UCLouvain-EPL

Algorithms in Data Science

Project 3: Privacy

Teachers:
Vincent Blondel
Jean-Charles Delvenne
Krings Gautier
Course assistant:
Alexey Medvedev

TEAM Members :
Nima Farnoodian - 68372000 - nima.farnoodian@student.uclouvain.be
Charles Rongione - 51841500 - charles.rongione@student.uclouvain.be
Breno Tiburcio - 74042000 - breno.tiburcio@student.uclouvain.be

Abstract

Our challenge is to find the best balance between data utility and anonymity. But first, we must recall What is sensitive information? Most of us would be comfortable with our medical information disclosed. But in many cases, what is sensitive to one might no be sensitive for others. Sensitive information has a cultural component. In some countries, many people would not like to have their taxes information available in extension to how much money they make. On the other hand, in Sweden, this information is public. Similarly, the data utility requires a fine-tune alignment with the anonymous data set purposes. What information are relevant for such works as building a hospital or study the stress impact on people lives. More than only statistical knowledge, we discovered that Anonymity is craft-work which demands a holistic understanding. Good work should take to account cultural components, statistics as well as technical aspects of the purposes to preserve data utility.

1 Strategy

1.1 k- Anonymity against Probability Attacks

The goal of K-anonymization is to solve the following problem: "Given person-specific field-structured data, produce a release of the data with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remain practically useful." (Sweeney et al. 2002)

1.2 l- Diversity against Homogeneity Attacks

K-anonymization is not sufficient because our Dataset is still vulnerable to homogeneity attacks. To avoid them, we need to be sure that every class of individuals contains at least two different sensitive attributes. If not, the attacker could guess what is the attribute of is target by knowing some quasi-identifiers about him.

1.3 Our Approach

In the beginning, we wanted to provide only one dataset for both questions because if an attacker could get the two different sources, so he could cross them and perform a matching attack and retrieve sensitive in formations. However, we figured out that it was impossible to both get sufficient anonymization and sufficient privacy without suppressing a considerable number of entries. We opted then to go for two different datasets cautiously selecting the columns we deem essential for each finality.

Moreover, we have chosen different quasi-identifiers for each study. Follow below a brief is a summary of our reasoning about each information each dataset should contain.

- Zip-code: Important in the Hospital problem because it is a geographical problem but not in the stress problem because few digits anonymity led us to large areas to infer on stress.
- Marital status: It's quite essential for stress because a relationship can be interpreted as a source of stress. But not so relevant on building hospitals.
- Commute time: pointless for hospitals but a long commute time can be a source of stress.
- Accommodation: pointless for hospitals but can probably be a source of stress.
- Gender: not useful for hospitals since the hospital treats individuals regardless of that. However, stress can be different for different genders.
- DOB: despite exist medical specialities are varying according to the age; we thought that this information would be more useful for the stress study since people of different ages reacts more evenly in regards the stress. Leave this information in both data-sets would increase the risk of a matching attack. And the hospital unity speciality information can be obtained by the column of diseases.
- Education: following the same logic, we do not differentiate individuals for the hospital. But more likely, the education level may correlate to the type of work of the person that might be a source of stress.

- Children: not useful for hospital. On the other hand, it implies more responsibility and consequently; children can become a source of stress.

Please, find below the table containing the initial data and the feature classification we have set for each proposed case:

Dataset/Finality	Stress Study	Hospital
Id	INSENSITIVE	INSENSITIVE
Gender	QUASIIDENTIFYING	-
DOB	QUASIIDENTIFYING	-
Zipcode	QUASIIDENTIFYING	QUASIIDENTIFYING
Education	QUASIIDENTIFYING	-
Employment	QUASIIDENTIFYING	-
Children	QUASIIDENTIFYING	-
Marital Status	QUASIIDENTIFYING	QUASIIDENTIFYING
Ancestry	-	-
Number Vehicles	-	-
Commute Time	INSENSITIVE	-
Accommodation	QUASIIDENTIFYING	QUASIIDENTIFYING
Disease	SENSITIVE	SENSITIVE

2 Implementation

2.1 Encryption function: Automata

Concealing the identity of patients or people in a clinical dataset that is meant to be publicly published is an inevitable and primary task for preserving the privacy of the corresponding people. Therefore, we decided to encrypt the ids (names) in the given dataset in order to prevent attacks on the identity of the patients. To this end, we use 1D-Cellular Automata-based encryption but with a modification to minimize the likelihood of retrieving the real ids in a dataset. We call it "CA batch Encryption" because it encrypts the whole ids in batch instead of encrypting each id. In what follows, we first explain the 1D-Cellular Automata (1D-CA) and then address the problem of encrypting ids in batch using 1D-CA in the dataset.

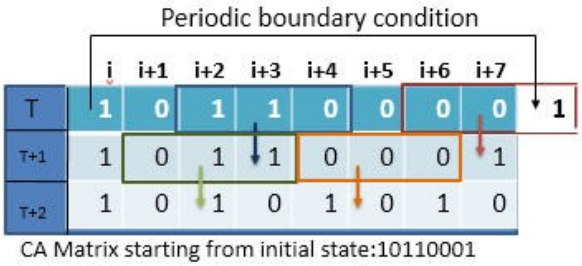
2.1.1 One-Dimensional Cellular Automata

John von Neumann first introduced cellular automata in the 1950s, and they were proposed as a model for investigating complex behaviors by Stanislaw Ulam, who, after World War II, used the very first computers to simulate examples of cellular automata. A cellular automaton (CA) consists of a regular lattice of cells, possibly of infinite size in theory but finite in practical simulation. This regular lattice can be of any dimension. Each cell can take on one of a finite number of values. The values of the cells are updated synchronously, in discrete time steps, according to a local rule, which is identical for all cells. This update rule considers the value of the cell itself and the values of neighboring cells within a certain radius.

A CA has three main properties, dimension d, states per cell k, and radius r. The dimension specifies the arrangement of cells; a one-dimensional line, two-dimensional plane, etc. The state per cell is the number of different values one cell can have. The radius defines the number of cells in each direction that will affect the update of a cell. For one-dimensional CA, a radius of r results in a neighborhood of size m = 2r + 1. For CA of higher dimension, it must be specified whether the radius refers only to directly adjacent cells or includes diagonally adjacent cells. For example, a two-dimensional CA of radius 1 will result in a neighborhood of either size 5 or 9, which are called von Neumann and Moore neighborhood, respectively. This project will deal with one-dimensional CA.

Each cell has index i, the state of a cell at time t is given by S_i^t. The state of cell i along with the state of each cell in the neighborhood of i is defined as n_i^t.

The elementary one-dimensional CA are those with k = 2, and r = 1. This yields a rule table of size 8 and 256 possible different rule tables. Rule tables for elementary CA are of the form $(t_7t_6t_5t_4t_3t_2t_1t_0)$, where the neighborhood (111) corresponds to t_7, (110) to t_7, ..., and (000) to t_0. The values t_7 through t_0 can be a binary number, which provides each elementary CA with a unique identifier, in the decimal range 0 to 255. For example, suppose the following configuration generated by one-dimensional CA.



CA Matrix starting from initial state:10110001

No.	7	6	5	4	3	2	1	0
Neighbors	111	110	101	100	011	010	001	000
Binary	0	1	0	1	1	1	1	0

Rule table for 90 (01011110)

Figure 1: Generating CA configuration with rule 90.

As seen in Figure 1, each cell is updated by their neighborhoods and the rule specified. To be more precise, consider cells bordered with red where $S_(i+7)^t = 0$ and $n_(i+7)^t$ shows the binary value of 001 corresponding to t_1. Therefore, the cell i+7 takes on the state 1 at time t+1 according to t_1, so it can be observed that $S_(i+7)^(t+1)$ is 1 and $n_(i+7)^(t+1)$ is 011. Notice that the boundary condition is here periodic.

2.1.2 Cellular Automata Encryption

The common approach of encrypting an input string with n characters using 1D-CA is to generate a 1D-CA structure that grows until time n with respect to an initial state with eight length. To this end, we should compute the binary string of each character. Notice that n is proportional to the length of the input string. After computing the binary strings, we obtain an array of binary code with n rows and 8 columns, which exactly matches the CA structure in terms of dimension. At the end, we perform an XOR operation on these two binary arrays to gain the encrypted text. Indeed, the encrypted text is the result of the XOR operation whose rows represent the binary code of encrypted characters. The cool fact about this technique is that, only one function is needed for encryption and decryption due to the associative property of XOR.

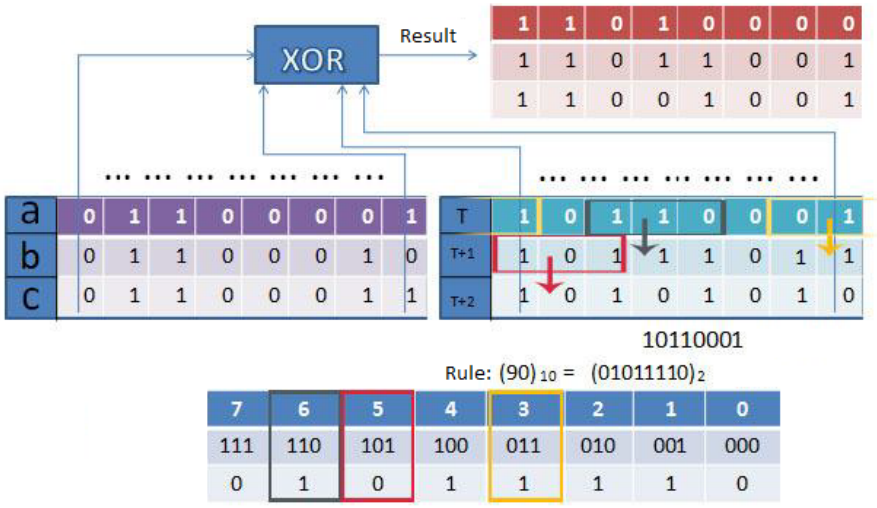


Figure 2: An example of 1D-CA cryptography.

To encrypt an input text using CA, an initial state should be fed into the structure. One way to do it is to use Logistic Function that is a polynomial mapping or recurrent relation. The formula for this recurrent function is as follows:

$$x_{n+1} = r x_n (1 - x_n) \tag{1}$$

In the formula (1), if r lies between 3.57 and 4, then the function follows a chaotic pattern over n times such that a tiny change in the initial number may have a butter fly effect in the long term. Therefore, we can obtain an initial state for the CA using this formula. The approach is as follow. First, we choose an initial number for logistic function and run it over n times (we call n as the privacy). Then, we multiply the final number by a large number, let's say, 10^4. Next, we can arbitrarily choose 8 bits of the resulting number and feed it into the structure of CA. In this case, after performing XOR, the initial value of the logistic function, the privacy, and the rule applied to the CA structure are required to retrieve the original text.

Although the CA technique may prevent attacks on the texts, it can be vulnerable if it is applied to preserve the privacy of the patients. The problem is that if an adversary is aware of the technique and know that a specific person is in the dataset, s/he can get the real ids back only by running some loops. Indeed, performing a XOR operation on the person name and the encrypted ids can help reveal the structure of CA that is all we are cautious about. Therefore, we extended and improved the original algorithm to prevent such an attack. As we have seen, the CA structure grows vertically and each row of it belongs to a character. So, now, what if we grow the CA structure vertically and in depth. Suppose, all the ids have the same length of x and there are y rows in the dataset. Then, for the first id, we create a CA structure of size (x,8). Now, take each row of the obtained CA structure as an initial state for other CA structures that grow y times. In this case, we will generate y CA structures each of which has x rows. Indeed, it is a three-dimensional CA structure with size of x,8,y. In this case, by performing an XOR operation on each id in position z and CA structure(:,8,z), we can encrypt all ids in the batch. The advantage of this approach is that the encryption relies on a lager structure instead of only a simple sentence. Therefore, it is going to be harder to find the CA structure by running for loops because each id has different structure.

We applied this approach on the un-anonymous database and realized that after encryption the number of repeated ids that are repeated 17 times become 3. In the un-anonymous database this number is 27, and the names are repeated 55 times. So, technically, there may be collisions after encryption, but honestly, it can get ids back simply and without loss. As a conclusion, this encryption function may give us an opportunity to cheat the adversary by misinformation although we are aware that almost all cryptography techniques are not safe enough to protect the privacy. We fully understand that our approach is not that perfect, yet powerful!

2.2 Anonymizations using Pyarxaas API

We opted to use a python library that encapsulates the connection to ARXaaS. It enabled us to apply and analyze different scenarios in terms of k-anonymity and l-diversity by providing us with various attacks risk calculations. Apart from that, at each step, we calculated the entropy to evaluate how far or short we were from the original data set.

Before using the Pyarxaas API, we had to create categorical hierarchy files for each Quasi-identifying column we have considered along with each trial. For further references see https://pyarxaas.readthedocs.io/en/latest/.

2.2.1 Hierarchy Categories for Anonymity

The API requires us to provide him with an anonymization hierarchy file. For example a zip code could be anonymized to different degrees : 11111 1111* 111** 11*** 1**** *****.

	0	1	2	3
0	2	[0-2[	[0-4[	Yes
1	1	[0-2[	[0-4[	Yes
2	4	[3-5[	[0-4[	Yes
3	0	[0-2[	[5-8[	No
4	3	[3-5[	[5-8[	Yes
5	5	[3-5[	[5-8[	Yes
6	6	[6-8[	[5-8[	Yes
7	7	[6-8[	[5-8[	Yes
8	8	[6-8[	[5-8[	Yes

Table 1: Example of Hierarchy : number of children

You will find the other Hierarchy files in the Data Folder we provided.

2.3 Manual Anonymizations

2.3.1 Diseases Categories

Another critical aspect of anonymity is external information. That can be a data-set that we are unaware of or general knowledge about a give features of our table. For instance, the types of disease that also reveal the gender of the person, such as Prostate or Breast cancer. As a consequence, we came up with a diseases category that would prevent an ordinary sense and still a couple of medical information.

Diseases Type	Categorization Hierarchy One	Categorization Hierarchy Two	Non-Hierarchiecal Category
multiple sclerosis	Immune Deficiency Disease	Brain Disease	Nervous system disease
Alzheimer's disease	Progressive Brain Disorder	Brain Disease	Degenerative disease
Breast Cancer	Cancer	Oncological Disease	Breast diseases
diabetes	Metabolic Disorder	Polygenic disorder	Endocrinological Disease
schizophrenia	Mental Disorder	Brain Disease	Nervous system disease
kidney disease	Renal Disorder	Urological Disease	Unirary system diseases
gastritis	Stomach Diseased	Infectious Disease	Digestive system Disease
HIV/AIDS	Immune Deficiency Disease	Infectious Disease	Immune system diseases
heart disease	Heart condition	Cardiovascular Disease	Circulatory system diseases
hypertension	Blood pressure disorder	Cardiovascular Disease	Circulatory system diseases
endometriosis	Gynecological Disorder	Ovarian diseases	Reproductive system diseases
prostate cancer	Cancer	Oncological Disease	Unirary system diseases
skin cancer	Cancer	Oncological Disease	Skin deseases

Table 2: Diseases Categories - Referenced by bmcbioinformatic:
https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03800-2

Since we reached the levels of risk and loss of information that we understood as sufficient, we deemed not to apply the diseases. We now that wrong generalization have serious entanglements. As we state at the beginning, sensitive information also has a cultural component as can create two levels of hierarchical anonymity and one third that has a more general and less invasive definition for each disease. Although the third column may not help us in the generalization, it softens the burden of carrying an illness that may lead to discrimination.

2.3.2 Suppressing vulnerable entries

One of the challenges was to maintain the information that we judge necessary and, at the same time, set a K-Anonymity and L-Diversity levels equal to two or three. Zip-code is one for the top, various features in our data set and also elementary information for building hospitals (as per deemed). To achieve the anonymity of two among all entries and keep the information of two digits (xx***) at zip-code, we had to allow the algorithm to suppress at most 10 percent of the dataset. The final data set designated to the hospital building studies have a reduced number of records of 1865 entries. Moreover, to achieve 3-diversity and 3-anonymity for the stress-related dataset, we also allowed the algorithm to suppress at most 10 percent of the dataset. The resulting dataset has 1858 records and benefits from 3-diversity and 3-anonymity.

3 Result and Comments

3.1 Entropy equation

The easiest way to have a measurement of the loss of information is to compute the variation of entropy between the original dataset and the anonymizated

$$H(D) = -\sum_{i=1}^{k} \frac{\#C_i}{N} \log \frac{\#C_i}{N} \tag{2}$$

This was a useful tool too compare the anonymity with the loss of information. But there something important to pay attention at : Some columns holds more entropy than the other. For example zip code or dob contain lot of entropy so the loss will be big if we anonymous these columns. However this is not necessarily a problem because they are not necessarily the most useful information. It depends on goal of the dataset.

3.2 Risks results

Our library provides us the vulnerability of our dataset against 3 types of risk :

- Prosecutor risk : In this case, an attacker wants to identify a specific person in a anonymized database. The identification risk is measured by finding the unique combinations of quasiidentifiers in the anonymized dataset. Thanks to K-anonymization we cannot predict which equivalence class an attacker will attempt to match. We must assume the worst-case scenario : the person they want to identify has the smallest k in the database. When de-identifying a dataset, a value of 5 for k (i.e., there are at least five records in any equivalence class) is often considered sufficient privacy protection. (this number is from : De-Identification Whitepaper 5 Privacy Analytics Inc)

- Journalist risk : Journalist risk is also concerned with the identification of an individual. However, in this case the journalist does not care which individual is identified. The probabilistic risk profile is different from that of prosecutor risk. The anonymized data is a subset of a larger public database. The journalist doesn't know a particular individual in the anonymized dataset but does know that all the people in the dataset exist in a larger public database.

- Marketer risk : In this case an intruder wants to identify as many individuals as possible in a database. Because of that the risk is calculated to everyone in the data set. The marketer risk is measured by calculating the probability of matching a record in an equivalence class of the anonimyzed set with those in the matching equivalence class in the marketer's database.

Too see if our anonymity level was sufficient, we looked at the average prosecutor risk. We aimed to get its maximum below 0.4 and the fraction of records affected by it below 0.05. And we managed to get an average risk of 0.07 which is, in our opinion, not so bad with an only 1.29 loss of entropy for the stress problem.

For the hospital data-set we got a highest prosecutor of 0.33, with only 0.04 of the data set affected by it. The average was only of 0.15 with is quite good because we lost only 0.84 entropy.

Parameter	Stress problem	Hospital problem
Entropy loss	1.29	0.84
Highest Prosecutor risk	0.33	0.33
Average prosecutor risk	0.07	0.15
People affected by highest Prosecutor risk	0.025	0.04

Table 3: Summary result of Anonymization

We achieved these results using a K-anonymity and l-diversity of 3 for both datasets.

3.3 Final Considerations

Anonymization is a very complex problem. It requires both pieces of knowledge in privacy and the theme of the data and their goal. We have an only good general sense. We can apply the algorithm to reach K-anonymity, l-diversity. We can even manage to manipulate them to get a specific entropy loss. But unfortunately, we can't attest that we did an excellent job since we are not specialists in regards to hospitals constructions or such complex psychological themes such as the origin of stress. This would require a dialogue with different experts such as statisticians, physicians, psychologists, sociologists. We are neither. We can make assumptions, of course. But they are not valuable because made by students, far away from having sufficient knowledge in any field previously mentioned.