Improving a De-identification Algorithm that Achieves both k-Anonymity and Differential Privacy

Tianyu Li

Master of Science
School of Informatics
University of Edinburgh
2019

Abstract

Data is playing a more and more important role in our lives. For citizens, data is all around them and data includes one's confidential information. People want their data safe for privacy concerns. However, for curators, they have to publish datasets for the sake of research or commerce, so there is a problem: how to preprocess the dataset before publishing in a way that can both protect individuals privacy and retain useful information for different uses. To deal with that, *k*-anonymity and differential privacy provide syntactic and semantic guarantees. The work of [22] combined k-anonymity with differential privacy and proposed a *k*-anonymous differential private algorithm: ε-safe lattice anonymization (ε-safe LA). In this project, we have reproduced the work of ε-safe LA with adapted utility function and sensitivity, which corrects the mistakes in his work and guarantee that ε-safe LA satisfies $(\beta, \varepsilon, \delta)$-differential privacy under sampling. We have carried out a series of experiments to explore the influence of parameters in ε-safe LA on the probability of outputting a useful dataset. We have also evaluated ε-safe LA by comparing it to other algorithms in terms of accuracy for learning tasks and running time, finding that ε-safe LA shows a moderate accuracy with a long running time. To improve the efficiency of ε-safe LA, we proposed and evaluated two interpolation methods which can half the running time of the algorithm.

Table of Contents

Chapter 1

Introduction

1.1 History

In an information era, privacy is a growing concern since data is being collected and used all around us. One can obtain lots of individual information from data, which causes privacy issues. Meanwhile, data is needed to be published for scientific research such as machine learning tasks, etc. When data curators publish a dataset in which the data is collected from the public, the dataset needs to strike a tradeoff between protecting people's privacy and retaining the information inside the data. To achieve this goal, different kinds of approaches have been proposed with anonymization being one among the most popular methods [16]. For anonymization, k-anonymity and differential privacy have been proposed by researchers and adopted by industry.

Samarati and Sweeney firstly introduced the concept of k-anonymity in 1998 [27]. A dataset is k-anonymous if for each individual in the dataset, there are no less than $k-1$ other individuals which show the same value. For example, in Table 1.1, the mini dataset is 2-anonymous for the attributes (*'age', 'gender', 'country'*) since each tuple, such as ('$20 < Age \leq 30$', 'Female', 'United States'), appears at least two times. The tuple contains all the identical messages and it is an ensemble of different attributes. The tuple is called the *quasi-identifiers*, which will be de-identified in the dataset, and the other attributes (*'disease'*) are the *sensitive* attributes which are assumed unknown from the attackers. In practice, the value of each attribute in a dataset can be known from an attacker, and in this project, all attributes are deemed as *quasi-identifiers* in the later experiments, which can enhance the privacy concerns.

Differential privacy was raised by Dwork in 2006 [9, 10], which protects the privacy in a different way. Differential privacy pays more attention to how much the

existence of a particular row of data influences the final output. If the output changes very small (less than a factor e^{ε}) when a row of data is changed, the observers can't distinguish whether the particular data was used in the computing, and thus protect the privacy.

Quasi-identifiers			Sensitive
Age	Gender	Country of domicile	Disease
$20 < Age \leq 30$	Female	United States	Cancer
$20 < Age \leq 30$	Female	United States	TB
$30 < Age \leq 40$	Male	England	No illness
$30 < Age \leq 40$	Male	England	TB
$30 < Age \leq 40$	Male	England	Heart-related

Table 1.1: Some fictitious patients data from Wikipedia in which k = 2.

1.2 Problem Statement

Both k-anonymity and differential privacy are proposed to preserve the privacy and information of a dataset, but they provide different guarantees. k-anonymity is syntactic while differential privacy provides semantic guarantees [20]. For k-anonymity, the guarantee is weak which only requires that each tuple appears no less than k times in the dataset. For differential privacy, it's algorithmic and probabilistic, in which the guarantee is stronger. However, the strong guarantee also makes differential privacy more difficult to be implemented in that it must limit the influence of any single row of data on the final output.

Both k-anonymity and differential privacy have different features and limitations, and the problem of the project is how to combine the use of k-anonymity and differential privacy in a more efficient way and to show how well the combined de-identification algorithm works.

1.3 Objective and Significance

The objective of the project is to find an algorithm that both better protects individual privacy in the dataset and limits the information loss. We want to achieved this from the following steps:

- Reproduce the result of a k-anonymous differential private algorithm (ε-safe LA) based on the work of [22]. ε-safe LA achieves differential privacy by sampling and node selection, and the node selection is implemented by a *k*-anonymous algorithm OLA [12] and the exponential mechanism [24]. ε-safe LA will be further introduced in Chapter 3.

- Compare the performance of ε-safe LA with different parameters (ε, penalty factor, etc) to find the best parameters.

- Improve the efficiency of ε-safe LA by using interpolation methods and compare the performance of ε-safe LA with or without interpolation methods. The interpolation methods include the level-based method which predicts the values of nodes in skipped levels and the dictionary-based method which predicts the values of nodes with same depth on certain attributes. The interpolation methods will be further explained in Chapter 3.

- Compare the accuracy of regression models to evaluate the output dataset published by ε-safe LA, different differential privacy methods, *k*-anonymity method, the raw dataset, etc. The actual information loss in a learning task is shown by the change of accuracy, and it's supposed to find out which algorithm both protects individual privacy and limits the information loss.

The outcome of the project can be used as reference for data curators such as governments or corporations to publish a dataset which is more useful for scientific research (learning tasks, etc.) and protect the privacy of individuals better. Consequently, my results benefits the owners of data such as citizens and the users of data such as researchers.

1.4 Main Contributions

- Do a literature review of related works in differential privacy and *k*-anonymity in Chapter 1 and Chapter 2.

- Reproduce the work of [22] with some improvements. In this project, we correct the calculation of sensitivity with an alternative utility function in Chapter 3

- Prove that ε-safe LA is ε-differential private in Chapter 3.

- Compute the value of ε, ε' and δ in Chapter 3, Chapter 4 and Chapter 5.

- Explore the mutual restraints between (k, β) pairs and the value of δ. Also, explore the influence of parameters (k, ε', β and penalty factor) on ε-safe LA in Chapter 4 and Chapter 5.
- Propose two different interpolation methods to improve the efficiency of ε-safe LA in Chapter 3, and design experiments to evaluate the performance of interpolation methods in Chapter 4 and Chapter 5.
- Evaluate ε-safe LA with learning tasks on Adult dataset in Chapter 4 and Chapter 5. Logistic regression models are trained to show the difference of accuracy between the raw dataset and the output dataset by ε-safe LA. In the learning task, *one-hot* encoding and standard scalar are applied.
- Compare the performance of ε-safe LA with other de-identification algorithms: OLA and DP functional logistic regression in Chapter 4 and Chapter 5. The comparison is carried out by learning tasks on Adult and IPUMS datasets in terms of accuracy and running time.
- Show the future directions after the project.

1.5 Report Structure

- **Chapter 2** introduces the related work about k-anonymity, differential privacy and the combination of k-anonymity and differential privacy.
- **Chapter 3** shows the theoretical basis of the project. It includes the process and algorithm of ε-safe LA with the concept of interpolation methods. Also, it contains the proof that ε-safe LA is differential private and the derivation of sensitivity based on the proposed utility function.
- **Chapter 4** describes the dataset used for test and how experiments are designed, including the sections of exploring ε-safe LA, the interpolation methods and the comparison results.
- **Chapter 5** illustrates the results for the designed experiments in the same aspects as in Chapter 4.
- **Chapter 6** further discusses the results of experiments and summarizes the work that remains to be done.

Chapter 2

Background

2.1 k-anonymity

After the concept of k-anonymity was proposed, some attacks occurred which demonstrated the weakness of k-anonymity. Two popular attacks are the homogeneity attack and the background knowledge attack [21]:

- **Homogeneity Attack.** Even if a dataset is k-anonymous, attackers can know the sensitive information if all the k tuples of quasi-identifiers share the same value in the sensitive attribute.
- **Background Knowledge Attack.** Similar to homogeneity attack, even if the data shows more than one values in sensitive attribute, attackers can know the sensitive information based on some background knowledge. For example, if an attacker is not sure whether a Japanese caught a virus or has heart disease, but the attacker knows that Japanese are seldom infected by heart disease. The attacker can infer that it's virus.

l-Diversity was proposed by Machanavajjhala, et al. [21] to deal with the attacks which are based on the similar sensitive attributes of the same k quasi-identifiers. A dataset has l-diversity if there are at least l different sensitive attributes for each quasi-identifier. The diversity of sensitive attributes is supposed to confuse the adversary with more interference, and thus protect against attacks.

However, Ninghui Li, et al. [19] found that l-diversity is not necessary and not sufficient for attribute disclosure prevention. The main reason is that l-diversity didn't consider the semantic relationships between different attributes. To fix the problem,

t-closeness is raised. By t-closeness, the distribution of sensitive attribute in each k-anonymity group is controlled to be similar to the distribution of sensitive attribute in the whole data set, and t is the distance threshold. From the same work, t-closeness and l-diversity can work together based on the generalization property of t-closeness [19]. However, there are still some limitations for t-closeness in that the value of t is difficult to pick and the value of some sensitive attributes are restricted to be higher than the values of other attributes [14].

t-Closeness and l-diversity always try to satisfy some strong guarantees. Based on the work of [7], the combination of k-anonymity and ε-differential privacy can also output a dataset which satisfies stochastic t-closeness, which is an extension to t-closeness.

There are some other k-anonymous algorithms. For example, Datafly [28] is a heuristic k-anonymous algorithm. The algorithm generalizes the quasi-identifiers which show the most distinct values. Then it repeats the process until the remained dataset is k-anonymous. The algorithm runs very fast with a time complexity of $\mathcal{O}(n \log n)$. However, the algorithm can't find the global optimal since it may generalize too many quasi-identifiers, and the information loss has never been calculated.

Mondrian [18] is another modern k-anonymous algorithm proposed by Kristen LeFevre, et al. This method took use of *kd-tree* [13] to implement a greedy approximation algorithm. The proposed multidimensional model for k-anonymity also runs very fast with a complexity of $\mathcal{O}(n \log n)$. Mondrian can split the dataset and reconstruct it with equivalence classes whose size is at least k. Mondrian works well with numerical data, but on categorical data it has been previously proposed and doesn't work well [30].

2.2 Differential Privacy

2.2.1 Definition

According to the work of Dwork [9, 10], differential privacy was raised to protect individual privacy and make better use for the dataset. Differential privacy cares about how a single row of data influences the final output, so the concept of neighbouring dataset is introduced. To denote it more clearly, l_1-distance ($||X - Y||_1$) shows the difference of the number of entries in two databases X and Y. And D and D' are neighbouring datasets if $||D - D'||_1 \leq 1$, which means that they only differ in one or

zero row of data. Based on that, an algorithm $\mathcal{A}$ satisfies ε-differential privacy if and only if for neighbouring datasets D, D' and any range $O \subseteq \text{range}(\mathcal{A})$:

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in O] \tag{2.1}$$

From Equation 2.1, it's shown that for the neighbouring datasets, the probability of whether the output belongs to O differs less than e^{ε}, which hides the existence of each individual. However, the guarantee is so strong that it's very hard to be implemented and it's excessive for most situations. To make it more practical, δ is added to the equation as a small error factor. Similarly, $\mathcal{A}$ satisfies (ε, δ)-differential privacy if:

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in O] + \delta \tag{2.2}$$

From Equation 2.2, the lower the values of ε and δ are, the better the algorithm protects the privacy. Based on the equation, the key question in differential privacy is how to better add the perturbation to get the low ε and δ while meet the strong guarantee. To achieve that, lots of studies are carried out and some mechanisms which satisfy (ε, δ)-differential privacy such as Laplace Mechanism in Section 2.2.2 and Exponential Mechanism in Section 2.2.3 have been proposed.

2.2.2 Laplace Mechanism

The Laplace mechanism is the most general mechanism for differential privacy and it adds Laplace noises [10]. For a query $f : \mathcal{D}^N \to \mathbb{R}$, the l_1-*sensitivity* (Δf) is:

$$\Delta f = \max_{X, X' \in \mathcal{D}^N : ||X - X'||_1 \leq 1} ||f(X) - f(X')||_1 \tag{2.3}$$

To add the noise, the Laplace distribution is applied and the distribution is centered at zero with a scale parameter b:

$$Lap(x \mid \mu = 0,\ b) = \frac{1}{2b} \exp(-\frac{|x|}{b}) \tag{2.4}$$

We use $Lap(b)$ to denote the $Lap(x \mid \mu = 0,\ b)$, and with a larger value of b, the $Lap(b)$ will be flatter as shown in Figure 2.1.

Then for the query $f : \mathcal{D}^N \to \mathbb{R}$, a randomized algorithm $\mathcal{A}$ satisfies ε-differential privacy if $\varepsilon > 0$ and:

$$\begin{aligned} \mathcal{A}(D,\ f,\ \varepsilon) &= f(D) + (Y_1\ \ldots\ Y_k) \\ \text{where} \quad (Y_1\ \ldots\ Y_k) &\sim Lap(\frac{\Delta f}{\varepsilon}) \text{ i.i.d.} \end{aligned} \tag{2.5}$$

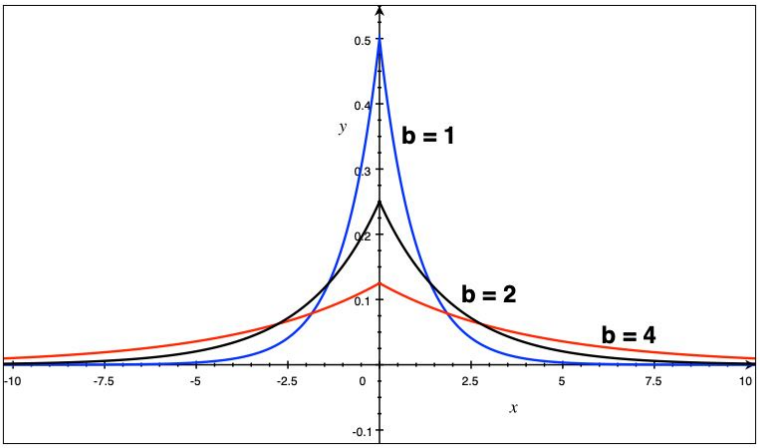


Figure 2.1: Probability density with different value of b.

2.2.3 Exponential Mechanism

The exponential mechanism [24] was proposed in 2007 as a technique for designing algorithms with differential privacy. In the exponential mechanism, a utility function $u : \mathcal{D}^N \times \mathcal{R} \to \mathbb{R}$ is defined to access the utility of each element $n \in \mathcal{R}$, and a measure μ is used to assign a large probability of elements with a large utility.

Among the calculation, the *sensitivity* (Δu) of the utility function need to be calculated first, which is defined as:

$$\Delta u = \max_{n \in \mathcal{R}} \max_{X, X' \in \mathcal{D}^N : ||X - X'||_1 \leq 1} |u(X, n) - u(X', n)| \tag{2.6}$$

And the output probability of exponential mechanism satisfies ε-differential privacy (where $\varepsilon = 2\varepsilon' \Delta u$) :

$$\Pr[\mathcal{A}^{\varepsilon'}_{u,\Delta u}(X) = t \in \mathcal{R}] = \frac{\exp(\varepsilon' \cdot u(X,t)) \cdot \mu(t)}{\int_{\mathcal{R}} \exp(\varepsilon' \cdot u(X,n)) \cdot \mu(r) dr} \tag{2.7}$$

2.3 Combination of k-Anonymity and Differential Privacy

As mentioned in Section 1.2, k-anonymity and differential privacy show different limitations and can work together to make better use of a dataset in terms of privacy protection and scientific use. Ninghui Li et al. [20] proposed a method in which k-anonymity is used as another way of adding perturbation. When the Laplace or Gaussian mechanism is applied, the noise is always added to the dataset directly. The proposed method

achieved differential privacy by doing a sampling and pruning the records which violate *k*-anonymity.

Firstly, the idea of differential privacy under sampling $(\beta, \varepsilon, \delta)$-DPS is proposed, and β is the probability of each row of data selected from the raw dataset. For an algorithm $\mathcal{A}$, if $\mathcal{A}^{\beta}$ is ε-differential private, $\mathcal{A}$ satisfies $(\beta, \varepsilon, \delta)$-DPS. Also, a smaller β will result in a smaller ε.

However, the existed *k*-anonymity algorithms are not "safe" enough since they do the mapping based on the data of individuals in the dataset. In the work of Ninghui Li, a strongly "safe" *k*-anonymity algorithm is defined which prunes all the individuals which appear less than k times in the same dataset. Such algorithm is compatible to all dataset but the output dataset is with extremely low utility because most individuals are removed. To deal with the problem, They introduced ε-safe *k*-anonymity.

An algorithm $\mathcal{A}$ is a ε-safe *k*-anonymity algorithm if the mapping function $\mathcal{A}_m$ is ε-differential private. Moreover, ε'-safe *k*-anonymity algorithms satisfy $(\beta, \varepsilon, \delta)$-DPS where $\varepsilon \geq -\ln(1-\beta) + \varepsilon'$. However, no concrete ε-safe *k*-anonymity algorithm has been proposed in the literature.

Other than the work of Ninghui Li, there are other techniques for combining *k*-anonymity and differential privacy. Naoise Holohan, et al. [16] applied k-anonymity to part of the quasi-identifiers and applied differential privacy to the rest. The approach shows a better result than simply using k-anonymity or differential privacy alone.

Chapter 3

Design and Implementation of Privacy-Enhancing Algorithms

3.1 Optimal Lattice Anonymization (OLA)

3.1.1 OLA

In Section 2.1, the basic ideas about *k*-anonymity are introduced which include the potential attacks and some improved methods. In our project, Optimal Lattice Anonymization (OLA) [12] was selected among different *k*-anonymous algorithms to serve as the basic algorithm of *k*-anonymity according to the work of [22].

OLA was proposed by Khaled El Emam, et al. [12] in 2009. In order to satisfy *k*-anonymity, the values of quasi-identifiers can be generalized to reduce the precision [3]. For each dataset, *generalization hierarchies* define the rules of generalization for each attributes, and Figure 3.1 shows an example. The precision is reduced when the attribute is with a higher level in the hierarchy.

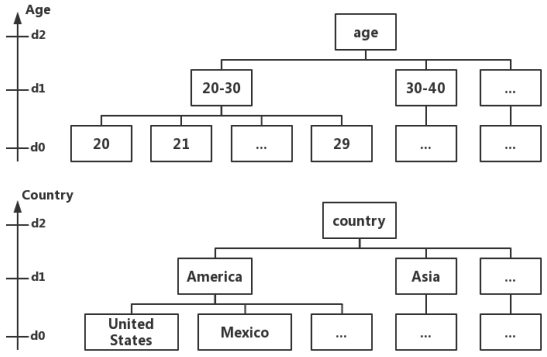


Figure 3.1: Example generalization hierarchies for quasi-identifiers *'age'* and *'country'*.

In OLA, a *lattice* is firstly generated as shown in Figure 3.2. Each *node* in the lattice represents the combination of levels of different attributes, and the lattice includes all the possible nodes. Also, a *generalization strategy* is defined as a connected path from the bottom to the top, and a binary search is carried out to find all the **k-anonymous** nodes.

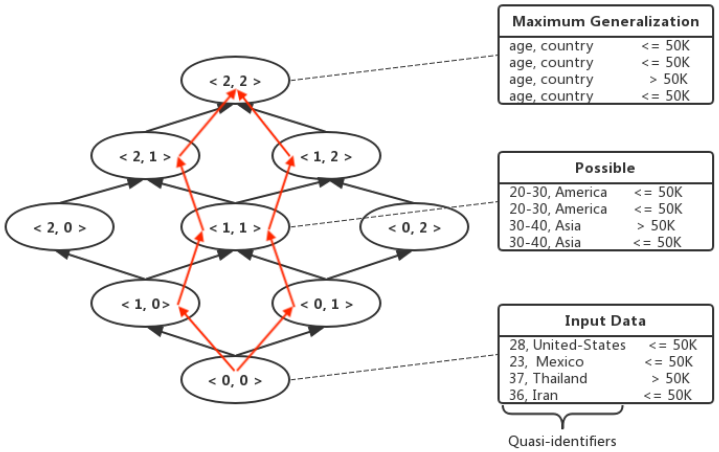


Figure 3.2: The example lattice and generalization strategies. In the left part of the figure, the lattice shows the generation levels for quasi-identifiers *'age'* and *'country'*, and the depth of each quasi-identifier is two. Two different strategies are shown from the red routes (from bottom to the top) and both routes go through the node (1, 1). In the right part of the figure, it shows the value of certain nodes.

When a node is considered, the rows which violate the *k*-anonymity criterion are pruned and the *suppression* is the proportion of the pruned rows in the whole dataset. Based on the value of suppression for each node, the node is said to be **k-anonymous** if its suppression is smaller than the ***MaxSuppression*** which is a constant (usually 5%). However, when the number of attributes increases, the number of nodes will increase exponentially, which results in a huge lattice and makes it time-consuming to check all the nodes. To mitigate the problem, the conception of *k-minimal node* is introduced. *k-minimal nodes* are *k*-anonymous nodes with the lowest depth for each generalization strategy. For a dataset with more than *k* records, *k*-minimal nodes exist and there is exactly one *k*-minimal node for each strategy. To compare the utility of each node, the value of information loss can be calculated from different information loss metrics as shown in Section 3.1.2. When all the *k*-minimal nodes are achieved, the *k*-minimal node which shows the least information loss is the global optimal for the dataset because of two important properties for OLA according to the work of [12]:

- If one node is *k*-anonymous, all its ancestor nodes are *k*-anonymous.

- For any specific strategy, the information loss of nodes in different levels is monotonic. A node with a lower height shows less information loss.

In this project, OLA is chosen as the algorithm for k-anonymity since it shows better performance and less information loss than some other k-anonymous algorithms [12]: Datafly [28], Samarati [26], etc. Mondrian [18] is also a good alternative algorithm as introduced in Section 2.1 and it shows better performance with numerical data while OLA is better with categorical data.

3.1.2 Information Loss Metric

Information loss metric $IL(X,n)$ is used in this project to use a numerical data to show the information loss of a dataset. It is applied to OLA to find the 'best' k-minimal node which shows the least information loss and it serves as a tool to evaluate the utility of each node. There are two common methods:

- **Precision**, or *prec* [29]. This method is appropriate for hierarchical data and it is dependent on generalization and suppression, which are about how much each attribute is generalized and how many rows of data are pruned. For a raw data X, we can generalize the dataset for a certain node if the generalization hierarchies are defined. For example, one generalization of X_g can be expressed as

$$X_g = g(X,n) \tag{3.1}$$

  where g is the generalization function to output the generalized dataset for node n with its defined hierarchies. Also, the equation of *prec* is shown in Equation 3.2 in which X_g is one generalization of the raw dataset X. X is with N rows of data and N_A attributes; A_i is an attribute; DGH_{A_i} is the domain generalization hierarchy of attribute A_i, which can also deemed as the maximum depth for each attribute; and g is the generalization function. In a node n, it is defined as $(n_{A_0}, n_{A_1}, \dots)$ where n_{A_i} is the depth of attribute A_i in n.

$$Prec(X,n) = 1 - \frac{\sum_{i=1}^{N_A} \sum_{j=1}^{||X_g||_1} \frac{n_{A_i}}{|DGH_{A_i}|}}{|X| \cdot N_A} \tag{3.2}$$

  In the equation, it calculates the total proportion of depth for each attribute and each row of data. In OLA, all rows of data are generalized with same depth for the same attribute. For example, if the depth of 'attribute 1' is *one* for *row1*, the

depth of 'attribute 1' for all other rows in the dataset is *one* as well. Such property in OLA can greatly help simplify the equation. If we define that *suppression* $sup(X,n)$ is the proportion of the pruned rows in the raw dataset for node n, the equation can be shown as:

$$\begin{aligned} Prec(X,n) &= 1-(1-sup(X,n))\frac{1}{N_A}\sum_{i=1}^{N_A}\frac{n_{A_i}}{|DGH_{A_i}|} \\ \text{where} \quad sup(X,n) &= 1-\frac{||X_g||_1}{|X|} = 1-\frac{||g(X,n)||_1}{|X|} \end{aligned} \tag{3.3}$$

Moreover, if we use $gen(X,n)$ to denote the proportion of the **remained** information from generalizing the attributes, (a low value of $gen(X,n)$ means a high value of information loss), the equation can be finally shown as:

$$\begin{aligned} Prec(X,n) &= 1-(1-sup(X,n))\cdot gen(X,n) \\ \text{where} \quad gen(X,n) &= \frac{1}{N_A}\sum_{i=1}^{N_A}\frac{n_{A_i}}{|DGH_{A_i}|} \end{aligned} \tag{3.4}$$

- Entropy-based metric [5]. This method is used for non-uniform distribution problems. From Equation 3.5, the $\Pr(a_j|b_{j'})$ is the conditional probability of two randomly selected records a_r from the original dataset X and $b_{j'}$ from the generalized dataset X_g. In Equation 3.6, I is the indicator function which returns 1 only if the condition is satisfied and it returns 0 in other conditions. i means the ith row of data and j means the jth attribute. In Equation 3.6, $b_j^{(i)}$ is the generalization of $a_j^{(i)}$ for node n. Note that the entropy-based metric is an alternative metric instead of *prec*, but it's not used in the project.

$$\Pr(a_j|b_{j'}) = \frac{\sum_{i=1}^{|X|} I(X_j^{(i)} = a_j)}{\sum_{i=1}^{||X_g||_1} I(X_{g\ j'}^{(i)} = b_{j'})} \tag{3.5}$$

$$\begin{aligned} H(X,n) &= -\sum_{i=1}^{|X|}\sum_{j=1}^{N_A} log_2(\Pr(a_j^{(i)}|b_j^{(i)})) \\ \text{where} \quad a_j^{(i)} &\in X \quad \text{and} \quad b_j^{(i)} = g(a_j^{(i)},n) \end{aligned} \tag{3.6}$$

3.2 ε-safe LA

3.2.1 Limitation of k-Anonymity

Besides the attacks mentioned in Section 2.1, it's possible for attackers to find and take use of additional information to re-identify the individual. There is a lot of unknown

background knowledge which could help attackers isolate one of the k individuals. Another point is that the extreme values can greatly influence the range of generalization. The generalization hierarchies is supposed to include all the range of values. For example, if the income of 95% individuals is smaller than \$500 while another 1% is larger than \$500,000, the large range can be used for attackers to infer the existence of the wealthy. To better protect the privacy of individuals, differential privacy provides a more precise definition that the probability of consequences change only by a small factor when a row of data is changed.

3.2.2 Combination with Differential Privacy

Based on the idea of $(\beta, \varepsilon, \delta)$-DPS and ε-safe k-anonymity algorithm introduced in Section 2.3, it's practical to combine OLA with differential privacy by transforming OLA to an ε-safe k-anonymity algorithm. The process is shown below:

Algorithm 1 Main process of ε-safe LA

Input: Input dataset for de-identification D_{in}

Output: Output dataset of ε-safe LA D_{out}

1: Apply sampling to D_{in}, and get D'_{in}
2: Construct the lattice of D'_{in}
3: Calculate the utility of each node by Equation 3.10
4: * Calculate the sensitivity for exponential mechanism by Equation 3.11
5: Output the probability by exponential mechanism
6: Output a node n_i according to the probability
7: Generalize the dataset D_{out} for n_i by using generalization hierarchies
8: **return** D_{out}

* Note: Line 4 (sensitivity) can be calculated anywhere before line 5 (if parameters are certain) because it is a constant and not changed.

From the definition of $(\beta, \varepsilon, \delta)$-DPS, the sampling should be applied before the k-anonymity algorithm, so we firstly compute the sampled dataset D'_{in}. We then construct the lattice of the dataset D'_{in} in the same way as OLA. After that, we use the exponential mechanism to serve as a mapping algorithm $\mathcal{A}_m$ to output a node in a differential private way. According to the conclusion of [20], the algorithm $\mathcal{A}$ is an ε-safe k-anonymity algorithm if it is built from a mapping function $\mathcal{A}_m$ that is ε-differential private. Also, it is proved that ε'-safe k-anonymity algorithm satisfies $(\beta, \varepsilon, \delta)$-DPS.

As a result, after using the exponential mechanism, the algorithm ε-safe LA satisfies $(\beta, \varepsilon, \delta)$-DPS and: ($f$ is the probability mass function for the binomial distribution)

$$\varepsilon \geq -\ln(1-\beta) + \varepsilon' \tag{3.7}$$

$$\delta = d(k, \beta, \varepsilon - \varepsilon') = \max_{n: n \geq \lceil \frac{k}{\gamma} \rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta), \gamma = \frac{(e^{\varepsilon - \varepsilon'} - 1 + \beta)}{e^{\varepsilon - \varepsilon'}} \tag{3.8}$$

In order to adopt the exponential mechanism as the mapping algorithm, as introduced in Section 2.2.3, we must define the utility function and calculate the sensitivity of utility. From the work of [22], the penalized information loss metric is used as the utility function to balance the influence of suppression and generalization. The utility function and sensitivity are defined as:

$$\begin{aligned} u(X,n) &= 1/\left(\underbrace{(1 - gen(X,n))}_{generalization} + \underbrace{f \cdot sup(X,n)}_{suppression} \right) \\ \text{and} \quad \Delta u &= f \cdot \frac{1}{|X|} \end{aligned} \tag{3.9}$$

where X is the dataset and n is the node. The first part calculates how much is generalized for all attributes and the second part is the penalized suppression in which f is the penalty factor (a constant). $gen(X,n)$ and $sup(X,n)$ are defined in Equation 3.4 and 3.3. A higher suppression or a deeper generalization will either increase the value of the first or the second part, so the inverse is applied to make sure that a higher value means a higher utility. However, the sensitivity doesn't match the utility function since it simply use the change of denominator as the sensitivity. Also, if one row of data is changed, the generalizing dataset can be with a change of maximum k rows of data, which is also not considered in his work.

To deal with the problem, the utility function is adapted to Equation 3.10. The new function has changed the *inverse* into *negative*, which can greatly reduce the complexity when computing the sensitivity. Also, although all the values of utility are negative numbers, the negative can guarantee that a higher value means a higher utility.

$$u(X,n) = -\left(\underbrace{(1 - gen(X,n))}_{generalization} + \underbrace{f \cdot sup(X,n)}_{suppression} \right) \tag{3.10}$$

Based on the utility function, the sensitivity can be calculated as Equation 3.11. When one row of data changes in the raw dataset X, the generalization part for the

same node remains the same in the utility function. However, the change of one row can result in the pruning of up to k rows of data. The sensitivity for the utility function is $f \cdot \frac{k}{|X|}$ where f is the penalty factor and $|X|$ is the size of the raw dataset.

$$\begin{aligned}\Delta u &= \max_{n\in\mathcal{R}} \max_{X,X'\in\mathcal{D}^N:||X-X'||_1\leq 1} |u(X,n)-u(X',n)| \\ &= \max_{n\in\mathcal{R}} \max_{X,X'\in\mathcal{D}^N:||X-X'||_1\leq 1} f\cdot|sup(X,n)-sup(X',n)| \\ &\leq f\cdot|sup(X,n)-\left(sup(X,n)+\frac{k}{|X|}\right)| = f\cdot\frac{k}{|X|}\end{aligned} \tag{3.11}$$

3.3 Interpolation Method

After ε-safe LA is proposed, another question occurs that the algorithm runs slowly when the number of attributes increases. To deal with such problem, we design two ways of computing the utility of nodes more efficiently by interpolating the utility from neighbouring nodes in the lattice.

- **Level-Based Interpolation**. As shown in Figure 3.3, we divide the lattice into nodes for which we compute a precise utility and nodes marked as '*' for which need interpolation. When the number of nodes is large (>10,000), the process of calculating the suppression will be time-consuming. By using the interpolation method, only about half the nodes are needed to calculate the utility. We confine the utility for the node as the average utility of its parent nodes and child nodes.

- **Dictionary-Based Interpolation**. The dictionary-based interpolation method saves time in another way. To avoid repeatedly calculating the generalized data, the generalized data for each attribute with each row is saved in a large dictionary:

$$\begin{aligned}d = \{&'attr0' : \{0 : [r1_{00}, r2_{00}\ldots], 1 : [r1_{01}\ldots]\ldots\}, \\ &'attr1' : \{0 : [r1_{10}, r2_{10}\ldots]\ldots\} \quad \ldots\}\end{aligned} \tag{3.12}$$

  In the dictionary, for rx_{ij}, x is the index of the row number, i is the index of attributes and j is the depth. The dictionary can store all the generalized values and avoid generalizing the same data. The dictionary-based interpolation method predicts the utility of certain nodes that share the same (attribute, depth) pair (the same ij). As shown in Figure 3.4, all the predicted nodes shows that the depth of 'attr2' is one. The predicted utility is also the average of its parent nodes and child nodes.

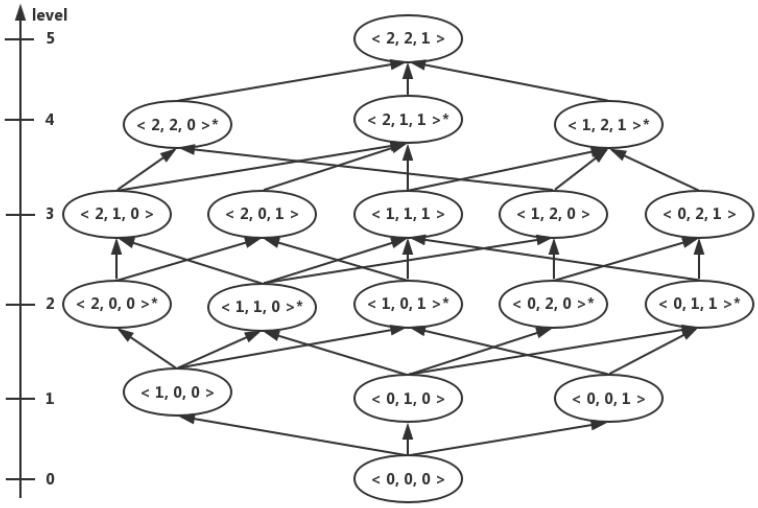


Figure 3.3: The example lattice for level-based interpolation method.

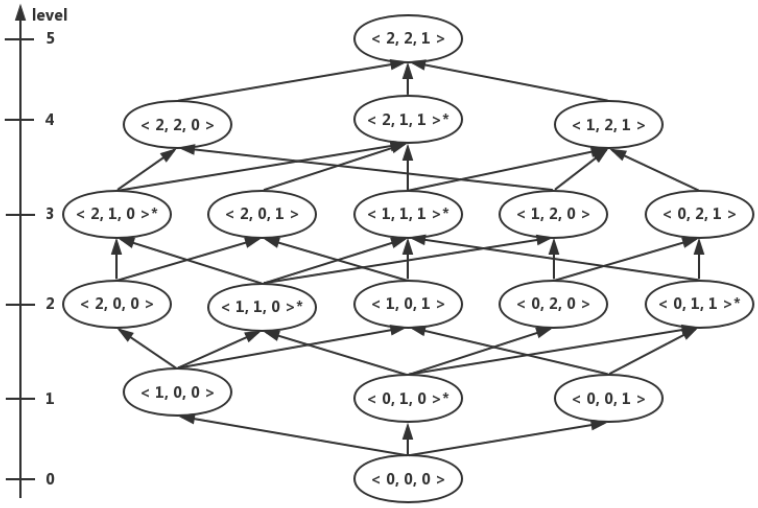


Figure 3.4: The example lattice for dictionary-based interpolation method. Here nodes in form of $< *, 1, * >$ are interpolated where * can be any depth.

After applying both interpolation methods, the sensitivity of the algorithm remains $f \cdot \frac{k}{|X|}$. The new utility function is shown in Equation 3.13.

$$u(X,n) = \begin{cases} -((1-gen(X,n)) + f \cdot sup(X,n)) & \textbf{NOT Interpolated Nodes} \\ \dfrac{\sum u(X,n_{parent}) + \sum u(X,n_{child})}{|n_{parent}| + |n_{child}|} & \text{Interpolated Nodes} \end{cases} \tag{3.13}$$

For the predicted nodes, the sensitivity remains $f \cdot \frac{k}{|X|}$, as shown in Equation 3.14.

$$\begin{aligned}\Delta u &= \max_{n\in\mathcal{R}} \max_{X,X'\in\mathcal{D}^N:||X-X'||_1\leq 1} |u(X,n)-u(X',n)| \\ &\leq \frac{\sum_{n_i\in n_{parent}\ or\ n_i\in n_{child}} \max_{n\in\mathcal{R}} \max_{X,X'\in\mathcal{D}^N:||X-X'||_1\leq 1} |u(X,n_i)-u(X',n_i)|}{|n_{parent}|+|n_{child}|} \\ &\leq \frac{(|n_{parent}|+|n_{child}|)\cdot f\cdot\frac{k}{|X|}}{|n_{parent}|+|n_{child}|} = f\cdot\frac{k}{|X|}\end{aligned} \tag{3.14}$$

3.4 Implementation Method

In practice, ε-safe LA is implemented by python. The computer experiment is MacOS Mojave, 1.7 GHz Intel Core i7 Processor and 8 GB memory.

About the codes, the structure of the lattice is defined as a two-dimension dictionary where the first key is the level of a node and the second key is the node, which can be simply shown as

$$\begin{aligned}lattice = \{level_0 : \{n_0 : \{[info, sup, u, prob]\}\}, \\ level_1 : \{n_1 : \{[info, sup, u, prob]\}, \\ n_2 : \{[info, sup, u, prob]\}, \dots\}, \\ \dots\}\end{aligned} \tag{3.15}$$

where $level_0 = 0$, $level_1 = 1\dots$ and n_0 is with a structure as $(0,0,0,0,0)$ for $level_0$. The value of each node is a list which includes the value of information loss, the suppression, the utility and the output probability.

Also, Equation 3.12 shows the dictionary which stores the transformed data for each row with each attribute and each depth. The dictionary can avoid repeatedly compute the generalized value for the same value.

Meanwhile, we have used some libraries to support the programming. We used *pandas* [23] to handle the input dataset, *numpy* to deal with lists and the *groupby* function to calculate the value of k.

More detailed, in the algorithm, we firstly load the dataset by *pandas* and selected the attributes. Construct the generalization hierarchies and program the functions for information loss metric and exponential mechanism. After that, confirm the values of parameters (ε', β, k and penalty factor) and fill in the lattice. To do that, the utility is calculated from the values of suppression and information loss, and final output probability is calculated from the value of utility. We can calculate information loss and suppression with the values of parameters, so we can get the final output.

Chapter 4

Experimental Methodology

4.1 Dataset

In the part about experiments, two datasets are used and tested for different purposes.

- UCI Adult Dataset [8]. The census dataset was extracted in 1996 for machine learning tasks, primarily to predict whether one's income is over 50K per year. There are 32,561 rows of data in the training set and 16,281 rows in the test set. Moreover, the dataset includes fifteen different attributes about the income of individuals, which are:

  'age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country', 'income'

  and *'fnlwgt'* is a final weight to estimate the civilian noninstitutional population in US, and other attributes involves the living information of individuals. Among the attributes, six of them are numerical and the other nine are categorical. In the project, the dataset is used as a primary dataset in all the experiments in Section 4.2, 4.3 and 4.4.

- Integrated Public Use Microdata Series (IPUMS) [2]. IPUMS contains different census datasets collected from 90 countries and the chosen dataset includes 188,846 rows of data from Brazil. The dataset has been coded consistently so as to facilitate comparative researches. It contains 12 attributes but only one attribute is categorical. The dataset is used in Section 4.4 to compare the performance of ε-safe LA on different datasets with other algorithms.

4.2 Evaluation of Parameters in ε-safe LA

4.2.1 Values of ε and Penalty Factor

Among the experiments, the property of ε-safe LA is firstly considered. In ε-safe LA, there are two constants which directly influence the output probability of each node: penalty factor and ε, so it's important to find the appropriate constant pairs because a bad choice can result in a uniformly distributed probability for all the nodes, which may output a node with much information loss and large suppression. To evaluate whether a constant pair is practical, the conception of 'good nodes' is introduced.

In the experiment, 'good nodes' are the nodes whose *suppression* is less than 15% and *generalization* is less than 0.6. If the total probability of 'good nodes' is close to one, the output node should be a 'good node' with good utility since the we can get a high utility from the utility function if both the values of *suppression* and *generalization* are small.

4.2.2 Choice of k

Taking advantage of 'good nodes', the values of ε and penalty factor are decided. Here the value of ε is the ε' in Equation 3.7, and the ε for ε-safe LA algorithm can be calculated. Also, in Equation 3.8, the value of δ is dependent on the value of k. δ should be very small and it is supposed to be smaller than $1/|D|$ where $|D|$ is the number of records in the dataset D [11]. With the inequality, the minimum value of k is achieved.

4.2.3 Learning Task

After all the parameters are determined, a final dataset D_{dp} is output by ε-safe LA. With the dataset, a learning task is carried out to compare the accuracy of different regression models trained on D_{dp} and the raw dataset. The purpose of the learning task is to show the real loss of usability.

The dataset for the learning task is the Adult dataset and logistic regression is used as the classifier. Logistic regression is similar to linear regression and it is the linear weights with the logistic function (Equation 4.1).

$$f(x) = \frac{1}{1+e^{-x}} \tag{4.1}$$

Moreover, logistic regression is binary and it is usually applied to dealing with binary classification problems [17]. It models the dependent variable and predicts whether an individual belongs to class '0' or '1'. Here the predicted attribute in the Adult dataset is 'income' and the task is to predict whether the 'income' of a individual is more than 50K.

Since there are categorical and numerical attributes in the dataset, data preprocessing is required before the training. In the dataset, one-hot encoding [15] is applied to categorical attributes and standard scalar is applied to the numerical attributes.

- One-hot encoding. As shown in Figure 4.1, each category in the categorical attribute is transformed to a new attribute, so the number of categories in the categorical attribute equals the number of new attributes. For an individual in the dataset, the values of these 'new' attributes are all '0' except only one '1' for the attribute with respect to it.

	Age	country
0	28	Brazil
1	23	Mexico
2	37	Thailand
3	36	Mexico

one-hot →

	Age	Country_B	Country_M	Country_T
0	28	1	0	0
1	23	0	1	0
2	37	0	0	1
3	36	0	1	0

Figure 4.1: A simple one-hot encoding example on attribute 'country'.

- Standard scalar. We used the standard scalar in *sklearn* [25] Python package to normalize the numerical data by Equation 4.2 where z is the output value; x is the sample; μ is the mean of x and s is the standard deviation of x. It moves the mean to zero and scale the variance to unit variance.

$$z_i = \frac{x_i - \mu}{s} \tag{4.2}$$

- Missing values. All the individuals with missing values are removed from the dataset during the preprocessing.

After the preprocessing of the dataset, two different logistic regression models are trained on: (1). the raw dataset with about 30,000 rows of data (some rows of data are pruned because of missing values) (2). the DP dataset which is with sampling, suppression and generalization. The same test set is transformed according to the rules used for training sets and the accuracy of different models on the test set are compared to show the results.

4.3 Interpolation Method

4.3.1 Parameters

To explore whether the two proposed interpolation methods are practical and helpful, some experiments are carried out. As in Section 4.2, we calculate the total probability of 'good nodes' and we compare the methods with and without interpolation method. If the results are similar, the use of interpolation method is supposed to predict a utility which is close to the real one.

However, it is difficult to distinguish whether a node is a 'good node' or not when interpolation method is used. When the utility is interpolated, there is no need to calculate the suppression nor generalization of the interpolated node, and it's unpractical to distinguish a node without these values. In the experiment, the set of 'good nodes' is defined by the original lattice. When we evaluate the interpolation method, we get the total probability of the nodes which are in the 'good nodes' set. It means that we don't judge whether a node is a 'good node' for the interpolated data, and we simply use the 'good nodes' from the original set.

4.3.2 Running Time

The use of 'good nodes' aims at finding out whether the predicted utility can replace the real utility. This part will focus on the running time to explore how much time is reduced by using the interpolation method.

The number of nodes will increase exponentially when the number of attributes increases, and the number of nodes shows instant impacts on the running time. Intuitively, when the number of attributes is small, the interpolation method may not help a lot. However, with a large number of attributes, the running time may be unacceptable and the interpolation method will be important. To verify the assumption and find the possible 'bottle neck' for the number of attributes, different number of attributes are tested to show the difference of running time after applying interpolation methods.

4.3.3 Learning Task

The learning task for the interpolation method is the same as that in Section 4.2, and it is used to show the influence of using interpolation method in terms of the final accuracy. Hopefully, the dataset output from the ε-safe LA with interpolation methods

should output a 'good node' with good utility for the learning task. This task is designed to show the difference in the final output before or after using the interpolation methods. For the logistic regression models, the datasets with and without the interpolation methods serve as the training set, and the accuracy on test sets are compared.

4.4 Comparison Between Algorithms

4.4.1 Compared Algorithms

In this section, the performance of different kinds of de-identification algorithms are compared and explored in learning tasks.

- ***k*-Anonymous algorithm**. Among different *k*-anonymous algorithms, OLA is picked to show the result of *k*-anonymous algorithms. OLA is the foundation of ε-safe LA and it is both efficient and able to find the full domain optimal [31]. Also OLA shows lower information loss than other algorithms (Datafly [28], Samarati [26], etc.), which means that it is a well-designed *k*-anonymous algorithm [12].

- **Differentially private algorithm by adding noise to the regression model**. In differential privacy, the key problem is not only how to add the noise, but also where the noise is added. In a learning task, the noise can be added to the regression model to makes the algorithm satisfy differential privacy with respect to the training dataset. Here DP Functional Logistic Regression [32] is introduced as an approach and it is introduced in Section 4.4.2.

- **Differentially private algorithm by adding noise to the training dataset**. The training dataset is another place to add noise. By using sampling and node selection methods, ε-safe LA adds noise to the training dataset.

4.4.2 DP Functional Logistic Regression

DP Functional Logistic Regression is a differential private regression method that adds Laplace noise to the objective function of the regression model [32]. To achieve that, it applies the functional mechanism to logistic regression models, as shown in Algorithm 2. Here t_i is the i-th individual in dataset D; $f(t_i, \omega)$ is the loss function and $f_D(\omega)$ is the objective function which equals $\sum_{t_i \in D} f(t_i, \omega)$; ω is the weight vector for

the model; Φ_j contains all the possible products of one or more values in ω and $\lambda_{\phi t_i}$ is the polynomial coefficient for $\phi \in \Phi_i$. The DP Functional Logistic Regression aims to minimize the perturbed loss function (with Laplace noise). To deal with the infinite polynomial representation in the objective function of logistic regression, the work pruned the higher orders (> 2) in the Taylor expansion of the objective function.

Algorithm 2 Functional mechanism for logistic regression.

Input: Input dataset: D, objective function: $f_D(\omega)$, ε

Output: The parameter vector $\overline{\omega}$ which minimize $\overline{f}_D(\omega)$

1: The loss function $f(t_i, \omega) = \sum_l f_l(g_l(t_i, \omega))$
2: Construct new objective function: $\hat{f}_D(\omega) = \sum_{l=1}^{m} \sum_{i=1}^{n} \sum_{k=0}^{2} \frac{f_l^{(k)}(z_l)}{k!} (g_l(t_i, \omega) - z_l)^k$
3: Let $f_D(\omega) = \hat{f}_D(\omega)$
4: $\Delta = 2 \max_t \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} ||\lambda_{\phi t}||_1$
5: **for** every $0 \leq j \leq J$ **do**
6: **for** every $\phi \in \Phi_j$ **do**
6: $\lambda_\phi = \sum_{t_i \in D} \lambda_{\phi t_i} + Lap\left(\frac{\Delta}{\varepsilon}\right)$
7: **end for**
8: **end for**
9: $\overline{f}_D(\omega) = \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \lambda_\phi \phi(\omega)$
10: $\overline{\omega} = \arg\min_\omega \overline{f}_D(\omega)$
11: **return** $\overline{\omega}$

In the mechanism, both the sensitivity concerns and the perturbation insertion are conducted. It also shows good accuracy on the IPUMS dataset as introduced in Section 4.1. However, the mechanism only works when the objective function is with the same form of $\sum_{i=1}^{n} f(t_i, \omega)$ and it is not compatible to more complicated objective functions such as Cox regression [4]. In this project, we only adopt the logistic regression part of the mechanism as shown in Algorithm 2.

4.4.3 Comparison Methods

To compare the performance of three different de-identification algorithms, different aspects are considered:

- **Accuracy**. The accuracy of different algorithms is one of the important part to show whether the approach can remain the information inside the dataset after paying privacy concerns. To be fair to different approaches, two different

datasets are considered. The Adult dataset contains more categorical attributes while most attributes in the IPUMS dataset are numerical. The type of attributes may influence the accuracy of different approaches, so it's crucial to run the tests with different datasets.

- **Privacy**. It's also significant for a de-identification algorithm to protect individuals' privacy in the dataset. To achieve that, the value of ε in differential private approaches and k in k-anonymous approaches are set to be the same so that the comparison is fair enough.

- **Efficiency**. Another point is the efficiency of each approach. The method may not be useful if it is too time consuming. In practice, an efficient approach is compatible with datasets at any size, which makes it more practical.

Chapter 5

Experimental Evaluation

5.1 Performance of the ε-safe LA

5.1.1 Dataset

Adult dataset is used to evaluate the performance of ε-safe LA. Seven attributes: 'age' (4), 'workclass'(3), 'education'(3), 'marital-status'(3), 'relationship'(3), 'race'(2), 'income'(1) are selected as the quasi-identifiers. The attribute 'income' is the attribute that we want to predict. As 'income' is binary, we do not generalize it. The numbers in the brackets are the depth of different attributes. All rows of data in the training set (after data cleaning) are used as the input of ε-safe LA and the test set is used as the test set for the learning task. We also defined the generalization hierarchies for all attributes in the Adult dataset.

5.1.2 Choice of k and β

As shown in Section 4.2, the value of k is required to be set before running experiments to find the relation between the total probability of 'good nodes' and the (ε', penalty factor) parameter pair. In the following experiments in Section 5.1.3, the tested ε' range from 0.01 to 1, so here the maximum δ in Table 5.1 is the maximum δ for the different values of ε'.

From Table 5.1, the β is the sampling probability that an individual is selected from the raw dataset as described in Section 2.3. Also, according to Section 4.2.2, it's very dangerous if the value of δ is on the same order as $1/||D||_1$ [11]. For the dataset, it's easy to calculate that $1/||D||_1 \approx 3.32 \times 10^{-5}$, and the values of δ shown in Table 5.1 are marked by different colors: 'red' if $\delta > 1/||D||_1$ which means that the (k, β) pair is

k	β	max δ ($\times 10^{-5}$)	k	β	max δ ($\times 10^{-5}$)	k	β	max δ ($\times 10^{-5}$)
55	0.60	1.59	60	0.60	0.71	65	0.60	0.32
55	0.65	2.40	60	0.65	1.29	65	0.65	0.80
55	0.70	5.00	60	0.70	3.30	65	0.70	1.12
55	0.75	9.72	60	0.75	9.72	65	0.75	3.09
55	0.80	34.4	60	0.80	13.2	65	0.80	6.73
70	0.60	0.14	75	0.60	0.06	80	0.60	0.03
70	0.65	0.26	75	0.65	0.11	80	0.65	0.07
70	0.70	0.74	**75**	**0.70**	**0.25**	80	0.70	0.17
70	0.75	1.23	75	0.75	1.23	80	0.75	0.39
70	0.80	6.73	75	0.80	3.86	80	0.80	1.51

Table 5.1: The maximum value of δ when k and β change.

unsafe, 'yellow' if $0.1/||D||_1 < \delta < 1/||D||_1$ which means that it's dangerous because they are on the same order, 'green' if $\delta < 0.1/||D||_1$ which means that it is safe.

Intuitively, in order to keep more information in the dataset, β should be as large as possible while k should be as small as possible so that the suppression of all the nodes can be smaller. Here we have to balance both sides with the consideration of δ. Finally $k = 75$ and $\beta = 0.7$ are chosen since they show a 'green' δ while k and β are reasonable.

5.1.3 Choice of ε and Penalty Factor

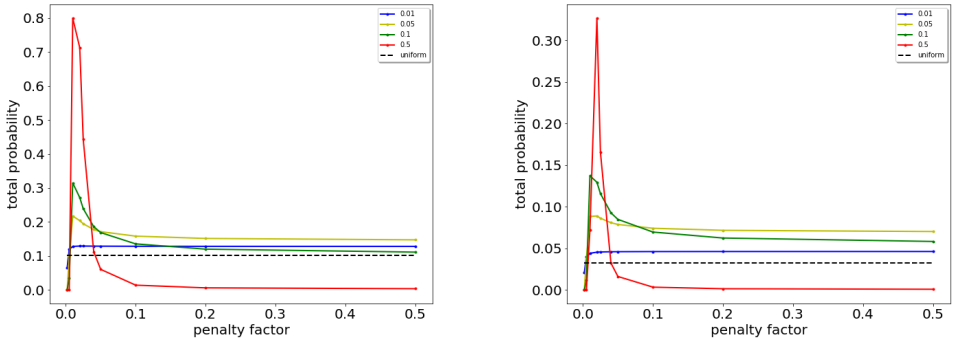
To choose the value of ε and penalty factor for ε-safe LA, the values which show a high total probability of 'good nodes' is considered to be a good pair. The ε in this section is the ε' for exponential mechanism instead of the ε for $(\beta, \varepsilon, \delta)$-DPS. When $\beta = 0.7$ and $k = 75$, 10 samples are drawn from the dataset and each sample contains on average 21,113 rows of data according to the value of β. Here different number of attributes are considered. The results of four attributes are shown in Figure 5.1(a), and the results of six attributes are shown in Figure 5.1(b).

In Figure 5.1, for each ε and penalty factor pair, the value of the total probability is the average of the 10 samples. It shows that the total probability decreased from 0.8 to 0.3 with an increasing number of attributes. The line for each value of ε starts at zero because the tiny penalty factor results that only the generalization part of a node is considered and the suppression part is neglected. When the value of penalty factor

grows larger, only the suppression of a node is considered (as shown in Equation 5.1). According to exponential mechanism in Section 2.2.3, with a high penalty factor, the output probability for a certain node is:

$$\begin{aligned}\lim_{f\to\infty} \exp\left(\frac{\varepsilon}{2\Delta u}\cdot u(X,t)\right) &= \exp\left(\frac{\varepsilon}{2\cdot f\cdot\frac{k}{|X|}}\cdot(f\cdot sup(X,n)+(1-gen(X,n)))\right)\\ &\approx \exp\left(\frac{\varepsilon}{2\cdot f\cdot\frac{k}{|X|}}\cdot f\cdot sup(X,n)\right)\\ &= \exp\left(\frac{\varepsilon\cdot sup(X,n)}{2\cdot\frac{k}{|X|}}\right)\end{aligned} \tag{5.1}$$

where f is the penalty factor and $g(X,n)$ is the generalization part of the utility function in Equation 3.10.



(a) 4 attributes (b) 6 attributes

Figure 5.1: The relation between the (ε, penalty factor) pair and the total probability of output a 'good node'. The value of penalty factor ranges from 0 to 0.5.

From Figure 5.1, the values of ε and penalty factor are decided. ε is selected as **0.5** because the total probability for $\varepsilon = 0.5$ is the highest and the final ε' for ε'-safe LA is $\varepsilon - \log(1-\beta) \approx 1.70$. With $\varepsilon = 0.5$, the value of penalty factor is **0.02** for 6 attributes and **0.01** for 4 attributes since they have shown the highest total probability.

5.1.4 Learning Task

A baseline is set in this part and it simply predicts the 'income' of all records as '$< 50K$'. With the parameters chosen in Section 5.1.3, a logistic regression model is trained for the output dataset of ε-safe LA. To achieve a higher accuracy for both sets, the dataset with 6 attributes is used as the training set. The accuracy is shown in

Table 5.2. The accuracy on the test set is about 82% when trained on the raw dataset while the accuracy is about 75% on the differential private dataset. Also, the accuracy of the DP Dataset is slightly lower than the Baseline. The suppression and generalization both result in the lower accuracy. After the generalization, some attributes might be completely generalized, which means that there is fewer real useful attributes for ε-safe LA. This can be dealt with by a better generalization hierarchy or better tuned parameters of k and β. A more precise information loss metric may help as well.

Dataset (D)	Accuracy	$\|D\|$
Raw Dataset	0.819	21,113
DP Dataset	0.752	19,518
Baseline	0.754	21,113

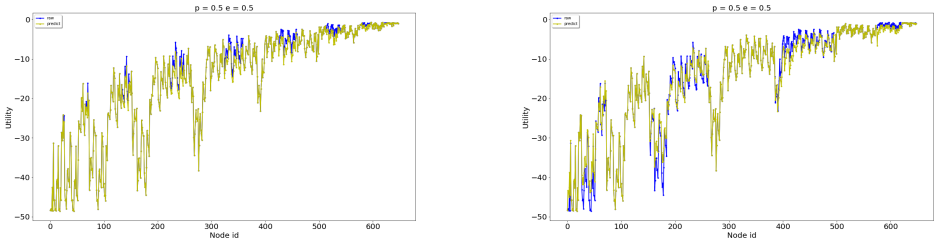
Table 5.2: The accuracy on the test set from the logistic regression models trained on the raw dataset and the differential private (DP) dataset.

5.2 Performance of Interpolation Method

5.2.1 Utility Difference

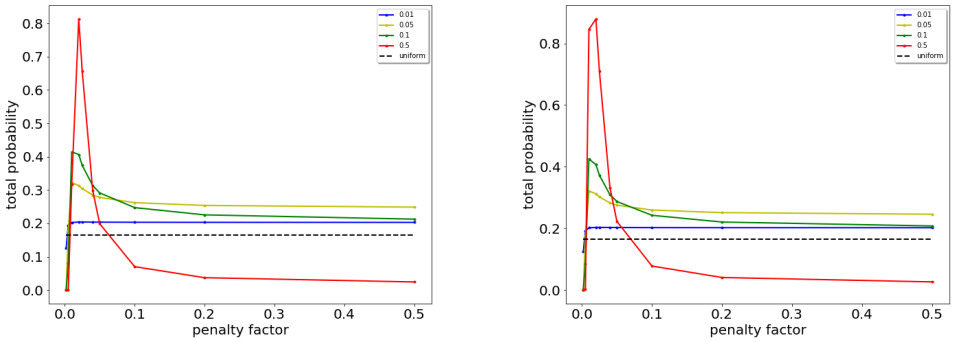
Firstly the dictionary-based interpolation method introduced in Section 3.3 is explored. Nodes with a certain depth of a certain attribute are interpolated. To compare the difference when interpolation methods are applied, Figure 5.2 is drawn to show the difference of utility when interpolation methods are applied. Also, the figures about the change of 'good nodes' are drawn in Figure 5.3 and Figure 5.4. Figure 5.4(a) shows the difference of total probability between the raw figure in 5.3(a) and interpolated attribute_0 with depth 2 in 5.3(b). They have shown significant difference for some penalty factors. Actually, the predicted figure is similar to the original one except when some 'special' depth and attributes (here attribute_0 with depth 2) are predicted. In common use, it's hard to know which attribute will negatively influence the results, which makes the method not so practical.

When the level-based interpolation method is applied, Figure 5.4(b) also shows the difference of total probability between the raw figure in 5.3(a) and skipped level interpolation method in 5.3(d), which shows a bit better results than the dictionary-based one. The level-based interpolation method is more balanced since it predicts the nodes based on their total levels while the dictionary-based method is based on

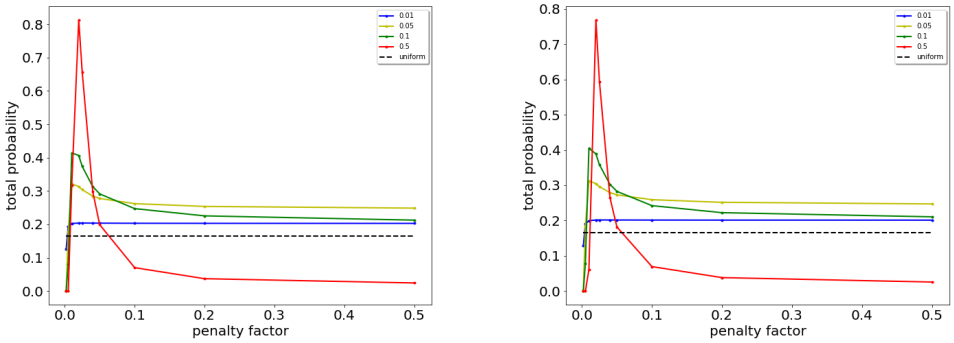


(a) Dictionary-based interpolation (b) Level-based interpolation

Figure 5.2: The difference of utility when dictionary-based interpolation method is applied.

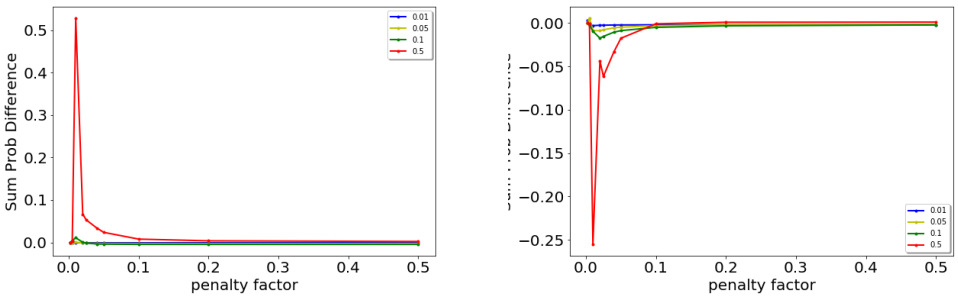


(a) Without interpolation (b) (Dictionary-based) Interpolate attribute_0 with depth 2



(c) (Dictionary-based) Interpolate attribute_3 with depth 1 (d) (Level-based) Skipped level interpolation

Figure 5.3: Figures of 'good nodes' for two interpolation methods with different interpolation depths and attributes.



(a) Difference of total probability (dictionary) (b) Difference of total probability (level)

Figure 5.4: The difference of total probability of 'good nodes' in Figure 5.4 and the difference of utility for each node. (a) is for the dictionary-based method and (b) is for the level-based method.

the depth of attributes. From the results, level-based interpolation method has shown less difference in terms of 'good nodes' figures and it is independent of the choice of attributes. The dictionary-based method is possible to be out of balance, which may lead to a great difference when the utility of the nodes with some attributes is interpolated. In the later part of this section, the level-based interpolation method is further studied. Also note that the interpolation methods are not very developed, which is still with large difference with some penalty factors. We will keep explore a better interpolation method in the future work.

5.2.2 Running Time

In this part, we want to explore how much time the interpolation method have reduced. Based on the work in Section 5.2.1, only the level-based interpolation method is considered and compared with the algorithm without prediction.

Time for / Method	4 attributes	6 attributes	8 attributes	10 attributes
Without interpolation	5.404s	19.279s	258.358s	2473.51s
Level-Based interpolation	4.217s	11.480s	141.733s	1336.08s
Number of nodes	108	648	7776	62208

Table 5.3: The running time of ε-safe LA with/without the interpolation method on the Adult dataset with different number of attributes.

From Table 5.3, It shows that when the number of attributes increases, the number of nodes increases exponentially, which makes the running time much longer. With a small number of attributes (four or six), the running time is less than half a minute and the influence of interpolation method is invisible. However, when the number becomes large, the interpolation method can save nearly half the time, which can greatly improve the efficiency of ε-safe LA for a large dataset with a large number of attributes or deep generalization hierarchies.

5.2.3 Learning Task

Here we have used the output dataset from the ε-safe LA with and without interpolation methods. Since the output node of ε-safe LA changes every time, we output 10 times from each algorithm and use the average as the accuracy. For a higher accuracy, we pick ten attributes from the Adult dataset as the input of each algorithm, and the test set is the test set in Adult dataset. In Table 5.4, both methods show very similar accuracy on the test set, which means that the interpolation method brings very small change to the final accuracy.

Algorithm	Accuracy
ε-safe LA without interpolation	0.7592
ε-safe LA with interpolation method	0.7576

Table 5.4: The accuracy of different algorithms on different datasets.

5.3 Results for Comparison Methods

In this section, the performance of different de-identification algorithms are compared. The results are tested on both the Adult dataset and the IPUMS dataset. In Adult dataset, **10** attributes are picked and the values of β and k are still 0.7 and 75. In IPUMS dataset, all the **12** attributes are contained. For the categorical attribute 'Martial Status', all the categories are transformed into *single* and *married*, so there are 13 attributes altogether. For IPUMS dataset, the value of β and k are chosen as 0.7 and 85 according to Table 5.5. In IPUMS dataset, there are 141,634 rows of data after splitting 25% as the test set, which means that $1/||D||_1 \approx 7.06 \times 10^{-6}$ and the values of δ in Table 5.5 are marked in colors the same as Table 5.1.

k	β	max δ $(\times 10^{-6})$	k	β	max δ $(\times 10^{-6})$	k	β	max δ $(\times 10^{-6})$
75	0.60	0.64	80	0.60	0.29	85	0.60	0.13
75	0.65	1.05	80	0.65	0.65	85	0.65	0.30
75	0.70	2.53	80	0.70	1.68	**85**	**0.70**	0.58
75	0.75	12.3	80	0.75	3.93	85	0.75	1.59
75	0.80	38.6	80	0.80	15.1	85	0.80	5.86

Table 5.5: The maximum value of δ when k and β change.

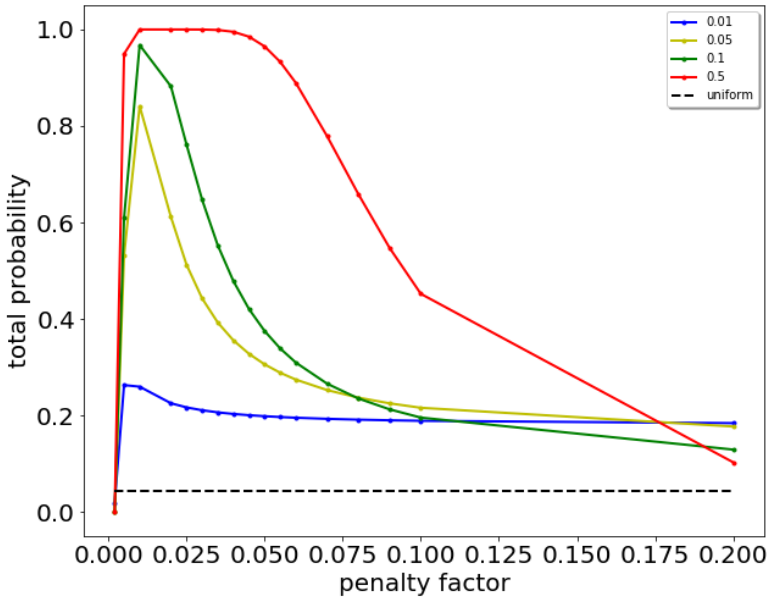


Figure 5.5: The relation between the total probability and the (ε, penalty factor) pair for the IPUMS dataset.

For ε-safe LA in IPUMS dataset, the value of ε′ for the exponential mechanism and the value of penalty factor are changed. In Figure 5.5, the chosen value of ε′ is 0.5 because it shows the highest total probability and penalty factor is picked as 0.03 because it is the midpoint of the penalty factor values which show a total probability very close to one.

For OLA, the maximum suppression is 15% for either dataset and *prec* is used as the information loss metric. Also, the value of k for OLA is the same as k for ε-safe LA, which is 75 for the Adult dataset and 85 for the IPUMS dataset.

The results of learning tasks are shown in Table 5.6. A baseline is also set that always predict False for all records in both datasets. In the experiments, the values of

ε are same in the two differential private algorithms. With the same ε, DP functional logistic regression shows a higher accuracy on IPUMS dataset while its accuracy on Adult dataset is much lower. In DP functional logistic regression, the main process is to add Laplace noise into the objective function. However, when there is a lot of categorical attributes (such as the Adult dataset), there are 100 attributes in the dataset after the one-hot encoding, which will result in a large number of weights. Much more noises are inserted when there are many categorical attributes, which can lead to a low accuracy on the Adult dataset.

For OLA and ε-safe LA, the accuracy are similar on both datasets. ε-safe LA shows a lower accuracy since the output dataset is with suppression and sampling, which means that the training set for ε-safe LA is with a smaller number of records than OLA. The difference of accuracy between these two algorithms is very small but ε-safe LA is ε-safe differential private. Their accuracy on the Adult dataset is slightly higher than the baseline and it remains to be improved by a better hierarchy or better tuned parameters. On the IPUMS dataset, their accuracy is more than 5% higher than the baseline and 1-2% lower than the raw dataset, which is a good result.

Algorithm	Acc on Adult	ε_{Adult}	Acc on IPUMS	ε_{IPUMS}
DP Functional LR	0.530	1.704	0.832	1.704
ε-safe LA	0.759	1.704	0.816	1.704
OLA	0.770	Not DP	0.826	Not DP
Raw dataset	0.830	Not DP	0.836	Not DP
Baseline	0.754	Not DP	0.751	Not DP

Table 5.6: The accuracy of different algorithms on different datasets. 'Not DP' means not differential private.

Algorithm	n_{attr} (after *one-hot*)	t_{Adult}	n_{attr}	t_{IPUMS}
DP Functional LR	100	0.082s	13	0.029s
ε-safe LA	19	2491.76s	13	10493.53s
OLA	63	3491.57s	13	13735.49s
Raw dataset	100	0.24s	13	1.48s

Table 5.7: The number of attributes after *one-hot* encoding for each dataset and the running time of different algorithms on Adult and IPUMS. Here the running time **doesn't** include the time of loading or pre-processing the raw dataset.

The number of attributes after the one-hot encoding and the running time of different algorithms is included in Table 5.7. It's shown that the running time of ε-safe LA and OLA is much longer than DP Functional LR or the raw dataset. To calculate the utility of ε-safe LA, the suppression of each node should be calculated to guarantee that the node is k-anonymous after the suppression, which consumes a lot of time. Meanwhile, with a large number of attributes, OLA also has to reach a deep level to find the k-anonymous nodes. In other words, there are less k-anonymous nodes when the level is shallow, which requires OLA to judge most nodes to find whether it is k-anonymous or not. However, the running time of OLA is longer than ε-safe LA. The reason is that OLA is with more records than ε-safe LA, which consumes more time than the time saved by the smaller number of nodes which need calculation in OLA. To verify the idea, ε-safe LA is also tested on the dataset without sampling only to show the difference of running time. And we find that the running time on the Adult dataset without sampling is 4640.50 seconds, which is much longer than OLA.

Chapter 6

Conclusions

6.1 Discussion of Experiments

In the project, we designed and carried out three different experiments to explore the properties and performance of ε-safe LA. Overall, ε-safe LA presents a new way of protecting privacy by combining OLA with differential privacy. From the experiments, ε-safe LA is practical and shows advantages in some aspects.

In Section 5.1, it shows how the values of the parameters are picked. Firstly, it's hard to choose the values of k and β because of the strict restriction for the value of δ as shown in Section 5.1.3. A higher value of k and a lower value of β can help reduce the value of δ, but it will result in a larger suppression and a smaller number of records, which will directly influence the usability of the output dataset. The selection is carefully considered and the constants are supposed to reserve both the privacy and the use of the dataset.

After choosing the values of k and β, we consider ε' and penalty factor. In reminder, ε' here is the ε' for the exponential mechanism. Figure 5.1 is drawn to show the relation between the total probability of 'good nodes' and the parameters. To better protect the privacy, the value of ε for ε-safe LA should be small, and the value of ε is calculated from ε' and β. Since the value of β is already confirmed, the value of ε' should be small for a small ε and it should also show a high total probability of using a 'good node' to probably output a useful dataset. However, it is hard to achieve a high total probability with a larger number of attributes. From Figure 5.1, the maximum probability is only about 0.3 for 7 attributes (including the sensitive attribute 'income'). Intuitively, the large value of suppression can reduce the number of 'good nodes', and the value of suppression is determined by the values of k, β and the number of attributes. Recall

that the values of k and β are restricted by the assumption that the value of δ should be smaller than $1/||D||_1$.

Section 5.2 shows the performance of interpolation methods. After finishing the experiments in Section 5.1, a problem occurs that the running time of the algorithm will be extremely long with a larger number of attributes. For the interpolation, we considered two different methods. One is to predict the nodes with the same attribute size, and another is to predict the nodes with the same level. However, the dictionary-based method is not stable when the predicted attribute is changed. In practice, we can't hope that we can luckily choose the 'right' predicted attributes. Alternatively, the level-based method shows a better performance. With the skip-level interpolation, it can output a similar chart of probabilities of 'good nodes' as the original one. More importantly, the interpolation method can help reduce the running time of ε-safe LA. In the programming, a 'groupby' function is used to get the number (k) of each different record, which will cost approximately 0.07 seconds for each node. Intuitively, 0.07 is not a long time, but if there are 60,000 nodes with 10 attributes, the running time will be more than an hour. The interpolation can nearly reduce half the running time by interpolating half the levels. What's more, for the learning task, the final accuracy is very close for the two different output dataset with/without interpolation, which also means that the interpolation method is practical. Due to the time limit, the interpolation method is not tested on other datasets or a larger number of attributes. For the Adult dataset with both the training set and test (about 45,000 rows of data), it will take one and a half hours to run ε-safe LA without interpolation. This part remains to be done in the future work and we will discuss it in Section 6.2.

In Section 5.3, the performance of ε-safe LA is compared to other de-identification algorithms. OLA and ε-safe LA show very similar accuracy on both datasets. OLA is with a higher accuracy since it's without sampling and ε-safe LA shows stronger guarantee for privacy protection. For differential private algorithms, there are three different methods of adding noises: to the raw dataset, to the regression model and to the result. ε-safe LA is the method of adding noise to the raw dataset while DP functional logistic regression adds noise to the regression model. With the same value of ε, we show that ε-safe LA shows a better accuracy on the categorical dataset (Adult), but on the numeric dataset (IPUMS), the accuracy of DP functional logistic regression is close to the accuracy of the original dataset and is 1.6% higher than ε-safe LA. DP functional logistic regression shows better performance only on numeric datasets while ε-safe LA works for both datasets. Another table shows that the running time of OLA

and ε-safe LA is much longer than DP functional logistic regression, which means that the interpolation method is with great importance.

In addition, when the noise is added to the raw dataset, the output dataset from the algorithm (ε-safe LA) can be published. For DP functional logistic regression, publishing the raw dataset is not 'safe'. Only the regression model is differential private. Publishing the dataset will definitely be more helpful for research studies than only publishing the regression model.

6.2 Future Work

With the three experiments, we revealed some properties of ε-safe LA, but there still remains a lot of work to do:

- Interpolation methods. We will better evaluate the performance of interpolation methods. Ideally the interpolation method in which only some nodes are evaluated is compared with the original method with a lattice that has the same number of evaluated nodes. To achieve that, two different generalization hierarchies are required such that the running time of the interpolation method and the original method should be similar. Also, the interpolation method is not developed, so we hope to find a better interpolation method or equation in the future to show a better results. What's more, the methods are only tested on the Adult dataset, so it's not clear about the performance of the interpolation method on other datasets. In the future, the experiments will be carried out on more datasets with a lrger number of attributes.

- We would like to have a larger number of differential private algorithms for comparison. Here we only compared the performance of ε-safe LA with DP functional logistic regression, which is a method of achieving differential privacy by adding noise to the regression model. In the future, ε-safe LA is supposed to be compared with other differential private algorithms which adds noise to the output or the raw input. By doing that, the performance of ε-safe LA can be analyzed more thoroughly.

- We would like to test that our implementation of ε-safe LA is ε-differential private. In this project, ε-safe LA is proved to be ε-differential private in theory. However, there are methods that provide an experimental approach to test

whether an algorithm is ε-differential private, such as the work of [6] and [1]. Such works cannot directly be used for ε-safe LA because they mainly aim at the Laplace mechanism which adds noises to the data in the dataset. In ε-safe LA, we add the noise by using sampling instead of directly adding noise to the dataset. It requires some changes to the experimental approaches to test ε-differential privacy.

- A better utility function and other information loss metric. In ε-safe LA, the utility function plays an important role since it leads to the final output probability of each node. Here we used a utility function whose sensitivity is easy to calculate. It will definitely be a problem to calculate the sensitivity when the utility function or the information loss metric is changed, but such changes may bring a good influence to the performance of ε-safe LA. The calculation of sensitivity is the most tricky part in ε-safe LA, and switching a better utility function or a new information loss metric might require considerable efforts.

- A better generalization hierarchy. In this project, the main purpose is to evaluate ε-safe LA and prove that it works in a differential private way, so we did not optimize the generalization hierarchy. A better generalization hierarchy will help improve the performance of ε-safe LA, but this was not the key point of this project and is future work.

- Use both ε-safe LA and DP functional logistic regression. It is an interesting open question what would be the effect of applying DP functional logistic regression to the output dataset of ε-safe LA. The key question is what is the value of ε after running two different differential private algorithms in a row?

Bibliography

[1] Benjamin Bichsel, Timon Gehr, Dana Drachsler-Cohen, Petar Tsankov, and Martin Vechev. Dp-finder: Finding differential privacy violations by sampling and optimization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 508–524. ACM, 2018.

[2] Minnesota Population Center. Integrated public use microdata series, international: Version 5.0, 2009.

[3] Valentina Ciriani, S De Capitani Di Vimercati, Sara Foresti, and Pierangela Samarati. κ-anonymity. In *Secure data management in decentralized systems*, pages 323–353. Springer, 2007.

[4] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[5] Ton De Waal and Leon Willenborg. Information loss through global recoding and local suppression. pages 17–20, January 1999.

[6] Zeyu Ding, Yuxin Wang, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. Detecting violations of differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 475–489. ACM, 2018.

[7] Josep Domingo-Ferrer and Jordi Soria-Comas. From t-closeness to differential privacy and vice versa in data anonymization. *Knowledge-Based Systems*, 74:151–158, 2015.

[8] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[9] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[11] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[12] Khaled El Emam, Fida Kamal Dankar, Romeo Issa, Elizabeth Jonker, Daniel Amyot, Elise Cogo, Jean-Pierre Corriveau, Mark Walker, Sadrul Chowdhury, Regis Vaillancourt, Tyson Roffey, and Jim Bottomley. A Globally Optimal k-Anonymity Method for the De-Identification of Health Data. *Journal of the American Medical Informatics Association*, 16(5):670–682, September 2009.

[13] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, September 1977.

[14] Keith B. Frikken and Yihua Zhang. Yet another privacy metric for publishing micro-data. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society - WPES '08*, page 117, Alexandria, Virginia, USA, 2008. ACM Press.

[15] Sarah Harris and David Harris. *Digital design and computer architecture: arm edition*. Morgan Kaufmann, 2015.

[16] Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pl Mac Aonghusa. (k,ε)-Anonymity: k-Anonymity with ε-Differential Privacy. *arXiv:1710.01615 [cs, math]*, October 2017. arXiv: 1710.01615.

[17] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[18] Kristen LeFevre, David J DeWitt, Raghu Ramakrishnan, et al. Mondrian multidimensional k-anonymity. In *ICDE*, volume 6, page 25, 2006.

[19] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[20] Ninghui Li, Wahbeh Qardaji, and Dong Su. On Sampling, Anonymization, and Differential Privacy: Or, k-Anonymization Meets Differential Privacy. *arXiv:1101.2604 [cs]*, January 2011. arXiv: 1101.2604.

[21] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24. IEEE, 2006.

[22] Leonardo Mazzone. Database de-identification for electronic health records. Bachelor thesis, The University of Edinburgh, April 2019.

[23] Wes Mckinney. pandas: a foundational python library for data analysis and statistics. *Python High Performance Science Computer*, 01 2011.

[24] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, volume 7, pages 94–103, 2007.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[27] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, technical report, SRI International, 1998.

[28] Latanya Sweeney. Datafly: A system for providing anonymity in medical data. In *Database Security XI*, pages 356–381. Springer, 1998.

[29] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, 2002.

[30] Vicenç Torra. Microaggregation for categorical variables: a median based approach. In *International Workshop on Privacy in Statistical Databases*, pages 162–174. Springer, 2004.

[31] Jing Zhang, Xiujun Gong, Zhipeng Han, and Siling Feng. An improved algorithm for k-anonymity. In *International Conference on E-business Technology and Strategy*, pages 352–360. Springer, 2012.

[32] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.