6-2016

An Adaptive Algorithm for Range Queries in Differential Privacy

Asma Alnemari

An Adaptive Algorithm for Range Queries in Differential Privacy

by

Asma Alnemari

A Thesis Submitted
in
Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Computer Science

Supervised by

Dr. Rajendra K. Raj

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

June 2016

The thesis “An Adaptive Algorithm for Range Queries in Differential Privacy” by Asma Alnemari has been examined and approved by the following Examination Committee:

Dr. Rajendra K. Raj
Professor
Thesis Committee Chair

Dr. Carol Romanowski
Associate Professor
Thesis Committee Co-Chair

Dr. Zack Butler
Associate Professor
Thesis Committee Observer

Dedication

To my parents, who have always been behind my success

Acknowledgments

First and foremost, I am grateful to God for the countless blessings that He has given me in this life.

I would like to express my sincere gratitude to my advisor, Professor Rajendra K. Raj for continually supporting my research and offering patience, motivation, and an immense depth of knowledge.

In addition to my advisor, I would like to thank Professor Carol Romanowski for her encouragement and insightful comments.

I also would like to thank Professor Zack Butler for acting as the observer for my defense and for providing valuable suggestions to help me improve my work.

My sincere thanks also goes to Professor Hans-Peter Bischof and Rebecca OConnor for the time that they both spent advising and assisting me in solving the issues that I faced from the first through the very last day of my masters program.

A special thanks to my family. Words cannot express how grateful I am to my husband, who always supports and motivates me. I am also so grateful to my sons, Abdullah and Abdulaziz, for their smiles, which make my life much easier.

I would like express my appreciation for my great siblings, who surround me with love and support all the time.

Last but not least, I want to thank my Uncle Mahmood and my Aunt Eman for all the moral support they provided.

Abstract

An Adaptive Algorithm for Range Queries in Differential Privacy

Asma Alnemari

Supervising Professor: Dr. Rajendra K. Raj

Preserving privacy while publishing data for analysis by researchers is an issue which has considerable attention recently. ε-differential privacy is a solution that guarantees the privacy while publishing sets of data or some information about them. This work aimed to develop a mechanism that answers a given workload of range queries efficiently under differential privacy. Therefore, an algorithm was implemented during this project to satisfy differential privacy by controlling the noise added to the answers according to the input data and the given workload. The algorithm first produces two partitions of the data domain. The first partition will be produced according to the relationships between the input data while the second partition will be produced according to the ranges of the given queries. When the domain is partitioned into buckets, the counts of each bucket are calculated privately and split among the vector's positions to answer the given query set. The performances of the proposed mechanisms were evaluated using different workloads over different attributes, and the algorithm produces satisfactory results in most cases.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Maintaining individual privacy while making data available for statistical purposes are major concerns for computer scientists. When some individuals have sensitive information in particular datasets, there are many ways by which their privacy may be compromised. Latanya Sweeney gave an example of how such datasets could threaten individuals' privacy when some datasets are joined to each other [8]. This observation encouraged her to propose k-anonymity model as an attempt to prevent the re-identification of an individuals data contained in a dataset [8]. The main goal of k-anonymity is releasing a version of a particular dataset with scientific guarantees that any individual whose data is in the original version cannot be re-identified while preserving the usefulness of the released data. Therefore, according to Sweeney, "A release of data is said to have the k-anonymity property if the information for each person contained in the release cannot be distinguished from at least $k - 1$ individuals whose information also appear in the release" [8]. However, Machanavajjhala et al. [7] pointed out that k-anonymity is not enough to preserve individuals' privacy in some cases, such as when an adversary has additional background knowledge about a victim. Thus, there is a need to find a model to prevent adversaries of inferring new information about individuals and maintain the usefulness of the data for more accurate statistics.

The differential privacy model has received considerable attention recently [2]. This model aims to preserve individuals' privacy while taking into consideration the background

knowledge that adversaries may have about some individuals. There have been many different attempts to find differential private algorithms while considering the accuracy of their outputs [1, 3, 5, 9]. However, no mechanism has yet been found to work without any restrictions to preserve individual's privacy because adversaries have a seemingly unlimited number of ways to infer sensitive information about their victims. In situations where data need to be analyzed statistically by researchers, such analyses may need to be restricted to ensure better privacy. One possible way to apply differential privacy is to anonymize the given datasets and then release them or apply some computations and release the results of these computations for some statistical purposes. However, this *non-interactive* setting is not likely to be used because when the dataset is released, control over privacy is lost. The other way to enforce differential privacy is taking queries from the database users and applying some computations over the real answers to anonymize them before giving them to the users. This study focuses on this *interactive* setting because it is more realistic and restricts the usage of the data, thus providing more privacy than a *non-interactive* setting.

To apply differential privacy over the *interactive* setting, users should not interact with the private data directly. So, there should be a mechanism between the database and users who issue queries. This mechanism would receive user queries and process them to ensure that the answers to these queries do not lead to the release of sensitive information from the database. Adversaries may expand their knowledge by asking multiple questions and using the answers to infer what they aim to know about their victims. To prevent that, there are two possible solutions. The first solution involves increasing the amount of noise added to the answers gradually as the user continues to issue queries over the database. The second solution is limiting the number of queries allowed to the database so users cannot continue asking questions beyond a determined point. While these solutions decrease the chances of inferring new information about the database's individuals, the released information may not be useful for the statistical purposes due to anonymization or the insufficient amounts of information required to make good decisions. Such solutions present no trade-off between privacy and utility.

One possible solution both to control privacy and provide more accurate answers involves providing the queries as a workload. In this situation, a mechanism should study the relationships between the given queries and the data to adjust the noise added to the real answers according to the sensitivity of the data and the given queries. There have been many attempts to find a mechanism that produces accurate results to workload queries depending on the given queries. Li et al. [5] proposed the *matrix mechanism*, which uses other query set called a strategy query and answers its queries by using the Laplace mechanism. The answers to the strategy query will be used to drive noisy answers to the given workload. This mechanism can produce satisfactory results, but the optimal query strategy of the given workload must be found to produce the best results. However, constructing the optimal strategy query requires very complex computations. Therefore, Li et al. [5] extended their work by finding a greedy algorithm that constructs a query strategy that is very close to the optimal query strategy, and enhances the performance of the matrix mechanism [6]. They proposed DAWA, a mechanism that combines many approaches so as to be data dependent and workload aware at the same time. DAWA takes advantage of every approach to produce more accurate answers as Li et al. claim [4].

DAWA is a ε-differential private algorithm for processing workload queries and produces better results than the current mechanisms in this domain according to results provided in [4]. The inputs of this algorithm consist of a workload of range queries and a database represented as a vector of counts. The algorithm goes through three stages to output a private estimate of the input database. In the first stage, the domain of the attribute's values is partitioned into buckets so that the counts of every value are approximately the same in every bucket. In the second stage, the count of every bucket is estimated privately with the *matrix mechanism* and the estimated count of every bucket will be spread uniformly among the bucket positions to produce a private vector of the input data. The last stage is using the produced private vector to answer the given workload queries and provide these answers for the database user.

For DAWA, Li et al. stated that using the Laplace mechanism to produce noisy counts

for each bucket of the domain partition could increase the error rates in some cases [4]. However, they did not specify why such a situation might occur or provide an explanation. Moreover, they did not mention the limitations of their algorithm or the situations in which this algorithm could produce worse results. Thus, this study investigates these issues to produce an algorithm that takes the shortages of the DAWA mechanism into consideration and corrects such issues to produce private answers for range queries. Because partitioning the domain of the attributes' values plays a considerable role in enhancing the results of DAWA, as stated by Li et al. [4], I have built a greedy algorithm that efficiently partitions the domain of the attributes' values. The differential privacy is taken into consideration to produce a partition that is quite close to the optimal partition. However, this algorithm does not produce the optimal partition due to the amount of noise added to the results that are used to partition the attributes' domain. Therefore, every time the algorithm is run it will produce a different partition. This is not a significant issue, however, because exact answers will not be produced in all cases. This is because the privacy of the database is the most significant issue that the algorithm seeks to preserve.

To capture the most sensitive areas on the vector of the values' counts, I created an algorithm to partition the domain of the value according to the given workload. This technique could reduce the chances of inferring new information from the answers of the given workload. The algorithm extracts the regions involved in the set of queries and intersects these regions to produce interval ranges. Thus, these ranges will be used to partition the vectors into buckets so the count of every bucket will be anonymized and then divided uniformly among the bucket position during the next stage. This algorithm aims to preserve the database's privacy so that the rate of error in some situations will be higher than the rate of error received when using the other technique to partition the counts domain.

To evaluate this studys efficiency, three approaches were used to answer a given workload query, and the squared error was computed for every set of answers to compare the performance of every approach. In the first approach, Laplace noise will be added to the count of every value to produce a private vector that will be used to answer the given set

of queries in the next step. The second approach is very similar to the approach used in DAWA to partition the data vector. So, it will partition the domain of the attribute's value according to the counts of these values. In the third approach, the domain of the attribute's values will be partitioned according to the given workload. To study the performance of each approach, the three mechanisms were run over different datasets to answer different workloads with different values of the parameter ϵ. In every situation, both of the proposed algorithms were competing to produce better results as discussed in the fourth chapter.

1.2 Background

1.2.1 Differential privacy

Originally proposed by Dwork [2], the model of differential privacy ensures that the presence of an individual in a database cannot be discovered regardless of the background knowledge that adversaries may have about this database or its individuals. To do so, the outcome of any analysis will be almost the same after adding or removing any record from the database [2]. Given an arbitrary query f with domain M and range $P(f : M- > P)$ and two databases D and D' that differ in one record, if K_f is a randomized function used to produce the response to query f, then K_f gives ε-differential privacy if for any $s \subseteq Range(K_f)$

$$Pr[K_f(D) \in s] = e^{\varepsilon} Pr[K_f(D') \in s]$$

Differential privacy assumes that a mechanism sits between users and the databases that they query. The mechanism receives the queries and modifies their answers to ensure the databases privacy before returning them to the users. Thus, the answers cannot be used to learn new information about individuals involved in the database. The most common method used to achieve differential privacy is adding noise to the real answers of the given queries. If f is a query on a database D and r is the correct answer to f, then the response to that query would be $r + y$, where y is an amount of noise that anonymizes results to

satisfy differential privacy. While adding noise to the real answers may protect individual privacy, it may also affect the quality of the results and cause them to be useless. Therefore, the amount of added noise must be adjusted carefully to ensure simultaneously individual privacy and data utility. Dwork [1] suggested using Laplace distribution noise, in which the noise is drawn from a Laplace distribution with mean 0 and scale $\Delta f/\varepsilon$, where Δf is the maximum value for $||f(D) - f(D')||$ and ε is the parameter used to control privacy (a smaller ε yields more privacy and consequently less accuracy) [3].

Definition 1 (The Laplace Distribution). *The Laplace Distribution (centered at* 0*) with scale* b *is the distribution with probability density function:*

$$Lap(x|b) = \frac{1}{2b}\exp(\frac{-|x|}{b})$$

Definition 2 (The Laplace Mechanism). *Given any function* f*, the Laplace mechanism is defined as:*

$$L(x, f) = f(x) + Y$$

where Y *is a random variable drawn from* $Lap(\Delta f/\varepsilon)$.

There are two possible settings to satisfy differential privacy: *interactive* and *non-interactive*. In the *interactive* setting, a trusted server holds the database and allows users to issue queries over this database. Then the server modifies the true answers to these queries to ensure that the answers do not reveal sensitive information about the database and its users. In the *non-interactive* setting, the data are anonymized and released, or some statistics are computed and published over this data. Thus, users can use the data for any purpose without the need to worry about the privacy of the individuals who have information contained within these released data.

Counting Queries

Counting queries are queries that ask about the number of instances that satisfy specific conditions. Such queries are important for statistical operations, whether in this pure form or in another form such as fractional or linear queries [3]]. Dwork et al. [3] emphasized that these queries are extremely powerful in primitive form because they contribute to statistical

learning models in terms of information. Therefore, the answers to these queries should be modified to decrease the chances of distinguishing individuals and recognizing their data. However, because the amount of added noise to the real answers could affect the usefulness of the results, this process should be carried out with caution.

Workload queries

The *interactive* setting provides better privacy than the *non-interactive* setting because the data remain under owner control and the answers to the given queries could be computed depending on the nature of the data and the given queries. However, many sequence queries issued over the database could cause information to be released if the answers are related. More privacy could be preserved, in some situations, by either decreasing the parameter ε or increasing the query sensitivity. For example, according to the number of issued queries, the amount of noise added to answers should be increased gradually to increase the uncertainty in regard to individuals within the database. This process may affect the usefulness of the results because the noise added to answers gradually increases according to the number of given queries. Therefore, giving the queries to the database as a workload would be a better solution for this issue. If this is done, it will be easier to find a mechanism that studies the relationships between these queries and measure their sensitivities to compute the amount of the noise that should be added to the answers. This will produce more accurate answers while maintaining the privacy of the database and its individuals.

1.3 Related Work

Dwork et al. [1] proposed a simple method to enforce differential privacy. This method first involves calculating the frequency distribution of the records in the input data. Then a noisy version of the distribution will be produced to be published. This method depends on the frequency matrix and provides accurate results for the queries about individual entries in this matrix because the amount of noise injected into every entry is very small. However, this method produces poor results for aggregate queries that involve a large number of

entries [9]. Therefore, researchers have proposed many methods to solve these issues and produce more accurate results.

Xiao et al. [9] proposed the *Privelet* technique using Wavelet transform to control the noise added to the frequency matrix and improve the accuracy of the results when answering the given range queries. Similarly, Li et al. [5] proposed *The Matrix Mechanism* to answer a counting queries workload. This mechanism requests noisy answers to a different set of queries called a *query strategy* to drive the noisy answers to the given workload. The noisy answers to the *query strategy* will be produced using a standard Laplace mechanism. The produced answers to the given workload will vary according to the *query strategy* used to produce the noisy answers. Thus, to ensure better results, this mechanism requires very complicated computations to find the optimal query strategy. This issue encouraged Li et al. [6] to extend their research and develop a greedy algorithm to construct a query strategy that is very close to the optimal *query strategy* to produce the best possible results.

Since there are many kinds of datasets, the mechanisms that have been developed to satisfy differential privacy are classified as either data dependent or data independent. Therefore, according to the nature of the data, the performance of the mechanism that has been chosen to satisfy differential privacy will be changed for better or worse. This observation encouraged Li et al. to develop a mechanism that considers the nature of the dataset and the given workload at the same time. Their algorithm, DAWA, produces an estimate of the input data vector where the noise added to the real counts is adapted to the input data and the workload [4]. The produced vector will then be used to answer the given workload queries.

1.4 Problem and hypothesis

Problem statement

The aim of this work was developing a mechanism to answer a set of range queries efficiently under differential privacy. Li et al [4] could come with a good mechanism "DAWA" to answer workloads of one-dimensional range queries. Their mechanism depends on partitioning the domain of the attribute values into buckets, so the values' counts in each bucket are almost the same. The goal of this process is decreasing the amounts of the noise that should be added to each count to preserve the database privacy which leads to more accurate answers. According to the results that Li et al. presented in their paper [4], DAWA produces outstanding results comparing to some current models. Li et al. emphasized that the partitioning stage plays a considerable role in the model performance. However, they did not mention the cases in which this model performs better or worse. While the condition of the partitioning process is the closeness of the values' counts, the authors did not specify the maximum difference between the values' counts to be in one bucket. Furthermore, they did not describe how DAWA will perform if the vector of the counts has many regions of density. In this case, according to their approach, each position of the vector will be a bucket. However, the next stage requires a set of computations over each bucket which means consuming a lot of resources to produce an amount of noise to be added to each values' count that is represented as a bucket. In the other hand, Li et al. claimed in their paper [4] that using Laplace Mechanism to produce noisy counts of the buckets increase the error rates of results in some cases but they did not specify these cases, and they did not provide good explanations to clarify the reasons for producing these results. In fact, there is a need to investigate the factors that affect the Laplace mechanism even worse or better because applying this mechanism is easier and does not consume too many resources. These observations encouraged me to come with a model to study these situations to find a suitable mechanism to solve these issues. So, I started by modifying the technique that Li et al. used to calculate the cost of the partitions to choose the best partition for the counts vector. After that, I built a greedy algorithm to find the best possible partition privately

instead of generating all the possible partitions and choose the best one of them according to their costs as Li et al. did in their work DAWA.

Most of the recent models that have been developed to answer sets of range queries under differential privacy aim to decrease the amounts of the noise added to the results to reduce the error rates. However, in some cases, we need to increase the amounts of added noise to increase the privacy level of the produced results. Therefore, the sensitive situations should be recognized to be considered while generating the noise. One possible way to recognize these situations and measure their sensitivity levels is studying the relationships between the set of queries that have been given to the database because the answers to these queries could be used together to infer new information about the database and its individuals. This fact gave me the insight to come with a mechanism that partitions the domain of the values' counts according to the given workload. This mechanism captures the sensitive areas of the vector because it intersects the ranges that the queries involve and partitions the vector according to these regions. Thus, the regions that the most queries focus on will be in separate buckets even if each bucket contains only one position of the vector. Since there is an amount of noise to be added to each bucket count, as the bucket contains fewer positions their counts will have larger amounts of noise which of course increase the privacy levels of the produced results.

My focus during this work was on the significance of the partitioning process and how it affects the efficiency of the produced results. Therefore, only the Laplace Mechanism is used to generate the amounts of noise that are required whether to be added to the counts of each bucket or to modify some results. This mechanism is differently private, and its parameters can be adjusted to change the privacy levels. In some situations, where every position is sensitive, the count of every position of the vector should have amounts of noise to ensure better privacy instead of grouping the counts and adding an amounts of noise to them. Therefore, this technique has been used in this work to be compared to the other techniques that depend on the partitioning process.

Hypothesis

The hypothesis underlying this thesis is that an adaptive algorithm can be developed to efficiently produce an optimal partition of attribute values and answer range queries under differential privacy with increased privacy levels. The partitioning process of the attribute counts affects the performance of the model that uses to answer a set of range queries under differential privacy. Moreover, the private greedy algorithm and the modified cost function can produce the best possible partition of the vector of the values' counts. Also, the algorithm that partitions the vector of the values' counts according to the given workload can increase the privacy levels of the produced answers.

Privacy

The main goal of this work is maintaining the privacy of databases and their individuals. Therefore, this concept is considered in every stage of the proposed model. This work involves two techniques to partition the domain of the counts. The first technique depends on the data itself, so the partitioning process preserves the privacy by adding an amount of noise to the count of each bucket using The Laplace Mechanism, which is a differently private mechanism. The second technique depends on the given workload and does not require any information about the database instances. Thus, partitioning the vector of the values' counts using the second technique does not impact the privacy of the database. However, the next stage of this work is estimating the count of each bucket, and that could leak some sensitive information of course. Therefore, there should be an amount of noise to be added to the count of each bucket. So, the Laplace mechanism is used to generate these mounts on noise to preserve the privacy while producing the results of the second stage of the proposed model.

Performance

The performance of the model is considered in each stage of this work. Therefore, a greedy algorithm was developed to partition the vector of the attribute counts according to the

closeness of these counts. Moreover, the next stage requires generating an amount of noise for each bucket which is a very simple process that does not consume a lot of resources. So, whether the bucket has one bucket or more, the performance of the algorithm will not be affected. Moreover, the mechanism of partitioning the count domain based on the given workload consumes fewer resources because it depends on the given workload which is much smaller that the data vector in most cases.

1.5 Roadmap

This thesis is organized in the following manner:

- Chapter 2 presents and explains the design of the proposed model in details.
- Chapter 3 explains how the proposed model and its components were implemented.
- Chapter 4 elucidates the details of the experiment and discusses the analysis of the results.
- Chapter 5 discusses the conclusions from the research and outlines the anticipated future work.

Chapter 2

Design

The main goal of the proposed model is developing an efficient mechanism to answer workloads of range queries under differential privacy. Since Li et al. [4] found that partitioning the vector of the values' counts could make a very considerable difference in the quality of the results, this model answers the given sets of range quires over single attribute using three approaches two of them partition the domain of the attribute values. That could contribute to investigate how the partitioning process could improve the results and in which cases it performs better than the other traditional mechanisms such as the Laplace Mechanism. Figure 2.1 presents the proposed model and how it produces its results. As shown, the first approach anonymizes the real counts of the values' vector by adding noise that is generated by Laplace mechanism. Thus, a private vector of the values' counts will be ready for the next stage of the model as shown in figure 2.2. The second and the third approaches produce their results based on partitioning the vector of the values' counts into buckets and anonymize the counts of the produced buckets. Then, the count of each bucket will be split uniformly among its positions to produce a private vector of the values' counts. The second approach partitions the vector according to the uniformity of its regions. Figure 2.3 overview of the mechanism that uses this approach with an example to clarify how it produces its results. Similarly, the third approach partitions the vector according to the regions that the given workload involves. This approach is also illustrated in figure 2.4. The last stage of the model is taking the private vectors produced by the previous mechanisms and answer the given workload according to the counts of these private vectors.

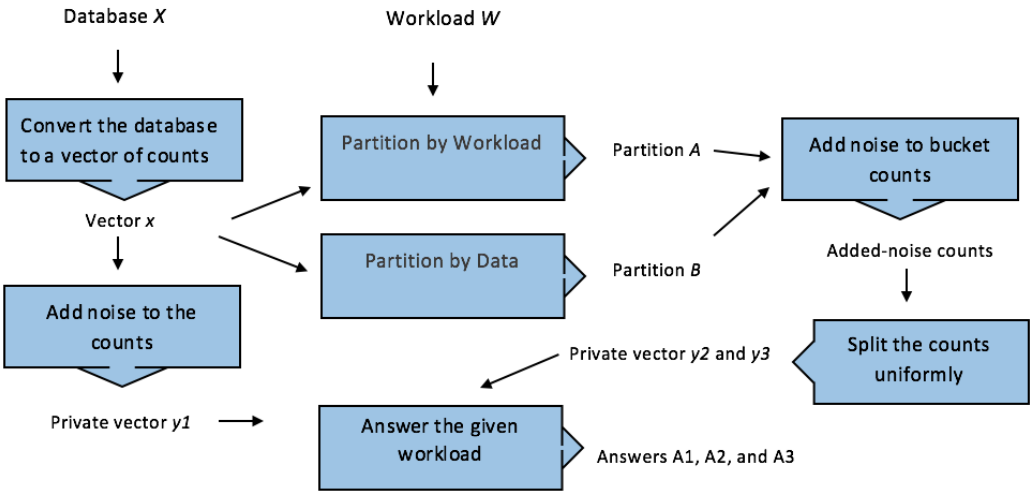
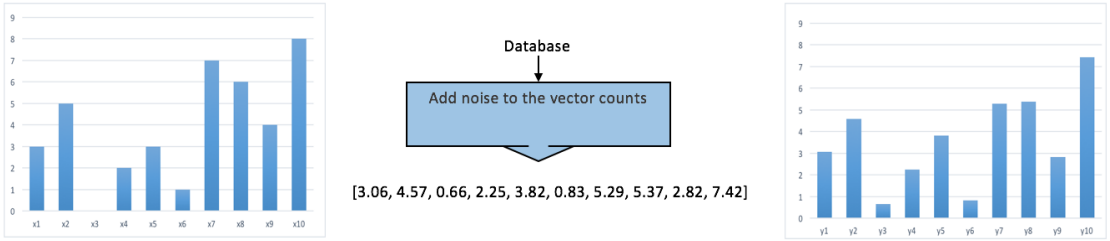


Figure 2.1: Overview for The proposed model.

2.1 Laplace Mechanism

Laplace mechanism is an effective mechanism to enforce differential privacy. In her book [1], Dwork proofed that this mechanism satisfies differential privacy and protect the individuals' privacy in most cases. Therefore, it has been used as a part of the proposed model to compare its results with the results of using the other mechanisms to recognize the situations in which using this mechanism produces the best results. Figure 2.2 illustrates how the Laplace mechanism has been used in the proposed model to produce private answers to the given workload. As shown, the database will be entered to the application as a vector of counts, and then an amount of noise will be generated and added to each position of the vector. Each amount of noise generated in this stage is drown from a Laplace distribution with mean 0 and scale $1/\varepsilon$ since every position will be affected by one in the case of adding or removing one record to its count.



(a) The original vector of the database (b) The Mechanism flow chart (c) The private vector of the database

Figure 2.2: Overview and example execution for The Laplace Mechanism.

2.2 Partitioning the counts' vector based on the Data

The second approach used to answer a given workload in the proposed model requires partitioning the domain of the values' counts according to the uniformity of its regions. That means the neighbor positions that have close counts will be grouped in one bucket. To check whether a particular position should be in a bucket with its neighbors or not, a function was built to calculate the cost of the bucket with this position and the cost of the same bucket without this position. Then, according to these two costs, the position will take a place in this bucket or not.

2.2.1 Cost function

This function was built similarly to the cost function that Li et al. used in their work DAWA [4]. According to their observations in that work, after the partition $B = \{b_1, ., b_n\}$ has been generated, an amount of noise z will be added to the counts of each bucket so the count of each bucket will be $c_i = b_i(x) + z_i$. After this process is done for each bucket, the counts of each bucket will be split uniformly among the bucket positions. Thus, the estimation for x_r where $r \in bi$ is:

$$y_i = \frac{b_i(x)}{|b_i|} + \frac{z_i}{|b_i|}$$

.

The bucket size and the degree of the uniformity within the bucket affect the accuracy

of estimation the bucket element x_r. That means, as the element x_r becomes close to the mean of the bucket, the estimation of x_r becomes more accurate [4].

As stated in [4], according to these observations, the deviation of the bucket b_i from being uniform is measured by dev as the following:

$$dev(x, b_i) = \sum_{r \in b_i} |x_r - \frac{b_i(x)}{b_i}|$$

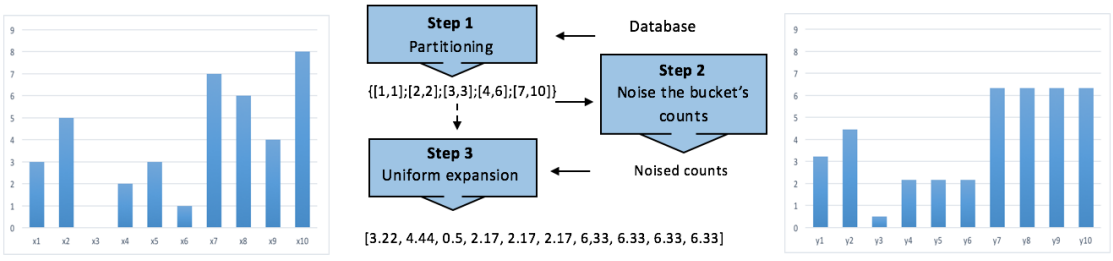
Using the only deviation functions to decide whether a particular position should be in a particular bucket or not will not contribute to produce a good partition. That because the deviation of the bucket which has only one element is less than the deviation of the bucket which has many elements unless the bucket elements have the same counts. In other words, using the deviation function alone to partition the vector will put each position of the vector in a separate bucket and will not group the neighbor positions that have close counts in one bucket unless they have the same count. Therefore, we should specify the maximum difference between the vector positions' counts to be in one bucket. Furthermore, since calculating the deviation of each bucket requires the counts of theses buckets, these counts will be anonymized by adding Laplace noise to their values and then the rest of the processes will be applied over these anonymized counts to enforce the differential privacy while calculating the deviations processes.

2.2.2 Greedy partitioning algorithm

To reduce the consumption of the resources while partitioning the domain of the values' counts, a greedy algorithm was developed as shown in Algorithm 1. The algorithm starts by the first position of the vector and constructs a partition with a bucket which consists of only this position and a partition with a bucket which consists of this position and the next position to it. After that, the deviations of these two buckets will be calculated to decide which bucket will be chosen as a part of the final partition. The process will be repeated over all the positions of the values' vector and after the loop is finished, the final partition will be ready to next processes.

Algorithm 1 The greedy algorithm for partitioning the counts' domain based on the data

1: **procedure** PARTITIONINGBYDATA(X, ϵ)
2: // let m be the maximum difference between the positions of X to be in one bucket
3: // let *firstpos* be the first position of the current bucket
4: // let *lastpos* be the last position of the current bucket
5: *firstpos* $\leftarrow 1$
6: *lastpos* $\leftarrow 1$
7: *loop*:
8: // if the loop reaches the last position of X
9: **if** *lastpos* $=$ *The size of X* **then**
10: $part_1 \leftarrow [firstpos, lastpos]$.
11: // add $part_1$ to the partition
12: $P \leftarrow P + part_1$.
13: // increase the last position
14: $lastpos \leftarrow lastpos + 1$.
15: **else**
16: $part_1 \leftarrow [firstpos, lastpos]$.
17: $part_2 \leftarrow [firstpos, lastpos + 1]$.
18: $dev_1 \leftarrow$ *Deviation*$(part_1)$.
19: $dev_2 \leftarrow$ *Deviation*$(part_2)$.
20: **if** $dev_1 + m < dev_2$ **then**
21: // add $part_1$ to the partition
22: $P \leftarrow P + part_1$.
23: // move to the next position
24: $lastpos \leftarrow lastpos + 1$.
25: $firstpos \leftarrow lastpos$.
26: **else**
27: // expand the currant bucket by adding the next position to it
28: $lastpos \leftarrow lastpos + 1$.
29: **if** *lastpos* $<=$ *The size of X* **then**
30: **goto** *loop*.
31: // return the partition
32: **return** P

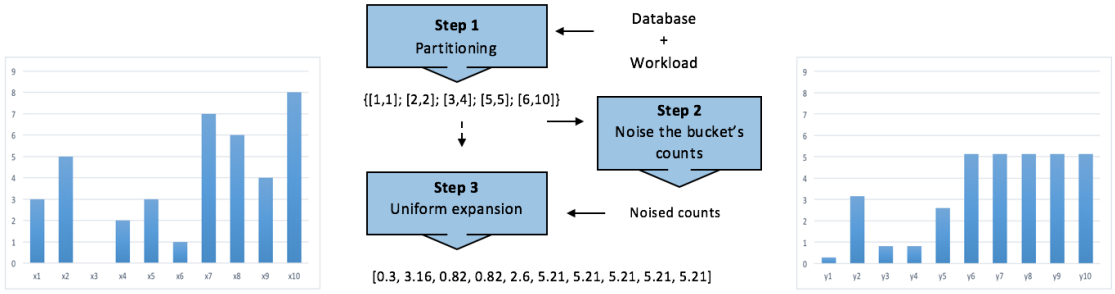


(a) The original vector of the database (b) The Mechanism flow chart (c) The private vector of the database

Figure 2.3: Overview and example execution for the Data Partitioning mechanism.

2.3 Partitioning the count's vector based on the Workload

The third approach that has been used in the proposed model to answer a given workload of range queries under differential privacy is partitioning the domain of the values' counts according to the given workload as shown in figure 2.4. As known, the given set of queries may contribute to leak sensitive information about the database if they were answered precisely or if the amounts of the added-noise to their answers are not enough to hide the private information. However, in some cases, a given set of queries involve big ranges, and they are unrelated to each other which means, adding big amounts of noise to their answers will decrease their quality with no need for that. In the other hand, we may have a set of queries with big ranges, but when these ranges get intersecting, the focus of these queries will be on very small and sensitive ranges. This observation was behind developing the mechanism of partitioning the counts' vector based on the given workload. This mechanism aims to decrease the chances of inferring new information about the database using the relationships between the set of queries and their answers. Therefore, this mechanism takes a database represented as a vector of counts with a workload of range queries (Figure 3.1 shows an example of a workload of range queries with a description for each one). To partition the given vector, the ranges of the workload queries will be intersected to produce interval ranges that will be transformed to buckets of the final partition. Moreover, this mechanism also focuses on the ranges that the given queries do not ask about and puts



(a) The original vector of the database

(b) The Mechanism flow chart

(c) The private vector of the database

Figure 2.4: Overview and example execution for the Workload Partitioning mechanism.

them in separate buckets to treat them like the other buckets. That because in some cases, adversaries may do not ask about the regions that they want to know their counts directly. Instead, they may ask about the neighboring regions and use their counts to infer what they want to know. Therefore, as these ranges become smaller, the amounts of the added noise to their count will be larger since every region will be in a separate bucket. This process decreases the chances of inferring some information from the answers to the given set of queries. Algorithm 2 describes how this mechanism partitions a given vector of counts according to a given workload.

Algorithm 2 The algorithm for partitioning the domain based on the workload

procedure PARTITIONINGBYW(X, W)
  for each query $q \in W$ **do**
    // extract the range of the query q
    $r_1, r_2 \leftarrow$ *ExtractTheRange*(q).
    // add the ranges to the $pinranges$ matrix
    $pinranges \leftarrow pinranges + (r_1, r_2)$.
  // sort the $pinranges$ and delete the repeated elements
  $pinranges \leftarrow$ *SortAndCheck*$(pinranges)$.
  // construct interval ranges from the $pinranges$ matrix
  $ranges \leftarrow$ *ConstructRanges*$(pinranges)$.
  // add the interval ranges to the partition
  $P \leftarrow ranges$.
  // return the partition
  return P

2.4 Error function

To measure the efficiency of each approach of the proposed model, a function has been developed to calculate the error of the produced answers by each mechanism. This function was drown from the function that Li et al. built to calculate the workload error under the Matrix mechanism [6]. For a given workload, the error of this workload is defined as the root mean square error of its answers [6].

Definition 1. (Workload error). *For each query q in a given workload W, let r_q be the true answer to the query q, p_q be the private answer to the query q produced by a mechanism M, and n the number of the queries in the given workload, the error of the given workload W by the mechanism M is:*

$$error_M(W) = \sqrt{\frac{1}{n}(\sum_{i \in W} \sqrt{(r_i - p_i)^2})}$$

This function is a very significant part of the proposed model because it returns the total error of the answers produced by each mechanism. Therefore, we can compare the performance of each mechanism by comparing their answers' error. Furthermore, we can recognize the situations in which a particular mechanism can produce the better results to investigate the factors that affect this mechanism and impact its performance.

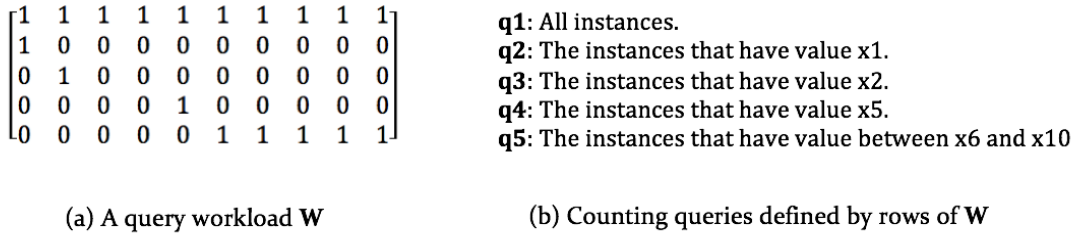
Chapter 3

Implementation

To verify the thesis's hypothesis, the proposed model, which answers a given workload over a particular database, was implemented using MATLAB. This model goes through four stages to complete its tasks and produces the final results. The first stage is preparing its inputs to be suitable for the next stages. Then, a private vector will be returned by each mechanism to answer the given workload. The final task of this model is computing the workload error under each mechanism to compare their performances. In this chapter, the implementation of the proposed model is elucidated including all of its stages.

3.1 Preparing the Model's Inputs

Two inputs are required for the proposed model. The first one is the database that the model's users would query on. So, to prepare this input for the next processes, the attribute that the model's user queries on will be converted to a vector of counts. The second input is a workload of range queries. To prepare this workload for the next processes, it will be converted to a matrix of two dimensions according to the number of the possible values of the attribute that the model's user query on and the number of the queries that the given workload has. Figure 3.1 shows an example of a workload that has been converted to a matrix with an explanation for each query of this workload.



(a) A query workload **W**

(b) Counting queries defined by rows of **W**

Figure 3.1: a sample workload matrix W consisting of 5 queries.

3.2 Constructing Private Vectors

This stage is the most significant stage of the proposed model. It requires sequences of processes to produce the private vectors that the model needs to answer the given workload. Since many of the processes in this stage need to reach sensitive data, differential privacy is considered and applied whenever the model needs to obtain any information from the database. Furthermore, Since the model answers the given workload under three different approaches, This stage returns three different private vectors. One of these vectors will be produced using the Laplace mechanism while the other vectors will be produced using the partitioning mechanisms.

3.2.1 The Laplace Mechanism

This mechanism takes the given vector produced by the previous stage and adds an amount of noise to each count of its positions to anonymize them. Therefore, a function was built to anonymize these counts. Furthermore, other function was built to generate and return an amount of noise using Laplace distribution with mean 0 and scale $1/\varepsilon$. Therefore, the noise-generation function will be called by the noise-addition function for every position of the given vector to construct one of the private vectors that the model requires.

3.2.2 The Partitioning Mechanisms

The second and the third approaches that have been used in the proposed model construct private vectors to answer the given workload based on partitioning the given vector to control the amounts of the added-noise to the counts of this vector to increase the accuracy of the produced results.

The Partitioning Mechanism Based on Data

To partition the given vector according to the data, algorithm 1 was implemented with a function that computes the deviation of a given bucket. The deviation function should enforce differential privacy during its work because it requires the total count of the given bucket which violates the database privacy. Therefore, this function anonymizes the count of the given bucket by adding Laplace noise to it at before giving this count to the next processes.

The Partitioning Mechanism Based on the Workload

Algorithm 2 was implemented to partition the given vector according to the regions that the given workload involves. As described in the previous chapter, a function is required to take the input workload and extract the regions that the queries involve. Then, the extracted regions should be intersected to produced interval regions that will be used to partition the counts' vector.

Anonymize the Bucket Counts

After partitioning the given vector to buckets, the counts of these buckets should be produced to be given to the next processes. Since these counts could contribute to leak some sensitive information about the database, they should be anonymized to ensure the database privacy. Therefore, the partitions will be sent to the noise-addition function to add an amount of Laplace noise to each bucket's count.

Splitting the Buckets' Counts Uniformly

After estimating the counts of the produced buckets, the count of each bucket will be split uniformly among its positions to produced a private vector. Therefore, a function was built to do this task by taking a particular partition as an input and applying these processes over its bucket to return a private vector that will be given to the next stage of the proposed model.

3.3 Answering the Given Workload

The proposed model requires implementing a function that takes a workload matrix with a vector of counts and returns the answers to the given workload. Thus, as long we have a workload over a vector of counts, we can use this function to answer this given workload whether the given vector has true or private counts. So, this function was implemented to be used to answer the given workload over each vector produced by each approach.

3.4 Computing the Workload Error

Computing the error of each workload under each mechanism requires finding the true and the private answers to this workload. Therefore, a function was implemented to calculate the workload error according to the error equation explained in the previous chapter.

Chapter 4

Analysis

4.1 Approach

The proposed model aims to validate the thesis hypothesis and verify that the partitioning mechanisms improve the model's performance and increase the privacy level of the database. The models analysis aimed to demonstrate the ability of the developed greedy algorithm to produce the optimal partition based on the distribution of the data, and the ability of the partitioning mechanism based on the given workload to increase the levels of the database privacy while producing satisfactory results.

The experiment was designed to measure and compare two significant factors of the three different approaches developed as parts of the proposed model. Therefore, during the experiment, there were three different set of answers for each given workload. The firs set was produced by the Laplace mechanism, and the other sets were produced by the partitioning mechanisms. Moreover, the workload error was computed over each set of answers produced by each mechanism. This process contributes to measure the privacy and the accuracy factors because the aim of the proposed model is improving these factors.

To measure the efficiency of the partitioning mechanisms, Two different vectors were generated. The first vector has many regions of density as shown in figure 4.1. The second vector has large uniform regions as shown in figure 4.2.

Since the individuals' privacy is the main concern of this work, three different workloads were constructed to test the privacy levels provided by the proposed model to protect its data. Figure 4.3 shows examples of the three kinds of workload used to evaluate the

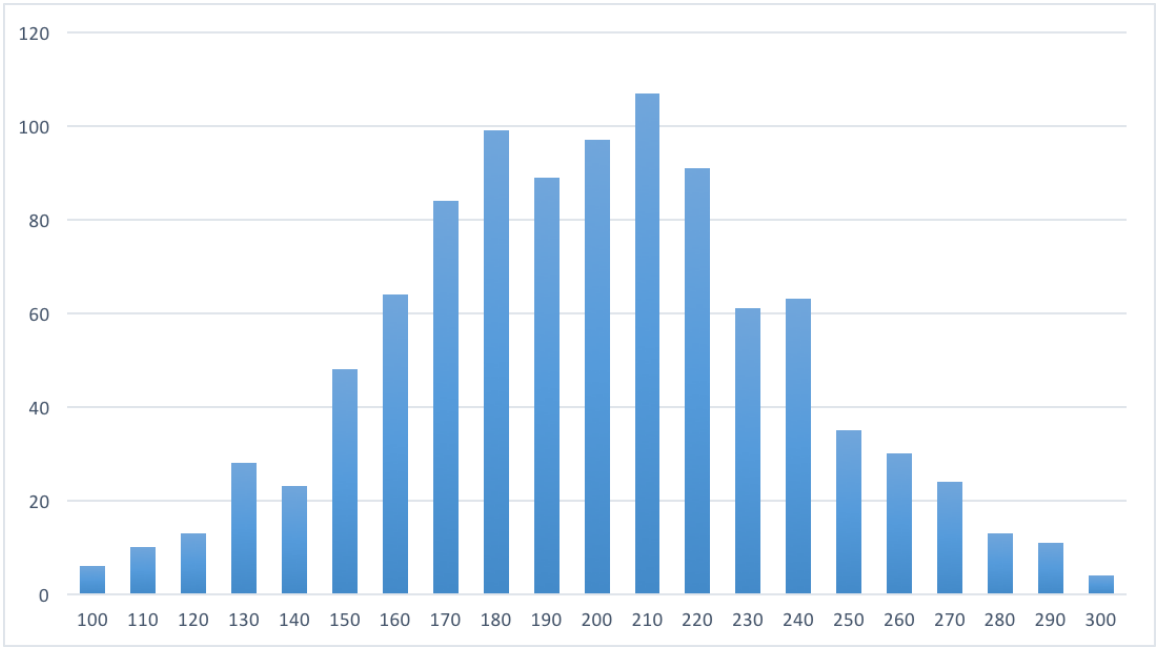


Figure 4.1: The vector that has many regions of density.

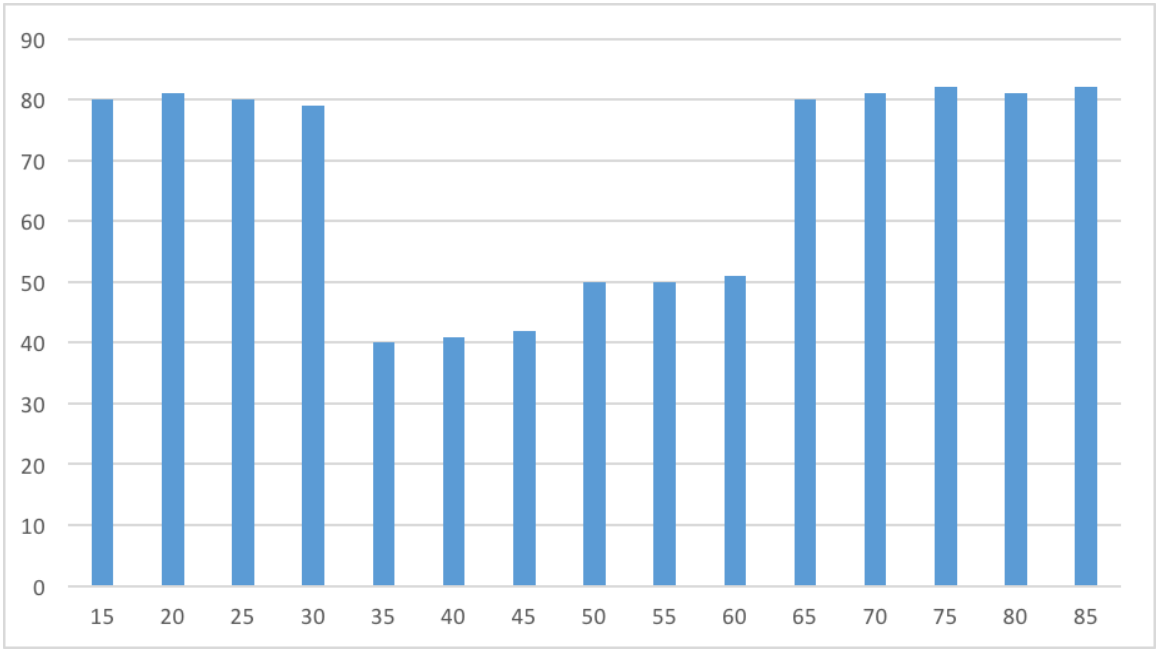


Figure 4.2: The vector that has large uniform regions.

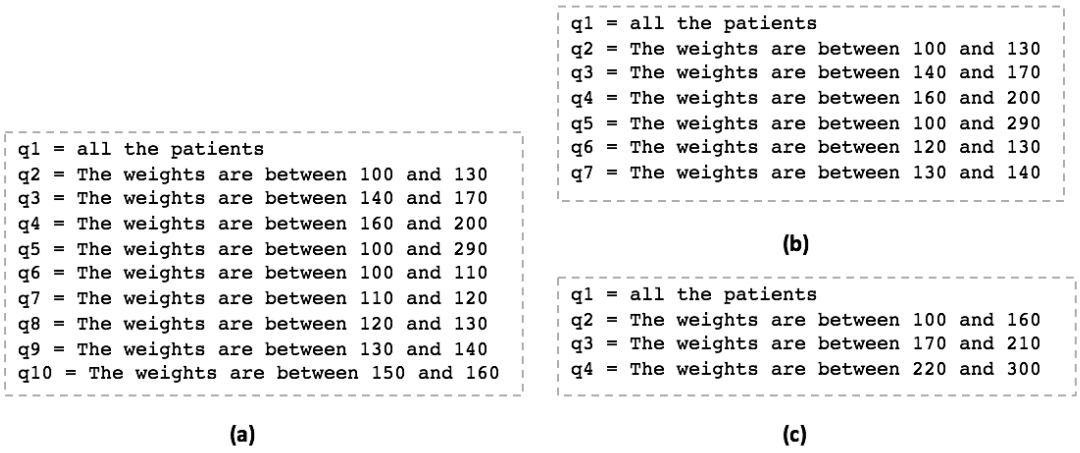


Figure 4.3: Three workloads that are different in their sensitivity levels (a) is very sensitive, (b) is sensitive, and (c) is normal.

proposed model. The first set of queries (a) is very sensitive because most of them ask about very small ranges which could be used to learn some information about the database individuals. The second set (b) is sensitive, and it has some queries that ask about small ranges and some ask about big ranges. The third set (c) is normal because its queries ask about big ranges which means their answers will be general and not sensitive.

Generating random noise from Laplace distributions has been used in different parts of this work. Therefore, since the amounts of the generated noise depend on the ϵ values, which indicates to the privacy budget, the experiment was repeated four times over each dataset with different values of the parameter ϵ. The values that have been chosen for the parameter ϵ are 0.1, 0.5, 0.01, and 0.05. Choosing these different values contributes to test the performance of each mechanism and how that related to the privacy budget.

4.2 Analysis

4.2.1 The partitioning Mechanism based on the Data

The Experiment over a Vector that has large uniform regions

This experiment was done over a vector that has large uniform regions as shown in figure 4.2. Since the vector has some regions that have close counts, the partitioning mechanism over this vector produces partitions with fewer buckets than the vectors that have many regions of density. As the number of the bucket become fewer, the number of the positions within each bucket will be more, so as the amount of the noise that has been generated for a particular bucket will be divided between the buckets' positions, each position would have a small amount of noise to be added to its count. In other words, As the buckets have more positions, the noise added to each position will be smaller and therefore this mechanism produces more accurate answers. For example: as illustrated in figure 4.4, to answer the sensitive workload (b) that have been shown in figure 4.3 over the vector presented in figure 4.2 by the partitioning mechanism based on the data, the vector of the counts was partitioned to four buckets. Each bucket has at least three positions. So, the amount of the added noise to each position will be divided at least by three which means less noise than if the bucket has only one position or if the Laplace mechanism is used over these kinds of vectors.

In these cases, the performance of the partitioning mechanism based on the data is better than the Laplace mechanism because the error rates of its results are fewer than the error rates of the results produced by the Laplace mechanism that generates an amount of noise to be added to each position of the vector. Table 4.1 and figure 4.5 present and show that in most cases the Partitioning mechanism based on the distribution of the data performs better than the Laplace Mechanism over the vector that has large uniform regions.

The ϵ parameter plays a role in the quality of the results produced by each mechanism. That because each mechanism requires generating amounts of noise from Laplace distributions to satisfy differential privacy. Therefore, as the values of this parameter become

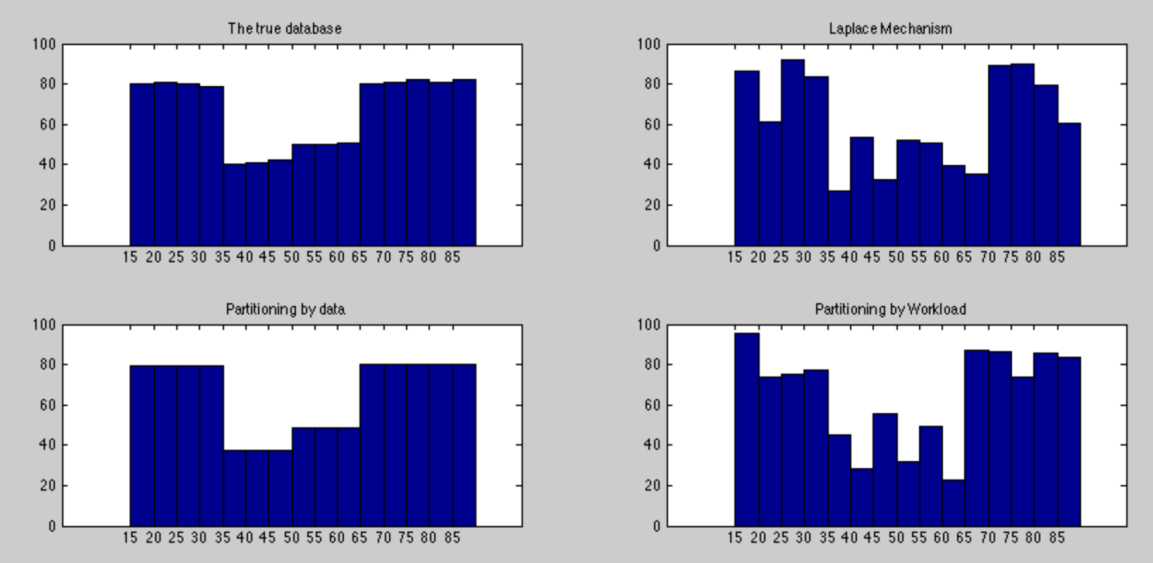


Figure 4.4: The true and private vectors produced by each mechanism of the proposed model to answer the workload (b) shown in figure 4.3 over the vector presented in figure 4.2.

smaller, the amounts of the generated noise become larger. Therefore, the partitioning mechanism would perform better than the Laplace mechanism as the values of this parameter become smaller while both mechanisms produce similar results as the values of the ϵ parameter become larger. This observation is also illustrated in figure 4.5.

ϵ	Workload	Laplace Mechanism	Partition by Data	Partition by Workload
0.1	V	4.6625	2.4678	4.1795
	S	7.6828	4.4239	4.2254
	N	11.5331	6.4001	1.1426
0.5	V	1.6612	1.047	2.0429
	S	3.0695	3.525	3.1458
	N	3.4644	2.0639	1.6342
0.01	V	17.7659	4.9794	13.5782
	S	13.9803	9.3311	11.1558
	N	16.2648	13.7127	4.0351
0.05	V	7.223	2.4523	7.5021
	S	6.7244	3.5396	4.8652
	N	6.1538	6.4107	3.2589

Table 4.1: Table of the error rates of answering three workloads (V:very sensitive, S:sensitive, N:normal) under each mechanism over the vector that has large uniform regions.

The Experiment over a Vector that has many regions of density

This experiment was done over a vector that has many regions of density as shown in figure 4.1. Since this vector has regions that have different counts, the partitioning mechanism based on the distribution of the data produces partitions with a large number of buckets. Therefore, as the number of the bucket become larger, the number of the positions within each bucket will be fewer. So, since the amount of the noise that has been generated for a particular bucket will be divided between the buckets' positions, each position would have a large amount of noise to be added to its count. In other words, As the buckets have fewer

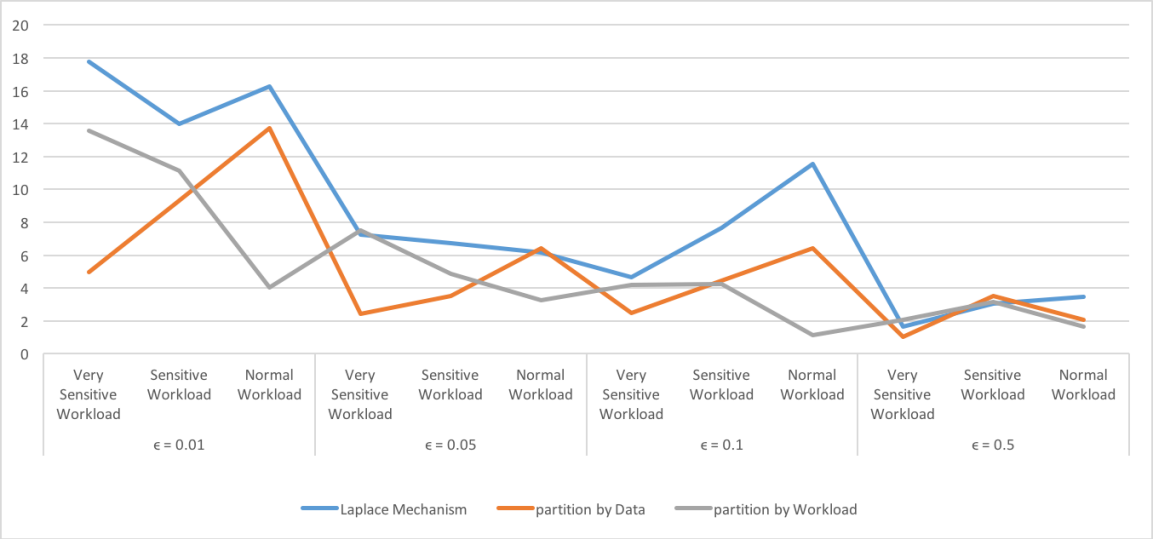


Figure 4.5: Error rates of answering three different workloads over the vector that has large uniform regions.

positions, the noise added to each position will be larger and therefore this mechanism performs worse. For example: as illustrated in figure 4.6, to answer the sensitive workload (a) that have been shown in figure 4.3 over the vector presented in figure 4.1 by the partitioning mechanism based on the data, the vector of the counts was partitioned to seventeen buckets most of them have only one position. So, the amount of the added noise to each position will be divided by one for most of the buckets which means more noise than if the bucket has more than one position like the situation described in the previous section.

In these cases, the performance of the partitioning mechanism based on the distribution of the data is very similar to the performance of the Laplace mechanism because most of the bucket has only one position because of the large differences between the vector counts. Table 4.2 and figure 4.7 present and show that in most cases the Partitioning mechanism based on the data performs as the Laplace Mechanism over the vector that has many regions of density. Since both mechanisms behave similarly as the number of the buckets become larger, the parameter ϵ will not contribute to make the performance of any mechanism better than the other. However, the error rates of both mechanisms will grow gradually according to the noise generated by Laplace distributions.

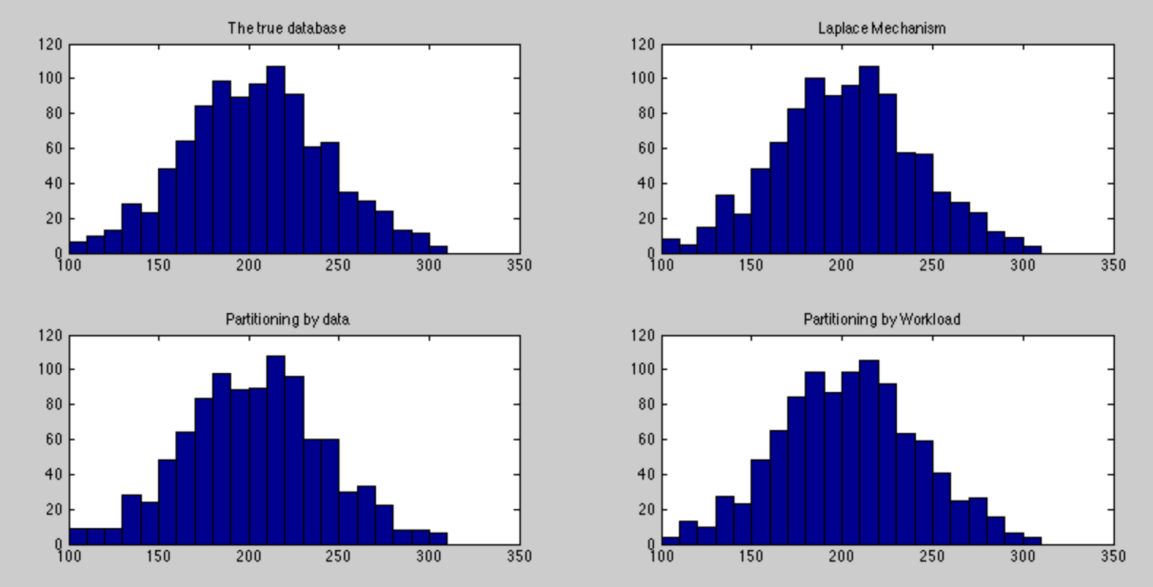


Figure 4.6: The true and private vectors produced by each mechanism of the proposed model to answer the workload (a) shown in figure 4.3 over the vector presented in figure 4.1.

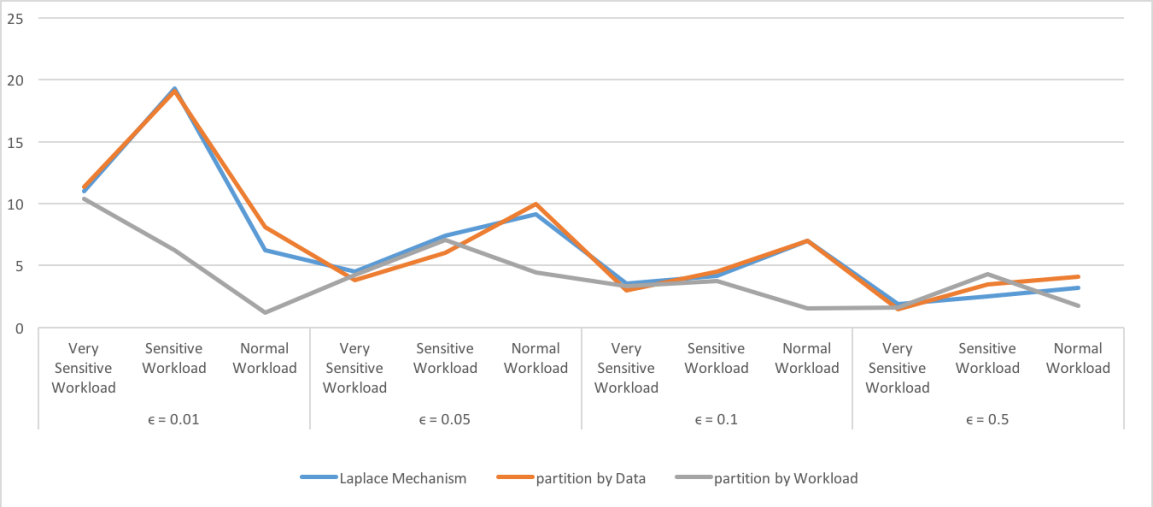


Figure 4.7: Error rates of answering three different workloads over the vector that has many regions of density.

ϵ	Workload	Laplace Mechanism	Partition by Data	Partition by Workload
0.1	V	3.5149	2.9623	3.3561
	S	4.1698	4.4906	3.7579
	N	7.0058	6.9771	1.529
0.5	V	1.8849	1.4922	1.6113
	S	2.4831	3.4746	4.3024
	N	3.1787	4.0677	1.769
0.01	V	11.0294	11.3886	10.3697
	S	19.306	19.1008	6.2201
	N	6.2267	8.083	1.1806
0.05	V	4.5272	3.8292	4.2514
	S	7.4267	6.0203	7.0993
	N	9.1475	9.9998	4.4158

Table 4.2: Table of the error rates of answering three workloads (V:very sensitive, S:sensitive, N:normal) under each mechanism over the vector that has many regions of density.

4.2.2 The partitioning Mechanism based on Workload

The partitioning mechanism based on the given workload performs better than the other mechanisms in most cases. Since this mechanism depends on the nature of the given workloads, its performance changes according to the kind of the given workload. As the given workload become more sensitive, the mechanism produces worse results because the sensitive workloads have more ranges with less number of positions than the normal workloads. Therefore, each position of the count vector would have a larger amount of noise to be added to its count as the given workload become more sensitive to preserve more privacy. That, of course, would increase the error rates and impact the quality of the produced results to provide more privacy for the database. For example workload (b) and (c) shown in figure 4.3 involve many ranges, and most of them are small. So, as illustrated in figure 4.4 and 4.6, the number of the buckets of the produced partition by this mechanism is very close to the number of its positions. So, most of the produced buckets have only one position therefore each position has an amount of noise to be added to its count to provide more privacy.

The values of the parameter ϵ affect the performance of the partitioning mechanism based on the given workload because the amounts of the generated noise depend on this parameter. So, as the values of this parameter become smaller, the error rates of this mechanism become larger according to the amounts of the generated noise as shown in figure 4.5 and 4.7.

4.3 Hypothesis Evaluation

The proposed model was designed to find an optimal partition of attribute values and answer range queries under differential privacy. Therefore, two partitioning mechanisms were developed based on the data itself and based on the given workload to investigate which one would perform better while preserving the privacy of the database. The results produced by these two mechanisms, which are discussed in the previous sections, show that

the accuracy of the produced answers to the given workload by the partitioning mechanism based the data depends on the distribution of that data. Moreover, the partitioning mechanism based on the given workload considers the sensitivity of the given workload, so the quality of the produced results will be based on that.

The partitioning mechanism based on the data produces better results than the Laplace mechanism when most of the vector's regions have close values because the positions of each region will be grouped in one bucket. So, the noise added to each position count will be small because of dividing the noisy count of that bucket uniformly among its positions. Therefore as the number of the buckets become smaller, the number of the buckets' items become larger and therefore, the amounts of the added noise to each position's count will be very small. That means the error rates will be smaller. However, in some cases, the given workload is very sensitive which requires more noisy answers to hide the individuals information and prevent inferring new information about these individuals. This observation indicates that the partitioning mechanism based on the data is not sufficient to preserve the individuals' privacy in some situations, and it needs to be improved to consider these situations.

Partitioning the counts vector based on the given workload contributes to decrease the error rates in most cases as shown on figure 4.5 and figure 4.7. Moreover, this mechanism consumes fewer resources because it depends on the given workload which is smaller than the data vector. The partitioning buckets are chosen based on the regions that the given workload focuses on. Therefore, as the number of the given workload become larger and the regions that these queries ask about become smaller, the buckets in the produced partition will be more. Thus, the error rates will be larger than if the given workload has fewer queries and larger ranges. So, this mechanism provides more privacy as the given workload has more sensitive queries. Therefore, this mechanism is better than the partitioning mechanism based on the data because it provides more privacy with fewer errors and consumes fewer resources.

Chapter 5

Conclusions

5.1 Current Status

The main goal of this work is verifying that the partitioning process of the data vector to answer a set of range quires under differential privacy is more efficient than the Laplace mechanism. Therefore, two different mechanisms were developed and tested to partition the data vector according to the data itself and according to the given workload.

The idea of Partitioning the count vector based on the data was inspired based on Li et al. approach [4]. To apply that, the vector of the values' counts is divided into buckets so within each bucket the counts of the positions are very close. Li et al. did that by generating all the possible partitions and choose the one that has the least cost. The cost of each bucket is calculated based on the bucket deviation and the noise that will be added to each bucket's count. However, instead of generating all the possible partitions, a greedy algorithm was developed to partition the vector which reduces the resources consumption and produces an optimal partition of the given vector. The algorithm takes each position of the vector and the last bucket that was generated form the previous process (if there is no bucket generated yet, a bucket that contains only the first position will be generated). Then two temporary buckets will be generated according to this bucket. The first temporary bucket contains the position that has been chosen by the algorithm with the positions that the last bucket has while the other temporary bucket has the same position that the last bucket has. The next step is calculating the cost of each bucket and the bucket that costs less than the other will be chosen to be a part of the final partition. This process will be repeated until

the algorithm tests all the positions of the counts' vector.By the end of these processes, an optimal partition will be generated as an output of this greedy algorithm.

Since a given workload may contribute to leak some information about the database if its queries ask about narrow ranges. The second partitioning mechanism was developed based on this observation. Therefore, according to the ranges that the given queries ask about, the vector will be partitioned. The algorithm that was built to do this task starts by the given workload and extracts the ranges of the queries and intersects them to see which ranges that most of the queries focus on to partition the data vector according to these ranges. So, the ranges that the given queries do not ask about will be in separate buckets to be treated like the other buckets.

After implementing these mechanisms and testing their performances, we found that the partitioning mechanism based on the given workload produce the best results in most cases. Moreover, when we looked at the cases that this mechanism produces worse results we found that the given workloads are very sensitive which means the amounts of the added noise to the real answers are big compared to the other mechanisms. That considered as an advantage of this mechanism because increasing the amounts of the added noise according to the sensitivity of the given workload provides more privacy.

5.2 Future Work

Since the partitioning mechanism based on the given workload produces outstanding results comparing to the partitioning mechanism based on the data and the Laplace mechanism, it can be a starting point for new research. Because the amount of the generated noise grows according to the sensitivity of the given workload, there should be a more sophisticated mechanism that measures the sensitivity of the given workload and link that to the sensitive ranges of the counts' vector. That could contribute to control the amounts of the added noise which provides more privacy and efficiency.

This work was done to answer workloads of range queries over single attribute. Therefore, this work should be extended to answer queries over multiple attributes. That would

contribute to make this model more realistic and provide more privacy since there are many related attributes in a particular database, and they may be used to learn sensitive information about the database and its individuals.

5.3 Conclusion

Partitioning the counts' vector could enhance the performance of answering a given workload of range queries under differential privacy. The partitioning mechanism based on the data could reduce the error rates unless the vector has many regions of density because in these cases the partitioning mechanism would put each position in a separate bucket and therefore it will behave exactly like the Laplace mechanism. The partitioning mechanism based on the given workload could reduce the error rates in most cases unless the given workload is very sensitive. That because the amounts of the added noise to the real counts are generated depending on the sensitivity of the given workload to provide more privacy.

Bibliography

[1] C.Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, 2006.

[2] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*. Springer Verlag, 2006.

[3] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. 2014.

[4] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. A data- and workload-aware algorithm for range queries under differential privacy. *Proc. VLDB Endow.*, 7(5):341–352, jan 2014.

[5] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *PODS*, 2010.

[6] Chao Li and Gerome Miklau. An adaptive mechanism for accurate query answering under differential privacy. *Proc. VLDB Endow.*, 5(6):514–525, feb 2012.

[7] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), mar 2007.

[8] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(15):557–570, oct 2002.

[9] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. In *ICDE*, 2010.