Private Reverse Top-k Algorithms Applied on Public Data of COVID-19 in the State of Ceará

Maria de Lourdes M. Silva[1], Iago C. Chaves[1], Javam C. Machado[1]

Computer Science Department
Federal University of Ceará, Brazil
{malu.maia,iago.chaves,javam.machado}@lsbd.ufc.br

Abstract. In this article we propose a differentially private reverse top-k query. Our strategy allows obtaining the less frequent data according to a search criteria, with a high guarantee of privacy of the individuals who contributed with personal data in the original database. We apply our strategy on public data for COVID-19 in the State of Ceará using two different queries. Our experimental results show that the result of the proposed top-k query returns a high degree of similarity to the result of a conventional top-k query, when the chosen budget is suitable, providing useful results for researchers, while ensuring a low probability of re-identification of individuals arising from the properties of differential privacy.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Authorization, Privacy and Security**]: Miscellaneous

Keywords: COVID-19, differentially private reverse top-k approaches, utility

1. INTRODUCTION

Publishing data is essential for scientific research, whether on validating experiments or developing new scientific tools. The importance of publishing data is even more evident in the health context, where experiments are intense, and epidemic control depends on comprehensive data analysis. The publication of health information, shared by public organizations, helps to increase transparency in this sector. However, health data must be shared carefully to make it challenging to re-identify patients, considering the sensitive nature of this type of data and providing high levels of utility so that data analysis may be correctly done.

The importance of publishing health data has been intensified after the COVID-19 pandemic (*Coronavirus Disease 2019*). Uncountable institutions published data about the patients with COVID-19 [SUS 2020] including the government of the state of Ceará, in Brazil. To illustrate the severity of the pandemic in the state of Ceará at municipal levels, Figure 1 represents the percentage of infected people in cities of the state using COVID-19 data updated on January 13, 2021. This percentage was calculated considering the total resident population in each city. Although Fortaleza has the highest number of infected citizens, it is not the highest percentage. On the other hand, Acarape and Crateus, the dark blue area in Figure 1, present a high level of contamination, considering that they are not among the most crowded cities.

Information about patients is published at a microdata level, where each tested individual is represented as a record in public data. At the same time, before releasing information, techniques to suppress some attributes are applied. The objective is to make it harder to re-identify the individuals tested for Covid-19 by suppressing the name and social security number of the dataset.

This research was partially supported by FUNCAP - Brazil and LSBD/UFC.

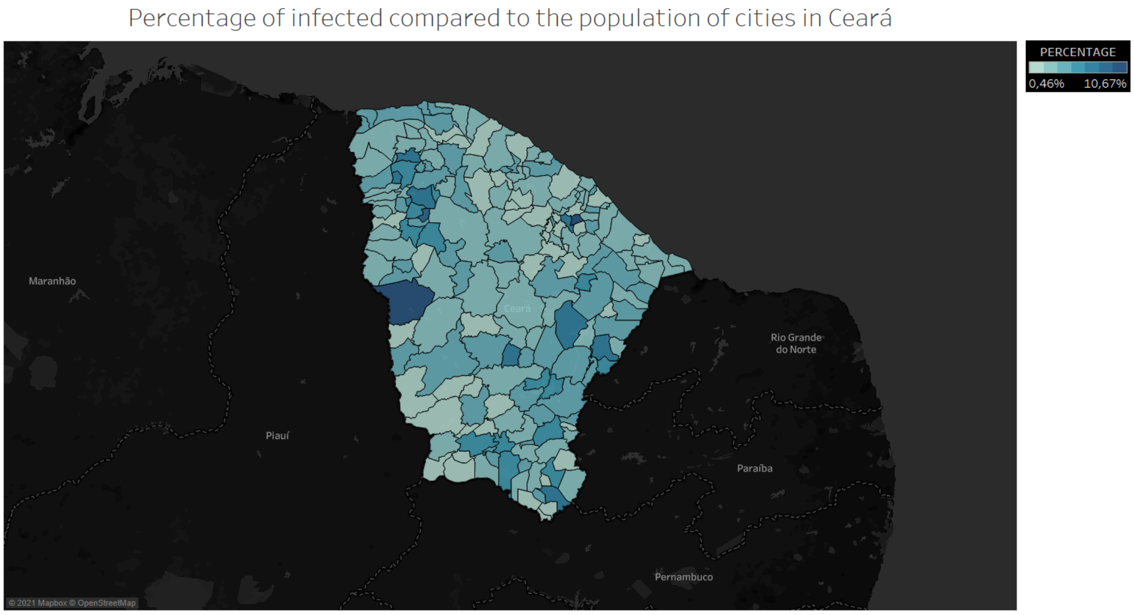
Percentage of infected compared to the population of cities in Ceará
PERCENTAGE
0,46%
10,67%
Maranhão
Ceará
Piauí
Rio Grande do Norte
Paraíba
Pernambuco
© 2021 Mapbox © OpenStreetMap

Fig. 1. Percentage of people infected by COVID-19 in Ceará.

Many studies show that suppression of identifiers is not sufficient to ensure the privacy of individuals [Narayanan and Shmatikov 2006]. Suppression and obfuscation techniques are not enough to guarantee patient's privacy, especially for individuals who have unique characteristics. For example, patients with addresses in geographic regions with a low disease incidence can be identified by mining the published dataset. Differential privacy [Dwork 2010] has been used successfully in publishing data that contains sensitive information about individuals. Its formal properties guarantee the privacy of individuals who contribute with their data while maintaining an acceptable level of use of data for scientific research.

Top-k queries are used to discover the k most frequent elements in a data set. These are queries that discriminate because they order the data according to their properties. In this sense, they have great potential to learn about the data, including discovering individuals who were included in the query output, affecting their privacy. Reverse top-k queries have even more significant potential, as they result in the least frequent k elements [Vlachou et al. 2010].

This article studies the reverse top-k query approach, seeking to extend it with differentially private mechanisms, such as Laplace, Exponential, and Permute-and-Flip. We apply top-k queries over Ceará's COVID-19 dataset to study its properties, considering different types of mechanisms. Our approach allows for the publication of health data, specifically when one uses the reverse top-k query to learn about the data, ensuring the privacy of individuals and acceptable levels of utility for this type of query. Our results can be published, and we specified the ideal budget value for each approach.

The main idea of our approach is divided into two steps:

(1) Achieving the reverse top-k query with differential privacy by adding noise to count values for Laplace mechanism, or adding randomness drawn by exponential distribution to the choice of an item for Exponential and Permute-and-Flip mechanisms;
(2) Selecting the k least frequent items based on their noisy count values, for Laplace, or select k items chosen by the first step, for Exponential and Permute-and-Flip.

This article is an extended version of [Silva et al. 2020], published on the 35th edition of the

Brazilian Symposium on Databases (SBBD); its remainder is organized as follows. Section 2 is a brief presentation of the related literature. Section 3 describes the theoretical background knowledge that is necessary to the understanding of the algorithm. Next, Section 4 presents the problem and the queries used in the algorithms, our approaches are presented in Section 5, and the results in Section 6. Finally, we conclude the article in Section 7.

2. RELATED WORK

Several works address the top-k problem privately; however, the literature lacks studies in the context of private reverse top-k. Those approaches deal with the common frequent itemset mining problem while the present work tackles the less frequent itemsets. The idea of first finding the frequent itemsets, in our case the less frequent ones, and then adding noise or randomness remains.

[Cheng et al. 2015] propose a new algorithm, which is called DP-Apriori, that achieves differential privacy in the problem of Frequent Itemset Mining (FIM). Their method provides a high level of utility and a high level of privacy while solving the problem of dealing with long transactions (transactions that contain lots of items). One possible way to solve it is by limiting the cardinality of the transactions using transaction truncating. The authors opted to use a novel transaction splitting approach that they proposed; it uses two techniques: smart weighted splitting and support estimation. The first one divides a transaction that exceeds the maximal cardinality into sub-transactions. Each sub-transaction is assigned a weight, and the sum of the weights of all sub-transactions is equal to the weights of the original transaction. The other technique estimates the actual support of an itemset, including its average support, to determine if an itemset is frequent. It also defines maximal support according to its noisy support in the transformed database to determine if it will generate frequent candidate itemsets. The Laplace mechanism is used to add noise to supports. The first stage of DP-Apriori extracts statistical information from the pre-processing phase and gets the matrix used by the support estimation technique. It is added geometric noise to each a_i, where a_i is the number of transactions with length i, and to each s_i, where s_i is the maximal support of i-itemsets. Then, the algorithm starts to mine frequent itemsets using differential privacy. For mining, the approach applies a weighted splitting operation to divide long transactions into sub-transactions randomly. It also applies a smart weighted splitting technique, calculates the support of a candidate, and adds Laplace noise. Therefore DP-Apriori estimates their maximal supports. All the supports are computed during the mining process. To evaluate the approach, the paper shows experimental results comparing to PrivBasis [Li et al. 2012] and Transaction Truncating [Zeng et al. 2012]. DP-Apriori presented better results, using the F1-Score metric to calculate the accuracy of the output.

[Lee and Clifton 2014] propose an approach called NoisyCut, that identifies the top-k frequent itemsets and uses them to construct a differentially private FP-tree. This algorithm uses a modified version of the general sparse vector technique [Hardt and Rothblum 2010], which was originally used to publish few counting queries above a threshold. In this technique, information disclosure spoils differential privacy only occurs for counting queries above the threshold, and negative answers do not count against the privacy budget. The authors modify the sparse vector technique to answer exponentially many thresholds queries given a fixed bound using differential privacy. Similar to the previous related work, the algorithm also has two phases, (1) discover the top-k frequent itemsets, where the most frequent itemsets are identified, and (2) add Laplace noise to support building FP-trees. Differential privacy is then applied, injecting noise in the supports of the frequent itemsets. In the sparse vector algorithm, k counting queries that are above a given threshold T are released. The first stage of this algorithm calculates the noisy threshold using part of the privacy budget, then calculates the noisy output of a query and compares it to the noisy threshold. Its advantage is that it only pays the privacy budget for above threshold queries; therefore, all queries below the bound are answered without wasting the privacy budget. The NoisyCut algorithm first discovers the k most frequent itemsets. Then, it learns the support of the k^{th} most frequent itemset and defines the threshold for its support. Next, a Laplacian noise is applied to get the noisy threshold to ensure differential privacy. For each itemset in the itemset lattice, a verification is done; if its noisy support is above the noisy threshold, the algorithm considers this itemset as frequent. The algorithm divides the itemset lattice into two partitions: one with frequent items and the other with infrequent ones. After this identification, the algorithm builds a noisy FP-tree to derive the chosen frequent itemsets' support. The authors evaluate their algorithm's performance comparing its F1-Score to PrivBasis' [Li et al. 2012] and SmartTruncation' (which is the following related work) using eight different datasets and ten values varying between 0.1 and 1. In all the cases, NoisyCut presented the highest values of accuracy.

In the work of [Zeng et al. 2012], the authors proposed an algorithm that deals with the problem of long transactions by truncating and adding noise to the errors introduced by truncation to achieve differential privacy. They realized that long transactions could cause the trade-off between privacy and utility in frequent itemset mining, so they try to improve the trade-off by limiting the cardinality of the transactions by deleting items in a differentially private way. For their utility model, the intuition is that if the support of an itemset is much larger than a given threshold, then the result should include that itemset. However, whether the support is much smaller than the threshold, this itemset is excluded from the output. To limit the maximal cardinality of transactions, they truncate a transaction that violates cardinality by keeping a subset of the transaction. This truncation spoils the utility so that information may be lost. In their algorithm, they first determine the maximal cardinality in which they set it to the value that the percentage of the transactions with cardinality is no greater than the maximal is at least 85%, it computes the percentage of transaction for each cardinality. It adds geometrical noise, which is a discrete variant of the Laplace mechanism, to its result. Suppose the cardinality of a transaction violates the support threshold constraint.

In that case, only a subset of its items is picked randomly without replacement to generate a new transaction, which its cardinality is equal to the maximal one, adding geometric noise to the number of transactions ensures ε-differential privacy. To their smart truncation, they first provide a heuristic method to predict if a candidate is frequent. To quantify it, they assign each candidate itemset to a frequency score which is the sum of all the noisy supports of its subsets, keeping the itemsets with higher frequency scores. The SmartTruncating algorithm finds the candidate itemsets and picks the itemset with the highest weight. Its items are added in the candidate itemset of an empty transaction T and deleted from the set of candidate itemsets. It then updates the weight of the remaining itemsets in the set of candidates, computes the average weight of an item in the candidate set, and repeats those steps until the cardinality of T exceeds the maximal value. The algorithm yielded higher F1-Score values when compared to PrivBasis [Li et al. 2012], providing utility and privacy to the frequent itemset mining problem.

Table I. Comparative between related work and present work based on used mechanisms.

Article	**Mechanisms**	**Problem approach**
[Cheng et al. 2015]	Laplace and Geometric	Frequent Itemset Mining
[Lee and Clifton 2014]	Laplace	Frequent Itemset Mining
[Zeng et al. 2012]	Geometric	Frequent Itemset Mining
This work	Laplace Exponential Permute and Flip	Reverse Top-k

3. PRELIMINARIES

Differential Privacy is a property of random algorithms, called mechanisms. Its purpose is to ensure that the presence or absence of each individual in the data set will not impact a query response, preventing an opponent from learning about an individual beyond what is already known about him. The most known mechanisms are Laplace [Dwork et al. 2006], to numerical or counting queries, and Exponential [McSherry and Talwar 2007] to categorical queries. In this article, we use the Laplace mechanism, Exponential mechanism, and Permute-and-Flip [McKenna and Sheldon 2020] which is based on the second one.

Definition 3.1 *ε-Differential Privacy.* A mechanism M satisfies ϵ-differential private where $\varepsilon \geq 0$, if and only if for any neighboring datasets D and D', i.e. datasets that differ in only one row, and $T \subseteq Range(M)$ denotes a possible output contained in the set of all possible outputs of the mechanism M, we have:

$$Pr[M(D) \in T] \leq \exp(\varepsilon) \times Pr[M(D') \in T], \tag{1}$$

Considering any $t \in Range(M)$, the condition (1) can be equivalently stated as:

$$\frac{Pr[M(D) = t]}{Pr[M(D') = t]} \leq \exp(\varepsilon), \tag{2}$$

where, for this definition, the fraction $\frac{0}{0}$ is defined to be 1, if it occurs.

This ε is called the budget, and it is the limit for the privacy loss in a query result, $\varepsilon \in (0, \infty)$, where $\varepsilon = 0$ represents the maximum guarantee of privacy and low utility of the data. Meanwhile, $\varepsilon = \infty$ represents the opposite, the perturbed data would be the same as the original, and consequently, the usefulness of the data would be maximum, but there would be no guarantee of privacy.

A property of ε-Differential Privacy that is useful when dealing with our problem is sequential composition [McSherry 2009] because we deal with the problem of the less frequent k items, then it is necessary to make k queries of the type "which is the least frequent item in the data set?".

THEOREM 3.2 SEQUENTIAL COMPOSITION. *Let M_i be a privacy mechanism which provides ε_i-differential privacy. Then, a sequence of $M_i(D)$ over the database D provides $(\sum_i \varepsilon_i)$-differential privacy.*

Readers may refer to the work of [McSherry 2009] for the proof of Theorem 3.2.

In our first approach, the noise is added by the Laplace mechanism. The answer to the problem of frequent itemsets returns a vector, so we consider the vector case of this mechanism. We consider $t \in Range(M)$ as one value of all possible outputs in this mechanism. To satisfy ε-differentially privacy, one can publish $\tilde{f}(D) = f(D) + X$, where $\tilde{f}(D)$ represents the noisy function, X is a random variable drawn from Laplace distribution and noise is added to the output of a function or query $f(D)$. To ensure that the mechanism guarantees ε-differential privacy, the sentence below has to be satisfied.

$$\frac{Pr[\tilde{f}(D) = t]}{Pr[\tilde{f}(D') = t]} \leq \exp \varepsilon \tag{3}$$

Definition 3.3 *Global sensitivity, vector case.* Let $D \simeq D'$ denote that D and D' are neighboring datasets. The global sensitivity of a function f is denoted by Δ_f, and defined as the equation below.

$$\Delta_f = \max_{D \simeq D'} ||f(D) - f(D')||_1 \tag{4}$$

When f is a counting query, the sensitivity is set as 1. It is because the addition or removal of an element will impact the output of f in one count.

Theorem 3.4 Laplace mechanism, vector case. *For any function f whose value is a k-dimensional vector, the Laplace mechanism, defined below, satisfies ε-differential privacy.*

$$M_L(D, f) = f(D) + \langle X_1, X_2, ..., X_k \rangle, \tag{5}$$

where $X_1, X_2, ..., X_k$, with $k = |D|$, are i.i.d. random variables drawn from Laplace distribution, with mean 0 and scale parameter equals to $\frac{\Delta_f}{\varepsilon}$.

Readers may refer to the work [Dwork et al. 2006] for the proof of Theorem 3.4.

We investigate two other mechanisms that do not add noise to output but randomize the answer based on exponential distribution.

Definition 3.5 *Exponential mechanism.* For any utility function u which captures for each record in the data set a score for a possible output, the exponential mechanism $M_E(D, u)$ outputs $t \in Range(M)$, where $Range(M)$ is the set of all possible outputs of the mechanism M, with probability proportional to $\exp(\frac{\varepsilon u(D,t)}{2\Delta u})$, where

$$\Delta u = \max_{\forall t, D \simeq D'} ||u(D, t) - u(D', t)||_1 \tag{6}$$

is the sensitivity of the utility function.

Theorem 3.6 Exponential mechanism. *The exponential mechanism satisfies ε-differential privacy.*

The proof of Theorem 3.6 is on [McSherry and Talwar 2007].

The permute-and-Flip mechanism works by iterating through the set of candidates in random order. The mechanism flips a biased coin for each item and returns that item if the coin comes up heads. The probability of heads is an exponential function of the score function.

Theorem 3.7. *[McKenna and Sheldon 2020] The Permute-and-Flip mechanism M_{PF} is regular and ε-differentially private.*

Readers may refer to the work of [McKenna and Sheldon 2020] for the proof of Theorem 3.7.

Top-k queries are defined based on a score function s that calculates the relevance of each record for a given query or task and results in an ordered list with the k most relevant records according to f,

Definition 3.8 *Top-k.* Let k be a positive integer number, D be a dataset, D' a subset of D and s a score function,

$$\text{TOP}(D, k, s) = \underset{D' \subseteq D, |D'| = k}{\arg\max} \sum_{d \in D'} s(d). \tag{7}$$

For reverse top-k queries, we will use a score function f similar to top-k; however, we will search for the lowest-scoring results. Therefore, the query will result in an inversely ordered list with k lowest-scoring records.

Definition 3.9 *Reverse Top-k.* Given a positive integer k, a data set D, a subset of D as D' and a score function s,

$$\text{RTOP}(D, k, s) = \underset{D' \subseteq D, |D'| = k}{\arg\min} \sum_{d \in D'} s(d). \tag{8}$$

4. METHODOLOGY

In this paper, we propose three private algorithms for the reverse top-k approach using differential privacy. The problem with using reverse top-k, without applying privacy techniques, is that if an opponent has external information, he can identify some sensitive information about a person whose some characteristic is at the output of reverse top-k through linkage attacks. As we saw in Section 3, these mechanisms have different processes. Laplace mechanism adds noise to the result of aggregation queries. For example, in a query that searches the k neighborhoods with the least amount of residents, the algorithm processes the number of residents per neighborhood and, for each neighborhood, adds a random noise drawn from the Laplace distribution. The reverse top-k version using the exponential mechanism has a similar result; however, the process is different. Using the same example, our possible answers are all k-size subsets of the neighborhood set. For each of those subsets, we calculate a score representing the similarity of the subset with the result of the initial response.

To evaluate the proposed algorithms, we used data published by the State of Ceará government on cases of COVID-19, updated on January 13, 2021. The dataset contains records of individuals who have tested for the disease in the state, with the identifying attributes of the patients removed. The data set has 1,048,575 records and 62 attributes. In this work, we will consider three attributes, which are: "resultadoFinalExame" (the result of the exam), "municipioCaso" (city of residence of infected people), and "idadeCaso" (age), two queries were executed and, consequently, two different pre-processes were needed. To compare the quality of the answer after applying privacy techniques, we propose a metric that calculates the error rate in the sequence of the reverse top-k with privacy compared to the real reverse top-k.

We used *Python* version 3.7.6 to implement our algorithms over *Jupyter Notebook*. The experiments were conducted on a PC with Intel Core i7-7800X CPU (3.50 GHz) and 16GB of memory. The code is available at `https://github.com/Brabissima/Private_Reverse_Top-k`. Committed on March 5, 2020.

4.1 Queries executed and Pre-processing

4.1.1 *Age Group Query.* Let us consider one wants to study the distribution of COVID-19 cases over age cohorts. We apply the query `SELECT COUNT(infected) TOP 10 REVERSE IN ages`, where we want the ten ages with fewer people infected by COVID-19. The ages column has eighty-one possible values, and each value represents an age. We created a particular category for elder people, and the last age class represents people aged eighty-one or older. The minimum value of age class is "0 years," and the maximum value of age class is "80 or more years old".

For this query, pre-processing was done as follows, for each value of age, the number of infected people was counted, and we created a new data frame from that, where the index of the data frame is the age range and the value is the count, sorted by the count in ascending order. The answer for this query using the mentioned dataset and not applying privacy techniques is in Table II.

Table II tells that only a few patients that tested positive for COVID-19 are older than 80 years old. That is probably because there are few people in this age group in the State population. One can also see that young patients did not test positive or did not test at all, which is probably the case. Elders and children are therefore more exposed to a privacy information leak.

4.1.2 *City Query.* The other query that we use to evaluate our approach is `SELECT COUNT(infected) TOP 10 REVERSE IN cities`, which is similar to the first one, but now we are searching the cities of infected people. The cities column has 184 possible values, which are the exact number of cities in Ceará. The only difference in pre-processing is that we are considering the count of infected people for each city.

Table II. Reverse top-10 ages of infected by COVID-19.

Top	Age
1	80 or more years old
2	6 years old
3	4 years old
4	7 years old
5	3 years old
6	5 years old
7	8 years old
8	9 years old
9	11 years old
10	10 years old

Table III. Reverse top-10 cities of infected by COVID-19.

Top	Cities
1	Antonina do Norte
2	Tarrafas
3	Penaforte
4	Arneiroz
5	Aiuaba
6	Baixio
7	Mulungu
8	Potengi
9	Jati
10	General Sampaio

The answer to this query is in Table III. It shows that "Antonina do Norte", "Tarrafas", and "Penaforte" are the cities with fewer positive reports for the COVID-19 test. That makes their citizens more suitable for re-identification if the dataset is associated with other sources of information.

5. PRIVATE REVERSE TOP-K

To ensure privacy, we propose three different approaches, applying the privacy mechanisms presented and defined in Section 3 with Sequential Composition. For all algorithms presented below, we used the example of the city query. Our queries are performed in the new dataset created in the pre-processing step of each query, and due to that, it is important to point some observations about differential privacy mechanisms applied over small data sets. The smaller the dataset, the more sensitive to randomness the data is [Sarathy and Muralidhar 2011], so the choice of budgets can vary. Although we are executing our approach in the new dataset, our sensitivity value is still one because our dataset has individual's data which means that the absence or addition of an individual causes an impact of the subtraction or sum of one counting unit in the class that this individual's characteristic is inserted. The correct choice of the score function is also essential, and it will depend on the type of query. For example, in our approach where we search the reverse top-k characteristics that follow some criteria, the ideal score function is the negative count of all the possible characteristics. The higher the score function, the more chances the value corresponding to it will be chosen first.

5.1 Private Reverse Top-k via Laplace Mechanism

Algorithm 1 outlines a query with the Laplace mechanism.

Algorithm 1: Reverse Top-k Via Laplace

```
Input: $\epsilon$, $\Delta f$, dataset, k.
Result: k cities ordered based on noisy_count.
original_count ← dataset.count;
cities ← dataset.cities;
noisy_count ← lenght(dataset) dimension zeros vector;
for $i$ ← 1...lenght(dataset) do
    noise ← laplace_variable(loc=0, scale=$\frac{\Delta f}{(\frac{\epsilon}{k})}$);
    noisy_count[i] ← original_count + noise
end
output ← ordered_by_noisy_count(cities, noisy_count)[:k];
```

We first store the amount of infected for each class. The classes are represented by Ceará's cities. For each city, we calculate the noise, which is a random variable drawn by Laplace distribution where the location parameter is 0 and the scale parameter is $\frac{\Delta f}{(\frac{\epsilon}{k})}$, due to the sequential composition. Then, the noises are added to the count values, and the output is the 184 cities with their respective count with noise, ordered by the new count values. The complexity of this algorithm is $O(n \log n)$.

5.2 Private Reverse Top-k via Exponential Mechanism

```
Algorithm 2: Reverse Top-$k$ Via Exponential
Input: $\epsilon$, $\Delta f$, dataset, scores, k.
Result: $reverse\_top$
$budget \leftarrow \frac{\epsilon}{k}$;
$reverse\_top \leftarrow$ k dimension vector;
for $i \leftarrow 1...k$ do
    probabilities ← k dimension zeros vector;
    for score in scores do
        probability ← exp $\left(\frac{budget \times score}{2 \times \Delta f}\right)$;
        probabilities[i] ← probability;
    end
    for $j \leftarrow 1...length(probabilities)$ do
        probabilities[j] ← $\frac{probabilities[j]}{sum(probabilities)}$;
    end
    Choose an unrepeated city following the calculated probabilities and add it to the vector:
      $reverse\_top$;
end
```

Algorithm 2 builds a vector of probabilities, in which each probability is calculated as exp $\left(\frac{budget \times score}{2 \times \Delta f}\right)$, each class is assigned to a score. The probabilities are assigned to a proportion of the sum of all probabilities, and an unrepeated class is chosen based on the probability associated with it. This process is repeated k times. The complexity of this algorithm is $O(kn)$.

5.3 Private Reverse Top-k via Permute-and-Flip Mechanism

```
Algorithm 3: Reverse Top-$k$ Via Permute-and-Flip
Input: $\epsilon$, $\Delta f$, dataset, scores, k.
Result: top
$budget \leftarrow \frac{\epsilon}{k}$;
R ← possible outputs;
top ← k dimension vector;
q* ← $\max_{r \in R} q_r$;
for $i \leftarrow 1...k$ do
    for r in RandomPermutation(R) do
        for score in scores do
            probability ← exp $\left(\frac{budget \times (q* - score)}{2 \times \Delta f}\right)$;
            if bernoulli(probability) then
                top[i] ← r;
                R ← R - {r};
        end
    end
end
```

In Algorithm 3, an item is sampled uniformly at random from the set R without replacement and returned with the same probability. Then iterate through a random permutation and added a Bernoulli condition. The complexity of this algorithm is the same as the exponential, $O(kn)$.

6. RESULTS

To obtain the results, six different budget values were applied in each approach. The error represents the mistakes in the ordination of the output of the private reverse top-k algorithm divided by k. The approaches were executed twenty times; from the results of the outputs, the errors were calculated, and these values were averaged so that the results are not skewed.

6.1 Age Query

For the query that selects the ten ages of infected people that appear less often, the chosen budgets were 0.01, 0.1, 0.5, 1, 1.5, and 2.

In the approximation that uses the Laplace mechanism to achieve ε-differential privacy, the averages of the errors obtained for each budget were 0.96, 0.75, 0.395, 0.215, 0.135, and 0.04, respectively, as shown in the first image of Figure 2. For $\epsilon = 0.01$, the error rate is almost one, which means that the answer returned by our algorithm with parameter 0.01 has much noise calculated by the Laplace distribution. It guarantees privacy, but this query's utility is close to zero, so the publication of age query with this budget choice is not helpful for scientific analysis but guarantees high privacy to infected individuals whose data are in the used dataset. A consequence of the achievement of differential privacy is the trade-off between privacy and utility. On the other hand, when the budget parameter is two, the error rate is close to zero, i.e., the utility of the algorithm's answer applied over the government dataset with this budget value is very high, but it can harm the data owners.

Similar to Laplace, the error rates of the exponential mechanism decline when ε values increase. The averages of these errors are 0.965, 0.815, 0.6, 0.395, 0.275, and 0.21 when $\varepsilon = 0.01$, $\varepsilon = 0.1$, $\varepsilon = 0.5$, $\varepsilon = 1$, $\varepsilon = 1.5$, and $\varepsilon = 2$, respectively. The main difference comparing Laplace and exponential approaches is that the error rate decays a little smoother and the applied randomness seems to be bigger for all budgets using the exponential mechanism. The accuracy of the outputs of the private algorithm via exponential, for the highest ε value of our tests, is worse than the accuracy of the private algorithm via Laplace. It shows that between the two forms of application, the most recommended is the Laplacian one, where the entity responsible for the publication can choose more precisely what it wants to prioritize: high utility or high privacy.

Running the algorithm for age query, using Permute-and-Flip mechanism represented by the third graph of Figure 2, the averages of errors are 0.9799, 0.845, 0.595, 0.33, 0.21, and 0.12. The results

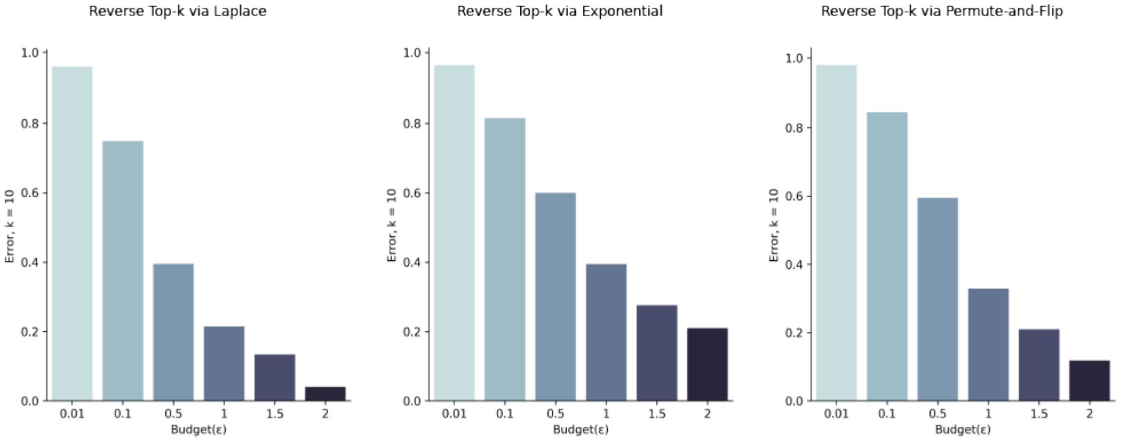

Fig. 2. Averages of the twenty errors calculated for the three different approaches that activate differential privacy over reverse top-k ages queries.

for this approximation are better than the exponential approximation but worse than the Laplacian approximation. The first proposal remains the best of the three for this age query.

In general, comparing the three graphs, it is possible to notice that, for all approximations, the error rate for the smallest budget remains similar, in contrast to the budget with the highest value.

6.2 City Query

The high budget values are because of the size of the dataset, as mentioned previously in Section 5, and the variation between the score functions that is greater than the variation of the previous query. Since there are several missing data regarding the age of the individuals whose data are in the dataset, as a consequence, the values of the frequency of ages are lower, and the variation between them is also lower. Different from what happens in cities query, where almost all cells in the column that indicate the cities' code are filled. Consequently, the sum of count values is much higher than the ages query. The chosen budget values were 0.1, 1, 5, 10, 30, and 75.

The error averages of the approach that uses the Laplace mechanism to achieve ε-differential privacy were represented in Figure 3 with an error rate of 0.939, 0.66, 0.29, 0.17, 0.11, and 0.09, respectively. Similar to the previous query, the lower the budget, the lower the accuracy of the algorithm's output, and error rates decrease as the budget value increases, as expected. Despite the distance of values between the last two budgets, the values obtained by the error rates were very close. Unsatisfactorily, even using $\epsilon = 75$, the error averages showed a higher value than the ages query.

The behavior of the second graphic of Figure 3 also declines when budget values increase, the averages errors were 0.98 for $\varepsilon = 0.1$, 0.835 for $\varepsilon = 1$, 0.4649 for $\varepsilon = 5$, 0.21 for $\varepsilon = 10$, 0.19 for $\varepsilon = 30$ and 0.1 for $\varepsilon = 75$. Compared to the Laplace approach, all errors were superior in the exponential algorithm, so the first approach seems better for this query. The entity that controls the budget value to publish the query can choose the criteria that are better to its interests, whether privacy or utility, so that still guarantees privacy, using the first approach where the graphic has a better behavior.

Permute-and-Flip approach results, which are in the third graph of Figure 3, presented the following error averages: 0.975, 0.8, 0.31, 0.275, 0.15, and 0.1. Comparing exponential and Permute-and-Flip mechanisms, the biggest difference between error rates was 0.1549, for $\varepsilon = 5$. When comparing the three algorithms, the version that uses the Laplace mechanism is also the most suitable for enabling ε-differential privacy.

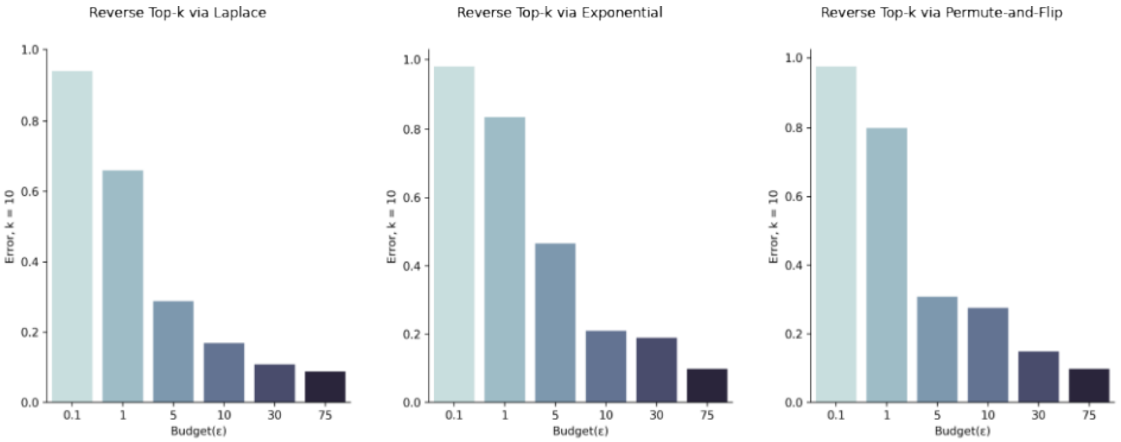

Fig. 3. Averages of the twenty errors calculated for the three different approaches that activate differential privacy over reverse top-k cities queries.

7. CONCLUSION AND OPEN QUESTIONS

This article studies COVID-19 patients' privacy problem on reverse top-k queries and proposes three different approaches that use differential privacy mechanisms. The algorithms were executed on the public data of the government of Ceará, which shows the evolution of coronavirus cases in the state. Although the reverse top-k queries were performed on a few categories, the results were satisfactory, and the choice of budgets for each query. We observed that the approach with the best performance in both queries was the one that uses the Laplace mechanism to achieve differential privacy. On the other hand, the worst approach for this query was the exponential mechanism. The curious fact is that, theoretically, the exponential mechanism performs better than the Laplace mechanism in categorical queries. It may have occurred due to the few elements or possible outputs from the query; the size of the new datasets where the queries were applied was eighty-one and a hundred eighty-four registers, respectively. Moreover, as mentioned before, the smaller the dataset, the more sensitive to randomness the data is.

The results also show that, in the age query, *budgets* around 1.0 achieve high utility, only around 20% of error. At that level, *budgets* also provide strong privacy guarantees, especially in the Laplace approach. On the other hand, *budgets* grow must faster in the city query, meaning that one has to give up much of patient privacy if he wants to keep data useful.

Some remaining open questions are:

(1) We focused on testing our approach in a database that is very large but has few elements to answer the chosen queries. Because of that, the counts of some categories had high values and the applied noise may have been very random. Applying approximations to datasets with more possible categories may yield much better results.
(2) For the second query, the chosen score function may not have been the most appropriate, because of that and the dataset size, the budget values needed to be much higher than the first query. Further investigation of different score functions should be suitable to mitigate this problem.
(3) One question that remains is "is there another privacy mechanism that has better results for this type of query?"

For future work, we intend to make an approximation using the Local Dampening mechanism [Farias et al. 2020] and apply our approaches in datasets that have more categories to solve the randomness problem.

REFERENCES

Cheng, X., Su, S., Xu, S., and Li, Z. Dp-apriori: A differentially private frequent itemset mining algorithm based on transaction splitting. *Computers & Security* vol. 50, pp. 74–90, 2015.

Dwork, C. Differential privacy in new settings. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms.* SIAM, Philadelphia, United States, pp. 174–183, 2010.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference.* Springer, New York, United States, pp. 265–284, 2006.

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

Hardt, M. and Rothblum, G. N. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science.* IEEE, Nevada, United States, pp. 61–70, 2010.

Lee, J. and Clifton, C. W. Top-k frequent itemsets via differentially private fp-trees. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* Association for Computing Machinery, New York, United States, pp. 931–940, 2014.

Li, N., Qardaji, W., Su, D., and Cao, J. Privbasis: Frequent itemset mining with differential privacy. *VLDB* 5 (11): 1340–1351, 2012.

McKenna, R. and Sheldon, D. Permute-and-flip: A new mechanism for differentially private selection. https://arxiv.org/pdf/2010.12603.pdf, 2020.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *48th Annual Symposium on Foundations of Computer Science*. Vol. 7. IEEE, Rhode Island, United States, pp. 94–103, 2007.

McSherry, F. D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *ACM SIGMOD Int. Conf. on Management of data*. Association for Computing Machinery, New York, United States, pp. 19–30, 2009.

Narayanan, A. and Shmatikov, V. How to break anonymity of the netflix prize dataset. https://arxiv.org/pdf/cs/0610105.pdf, 2006.

Sarathy, R. and Muralidhar, K. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Trans. Data Priv.* 4 (1): 1–17, 2011.

Silva, M. d. L. M., Chaves, I. C., and Machado, J. C. Aplicação de top-k reverso com privacidade sobre os dados públicos de covid-19 no estado do ceará. In *Anais do XXXV Simpósio Brasileiro de Bancos de Dados*. SBC, SBC Open Lib, Cear´a, Brazil, pp. 193–198, 2020.

SUS. Boletim epidemiológico novo coronavírus (covid-19). https://bit.ly/32yFY7a, 2020.

Vlachou, A., Doulkeridis, C., Kotidis, Y., and Nørvåg, K. Reverse top-k queries. In *International Conference on Data Engineering*. Piscataway, NJ IEEE 2010, Long Beach, CA, United States, pp. 365–376, 2010.

Zeng, C., Naughton, J. F., and Cai, J.-Y. On differentially private frequent itemset mining. *The VLDB journal: very large data bases: a publication of the VLDB Endowment* 6 (1): 25, 2012.