is available in the literature on answer linear queries, i.e. queries that can be answered by linear aggregation, over relational databases using SQL unde differential privacy.

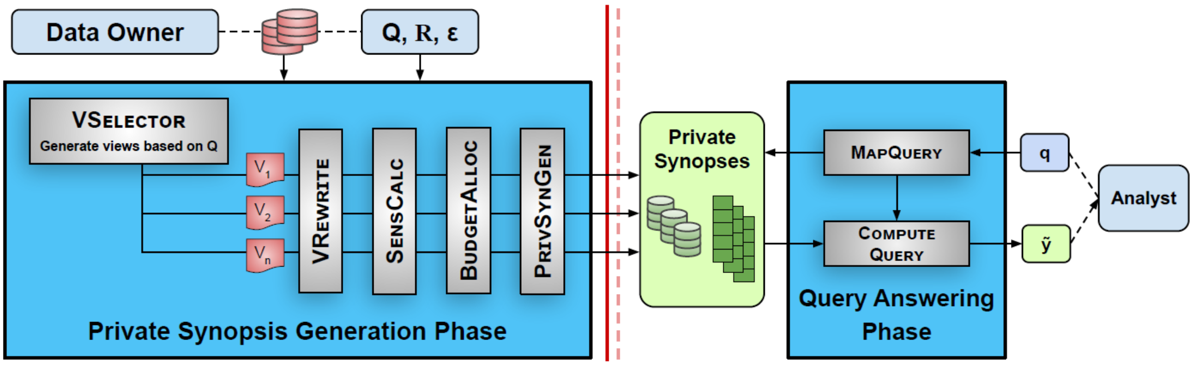
As will be shown in Section 6.5.1, the EBC metric can be computed by issuing a set of count SQL queries with cyclic joins and GROUP BY clauses over a graph stored in relational database. Thus we survey works that can possibly answer to this kind of query.

Local sensitivity has been used for answering full acyclic join queries (TAO *et al.*, 2020). This approach lacks generality since it cannot compute SQL queries with cyclic joins and GROUP BY clauses which it is not the case for EBC queries. The recursive mechanism (CHEN; ZHOU, 2013) can answer linear queries with unrestricted joins with GROUP BY clauses, however it requires the target function f to be monotonic, i.e., inserting a new individual in the database always causes f to increase (or always decrease). This monotonicity condition is not satisfied by EBC.

6.4.2.1 Privatesql

PrivateSQL (KOTSOGIANNIS *et al.*, 2019) is an approach that can answer linear queries with cyclic joins and correlated subqueries with GROUPBY clauses. The architecture of PrivateSQL is displayed in Figure 19.

Figure 19 – PrivateSQL's System Architecture (KOTSOGIANNIS *et al.*, 2019).



Source: elaborated by Kotsogiannis *et al.* (2019).

This approach requires a representative workload Q as input, a primary relation in the database. The representative workload is used in the VSelector module to identify a set of views over the base relations that support the analyst queries and then generates a set of views that can answer all the the analyst's queries. The VRewrite module rewrite using truncation operators and semijoin operators to bound the sensitivity. Therefore, the sensitivity for each query is computed in the SensCalc module and a fraction of the budget is given to each query in the BudgetAllocator module. The PrivSynGen generates a synopsis for each view. A synopsis is a compact representation of a view that capture important information from it and can approximate the answer of a given query on the view. The private synopses are released under differential privacy and queries can be executed over them without using privacy budget. The synopsis generation is based on non-negative least squares inference (LI *et al.*, 2015). The sensitivity computation for each view is based on Flex (JOHNSON *et al.*, 2018) augmented with truncation operators.

Then when a new query is issued to the system, this query is rewritten as linear queries over the private views' synopsis in the MapQuery module. Since graph databases can be modeled as a table Node(id) and a table Edge(source, target), we carry out an experimental evaluation in Section 6.5.1.

6.5 Experimental Evaluation

Datasets. We use three real-world graph datasets: 1) *Enron* is a network of email communication obtained from around half million emails. Each node is an email address and an edge connects a pair of email addresses that exchanges emails ($|V| = 36,692$, $|E| = 183,831$ and $\Delta(G) = 1,383$); 2) *DBLP* is a co-authorship network where two authors (nodes) are connected if they published at least one paper together ($|V| = 317{,}080$, $|E| = 1,049,866$ and $\Delta(G) = 343$); 3) *Github* is a network of developers with at least 10 stars on the platform. Developers are represented as nodes and an edge indicates that two developers follow each other ($|V| = 37,700$, $|E| = 289{,}003$ and $\Delta(G) = 9{,}458$). V is the set of vertices, E is the set of edges and $\Delta(G)$ is the maximum degree of a graph G. All datasets can be found on Stanford Network Dataset Collection (LESKOVEC; KREVL, 2014).

Methods. We compare the four versions of PrivTopk (algorithm 3): 1) EMPrivTopk, using the exponential mechanism, 2) PFPrivTopk, using permute-and-flip mechanism, 3) LDPrivTopk using local dampening mechanism and 4) SLDPrivTopk using shifted local dampening mechanism.

Evaluation. We evaluate the accuracy by the percentage of common nodes to the retrieved top-k set and the true top-k set, i.e., $(|\text{retrieved\_topk} \cap \text{true\_topk}|)/k$. We report the mean accuracy in 100 simulations. We set $k \in \{5, 10, 20\}$ and a range for privacy budget $B \in [10^{-3}, 10^4]$.

Figure 20 displays the results. We first note that the global sensitivity based approaches, EMPrivTopk and PFPrivTopk, exhibit low accuracy for low values of B. This is due to the high global sensitivity: $\Delta EBC = 22,361,076.5$ for github dataset, $\Delta EBC = 477,826.5$ for DBLP dataset and $\Delta EBC = 29,326.5$ for Enron dataset. Also, the LDPrivTopk algorithm suffers from the inversion problem (Section 4.1) while SLDPrivTopk could exploit the correlation between EBC and δ^{EBC} to fix this problem.

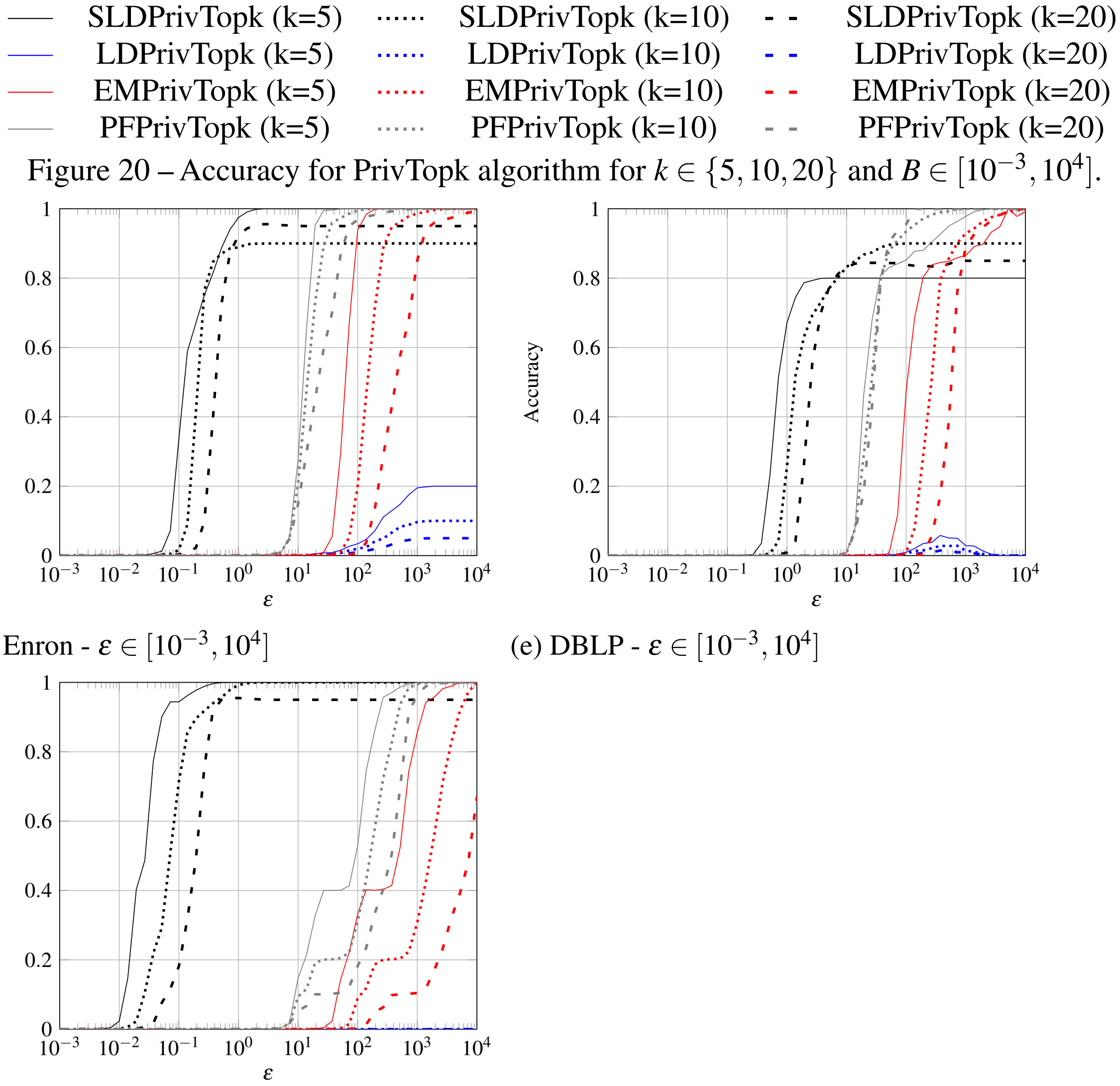


Figure 20 – Accuracy for PrivTopk algorithm for $k \in \{5, 10, 20\}$ and $B \in [10^{-3}, 10^{4}]$.

(d) Enron - $\varepsilon \in [10^{-3}, 10^{4}]$

(e) DBLP - $\varepsilon \in [10^{-3}, 10^{4}]$

(f) Github - $\varepsilon \in [10^{-3}, 10^{4}]$

Source: elaborated by the author.

We observe a clear pattern where the methods perform worse as k grows. This is explained by the fact that each call to the non-numeric mechanism uses B/k of the total privacy budget B (Algorithm 3). Thus, larger k implies that less of the privacy budget is used in each non-numeric mechanism call which hurts accuracy.

For SLDPrivTopk, we note that we need different level of privacy budget for each dataset reasonable accuracy. This is explained by a number of factors. For Github dataset, the distribution of EBC is heavy tailed thus the nodes with high EBC have a higher probability to be correctly picked with low privacy budget. On the other hand, for the DBLP dataset we need need more privacy budget as it has roughly 10 times more nodes than the other datasets, which dilute the probability of the nodes with higher EBC.

Our approach SLDPrivTopk achieves the same level of accuracy with privacy values 3 to 4 orders of magnitude less than EMPrivTopk and 2 to 4 orders of magnitude less than PFPrivTopk.

6.5.1 Comparison to PrivateSQL

We carry out an experimental comparison of local dampening mechanism to the PrivateSQL on Influential Node Analysis problem. For this application, PrivateSQL is not particularly scalable (as discussed further) so we performed the experiments with smaller datasets and test with fewer values for the privacy budgets.

PrivateSQL addresses the Influential Node Analysis problem by computing the counts $q_{uv}(c)$ and $p_{uv}(c)$ for all $u,v \in N_c$ (Definition 11) to compute EBC and takes the top-k nodes with highest EBC score. Note that it considers only for the terms $p_{uv}(c)/q_{uv}(c)$ where the distance from u to v in $G[N_c \cup \{c\}]$ is 2 which is the maximum possible distance as $u,v \in N_c$. If their distance is 1 (i.e. u and v are neighbors), the term $p_{uv}(c)$ is 0 since the geodesic path of length 1 from u to v ($u,v \neq c$) cannot contain c. Thus, for PrivateSQL, we pose private queries $\mathscr{Q}(u,v,c)$ (see Appendix B) that returns 1) 0 if u and v are neighbors, 2) 0 if the distance from u to v is larger than 2 and 3) $q_{uv}(c)$, otherwise.

Therefore, when $\mathscr{Q}(u,v,c)$ is not equal to 0, it means that the term $p_{uv}(c)/q_{uv}(c)$ should be accounted in the EBC definition. In that case, we obtain a noisy estimate for $q_{uv}(c)$ from $\mathscr{Q}(u,v,c)$. A noisy estimate for $p_{uv}(c)$ can be derived from noisy $q_{uv}(c)$ by setting p_{uv} to 0 if $q_{uv}(c) = 0$ or to 1 if $q_{uv}(c) > 0$. The rationale is that exactly one of the paths of length 2 from u to v contains c as $u,v \in N_c$.

The set N_c is itself private information. Hence, to compute $EBC(c)$ for every $c \in V(G)$ we need to compute $\mathscr{Q}(u,v,c)$ for every $u,v \in V(G)$. This results in a total number of $O(n^3)$ queries which poses a scalability problem. For this reason, we perform experiments with samples S of the graphs which are obtained by choosing a node sample S_n in breadth-first search fashion with a random seed node and then we set $S = G[S_n]$.

Table 1 displays the mean accuracy for 10 runs on 10 sample graphs with $|S_n| =$ 50 nodes with $k \in \{1,2,3\}$ for each $B \in \{0.1, 0.5, 1.0, 5.0, 10.0\}$. We compare the best local dampening based algorithm SLDPrivTopk (PTK) with the PrivateSQL based approach (PSQL).

PrivateSQL approach generated one private view for each node in the graph. Thus, the privacy budget needs to be divided by the number of nodes n which implies that accuracy is hurt as n grows. Moreover, the sensitivity for each view is high, e.g, sensitivity is 1448 when $\Delta(G) = 10$. This entails in a poor performance for the PrivateSQL based approach.

Table 1 – Mean accuracy for SLDPrivTopk (PTK) and PrivateSQL (PSQL) over 10 runs on 10 sample graphs with 50 nodes with $k \in \{1,2,3\}$ for each $B \in \{0.1, 0.5, 1.0, 5.0, 10.0\}$.

	Enron		DBLP		Github	
B	PTK	PSQL	PTK	PSQL	PTK	PSQL
0.1	0.06	0.01	0.05	0.02	0.07	0.02
0.5	0.45	0.01	0.27	0.02	0.49	0.02
1.0	0.60	0.16	0.44	0.05	0.69	0.02
5.0	0.84	0.20	0.87	0.11	0.86	0.03
10.0	0.88	0.21	0.92	0.21	0.91	0.07

Source: elaborated by the author

6.6 Conclusion

In the chapter, we defined the Influential Node Analysis application where the goal is to release the top-k most influential nodes of a given graph according to a influence metric. In this work, we use EBC centrality as influence metric.

Then we have introduced a template of a private mechanism that releases the top-k influential nodes. In this template we can apply any non-numeric mechanism. We applied the exponential mechanism, the permute-and-flip mechanism, the local dampening mechanism and the shifted local dampening mechanism. To apply those mechanisms we provided the global sensitivity and upper on the local sensitivity of the EBC metric and showed to be admissible.

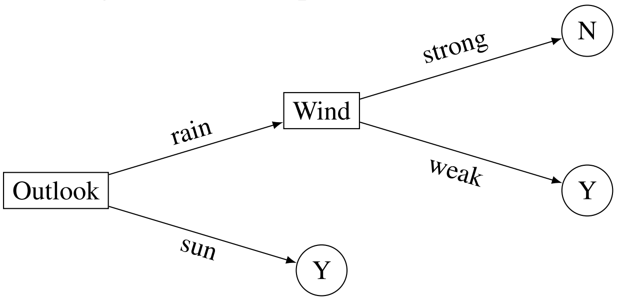
In our experimental evaluation, we first compare the proposed mechanism with the exponential mechanism, the permute-and-flip mechanism, the local dampening mechanism and the shifted local dampening mechanism. The results show that the shifted local dampening mechanism significantly reduces the use of privacy budget by 2 to 4 with the same level of accuracy compared to the global sensitivity based approaches (exponential mechanism and permute-and-flip mechanism).

After this, we compared the local dampening mechanism to the PrivateSQL approach. PrivateSQL had poor performance since it needs to generate a view for each node. Then the privacy budget used to construct each view is divided by the number of nodes.

7 APPLICATION 3: ID3 DECISION TREE INDUCTION

Classification based on decision tree is an important tool for data mining (KOTSIANTIS *et al.*, 2007). Specifically, decision trees are a set of rules that are applied to the input attributes to decide to which class a given instance belongs. Figure 22 shows an example of a decision (Y or N) that is taken by checking the attributes Outlook and Wind of the input row.

Figure 22 – Example of Decision Tree.



Source: elaborated by the author.

Creating a decision tree manually is a burden. Thus many approaches for automatically building decision trees were proposed. One of the most known tree induction algorithms is the ID3 algorithm (QUINLAN, 1986). A tree induction algorithm receives a dataset and outputs a decision tree. Table 2 shows an example of a dataset regarding weather.

Table 2 – Sample dataset.

Outlook	**Wind**	**Decision**
rain	strong	Y
rain	weak	N
sun	strong	Y
sun	weak	N
sun	strong	N

Source: elaborated by the author

The ID3 algorithm (QUINLAN, 1986) starts with the root node containing the original set. Then the algorithm greedly chooses an unused attribute to split the set and generate child nodes. The selection criterion is Information Gain (IG), given by the entropy before splitting minus the entropy after splitting. It expresses how much entropy was gained after the split. This process continues recursively for the child node until splitting does not reduce entropy or the maximum depth is reached.

7.1 Problem Statement

A decision tree induction algorithm takes as input a dataset $\mathscr{T}$ with attributes $\mathscr{A} = \{A_1, \ldots, A_d\}$ 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 in one of the most known tree induction algorithms, the ID3 algorithm.

The notation for this chapter is summarized in Table 3. All logarithms are in base 2. When it is clear from the context, we drop the superscript $\mathscr{T}$ from the notations.

Table 3 – Notation table for private decision tree induction

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

Source: elaborated by the author

7.2 Related Work

Blum et al (BLUM *et al.*, 2005) presented the SuLQ framework that provides primitives for data mining algorithms. As an application, they introduced a differentially private adaptation for ID3 algorithm where it computes the information gain based on the noisy counts provided by SuLQ. This approach applies two operators from SuLQ: 1) **NoisyCount** that uses Laplace mechanism to return private estimate of a count query, and 2) **Partition** that splits the dataset in disjoint subsets so that the privacy budgets for the queries over each subset do not sum up (parallel composition, Theorem 2.5.2) meaning that we can make a more efficient use of the privacy budget. A major drawback of this algorithm is that it requires to query the noisy counts (via **NoisyCounts**) for each attribute, so the privacy budget needs to be split to those queries.

This entails in a small budget per query and in a larger noise magnitude.

To overcome this, Friedman and Schuster (2010) introduced a variation of the SuLQ algorithm (Algorithm 4). This algorithm replaces the many **NoisyCount** calls for a single call to the exponential mechanism. Line 12 (Algorithm 4) is the call for the exponential mechanism that was previously several calls to **NoisyCount**.

The procedure *Build_DiffPID*3 in algorithm 4 starts by checking for the construction of a leaf node (Line 5-7). It verifies if there the dataset has any remaining attribute and if there is enough instances to make new splits. In lines 8-10, it partitions the data based on the class attribute C, it privately queries the count of each partition and create a leaf with the class label of the largest partition. The Lines 12-15 creates a new decision rule recursively. It starts by privately choosing the attribute with largest IG using the exponential mechanism then it partition the dataset using the chosen attribute and call *Build_DiffPID*3 with each partition to create the sub-trees.

Algorithm 4: GlobalDiffPID3

```
1  Procedure GlobalDiffPID3(Dataset $\mathscr{T}$, Attribute Set $\mathscr{A}$, Class
     attribute $C$, Depth $d$, Privacy Budget $B$)
2      $\varepsilon = B/(2(d+1))$
3      return Build_DiffPID3($\mathscr{T}$, $\mathscr{A}$, $C$, $d$, $\varepsilon$)
4  Procedure Build_DiffPID3($\mathscr{T}$, $\mathscr{A}$, $C$, $d$, $\varepsilon$)
5      $t = \max_{A \in \mathscr{A}} |A|$
6      $N_{\mathscr{T}} = \text{NoisyCount}_{\varepsilon}(\mathscr{T})$
7      if $\mathscr{A} = \emptyset$ or $d = 0$ or $\frac{N_{\mathscr{T}}}{t|C|} < \frac{\sqrt{2}}{2}$ then
8          $\mathscr{T}_c = \text{Partition}(\mathscr{T}, \forall c \in C : r_c = c)$
9          $\forall c \in C : N_c = \text{NoisyCount}_{\varepsilon}(\mathscr{T}_c)$
10         return a leaf labeled with $\arg\max_c(N_c)$
11     end
12     $\bar{A} = \mathscr{E}(\mathscr{T}, \varepsilon, IG, \mathscr{A})$ // Exp.  mechanism call
13     $\mathscr{T}_i = \text{Partition}(\mathscr{T}, \forall i \in \bar{A} : r_{\bar{A}} = i)$
14     $\forall i \in \bar{A} : \text{Subtree}_i = \text{Build\_DiffPID3}(\mathscr{T}_i, \mathscr{A} \setminus \bar{A}, C, d-1, \varepsilon)$
15     return a tree with a root node labeled $\bar{A}$ and edges labeled 1 to $\bar{A}$ each going to
         $\text{Subtree}_i$
```

To the best of our knowledge, our work is the first to apply local sensitivity to single trees which was an open question pointed out in a recent survey on private decision trees (FLETCHER; ISLAM, 2019).

Many other works address the private construction of random forests (FLETCHER; ISLAM, 2015a; FLETCHER; ISLAM, 2015b; FLETCHER; ISLAM, 2017; JAGANNATHAN *et al.*, 2009; PATIL; SINGH, 2014; RANA *et al.*, 2015). Interestingly, local sensitivity was used for building random forests (FLETCHER; ISLAM, 2015b; FLETCHER; ISLAM, 2017) using smooth sensitivity. This shows a promising future direction of our work which is applying local dampening to construct random forests.

7.3 Private Mechanism

We use the algorithm GlobalDiffPID3 (FRIEDMAN; SCHUSTER, 2010) (Algorithm 4) as a template. We aim to adapt it for the use of the local dampening mechanism and to the shifted local dampening producing the *LocalDiffPID3* and *ShiftedLocalDiffPID3*, respectively. In the following, we make a discussion about the split criterion, the global sensitivity of the split criterion for the exponential mechanism and the element local sensitivity for the local dampening.

Split criterion. In this work, we address one of the most traditional split criterion, *information gain* (IG). It is given by the entropy of the class attribute C in $\mathscr{T}$ minus the obtained entropy of C splitting the tuples according to an attribute $A \in \mathscr{A}$.

$$IG(\mathscr{T},A) = H_C(\mathscr{T}) - H_{C|A}(\mathscr{T}),$$

where $H_C(\mathscr{T})$ is entropy with respect to the classe attribute C

$$H_C(\mathscr{T}) = -\sum_{c \in C} \frac{\tau_c}{\tau} \log \frac{\tau_c}{\tau},$$

and $H_{C|A}(\mathscr{T})$ is the entropy obtained by splitting the instances according to attribute A

$$H_{C|A}(\mathscr{T}) = \sum_{j \in A} \frac{\tau_j^A}{\tau} . H_C(\mathscr{T}_j^A).$$

Since $H_C(\mathscr{T})$ does not depend on A, we can further simplify the utility function IG:

$$\begin{aligned} IG(\mathscr{T},A) &= -\tau . H_{C|A}(\mathscr{T}) && (7.1) \\ &= -\sum_{j \in A} \sum_{c \in C} \tau_{j,c}^A . \log(\frac{\tau_{j,c}^A}{\tau_j^A}). && (7.2) \end{aligned}$$

Global sensitivity. The exponential mechanism requires the computation of the global sensitivity for IG. It is given by $\Delta IG = \log(N+1) + 1/\ln 2$ (FRIEDMAN; SCHUSTER,

2010) where N is the size of the dataset $\mathscr{T}$. The global sensitivity case can be achieved by $\mathscr{T}$ and $\mathscr{T}'$ where:

1. $\mathscr{T}$ has all tuples with values for A equal to a single value $j \in A$ and all tuples class attribute C are set to a value different from a given value $c \in C$ (i.e. $\tau_j^A = \tau$ and $\tau_{j,c}^A = 0$);
2. $\mathscr{T}'$ is obtained from $\mathscr{T}$ by adding a tuple r where $r_A = j$ and $r_C = c$.

Element local sensitivity. In our experiments, we observed that this mentioned case for the global sensitivity is not frequent in real datasets. For those datasets, a local measurement of the sensitivity can be about one order of magnitude lower than the global sensitivity. To this matter, we replace line 12 of algorithm 4 for a local dampening mechanism call producing the algorithm LocalDiffPID3.

Element Local Sensitivity at distance 0. To use local dampening mechanism, we provide means to efficiently compute the element local sensitivity for IG (Lemma 7.3.2). The element local sensitivity at distance t measures $LS^{IG}(\mathscr{T}',0,A)$ for all datasets $\mathscr{T}'$ such that $d(\mathscr{T},\mathscr{T}') \leq t$. We first show how to obtain $LS^{IG}(\mathscr{T}',0,A)$:

Lemma 7.3.1. *(Element local sensitivity at distance 0 for IG). Given a dataset $\mathscr{T}$ and the attribute set A, $LS^{IG}(\mathscr{T},0,A)$ produces the element local sensitivity for IG at distance* 0*:*

$$LS^{IG}(\mathscr{T},0,A) = \max_{j \in A, c \in C} h(\tau_j^{A,\mathscr{T}}, \tau_{j,c}^{A}\mathscr{T}),$$

where

$$h(a,b) = \max(f(a) - f(b), g(b) - g(a)),$$
$$g(x) = x.log((x-1)/x) - log(x-1),$$
$$f(x) = x.log((x+1)/x) + log(x+1).$$

Assume that $g(x) = 0$ for $x \leq 1$ and $f(x) = 0$ for $x \leq 0$. Note that, the expression $g(\tau_{j,c}^{A,\mathscr{T}}) - g(\tau_j^{A,\mathscr{T}})$ measures the impact of the removal of a tuple r such that $r_A = j$ and $r_C = c$ and the expression $f(\tau_j^{A,\mathscr{T}}) - f(\tau_{j,c}^{A,\mathscr{T}})$ measures the addition of tuple r. Thus to obtain $LS^{IG}(\mathscr{T},0,A)$, we need to measure, for all $j \in A$ and $c \in C$, the addition or removal of the tuple r where $r_A = j$ and $r_C = c$, i.e. $h(\tau_j^{A,\mathscr{T}}, \tau_{j,c}^{A,\mathscr{T}}) = \max(f(\tau_j^{A,\mathscr{T}}) - f(\tau_{j,c}^{A,\mathscr{T}}), g(\tau_{j,c}^{A,\mathscr{T}}) - g(\tau_j^{A,\mathscr{T}}))$.

Element local sensitivity at distance t. We use a similar idea to compute $LS^{IG}(\mathscr{T},t,A)$. $LS^{IG}(\mathscr{T},t,A)$ searches for the largest $LS^{u}(\mathscr{T}',0,A)$ over all datasets $\mathscr{T}'$ where $d(\mathscr{T},\mathscr{T}') \leq t$:

$$\begin{aligned} LS^{IG}(\mathscr{T},t,A) &= \max_{\mathscr{T}'|d(\mathscr{T},\mathscr{T}')\leq t} LS^{u}(\mathscr{T}',0,A) \\ &= \max_{c\in C, j\in A} \max_{\mathscr{T}'|d(\mathscr{T},\mathscr{T}')\leq t} h(\tau_j^{A,\mathscr{T}'},\tau_{j,c}^{A,\mathscr{T}'}). \end{aligned}$$

Exhaustively iterating over all $\mathscr{T}'$ to compute $h(\tau_j^{A,\mathscr{T}'},\tau_{j,c}^{A,\mathscr{T}'})$ is not feasible since the number of datasets $\mathscr{T}'$ grows exponentially with respect to t. However, we can restrict the number of evaluations of h by discarding some of the datasets $\mathscr{T}'$.

To this end, we introduce the algorithm *Candidates*$(\mathscr{T},t,j,c)$ (Algorithm 5) that produces a subset of the set of the pairs $(\tau_j^{A,\mathscr{T}'}, \tau_{j,c}^{A,\mathscr{T}'})$ of all datasets $\mathscr{T}'$ such that $d(\mathscr{T},\mathscr{T}')=t$, i.e., *Candidates*$(\mathscr{T},t,j,c) \subseteq \{(\tau_j^{A,\mathscr{T}'}, \tau_{j,c}^{A,\mathscr{T}'}) \mid d(\mathscr{T},\mathscr{T}')=t\}$.

Algorithm 5: Candidates Algorithm

```
1  Procedure Candidates(Dataset $\mathscr{T}$, distance $t$, attribute value $j$, class
   attribute value $c$)
2      if $t = 0$ then
3          return $\{(\tau_j^A, \tau_{j,c}^A)\}$
4      end
5      $candidates = \emptyset$
6      for each pair $(a,b) \in$Candidates($\mathscr{T}$,$t-1,j,c$) do
7          if $a > 0$ and $b > 0$ then
8              $candidates = candidates \cup \{(a-1,b-1)\}$
9          end
10         if $a < \tau$ then
11             $candidates = candidates \cup \{(a+1,b)\}$
12         end
13     end
14     return $candidates$
```

The Candidates algorithm has two important properties:

1. *Candidates*$(\mathscr{T},t,j,c)$ contains the pair $(\tau_j^{A,\mathscr{T}'}, \tau_{j,c}^{A,\mathscr{T}'})$ such that $h(\tau_j^{A,\mathscr{T}'}, \tau_{j,c}^{A,\mathscr{T}'})$ is maximum, i.e., $h(\tau_j^{A,\mathscr{T}'}, \tau_{j,c}^{A,\mathscr{T}'}) = \max_{\mathscr{T}'|d(\mathscr{T},\mathscr{T}')=t} h(\tau_j^{A,\mathscr{T}'}, \tau_{j,c}^{A,\mathscr{T}'})$ (Lemma 7.3.2);
2. It is cacheable, when computing $D_{IG,\delta^{IG}}$, we evaluate $LS^{IG}(\mathscr{T}',t,A)$ several times in increasing order of t then one can cache calls to *Candidates*$(\mathscr{T},t-1,j,c)$ to execute *Candidates*$(\mathscr{T},t,j,c)$ (line 6) efficiently.

Thus $LS^{IG}(\mathscr{T},t,A)$ is given by:

Lemma 7.3.2. *(Element local sensitivity at distance t for IG) Given an input table $\mathscr{T}$, a distance t and an attribute set A, $LS^{IG}(\mathscr{T},t,A)$ produces the element local sensitivity at distance t for IG.*

$$LS^{IG}(\mathscr{T},t,A) = \max_{\substack{j\in A, c\in C,\\ 0\le t'\le t}} \max_{(a,b)\in Candidates(\mathscr{T},t',j,c)} h(a,b).$$

In turn, the computation of $LS^{IG}(\mathscr{T},t,A)$ is also cacheable. One can store a previous call to $LS^{IG}(\mathscr{T},t-1,A)$ to compute $LS^{IG}(\mathscr{T},t,A)$ as:

$$LS^{IG}(\mathscr{T},t,A) = \max\left(\max_{\substack{j\in A, c\in C,\\ (a,b)\in Candidates(\mathscr{T},t,j,c)}} h(a,b), \qquad LS^{IG}(\mathscr{T},t-1,A) \right).$$

In the datasets used in our experiments, LS^{IG} and IG shows correlation. Consequently, we also replace the exponential mechanism call on line 12 in algorithm 4 by a call to the Shifted local dampening with LS^{IG}, which produces the ShiftedLocalDiffPID3.

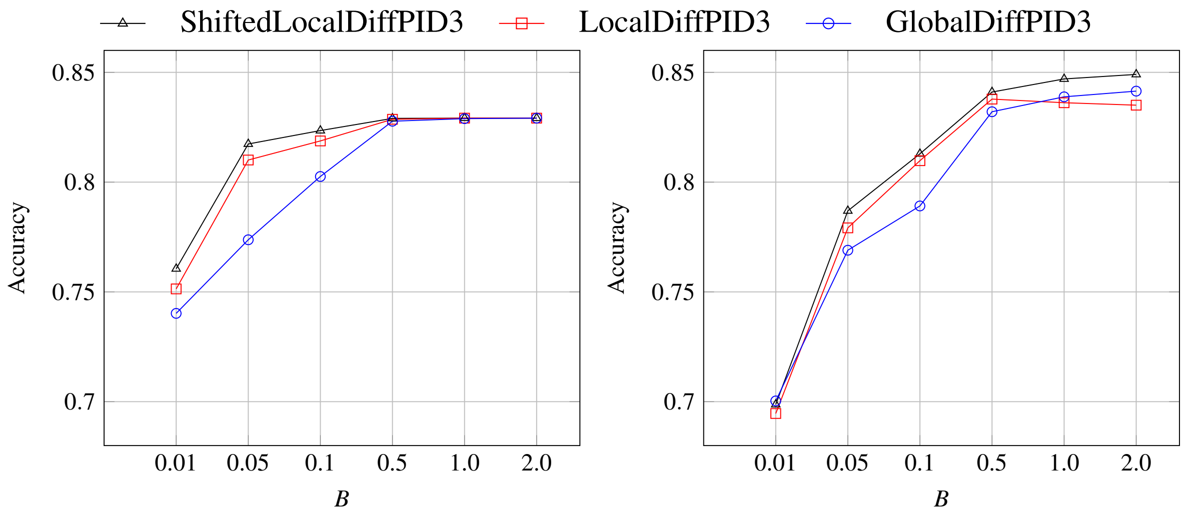
Continuous attributes support. An important feature introduced in C4.5 algorithm (SALZBERG, 1993) is the support for continuous attributes. To support continuous attributes, we use a simpler approach that performed well in our experiments and it is also used in (FRIEDMAN; SCHUSTER, 2010). We discretize the continuous attributes in b evenly spaced bins on the dataset and use them as discrete attributes.

7.4 Experimental Evaluation

Datasets. We use of three tabular datasets: 1) *National Long Term Care Survey (NLTCS)* (MANTON, 2010) is a dataset that contains 16 binary attributes of $21,574$ individuals that participated in the survey, 2) *American Community Surveys (ACS)* dataset (SERIES, 2015) includes the information of $47,461$ rows with 23 binary attributes obtained from 2013 and 2014 ACS sample sets in IPUMS-USA and 3) *Adult* dataset (BLAKE; MERZ, 1998) contains $45,222$ records (excluding records with missing values) with 12 attributes where 8 are discrete and 4 are continuous.

Methods. We compare the three versions of the DiffPID3 (algorithm 4): 1) *GlobalDiffPID3* using exponential mechanism, 2) *LocalDiffPID3* using local dampening mechanism and 3) *ShiftedLocalDiffPID3* using local dampening mechanism with shifting.

Evaluation. We evaluate the accuracy of the approach by reporting the mean accuracy across the 10 runs of a 10-fold cross validation. We set $depth \in \{2,5\}$ and $\varepsilon \in \{0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$.



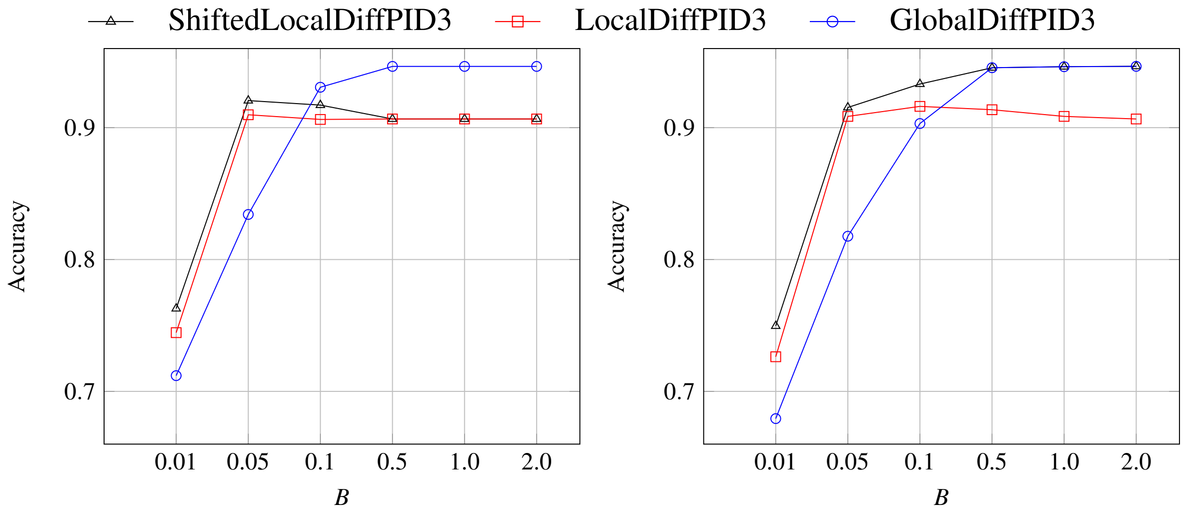
(c) Depth=2

(d) Depth=5

Source: elaborated by the author.

Figure 23 – NLTCS dataset

Figure 23 presents the results for NLCTS dataset. We observe that the LocalDiffPID3 improves on the GlobalDiffPID3 in almost every privacy budget value, up to 5%. While ShiftedLocalDiffPID3 improves a little more in relation to the LocalDiffPID3, up to 1%.



(c) ACS dataset - Depth=2

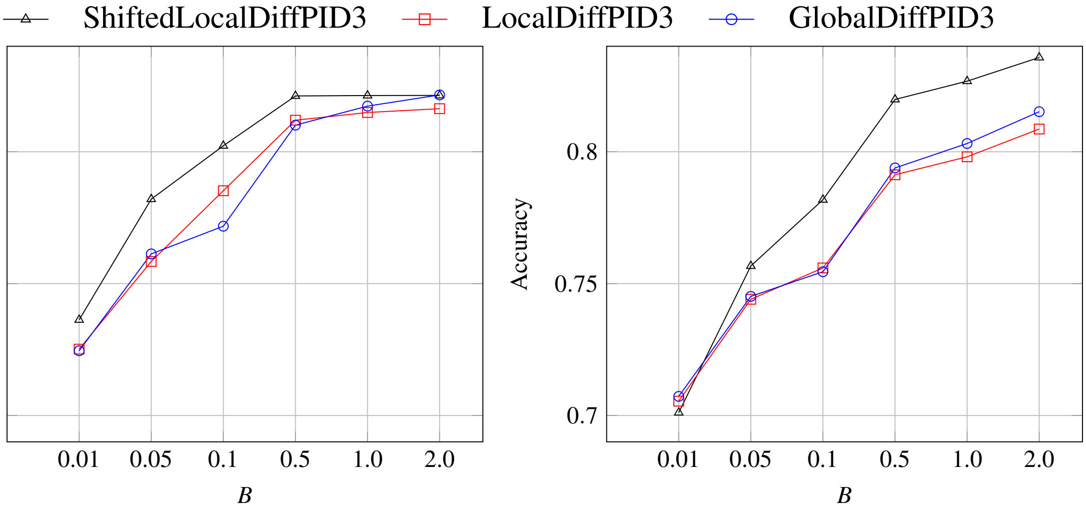
(d) ACS dataset - Depth=5

Source: elaborated by the author.

Figure 24 – ACS dataset

For the ACS dataset, Figure 24, the inversion problem (Section 4.1) appears. Specifically, for $depth = 2$, the second and the third attributes with largest IG become the third and second attributes, respectively, with larger Dampened IG. As a consequence, as B grows, LocalDiffPID3 tends to pick the first and the third attributes with largest Information which is sub-optimal. ShiftedLocalDiffPID3 is less prone to suffer from the inversion problem in larger depths, i.e. depth=5, since it can pick, in deeper levels, those attributes that loose rank (see Figure 24d). ShiftedLocalDiffPID3 improves at most 8% on GlobalDiffPID3.

Figure 25 – Adult dataset



(c) Depth=2

(d) Depth=5

Source: elaborated by the author.

Fot the Adult Dataset, Figure 25, the LocalDiffPID3 improves just a little over the GlobalDiffPID3. However, ShiftedLocalDiffPID3 improves over GlobalDiffPID3 up to 4%.

7.5 Conclusion

In this section, we introduced the notions of decision tree and automatic tree induction. We state the problem for this section which is to build private tree induction algorithm based on ID3 algorithm.

We use the same algorithm template as (FRIEDMAN; SCHUSTER, 2010). We replace the exponential mechanism call to choose the attribute to split the dataset using IG for a call to the local dampening mechanism or the shifted local dampening mechanisms. For that, we provided the element local sensitivity.

The experimental results compare the algorithm using the local dampening mechanism and the shifted local dampening mechanism to one using exponential mechanism using 3 real world datasets. The results show that our approach outperforms the exponential mechanism based algorithm in most of the values for privacy budget, values for depth and datasets tested in terms of accuracy. The improvement on accuracy was up to 8%.

8 CONCLUSION AND FUTURE WORK

In this thesis, we introduced the local dampening mechanism, a novel framework to provide differential privacy for non-numeric queries using local sensitivity. We have shown that using local sensitivity on non-numeric queries reduces the noise added to achieve differential privacy which makes the answer of those queries more useful. We discuss that we can use the local sensitivity or an upper bound for it through a sensitivity function. We classify than sensitivity functions according to four aspects: admissibility, boundedness, monotonicity and stability.

We proposed a second version of the local dampening mechanism called the shifted local dampening where we can benefit from monotonic sensitivity functions. We provide a theoretical accuracy analysis and guide to construct sensitivity functions that provide better accuracy on the shifted local dampening mechanism.

We evaluated our approach on three applications:

1. Median selection. We showed that local dampening mechanism can select elements 18% closer to the median compared to global sensitivity based approaches.
2. Influential node analysis. This application benefited greatly from the use of local sensitivity. In our experiments we could reduce the use of privacy budget by 2 to 4 while keeping the same accuracy level.
3. Decision Tree induction. Our approach improves on approaches that use the exponential mechanism for this task based on Information Gain. In our experiments, the improvement of accuracy is up to 8%.

8.1 Future Work

Our work has laid the foundations for providing differential privacy for non-numeric queries using local sensitivity. There are many interesting directions of future work. Any problem in the literature that has used the Exponential mechanism for non-numeric queries to guarantee DP is a candidate problem that could potentially benefit from using our local dampening mechanism instead, and worthy of future work. Some example of future direction include:

1. Address other graph influence/centrality metrics for Influential Node analysis. Specifically, egocentric measures are good candidates since many of them have low local sensitivity as

the egocentric density;

2. Apply the local dampening mechanism to private random forest algorithms. Many private random forest algorithms already use local sensitivity through the smooth sensitivity framework to add numeric noise. For some of those algorithms, replacing the numeric mechanism to a non-numeric mechanism can reduce the number of differentially private queries issued which saves privacy budget. This is the same improvement that (FRIEDMAN; SCHUSTER, 2010) provided over (BLUM *et al.*, 2005);
3. We envision to tackle the problem of differentially private multivariate non-numeric queries. The setting to this problem is similar to the non-numeric setting. However, the difference is that the analyst has more than one utility functions and he/she wants to chose the best answer that maximizes all utility functions.

BIBLIOGRAPHY

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

BLOCKI, J.; BLUM, A.; DATTA, A.; SHEFFET, O. Differentially private data analysis of social networks via restricted sensitivity. In: ACM. **Proceedings of the 4th Conference on Innovations in Theoretical Computer Science**. [*s.l.*], 2013. p. 87–96.

BLUM, A.; DWORK, C.; MCSHERRY, F.; NISSIM, K. Practical privacy: the sulq framework. In: **Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems**. [*s.l.*: *s.n.*], 2005. p. 128–138.

BRASIL. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. 2018. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm. Acesso em: 09/08/2021.

CHEN, S.; ZHOU, S. Recursive mechanism: towards node differential privacy and unrestricted joins. In: **Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data**. [*s.l.*: *s.n.*], 2013. p. 653–664.

CORMODE, G.; PROCOPIUC, C.; SRIVASTAVA, D.; SHEN, E.; YU, T. Differentially private spatial decompositions. In: IEEE. **2012 IEEE 28th International Conference on Data Engineering**. [*s.l.*], 2012. p. 20–31.

CORMODE, G.; SRIVASTAVA, D.; LI, N.; LI, T. Minimizing minimality and maximizing utility: analyzing method-based attacks on anonymized data. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 3, n. 1-2, p. 1045–1056, 2010.

DWORK, C. Differential privacy. **Encyclopedia of Cryptography and Security**, Springer, p. 338–340, 2011.

DWORK, C.; KENTHAPADI, K.; MCSHERRY, F.; MIRONOV, I.; NAOR, M. Our data, ourselves: Privacy via distributed noise generation. In: SPRINGER. **Annual International Conference on the Theory and Applications of Cryptographic Techniques**. [*s.l.*], 2006. p. 486–503.

DWORK, C.; MCSHERRY, F.; NISSIM, K.; SMITH, A. Calibrating noise to sensitivity in private data analysis. In: SPRINGER. **Theory of cryptography conference**. [*s.l.*], 2006. p. 265–284.

DWORK, C.; ROTH, A. *et al.* The algorithmic foundations of differential privacy. **Foundations and Trends in Theoretical Computer Science**, v. 9, n. 3-4, p. 211–407, 2014.

EUROPEAN COMMISSION. **2018 reform of EU data protection rules**. 2018. Disponível em: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf. Acesso em: 09/08/2021.

EVERETT, M.; BORGATTI, S. P. Ego network betweenness. **Social networks**, Elsevier, v. 27, n. 1, p. 31–38, 2005.

FARIAS, V. A. E. de; BRITO, F. T.; FLYNN, C.; MACHADO, J. C.; MAJUMDAR, S.; SRIVASTAVA, D. Local dampening: Differential privacy for non-numeric queries via local sensitivity. **Proc. VLDB Endow.**, v. 14, n. 4, p. 521–533, 2020. Disponível em: http://www.vldb.org/pvldb/vol14/p521-farias.pdf. Acesso em: 09/08/2021.

FLETCHER, S.; ISLAM, M. Z. A differentially private decision forest. **AusDM**, v. 15, p. 99–108, 2015.

FLETCHER, S.; ISLAM, M. Z. A differentially private random decision forest using reliable signal-to-noise ratios. In: SPRINGER. **Australasian joint conference on artificial intelligence**. [*s.l.*], 2015. p. 192–203.

FLETCHER, S.; ISLAM, M. Z. Differentially private random decision forests using smooth sensitivity. **Expert Systems with Applications**, Elsevier, v. 78, p. 16–31, 2017.

FLETCHER, S.; ISLAM, M. Z. Decision tree classification with differential privacy: A survey. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 52, n. 4, p. 1–33, 2019.

FREEMAN, L. C. Centrality in social networks conceptual clarification. **Social networks**, North-Holland, v. 1, n. 3, p. 215–239, 1978.

FRIEDMAN, A.; SCHUSTER, A. Data mining with differential privacy. In: **Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [*s.l.*: *s.n.*], 2010. p. 493–502.

GANTA, S. R.; KASIVISWANATHAN, S. P.; SMITH, A. Composition attacks and auxiliary information in data privacy. In: **Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [*s.l.*: *s.n.*], 2008. p. 265–273.

HARDT, M.; LIGETT, K.; MCSHERRY, F. A simple and practical algorithm for differentially private data release. **arXiv preprint arXiv:1012.4763**, 2010.

HARDT, M.; LIGETT, K.; MCSHERRY, F. A simple and practical algorithm for differentially private data release. In: PEREIRA, F.; BURGES, C. J. C.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2012. v. 25. Disponível em: https://proceedings.neurips.cc/paper/2012/file/208e43f0e45c4c78cafadb83d2888cb6-Paper.pdf. Acesso em: 09/08/2021.

HAY, M.; MACHANAVAJJHALA, A.; MIKLAU, G.; CHEN, Y.; ZHANG, D. Principled evaluation of differentially private algorithms using dpbench. In: **Proceedings of the 2016 International Conference on Management of Data**. New York, NY, USA: Association for Computing Machinery, 2016. (SIGMOD ’16), p. 139–154. ISBN 9781450335317. Disponível em: https://doi.org/10.1145/2882903.2882931. Acesso em: 09/08/2021.

JAGANNATHAN, G.; PILLAIPAKKAMNATT, K.; WRIGHT, R. N. A practical differentially private random decision tree classifier. In: IEEE. **2009 IEEE International Conference on Data Mining Workshops**. [*s.l.*], 2009. p. 114–121.

JIN, X.; ZHANG, N.; DAS, G. Algorithm-safe privacy-preserving data publishing. In: **Proceedings of the 13th International Conference on Extending Database Technology**. [*s.l.*: *s.n.*], 2010. p. 633–644.

JOHNSON, N.; NEAR, J. P.; SONG, D. Towards practical differential privacy for sql queries. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 11, n. 5, p. 526–539, 2018.

KARWA, V.; RASKHODNIKOVA, S.; SMITH, A.; YAROSLAVTSEV, G. Private analysis of graph structure. **Proceedings of the VLDB Endowment**, Very Large Data Base Endowment Inc., v. 4, n. 11, p. 1146–1157, 2011.

KASIVISWANATHAN, S. P.; NISSIM, K.; RASKHODNIKOVA, S.; SMITH, A. Analyzing graphs with node differential privacy. In: SPRINGER. **Theory of Cryptography Conference**. [*s.l.*], 2013. p. 457–476.

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

KOTSOGIANNIS, I.; TAO, Y.; HE, X.; FANAEEPOUR, M.; MACHANAVAJJHALA, A.; HAY, M.; MIKLAU, G. Privatesql: a differentially private sql query engine. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 12, n. 11, p. 1371–1384, 2019.

LAEUCHLI, J.; RAMÍREZ-CRUZ, Y.; TRUJILLO-RASUA, R. Analysis of centrality measures under differential privacy models. **arXiv preprint arXiv:2103.03556**, 2021.

LESKOVEC, J.; KREVL, A. **SNAP Datasets: Stanford Large Network Dataset Collection**. 2014. Http://snap.stanford.edu/data.

LI, C.; MIKLAU, G.; HAY, M.; MCGREGOR, A.; RASTOGI, V. The matrix mechanism: optimizing linear counting queries under differential privacy. **The VLDB journal**, Springer, v. 24, n. 6, p. 757–781, 2015.

LI, N.; LI, T.; VENKATASUBRAMANIAN, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In: IEEE. **2007 IEEE 23rd International Conference on Data Engineering**. [*s.l.*], 2007. p. 106–115.

LI, N.; LI, T.; VENKATASUBRAMANIAN, S. Closeness: A new privacy measure for data publishing. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 22, n. 7, p. 943–956, 2009.

LU, W.; MIKLAU, G. Exponential random graph estimation under differential privacy. In: ACM. **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [*s.l.*], 2014. p. 921–930.

MA, H.; YANG, H.; LYU, M. R.; KING, I. Mining social networks using heat diffusion processes for marketing candidates selection. In: **Proceedings of the 17th ACM Conference on Information and Knowledge Management**. [*s.l.*: *s.n.*], 2008. p. 233–242.

MACHANAVAJJHALA, A.; HE, X.; HAY, M. Differential privacy in the wild: A tutorial on current practices & open challenges. In: **Proceedings of the 2017 ACM SIGMOD International Conference on Management of data**. [*s.l.*]: ACM, 2017. p. 1727–1730.

MACHANAVAJJHALA, A.; KIFER, D.; GEHRKE, J.; VENKITASUBRAMANIAM, M. l-diversity: Privacy beyond k-anonymity. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM New York, NY, USA, v. 1, n. 1, p. 3–es, 2007.

MANTON, K. G. National long-term care survey: 1982, 1984, 1989, 1994, 1999, and 2004. **Inter-university Consortium for Political and Social Research**, 2010.

MARSDEN, P. V. Egocentric and sociocentric measures of network centrality. **Social networks**, Elsevier, v. 24, n. 4, p. 407–422, 2002.

MCKENNA, R.; SHELDON, D. R. Permute-and-flip: A new mechanism for differentially private selection. **Advances in Neural Information Processing Systems**, v. 33, 2020.

MCSHERRY, F.; TALWAR, K. Mechanism design via differential privacy. In: **48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)**. [*s.l.*: *s.n.*], 2007. p. 94–103.

MCSHERRY, F. D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: **Proceedings of the 2009 ACM SIGMOD International Conference on Management of data**. [*s.l.*: *s.n.*], 2009. p. 19–30.

MOHAMMED, N.; CHEN, R.; FUNG, B.; YU, P. S. Differentially private data release for data mining. In: ACM. **Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [*s.l.*], 2011. p. 493–501.

NARAYANAN, A.; SHMATIKOV, V. Robust de-anonymization of large sparse datasets. In: IEEE. **2008 IEEE Symposium on Security and Privacy (sp 2008)**. [*s.l.*], 2008. p. 111–125.

NARAYANAN, A.; SHMATIKOV, V. De-anonymizing social networks. In: IEEE. **2009 30th IEEE symposium on security and privacy**. [*s.l.*], 2009. p. 173–187.

NERGIZ, M. E.; ATZORI, M.; CLIFTON, C. Hiding the presence of individuals from shared databases. In: **Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data**. [*s.l.*: *s.n.*], 2007. p. 665–676.

NISSIM, K.; RASKHODNIKOVA, S.; SMITH, A. Smooth sensitivity and sampling in private data analysis. In: ACM. **Proceedings of the thirty-ninth annual ACM symposium on Theory of computing**. [*s.l.*], 2007. p. 75–84.

PATIL, A.; SINGH, S. Differential private random forest. In: IEEE. **2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)**. [*s.l.*], 2014. p. 2623–2630.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, n. 1, p. 81–106, 1986.

RANA, S.; GUPTA, S. K.; VENKATESH, S. Differentially private random forest with high utility. In: IEEE. **2015 IEEE International Conference on Data Mining**. [*s.l.*], 2015. p. 955–960.

RASTOGI, V.; HAY, M.; MIKLAU, G.; SUCIU, D. Relationship privacy: output perturbation for queries with joins. In: **Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems**. [*s.l.*: *s.n.*], 2009. p. 107–116.

SALZBERG, S. L. **C4. 5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann publishers, inc.** [*s.l.*]: Kluwer Academic Publishers, 1993.

SAMARATI, P.; SWEENEY, L. Generalizing data to provide anonymity when disclosing information (abstract). In: MENDELZON, A. O.; PAREDAENS, J. (Ed.). **Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1-3, 1998, Seattle, Washington, USA**. ACM Press, 1998. p. 188. Disponível em: https://doi.org/10.1145/275487.275508. Acesso em: 09/08/2021.

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

SWEENEY, L. k-anonymity: A model for protecting privacy. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, World Scientific, v. 10, n. 05, p. 557–570, 2002.

TAO, Y.; HE, X.; MACHANAVAJJHALA, A.; ROY, S. Computing local sensitivities of counting queries with joins. In: **Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data**. [*s.l.*: *s.n.*], 2020. p. 479–494.

WONG, R. C.-W.; FU, A. W.-C.; WANG, K.; YU, P. S.; PEI, J. Can the utility of anonymized data be used for privacy breaches? **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM New York, NY, USA, v. 5, n. 3, p. 1–24, 2011.

XIAO, X.; TAO, Y.; KOUDAS, N. Transparent anonymization: Thwarting adversaries who know the algorithm. **ACM Transactions on Database Systems (TODS)**, ACM New York, NY, USA, v. 35, n. 2, p. 1–48, 2010.

ZHANG, J.; CORMODE, G.; PROCOPIUC, C. M.; SRIVASTAVA, D.; XIAO, X. Private release of graph statistics using ladder functions. In: ACM. **Proceedings of the 2015 ACM SIGMOD international conference on management of data**. [*s.l.*], 2015. p. 731–745.

ZHANG, J.; CORMODE, G.; PROCOPIUC, C. M.; SRIVASTAVA, D.; XIAO, X. Privbayes: Private data release via bayesian networks. **ACM Transactions on Database Systems (TODS)**, ACM, v. 42, n. 4, p. 25, 2017.

APPENDIX A – PROOFS

A.1 Proof of Lemma 4.4.1

Lemma 4.4.1. *(Local Dampening Accuracy) Let $\delta^u(x,t,r)$ and $\bar{\delta}^u(x,t,r)$ be two stable functions and x be a database. If $\delta^u(x,t,r)$ dominates $\bar{\delta}^u(x,t,r)$ then:*

1. *$Pr[\mathscr{E}(M_{SLD},x) \geq \theta] \leq Pr[\mathscr{E}(\overline{M}_{SLD},x) \geq t]$ for all $\theta \geq 0$,*
2. *$\mathbb{E}[\mathscr{E}(M_{SLD},x)] \leq \mathbb{E}[\mathscr{E}(\overline{M}_{SLD},x)]$,*

where M_{SLD} represents an instance of the shifted local dampening mechanism using δ^u as sensitivity function while $\overline{M_{SLD}}$ is an instance using $\bar{\delta}^u$.

Proof. We first prove point 1. Let t be a real number larger than 0. we need to show that the following expression if non-positive.

$$Pr[\mathscr{E}(M_{SLD},x) \geq \theta] - Pr[\mathscr{E}(\overline{M}_{SLD},x) \geq \theta] \tag{A.1}$$

$$= \sum_{r\in\mathscr{R}|u^*-u(x,r)\geq\theta} Pr[M_{SLD}(x) = r] - Pr[\overline{M}_{SLD}(x) = r] \tag{A.2}$$

$$= \sum_{r\in\mathscr{R}|u^*-u(x,r)\geq\theta} \lim_{s\to\infty} \left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\delta}(x,r)}{2}\right)}{\sum_{r'\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^s,\delta}(x,r')}{2}\right)} \right) - \lim_{s\to\infty} \left(\frac{\exp\left(\frac{\varepsilon D_{u^s,\bar{\delta}}(x,r)}{2}\right)}{\sum_{r'\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^s,\bar{\delta}}(x,r')}{2}\right)} \right) \tag{A.3}$$

$$= \sum_{r\in\mathscr{R}|u^*-u(x,r)\geq\theta} \left(\frac{\exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r)}{2}\right)}{\sum_{r'\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right)} - \frac{\exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r)}{2}\right)}{\sum_{r'\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r')}{2}\right)} \right) \tag{A.4}$$

$$= \sum_{r\in\mathscr{R}|u^*-u(x,r)\geq\theta} \left(\frac{\exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r)}{2}\right) \sum_{r'\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r')}{2}\right)}{\sum_{r',r''\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right) \exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r'')}{2}\right)} \right. \tag{A.5}$$

$$\left. - \frac{\exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r)}{2}\right) \sum_{r'\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right)}{\sum_{r',r''\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right) \exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r'')}{2}\right)} \right) \tag{A.6}$$

$$= \frac{\sum_{r|u^*-u(x,r)\geq\theta} \sum_{r'\in\mathscr{R}} \exp\left(\frac{\varepsilon\left(D_{u^{s_0},\delta}(x,r) + D_{u^{s_0},\bar{\delta}}(x,r')\right)}{2}\right)}{\sum_{r',r''\in\mathscr{R}} \exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right) \exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r'')}{2}\right)} \tag{A.7}$$

$$-\frac{\sum_{r|u^*-u(x,r)\geq\theta}\sum_{r'\in\mathscr{R}}\exp\left(\frac{\varepsilon\,(D_{u^s,\bar{\delta}}(x,r)+D_{u^s,\delta}(x,r'))}{2}\right)}{\sum_{r',r''\in\mathscr{R}}\exp\left(\frac{\varepsilon D_{u^s,\delta}(x,r')}{2}\right)\exp\left(\frac{\varepsilon D_{u^s,\bar{\delta}}(x,r'')}{2}\right)} \tag{A.8}$$

$$=\frac{\sum_{\substack{r|u^*-u(x,r)\geq\theta\\ r'|u^*-u(x,r)<\theta}}\exp\left(\frac{\varepsilon\,(D_{u^{s_0},\delta}(x,r)+D_{u^{s_0},\bar{\delta}}(x,r'))}{2}\right)+\sum_{\substack{r|u^*-u(x,r)\geq\theta\\ r'|u^*-u(x,r)\geq\theta}}\exp\left(\frac{\varepsilon\,(D_{u^{s_0},\delta}(x,r)+D_{u^{s_0},\bar{\delta}}(x,r'))}{2}\right)}{\sum_{r',r''\in\mathscr{R}}\exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right)\exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r'')}{2}\right)}$$
$$-\frac{\sum_{\substack{r|u^*-u(x,r)\geq\theta\\ r'|u^*-u(x,r)<\theta}}\exp\left(\frac{\varepsilon\,(D_{u^{s_0},\bar{\delta}}(x,r)+D_{u^{s_0},\delta}(x,r'))}{2}\right)+\sum_{\substack{r|u^*-u(x,r)\geq\theta\\ r'|u^*-u(x,r)\geq\theta}}\exp\left(\frac{\varepsilon\,(D_{u^{s_0},\bar{\delta}}(x,r)+D_{u^{s_0},\delta}(x,r'))}{2}\right)}{\sum_{r',r''\in\mathscr{R}}\exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right)\exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r'')}{2}\right)} \tag{A.9}$$

$$=\frac{\sum_{\substack{r|u^*-u(x,r)\geq\theta\\ r'|u^*-u(x,r')<\theta}}\left(\exp\left(\frac{\varepsilon\,(D_{u^{s_0},\delta}(x,r)+D_{u^s,\bar{\delta}}(x,r'))}{2}\right)-\exp\left(\frac{\varepsilon\,(D_{u^{s_0},\bar{\delta}}(x,r)+D_{u^s,\delta}(x,r'))}{2}\right)\right)}{\sum_{r',r''\in\mathscr{R}}\exp\left(\frac{\varepsilon D_{u^{s_0},\delta}(x,r')}{2}\right)\exp\left(\frac{\varepsilon D_{u^{s_0},\bar{\delta}}(x,r'')}{2}\right)} \tag{A.10}$$

$$\leq 0 \tag{A.11}$$

The Line A.4 follows from the Lemma 4.3.1 where $s_0 = \Delta u + u*$. The last inequality (Line A.11) follows from the following:

$$D_{u^s,\delta}(x,r') - D_{u^s,\delta}(x,r) \tag{A.12}$$

$$= \frac{u^{s_0}(x,r') - \sum_{t=0}^{n-1}\delta(x,t,r')}{\Delta u} + n - \frac{u^{s_0}(x,r) - \sum_{t=0}^{n-1}\delta(x,t,r)}{\Delta u} - n \tag{A.13}$$

$$= \frac{u^{s_0}(x,r') + \sum_{t=0}^{n-1}(\bar{\delta}(x,t,r') + \alpha_{x,t,r'})}{\Delta u} - \frac{u^{s_0}(x,r) + \sum_{t=0}^{n-1}(\bar{\delta}(x,t,r) + \alpha_{x,t,r})}{\Delta u} \tag{A.14}$$

$$= \frac{u^{s_0}(x,r') + \sum_{t=0}^{n-1}\bar{\delta}(x,t,r') + \sum_{t=0}^{n-1}\alpha_{x,t,r'}}{\Delta u} - \frac{u^{s_0}(x,r) + \sum_{t=0}^{n-1}\bar{\delta}(x,t,r) + \sum_{t=0}^{n-1}\alpha_{x,t,r}}{\Delta u} \tag{A.15}$$

$$= D_{u^s,\bar{\delta}}(x,r') + \frac{\sum_{t=0}^{n-1}\alpha_{x,t,r'}}{\Delta u} - D_{u^s,\bar{\delta}}(x,r) - \frac{\sum_{t=0}^{n-1}\alpha_{x,t,r}}{\Delta u} \tag{A.16}$$

$$\geq D_{u^s,\bar{\delta}}(x,r') - D_{u^s,\bar{\delta}}(x,r) \tag{A.17}$$

□

Line A.13 follows from the definition of the shifted local dampening when using shifting by s_0. The last inequality (Line A.17) is due to the dominance of δ^u over $\bar{\delta}^u$ and as $u(x,r') > u(x,r)$.

A.2 Proof of Lemma 5.2.2

Lemma 5.2.2. *(Median Selection Element Local Sensitivity at distance 0)*

$$LS^{u_{med}}(x,0,i) = \max(|x_m - x_i|, x_{m+1} - x_m, x_m - x_{m-1}, p(x,i), q(x,i)),$$

where

$$p(x,i) = \max \begin{cases} \Lambda - x_i & \textit{if } i > m \\ \Lambda - x_{m+1} & \textit{if } i = m \\ \Lambda + x_i - 3x_m + x_{m+1} & i < m \end{cases},$$

$$q(x,i) = \max \begin{cases} x_i & \textit{if } i > m \\ x_{m-1} & \textit{if } i = m \\ 3x_m - x_i - x_{m-1} & i < m \end{cases},$$

and $0 \le x_1 \le \cdots \le x_n \le \Lambda$.

Proof. For this proof, we restate the definitions. The utility function is given by

$$u_{med}(x,r) = |v_x(med_x) - v_x(r)|$$

, where $v_x(r)$ is the value of the element r in x, $med_x \in \mathscr{R}$ is the element with $m-th$ largest value in x and $m = \lceil \frac{n}{2} \rceil$.

Thus, the element local sensitivity $LS^{u_{med}}(x,0,r)$ for u_{med} is rewritten as:

$$LS^{u_{med}}(x,0,r) = \max(\underbrace{|v_x(med_x) - v_x(r)|}_{(1)}, \underbrace{||v_x(med_x) - v_x(r)| - |v_x(med_x^+) - v_x(r)||}_{(2)},$$
$$\underbrace{||v_x(med_x) - v_x(r)| - |v_x(med_x^-) - v_x(r)||}_{(3)}, \underbrace{p(x,r)}_{(4)}, \underbrace{q(x,r)}_{(5)})$$

where

$$p(x,r) = \max \begin{cases} \underbrace{\Lambda - v_x(med_x^+)}_{(4.1)} & \text{if } r = med \\ \underbrace{\Lambda - v_x(r)}_{(4.2)} & \text{else if } v_x(r) >= v_x(med) \\ \underbrace{|v_x(med_x) - v_x(r) - \Lambda + v_x(med_x^+)|}_{(4.3)} & \text{otherwise} \end{cases},$$

$$q(x,r) = \max \begin{cases} \underbrace{v^x(med_x^-)}_{(5.1)} & \text{if } r = med \\ \underbrace{|v_x(r) - v_x(med_x) - v_x(med_x^-)|}_{(5.2)} & \text{if } v_x(r) >= v_x(med) \\ \underbrace{v^x(r)}_{(5.3)} & \text{otherwise} \end{cases},$$

where $med_x^+ \in \mathscr{R}$ and $med_x^- \in \mathscr{R}$ are the elements with the $m+1$-ith and $m-1$-ith largest value, respectively.

We first prove that for every case above there is a neighboring database y of x which $|u(x,r) - u(y,r)|$ is larger than it:

(1) Let y be a database obtained from x by changing the value of r to $v_x(med_x)$, $v_y(r) = v_x(med_x)$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_x(med_x) - v_y(r)|| = |v_x(med_x) - v_x(r)|$ as $v_y(med_y) = v_x(med_x)$ and $v^y(r) = v_x(med_x)$.

(2) Let y be a database obtained from x by changing the value of med_x to Λ, $v_y(med_x) = \Lambda$. This way, $v_y(med_y) = v_x(med_x^+)$ and $v_y(r) = v_x(r)$. Then $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = ||v_x(med_x) - v_x(r)| - |v_x(med_x^+) - v_x(r)||$.

(3) Let y be a database obtained from x by changing the value of med_x to 0, $v_y(med_x) = 0$. This way, $v_y(med_y) = v_x(med_x^-)$ and $v_y(r) = v_x(r)$. Then $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = ||v_x(med_x) - v_x(r)| - |v_x(med_x^-) - v_x(r)||$.

(4.1) Let y be a database obtained from x by changing the value of r to Λ, $v_y(r) = \Lambda$. In this case, $r = med_x$, so $v_y(med_y) = v_x(med_x^+)$. Then, we have that $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = ||v_x(med_x) - v_x(med_x)| - |v_x(med_x^+) - \Lambda|| = |v_x(med_x^+) - \Lambda| = \Lambda = v_x(med_x^+)$.

(4.2) Let y be a database obtained from x by changing the value of r to Λ, $v_y(r) = \Lambda$. In this case, $v_x(r) >= v_x(med)$, so $v_y(med_y) = v_x(med_x)$. Then, we have that $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(r) - v_x(med_x) - \Lambda + v_x(med_x)| = \Lambda - v_x(r)$.

(4.3) Let y be a database obtained from x by changing the value of r to Λ, $v_y(r) = \Lambda$. In this case, $v_x(r) < v_x(med)$, so $v_y(med_y) = v_x(med_x^+)$. Then, we have that $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(med_x) - v_x(r) - \Lambda + v_x(med_x^+)|$.

(5.1) Let y be a database obtained from x by changing the value of r to 0, $v_y(r) = 0$. In this case, $r = med_x$, so $v_y(med_y) = v_x(med_x^-)$. Then, we have that $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(med_x^-) - 0| = v_x(med_x^-)$.

(5.2) Let y be a database obtained from x by changing the value of r to 0, $v_y(r) = 0$. In this case, $v_x(r) >= v_x(med)$, so $v_y(med_y) = v_x(med_x^-)$. Then, we have that $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(r) - v_x(med_x) - v_x(med_x^-)|$.

(5.3) Let y be a database obtained from x by changing the value of r to 0, $v_y(r) = 0$. In this case, $v_x(r) < v_x(med)$, so $v_y(med_y) = v_x(med_x)$. Then, we have that $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(med_x) - v_x(r) - v_x(med_x)| = v_x(r)$.

Now we prove that for every neighboring database y of x, the exists a case where $|u(x,r) - u(y,r)|$ is smaller or equal than it. The database y can be obtained from x by choosing a element $r' \in \mathscr{R}$ and then choosing a new value $v_y(r')$ for it. We divide the set of neighboring databases in some cases:

(i) The element r' is such that $r' \neq r$. This operation cannot change the value of r, $v_y(r) = v_x(r)$. However, this operation can set the value of med_y. The element r' can move freely along the domain $[0, \Lambda]$. However, there are only three possibilities for med_y: it can keep the value of med_x, it can change to the value of med_x^+ or it can change to med_x^-.

(i.i) if $v_y(med_y) = v_x(med_x)$ then $|u(x,r) - u(y,r)| = 0$;

(i.ii) if $v_y(med_y) = v_x(med_x^+)$ then $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = ||v_x(med_x) - v_x(r)| - |v_x(med_x^+) - v_x(r)||$ (case 2); and

(i.iii) if $v_y(med_y) = v_x(med_x^-)$ then $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = ||v_x(med_x) - v_x(r)| - |v_x(med_x^-) - v_x(r)||$ (case 3).

(ii) The element r' is such that $r' = r$. We split this argument in two subcases:

(ii.i) $r' = r = med_x$. Here we divide again in 3 subcases to set the value of r' on y:

(ii.i.i) $v_y(r') = med_x$. It means that $x = y$ and $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = ||v_x(med_x) - v_x(r)| - |v_x(med_x) - v_x(r)|| = 0$.

(ii.i.ii) $v_y(r') = v_y(r') \geq med_x$. So we have that the median element of y is at maximum $v_x(med_x^+)$ and that $v_y(r') \geq v_y(med_y)$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |0 - |v_y(med_y) - v_y(r)|| = v_y(r) - v_y(med_y) \leq \Lambda - v_x(med_x^+)$ (case 4.1).

(ii.i.iii) $v_y(r') = v_y(r') \leq med_x$. So we have that the median element of y is at minimum $v_x(med_x^-)$ and that $v_y(r') \leq v_y(med_y)$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |0 - |v_y(med_y) - v_y(r)|| = v_y(med_y) - v_y(r) \leq v_x(med_x^-)$ (case 5.1).

(ii.ii) $r' = r \neq med_x$. We divide in two subcases:

(ii.ii.i) $v_x(r) \geq v_x(med_x)$. We divide again in three cases:

(ii.ii.i.i) $v_y(r) = v_y(r') \geq v_x(r) = v_x(r')$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) -$

$v_x(r)| - |v_y(med_y) - v_y(r)|| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = v_x(med_x) - v_x(r) + v_y(r) - v_y(med_y) \leq \Lambda - v_x(r)$ since $v_x(med_x) = v_y(med_y)$ (case 4.2).

(ii.ii.i.ii) $v_x(med_x) \leq v_y(r) = v_y(r') < v_x(r) = v_x(r')$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(r) - v_x(med_x) - |v_x(med_x) - v_y(r)|| = |v_x(med_x) - v_x(r) + v_y(r) - v_x(med_x)| = v_x(r) - v_y(r) \leq v_x(r) - v_x(med_x) \leq |v_x(med_x) - v_x(r)|$ (case 1).

(ii.ii.i.iii) $0 \leq v_y(r) < v_x(med_x)$. So we have that $v_y(med_y) = v_x(med_x^-)$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(r) - v_x(med_x) - v_x(med_x^-) + v_y(r)| \leq |v_x(r) - v_x(med_x) - v_x(med_x^-)|$ (case 5.2). The last inequality follows by setting $v_y(r) = 0$.

(ii.ii.ii) $v_x(r) \leq v_x(med_x)$. We divide in three cases:

(ii.ii.ii.i) $v_y(r) = v_y(r') > v_x(med_x)$. So we have that $v_y(med_y) = v_x(med_x^+)$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(med_x) - v_x(r) - v_y(r) + v_x(med_x^+)| \leq |v_x(med_x) - v_x(r) - \Lambda + v_x(med_x^+)$ (case 4.3).

(ii.ii.ii.ii) $v_x(r) = v_x(r') < v_y(r) = v_y(r') \leq v_x(med_x)$. So we have that $v_y(med_y) = v_x(med_x)$ Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(med_x) - v_x(r) - v_x(med_x) + v_y(r)| = |v_y(r) - v_x(r)| \leq |v_x(med_x) - v_x(r)|$ (case 1).

(ii.ii.ii.i) $0 \leq v_y(r) = v_y(r') \leq v_x(r) = v_x(r')$. So we have that $v_y(med_y) = v_x(med_x)$. Thus $|u(x,r) - u(y,r)| = ||v_x(med_x) - v_x(r)| - |v_y(med_y) - v_y(r)|| = |v_x(med_x) - v_x(r) - v_x(med_x) + v_y(r)| = |v_y(r) - v_x(r)| \leq v_x(r)$ (case 5.3).

□

A.3 Proof of Lemma 5.2.3

Lemma 5.2.3. *(Element local sensitivity at distance t for median selection)*

$$LS^{u_{med}}(x,t,r) = \max_{candidates(x,t,r)} LS^{u_{med}}(y,0,r).$$

A.4 Proof of Lemma 6.3.1

Lemma 6.3.1. *Let G and G' be two neighboring graphs and v a node belonging to $V(G)$ and $V(G')$, we have that:*

$$\max_{G,G'|d(G,G')\leq 1} |EBC^G(v) - EBC^{G'}(v)| = \max\left(d^G(v)(d^G(v)-1)/4, d^G(v)\right),$$

where $d^G(v)$ denotes the degree of v in G, i.e., $d^G(v) = |N_v^G|$.

Proof. Let $\Delta(v)$ be defined as

$$\begin{aligned}\Delta(v) &= \max_{G,G'|d(G,G')} |EBC^G(v) - EBC^{G'}(v)| \\ &= \max_{G,G'|d(G,G')} \left| \sum_{x,y\in N_v^G|x\neq y} b_{xy}^G(v) - \sum_{x,y\in N_v^{G'}|x\neq y} b_{xy}^{G'}(v) \right|\end{aligned}$$

where $b_{uy}^{G'}(v) = g_{uy}^{G'}(c)/g_{uy}^{G'}$. Without loss of generality, let $e \in V(G)$ the edge that belongs to G' and not to G, i.e., $E(G') = E(G) \cup \{e\}$. We analyse two cases for e:

Case (1). One end of e is v, $e = (vu)$, i.e., $N(v)^{G'} = N(v)^G \cup \{u\}$. Since e does not belong to G then u is not a neighbor of v in G which means that the terms b_{uy} for all $y \in N_v^G$ are the only terms that do not exist on the expression for $EBC^G(v)$. So we rewrite $EBC^{G'}(v)$ as $\sum_{x,y\in N_v^{G'}|x\neq y} b_{xy}^G(v) + \sum_{y\in N_v^G} b_{uy}^{G'}(v)$ and $\Delta(v)$ as

$$\begin{aligned}&\max_{G,G'|d(G,G')} \left| \sum_{x,y\in N_v^G|x\neq y} b_{xy}^G(v) - \sum_{x,y\in N_v^G|x\neq y} b_{xy}^{G'}(v) - \sum_{y\in N_v^G} b_{uy}^{G'}(v) \right| \\ &= \max_{G,G'|d(G,G')} \left| \sum_{x,y\in N_v^G|x\neq y} (b_{xy}^G(v) - b_{xy}^{G'}(v)) - \sum_{y\in N_v^G} b_{uy}^{G'}(v) \right|\end{aligned}$$

We find bounds for $\sum_{y\in N_v^G} b_{uy}^{G'}(v)$. $b_{uy}^{G'}(v)$ is non-negative as $g_{uy}^{G'}$ is positive and $g_{uy}^{G'}(c)$ is non-negative. $b_{uy}^{G'}(v) \leq 1$ since $g_{uy}^{G'} \geq g_{uy}^{G'}(c)$. Moreover there are $|N_v^G| = d$ pairs u,y since $y \in N(v)^G$. Thus

$$0 \leq \sum_{y\in N_v^G} b_{uy}^{G'}(v) \leq d$$

Now we find bounds for $\sum_{x,y\in N_v^G|x\neq y}(b_{xy}^G(v) - b_{xy}^{G'}(v))$. If a geodesic path from x to y ($x,y \in N_v^G$) has size 1 in G', i.e., the edge (xy) belongs to $E(G')$, then there is only one geodesic path and it does not contain the central node v which implies that $b_{xy}^{G'}(v) = 0$. That also holds for G since the edge (xy) also exists in G so $b_{xy}^G(v) = 0$. Therefore $b_{xy}^G(v) - b_{xy}^{G'}(v) = 0$ for a pair $x,y \in N_v^G$ at distance 1. Also, there is no pairs of nodes x,y at distance 3 or more since it exist the path $< x v y >$ in both G and G'.

Consider a pair of nodes $x,y \in N_v^G$ where x is at distance 2 from y in G'. If none of the geodesic paths from x to y contains u in G', then the number of geodesic paths from x

to y (containing v or not) does not change from G to G'. So we have that $b^G_{xy}(v) - b^{G'}_{xy}(v) = 0$. Thus we are interested in the case that a given pair $x, y \in N^G_v$ is at distance 2 where u belongs to a geodesic path from x to y in G' (consequently, also in G'). All geodesic paths from x to y from G are preserved in G' as no edges were removed. But there is a new path $< xuy >$ in G' so $g^{G'}_{xy} = g^G_{xy} + 1$ and as there is only one path $< xvz >$ that contains the central node V in G and G', $g^G_{xy}(c) = g^{G'}_{xy}(c) = 1$. Then

$$b^G_{xy}(v) - b^{G'}_{xy}(v) = \frac{1}{g^G_{xy}} - \frac{1}{g^G_{xy}+1} \tag{A.18}$$

Note that $(1/g^G_{xy}) - (1/(g^G_{xy}+1))$ is monotonically decreasing on g^G_{xy} since

$$\frac{d}{dg^G_{xy}}\left[\frac{1}{g^G_{xy}} - \frac{1}{g^G_{xy}+1}\right] = -\frac{1}{(g^G_{xy}+1)^2} - \frac{1}{(g^G_{xy})^2} < 0$$

so since $b^G_{xy}(v) \leq 1$, we have $b^G_{xy}(v) - b^{G'}_{xy}(v) \leq 1/2$. Besides that, we count how many possible path of the form $< xuy >$ for $x, y \in N^G_c$ where $x \neq y$. Since there are $d = |N^G_c|$ there exists at most $\binom{d}{2} = d(d-1)/2$ of those paths. Thus we have:

$$0 \leq \sum_{x,y \in N^G_v | x \neq y} (b^G_{xy}(v) - b^{G'}_{xy}(v)) \leq \frac{d(d-1)}{2}.\frac{1}{2} = \frac{d(d-1)}{4}$$

Thus

$$\begin{aligned} d &\leq \sum_{x,y \in N^G_v | x \neq y} (b^G_{xy}(v) - b^{G'}_{xy}(v)) - \sum_{y \in N^G_v} b^{G'}_{uy}(v) \leq \frac{d(d-1)}{4} \\ \implies & \left| \sum_{x,y \in N^G_v | x \neq y} (b^G_{xy}(v) - b^{G'}_{xy}(v)) - \sum_{y \in N^G_v} b^{G'}_{uy}(v) \right| \\ & \leq \max\left(\frac{d(d-1)}{4}, d\right) \\ \implies & \Delta(x) = \max\left(\frac{d(d-1)}{4}, d\right) \end{aligned}$$

Case (2): None of the ends of e is v. We omit this part of the proof since it has similar reasoning to case 1 and it yields a lower sensitivity than case 1. □

APPENDIX B – PRIVATESQL'S SQL QUERY

The SQL query $\mathcal{Q}(u,v,c)$ over the table of nodes node(id), the table of all pairs of nodes node_pair(id1, id2) and private table edge(a,b) is given as:

```sql
SELECT COUNT(*)
FROM edge e1, edge e2,
  (SELECT np1.id1 AS idd1, np1.id2 as idd2, COALESCE (CNT_2,0) AS CNT_1
     FROM node_pair AS np1 LEFT OUTER JOIN
        (SELECT np2.id1 AS id1, np2.id2 as
                id2, COUNT (*) AS CNT_2
        FROM node_pair AS np2, edge e3
        WHERE e3.a = np2.id1 AND
              e3.b = np2.id2
        GROUP BY np2.id1, np2.id2) AS pair2
        ON np1.id1 = pair2.id1
        AND np1.id2 = pair2.id2) AS magic1,
    (SELECT e4.a AS m2id,
            count(*) AS CNT_4
    FROM edge e4
    WHERE e4.b = c
    GROUP BY e4.a) AS magic2,
    (SELECT e5.a AS m3id,
            count(*) AS CNT_5
    FROM edge e5
    WHERE e5.b = c
    GROUP BY e5.a) AS magic3,
    (SELECT node.id as m4id,
            COALESCE(CNT_7, 0) as CNT_6
    FROM node left outer join
        (SELECT e6.a AS m5id,
                count(*) AS CNT_7
        FROM edge e6
        WHERE e6.b = c
        GROUP BY e6.a)
        AS magic5 on node.id=magic5.m5id
    WHERE magic5.CNT_7>0 OR
          node.id=c) AS magic4
WHERE e1.b = e2.a AND
```

```
        e1.a = u AND
        e2.b = v AND
        e1.a = idd1 AND
        e2.b = idd2 AND
        CNT_1 = 0 AND
        m2id = e1.a AND
        m3id = e2.b AND
        m4id = e2.a;
```