Local Dampening: Differential Privacy for Non-numeric Queries via Local Sensitivity

Victor Aguiar Evangelista de Farias[1]
Advisor: Javam de Castro Machado[1]

[1]Mestrado e Doutorado em Ciência da Computação
Universidade Federal do Ceará (UFC)
Campus do Pici, Fortaleza - CE – Brazil

{victor.farias, javam.machado}@lsbd.ufc.br

Abstract. *Differential privacy is the state-of-the-art formal definition for data release under strong privacy guarantees. We present the local dampening mechanism a differentially private mechanism for non-numeric queries. Our approach is the first to leverage the notion of local sensitivity to reduce noise injected to the output. We develop a theoretical accuracy analysis to show the conditions that our approach performs accurately and we conduct a experimental evaluation with competitors on diverse problems. Those contributions were published on VDLB conference and on the special issue of the VLDB journal. Non-related contributions were published in SBBD, SBRC, CLOSER and FGCS. This work was carried out in cooperation with AT&T Labs Research - USA.*

1. Introduction

Recent regulations on data privacy, such as *General Data Protection Regulation (GPDR)* [Commission 2018] and *Lei Geral de Proteção de Dados Pessoais (LGPD)* [Brasil 2018], pose strict privacy requirements when gathering, storing and sharing data. Specifically, they require that an individual's information be rendered anonymous so that the individual is no longer identifiable in the published information.

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. The key intuition is that the original analyst's query is replaced by a random algorithm where the output distribution of answers should not change significantly based on the presence or absence of an individual. It provides statistical guarantees against the inference of private information through the use of auxiliary information.

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

For a numeric query (i.e., query with numeric output) f, the *Laplace mechanism* [Dwork et al. 2006b] is a well-known output perturbing private method. It adds numeric noise to the output of f and calibrates the noise based only on f and not on x. The noise magnitude is proportional to the concept of *global sensitivity*, which measures the worst-case impact on f's output of the addition or removal of an individual over the set of possible input databases. This may result in an unreasonably high amount of noise when x is far from the database with the worst-case impact, which is the case for many realistic databases. To remedy this, Nissim et al. [Nissim et al. 2007] proposed to add 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 database instance x, resulting in a lower bound to the global sensitivity.

For the class of non-numeric queries f, i.e. f has a non-numeric range $\mathcal{R}$, the *exponential mechanism* [McSherry and Talwar 2007] ensures differential privacy by sampling elements from $\mathcal{R}$ using the exponential distribution. This requires a utility function $u(x, r)$ that takes as input a database x and an element $r \in \mathcal{R}$ and outputs a numeric score that measures the utility of r. The larger $u(x, r)$, the higher the probability of the exponential mechanism outputting r. The exponential mechanism uses a similar notion of global sensitivity to that found in [Dwork et al. 2006b] where it measures the worst-case impact on the utility $u(x, r)$ for all elements $r \in \mathcal{R}$ by adding or removing an individual from all databases. However, to the best of our knowledge, the literature lacks generic mechanisms that apply local sensitivity to the non-numeric setting.

1.1. 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 $x \in \mathcal{D}^n$ where each entry represents an individual tuple, and $\mathcal{D}$ is the set of all possible tuple values. The function $f : \mathcal{D}^n \rightarrow \mathcal{R}$ receives the dataset $x \in \mathcal{D}^n$ to be evaluated and outputs an element r in its non-numeric range $\mathcal{R}$.

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 1).

Definition 1. *(ϵ-Differential Privacy [Dwork et al. 2006a, Dwork et al. 2006b]). A randomized algorithm $\mathcal{M}$ satisfies ϵ-differential privacy, if for any two databases x and y satisfying $d(x, y) \leq 1$ and for any possible output O of $\mathcal{M}$, we have*

$$Pr[\mathcal{M}(x) = O] \leq \exp(\epsilon) Pr[\mathcal{M}(y) = O],$$

where $Pr[\cdot]$ denotes the probability of an event and d denotes the hamming distance between the two databases, i.e. the number of tuples of individuals that changed value, i.e., $d(x, y) = |\{i \mid x_i \neq y_i\}|$. We refer to d as the distance between two given databases.

1.2. Contributions

We propose the *local dampening mechanism*, which adapts the notion of local sensitivity to the non-numeric setting and uses it to dampen the utility function u in order to increase the signal-to-noise ratio. Local dampening also employs the exponential distribution as the exponential mechanism [McSherry and Talwar 2007]. Applications in which local sensitivity is significantly smaller than global sensitivity can benefit from our approach. For the scenario where local sensitivity is near the global sensitivity, the local dampening mechanism reverts to the exponential mechanism.

To this end, we present a new version of the local sensitivity, called *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 $r \in \mathcal{R}$. Element local sensitivity computes this impact, but only for some given element $r \in \mathcal{R}$. This 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 $\mathcal{R}$, the element local sensitivity is low.

We illustrate the effectiveness of the local dampening mechanism by applying it to three diverse problems: (i) Influential Node analysis, which searches 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; (ii) We also provide an application on tabular data that is a private adaptation to the ID3 algorithm to build a decision tree from a given tabular dataset based on the information gain for each attribute and; (iii) Percentile selection problem, where the task is to release the label of the p-th percentile element.

Our contributions are summarized as follows:

- We adapt the local sensitivity definition to the non-numeric setting and we introduce a new definition of local sensitivity that measures sensitivity per element.
- We introduce the Local Dampening mechanism, a novel differentially private mechanism to answer non-numeric queries that applies local sensitivity to attenuate the utility function to increase the signal-to-sensitivity ratio to reduce noise.
- We present a second version of our approach which we call 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 under 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 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. Additionally, we perform a empirical scalability analysis of the proposed algorithm and show a sub-quadratic runtime behavior.
- 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 improve accuracy up to 12% compared to previous works.
- We tackle the Percentile Selection problem where a private mechanism should report the label of the p-th percentile element. Empirical results show that the

local dampening mechanism can improve up to 73% over global sensitivity based approaches.

The main results of this thesis were published in [de Farias et al. 2020] (PVLDB) and [Farias et al. 2023] (VLDB Journal). Additionally, side contributions on cloud computing were developed and published in [Farias et al. 2017] (SBBD), [Paula et al. 2017] (SBRC), [Cavalcante et al. 2018] (CLOSER), [Farias et al. 2018] (FGCS) and [Lima et al. 2018] (SBBD).

In this paper, we summarize the main contributions of the thesis. We omit the percentile selection problem, the scalability analysis and the comparison to the PrivateSQL approach in the node influential selection. All the proofs are deferred to the thesis [Farias 2021].

Section 2 surveys related work. The local dampening mechanism is presented in Section 3. Section 4 addresses the influential node analysis problem and the Section 5 discuss the decision tree induction problem. Section 6 concludes the paper.

2. 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 the two available differential privacy approaches for the non-numeric, also known as the selection problem, setting in the literature, the exponential mechanism and the permute-and-flip mechanism.

2.1. Exponential Mechanism

The exponential mechanism [McSherry and Talwar 2007] privately answers a function $f : \mathcal{D}^n \rightarrow \mathcal{R}$ applied to database x by sampling an element $r \in \mathcal{R}$ with probability proportional to its utility score $u(x, r)$. It uses the exponential distribution to assign probabilities for each $r \in \mathcal{R}$. The exponential mechanism is stated as follows:

Definition 2. *(Exponential Mechanism [McSherry and Talwar 2007]). The exponential mechanism $M_{EM}(x, \epsilon, u, \mathcal{R})$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to* $\exp\left(\frac{\epsilon\, u(x,r)}{2\Delta u}\right)$.

2.2. Permute-and-Flip

The *permute-and-flip* mechanism,M_{PF}, [McKenna and Sheldon 2020] is recent work that also addresses 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 defines permute-and-flip approach.

3. Local Dampening Mechanism

We present the *local dampening mechanism*, an output perturbing differentially private mechanism for the non-numeric setting that uses local sensitivity to reduce the noise injected to the true answer. Our approach uses the global and local sensitivity notions:

Definition 3. *(Global Sensitivity Δu [McSherry and Talwar 2007]). Given a utility function $u : \mathcal{D}^n \times \mathcal{R} \rightarrow \mathbb{R}$ that takes as input a database $x \in \mathcal{D}^n$, where n is the size of*

Algorithm 1: Permute-and-Flip

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

the database, and an element $r \in \mathcal{R}$ and outputs a numeric score for r in x. The global sensitivity of u is defined as:

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

Definition 4. *(Local Sensitivity, adapted from [Nissim et al. 2007]). Given a utility function $u(x, r)$ that takes as input a database x and an element r and outputs a numeric score, the local sensitivity of u is defined as*

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

Definition 5. *(Local Sensitivity at distance t, adapted from [Nissim et al. 2007]). Given a utility function $u : \mathcal{D}^n \times \mathcal{R} \to \mathbb{R}$ that takes as input a database $x \in \mathcal{D}^n$ and an element $r \in \mathcal{R}$ and outputs a numeric score for r in x, the local sensitivity at distance t of u is defined as*

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

Additionally, we introduce a new notion of sensitivity called *element local sensitivity*. It measures the worst impact on the sensitivity for a given element $r \in \mathcal{R}$ 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.

More broadly, we coin the notion of *sensitivity function* that generalizes local sensitivity definitions. A sensitivity function is a function that computes one of the notions of sensitivity or an upper bound on it.

The local dampening mechanism employs 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.

3.1. Element Local Sensitivity

The local sensitivity at distance t, $LS^u(x, t)$, quantifies the maximum sensitivity of u over all elements $r \in \mathcal{R}$ for an input database x with t modifications (Definition 5). That gives a high-level description of the variation of u in neighboring databases. However, if just one element in $\mathcal{R}$ 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.

Definition 6. *(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*

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

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

Note that we can obtain $LS^u(x,t)$ from this definition: $LS^u(x,t) = \max_{r\in\mathcal{R}} LS^u(y,t,r)$ as $LS^u(x,t,r) = \max_{y|d(x,y)\leq t} LS^u(y,0,r)$.

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 $\delta^u : \mathcal{D}^n \times \mathbb{N}^0 \times \mathcal{R} \rightarrow \mathbb{R}$. 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 of sensitivity functions based on four properties: admissibility, boundedness, monotonicity and stability.

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

Definition 7. *(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 $x \in \mathcal{D}^n$ and all $r \in \mathcal{R}$*
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 admissible as $\Delta u \geq LS^u(x,0,r)$, for all x and a constant value would satisfy the second requirement of Definition 7. We also show that the sensitivity functions $LS^u(x,t)$ $LS^u(x,t,r)$ are admissible.

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

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 $r \in \mathcal{R}$, we say that $\delta^u(x,t,r)$ is monotonic. We have two classifications for monotonicity: i) Non-decreasing Monotonicity, presented below, and; ii) Non-increasing Monotonicity which is the symmetric version of the Non-decreasing Monotonicity. The definition of the latter is not provided because of space constraints.

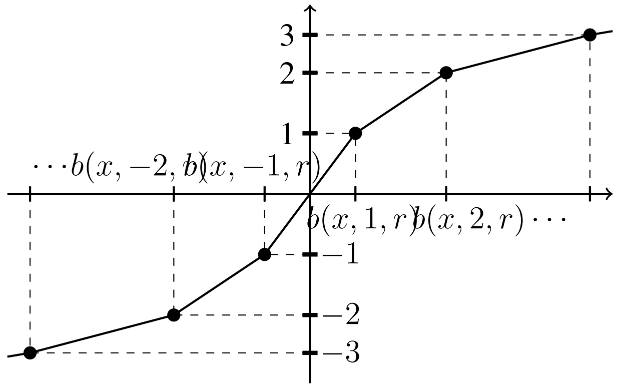


Figura 1. Dampening function D_{u,δ^u}

Definition 9. *(Non-decreasing Monotonicity) Let $u(x,r)$ be a 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 $x \in \mathcal{D}^n$, $r, r' \in \mathcal{R}$, $t \geq 0$ such that $u(x,r) \geq u(x,r')$.*

Stability. Satisfying 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.

Definition 10. *(Stability) A sensitivity function $\delta^u(x,t,r)$ is stable if δ^u is admissible, bounded and monotonic.*

3.3. Local Dampening Mechanism

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.

Definition 11. *(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:

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

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 $u(x,r) \in [b(x,i,r), b(x,i+1,r))$.

Thus, the local dampening mechanism is defined as:

Definition 12. *(Local dampening mechanism). The local dampening mechanism $M_{LD}(x,\epsilon,u,\delta^u,\mathcal{R})$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\epsilon\, D_{u,\delta^u}(x,r)}{2}\right)$.*

3.4. Shifted Dampening Mechanism

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.

The key idea for this version 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 in the experiments that the shifted local dampening also performs well for non strictly monotonic functions.

We propose to replace the original utility function u with its shifted version $u^s = u(x, r) - s$. where s is the utility score shift.

In what follows, the shifted local dampening mechanism is stated as follows:
Definition 13. *(Shifted Local Dampening Mechanism for non-decreasing sensitivity function). The shifted local dampening mechanism $M_{SLD}(x, \epsilon, u, \delta^u, \mathcal{R})$ outputs an element $r \in \mathcal{R}$ with probability equals to*

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

The Shifted Local Dampening Mechanism for non-decreasing sensitivity function is symmetric to the Definition 13.

3.5. 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 $\mathcal{M}$ by studying the error random variable $\mathcal{E} = u^* - u(x, \mathcal{M}(x))$ where u^* is the optimal utility score, $u^* = \max_{r\in\mathcal{R}} u(x, r)$ where u^* is the optimal utility score, $u^* = \max_{r\in\mathcal{R}} u(x, r)$.

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.

3.5.1. 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 14. *(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 $\alpha(x, t, r)$ refer to the gap between $\delta^u(x, t, r)$ and $\bar{\delta}^u(x, t, r)$: $\alpha(x, t, r) = \bar{\delta}^u(x, t, r) - \delta^u(x, t, r)$. Assume that $\mathcal{R} = \{r_1, ..., r_q\}$ 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 1. *(Shifted 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. *$Pr[\mathcal{E}(M_{SLD}, x) \geq \theta] \leq Pr[\mathcal{E}(\overline{M}_{SLD}, x) \geq t]$ for all $\theta \geq 0$,*
2. *$\mathbb{E}[\mathcal{E}(M_{SLD}, x)] \leq \mathbb{E}[\mathcal{E}(\overline{M}_{SLD}, x)]$,*

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

Strict monotonicity may be not be satisfy for given $LS^u(x, t, r)$ or any $\delta^u(x, t, r)$. However, we argue that if $\delta^u(x, t, r)$ exhibit a correlation of $u(x, r)$ and $\delta^u(x, t, r)$ with respect to r, the shifted local dampening stills yield good results.

Our empirical results corroborates with this argument. In the applications and datasets analyzed 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.

3.5.2. 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 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 $r \in \mathcal{R}$. It implies that $\delta^u(x, t, r)$ dominates Δu. Thus the following corollary holds:

Corollary 1. *Let δ^u be a stable function. The shifted local dampening mechanism $M_{SLD}(x, \epsilon, u, \delta^u, \mathcal{R})$ is never worse than the exponential mechanism $M_{EM}(x, \epsilon, u, \mathcal{R})$. That is:*

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

This result 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 2. *$LS^u(x, t, r)$ is minimum admissable, i.e. $LS^u(x, t, r)$ dominates any admissible sensitivity function $\delta^u(x, t, r)$.*

4. Application 1: Influential Node Analysis

Identifying influential nodes in a network is an important task for social network marketing [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.

4.1. Problem statement

The influential node analysis problem is a query over an input graph database $G = (V, E)$ that releases the labels of k nodes that maximize a given influence metric. In this work, we use the Egocentric Betweenness Centrality metric (EBC, Definition 15).
Definition 15. *(Egocentric Betweenness Centrality (EBC))*

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

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.

4.2. Private Mechanism

We propose *PrivTopk*, a top-k algorithm template that chooses iteratively k nodes that maximize EBC. In each iteration, the algorithm makes a call to a non-numeric mechanism that returns a node that maximizes EBC that was not previously chosen. We experiment with four instances of this algorithm template: i) *EMPrivTopk*, where the non-numeric mechanism is the exponential mechanism; ii) *PFPrivTopk*, where the non-numeric mechanism is the permute-and-flip mechanism; iii) *LDPrivTopk* where the non-numeric mechanism is the local dampening mechanism; iv) *SLDPrivTopk* where the non-numeric mechanism is the shifted local dampening mechanism.

4.3. Sensitivity Analysis

Global Sensitivity. We need to provide the global sensitivity for EBC to the Exponential Mechanism and the permute-and-flip mechanism:
Lemma 3. *(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.

Element Local Sensitivity. For the local dampening call, we provide an upper bound to the element local sensitivity using the sensitivity function δ^{EBC}:
Definition 16. *(Sensitivity function $\delta^{EBC}(G, t, v)$). The sensitivity function δ^{EBC} for EBC is defined as*

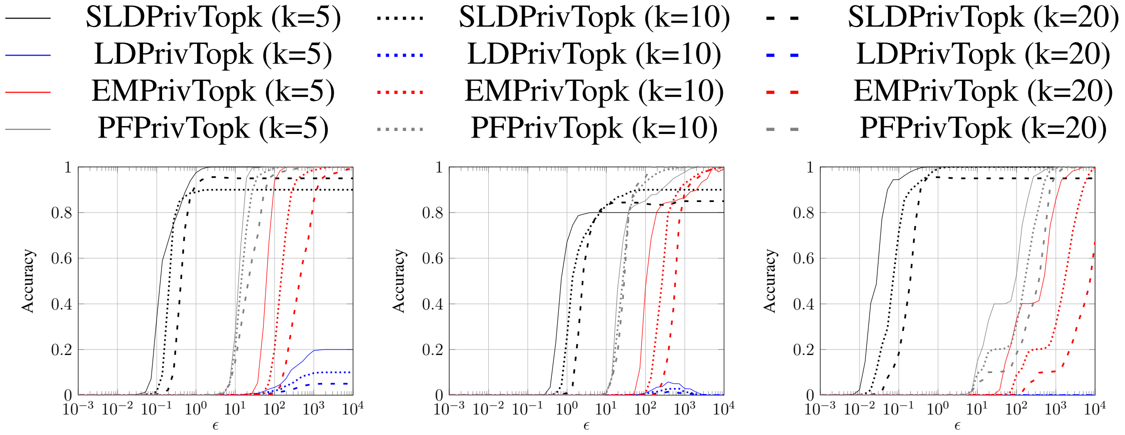
$$\delta^{EBC}(G, t, v) = \max\left(\tfrac{(d^G(v)+t)(d^G(v)+t-1)}{4}, d^G(v) + t\right),$$

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

4.4. 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 and Krevl 2014].

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 $\epsilon \in [10^{-3}, 10^4]$.



(a) Enron Dataset **(b) DBLP Dataset** **(c) Github Dataset**

Figura 2. Accuracy for PrivTopk algorithm for $k \in \{5, 10, 20\}$ and $B \in [10^{-3}, 10^4]$.

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

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

5. Application 2: ID3 Decision Tree Induction

Classification based on decision tree is an important tool for data mining [Kotsiantis et al. 2007]. 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].

The ID3 algorithm [Quinlan 1986] starts with the root node containing the original set. Then the algorithm greedily 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.

5.1. Problem Statement

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

5.2. Private Mechanism

We use the algorithm DiffPID3 [Friedman and Schuster 2010] (referred as GlobalDiffPID3) as a template. We aim to adapt it for the use of the local dampening mechanism and to the shifted local dampening mechanism producing the *LocalDiffPID3* and *ShiftedLocalDiffPID3*, respectively. In the following, we need to provide the global sensitivity of the split criterion (Information Gain IG) for the exponential mechanism and the element local sensitivity for the local dampening. The global sensitivity for IG is given by $\Delta IG = \log(N+1) + 1/\ln 2$ [Friedman and Schuster 2010] where N is the size of the dataset $\mathcal{T}$. First, we show that the element local sensitivity at distance 0 is given by:

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

where

$$\begin{aligned} h(a,b) &= \max(f(a) - f(b), g(b) - g(a)), \\ g(x) &= x.log((x-1)/x) - log(x-1), \\ f(x) &= x.log((x+1)/x) + log(x+1). \end{aligned}$$

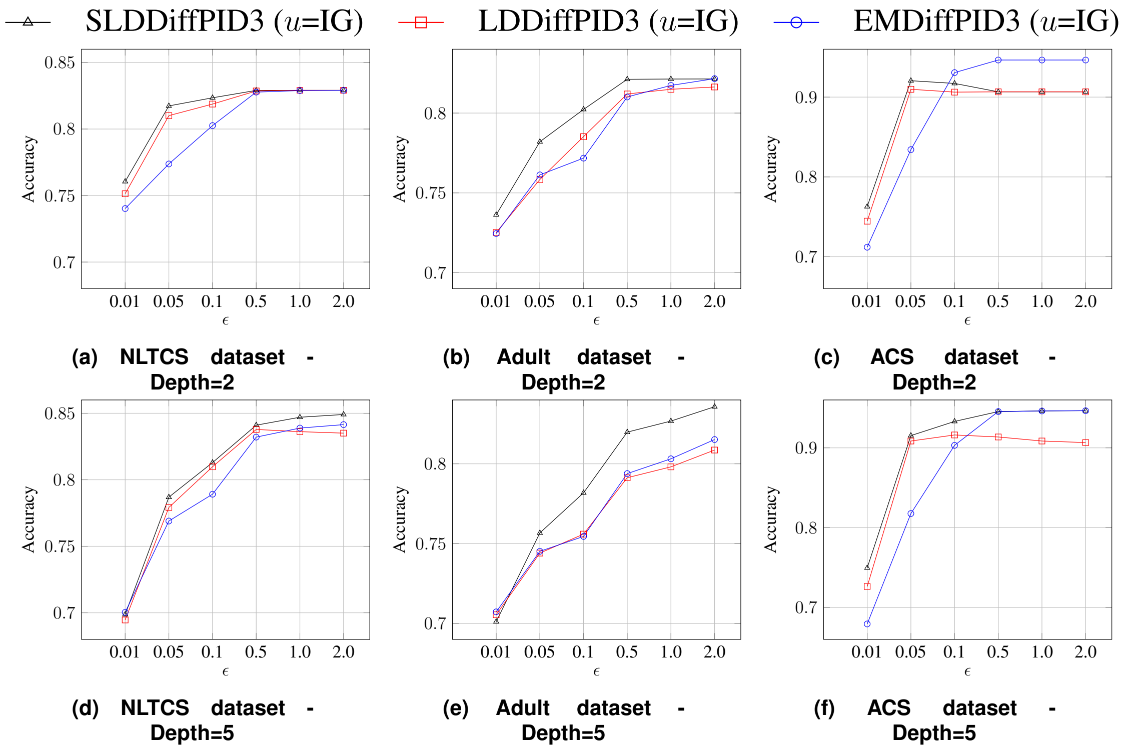
Then, element local sensitivity at distance t is:

$$LS^{IG}(\mathcal{T}, t, A) = \max_{c \in C, j \in A} \max_{\mathcal{T}' | d(\mathcal{T}, \mathcal{T}') \leq t} h(\tau_j^{A,\mathcal{T}'}, \tau_{j,c}^{A,\mathcal{T}'}).$$

5.3. 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 and Merz 1998] contains $45,222$ records (excluding records with missing values) with 12 attributes where 8 are discrete and 4 are continuous.

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 typical values for $depth$ and ϵ: $depth \in \{2, 5\}$ and $\epsilon \in \{0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$.



(a) NLTCS dataset - Depth=2

(b) Adult dataset - Depth=2

(c) ACS dataset - Depth=2

(d) NLTCS dataset - Depth=5

(e) Adult dataset - Depth=5

(f) ACS dataset - Depth=5

Figura 3. Accuracy for DiffID3 algorithm

Figure 3 presents the results. 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%. Fot the Adult Dataset, the LocalDiffPID3 improves a little over the GlobalDiffPID3. However, ShiftedLocalDiffPID3 improves over GlobalDiffPID3 up to 4%.

6. Conclusion

In this paper, 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 magnitude of the noise added to achieve Differential Privacy which makes the answer of those queries more useful. We evaluated our approach on three applications: 1) Influential node analysis; 2) Decision Tree induction and; 3) Percentile Selection (omitted).

The thesis results described in this paper have laid the foundations for providing DP for non-numeric queries using local sensitivity. We have achieved a deeper theoretical understanding of the Local Dampening mechanism to understand the class of problems for which it can provide significant gains over the Exponential mechanism. There are many stimulating directions for future work. First, any problem in the literature that has used the Exponential mechanism for non-numeric queries to guarantee DP is a candidate problem that could benefit from using our local dampening mechanism instead and is worthy of future work. Second, tackling other graph influence/centrality metrics for Influential Node analysis, such as PageRank, would be interesting. Third, applying the local dampening mechanism for private evolutionary algorithms is a promising future direction.

Referências

Blake, C. L. and Merz, C. J. (1998). Uci repository of machine learning databases.

Brasil (2018). Lei geral de proteção de dados pessoais (lgpd).

Cavalcante, D. M., de Farias, V. A., Sousa, F. R., Paula, M. R. P., Machado, J. C., and de Souza, J. N. (2018). Popring: A popularity-aware replica placement for distributed key-value store. *CLOSER*, 2018:440–447.

Commission, E. (2018). 2018 reform of eu data protection rules.

de Farias, V. A. E., Brito, F. T., Flynn, C., Machado, J. C., Majumdar, S., and Srivastava, D. (2020). Local dampening: Differential privacy for non-numeric queries via local sensitivity. *Proc. VLDB Endow.*, 14(4):521–533.

Dwork, C. (2011). Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Farias, V. (2021). *Local dampening: differential privacy for non-numeric queries via local sensitivity*. PhD thesis, Universidade Federal do Ceará.

Farias, V., Pinheiro, P., Sousa, F., Gomes, J., and Machado, J. (2017). Online performance modeling for nosql databases using extreme learning machines. In *Anais do XXXII Simpósio Brasileiro de Bancos de Dados*, pages 276–281, Porto Alegre, RS, Brasil. SBC.

Farias, V. A., Brito, F. T., Flynn, C., Machado, J. C., Majumdar, S., and Srivastava, D. (2023). Local dampening: Differential privacy for non-numeric queries via local sensitivity. *The VLDB Journal*, pages 1–24.

Farias, V. A., Sousa, F. R., Maia, J. G. R., Gomes, J. P. P., and Machado, J. C. (2018). Regression based performance modeling and provisioning for nosql cloud databases. *Future Generation Computer Systems*, 79:72–81.

Friedman, A. and Schuster, A. (2010). Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–502.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Lima, M. I., de Farias, V. A., Praciano, F. D., and Machado, J. C. (2018). Workload-aware parameter selection and performance prediction for in-memory databases. In *Anais do XXXIII Simpósio Brasileiro de Banco de Dados*, pages 169–180. SBC.

Ma, H., Yang, H., Lyu, M. R., and King, I. (2008). Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 233–242.

Machanavajjhala, A., He, X., and Hay, M. (2017). 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*, pages 1727–1730. ACM.

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

McKenna, R. and Sheldon, D. R. (2020). Permute-and-flip: A new mechanism for differentially private selection. *Advances in Neural Information Processing Systems*, 33.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103.

Nissim, K., Raskhodnikova, S., and Smith, A. (2007). Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM.

Paula, M. R. P., Rodrigues, E., Farias, V. A., Sousa, F. R., and Machado, J. C. (2017). Bacos: A dynamic load balancing strategy for cloud object storage. In *Anais do XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Series, I. P. U. M. (2015). Version 6.0. *Minneapolis: University of.*

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. (2015). Private release of graph statistics using ladder functions. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 731–745. ACM.