Random Aggregation: Differential Privacy in k-Anonymity

Liyuan Yang
yangly7@chinatelecom.cn

Security Technology Research Division Research Institute of China Telecom Corporation Limited
Shanghai 201315, China

Abstract—Microaggregation is used to solve the problem of privacy protection. A new method of microaggregation is proposed in this paper to satisfy differential privacy. It adds random noises to the centroids of equivalence classes, which reduces the risks of differential attacks and link attacks. Since the process of k-partition is very time-consuming, a top-down method is used to divide the original dataset into equivalence classes. Firstly, a matrix of distances is built and the original dataset is divided into several smaller datasets. And then the equivalence classes are formed by splitting datasets repeatedly. The experiments are conducted on two different datasets. Comparing with the existing methods, the results show that the proposed methods have the advantage of high efficiency and less information loss.

Keywords—random k-anonymity, privacy protection, statistical disclosure control, differential privacy, random aggregation, microaggregation

1 INTRODUCTION

In recent years, people pay more and more attention to the protection of personal data. As releasing microdata about individuals poses privacy threat due to the privacy-related attributes, existing laws and regulations require that statistical disclosure control (SDC) technology should be used to protect microdata in cloud computing [1]. The purpose of SDC is precisely to ensure that only useful macrotrends are learned by the recipients of such data and individual privacy is therefore protected. Among them, *k*-anonymity is a widely used method. It requires a certain number (at least *k*) of records that are indistinguishable on quasi-identifiers presence in the published data. The set of these records is called equivalence class. It only has 1/*k* probability for an attacker to link to the correct result. The larger *k* is, the more privacy information is protected, but the more information is lost.

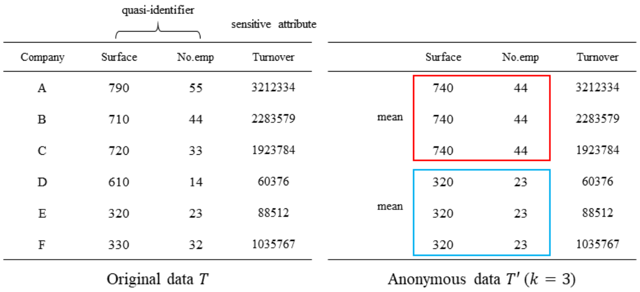
There are two common ways to implement *k*-anonymity. One is generalization, and the other is microaggregation. Generalization is a method of expanding the value of a specific quasi-identifier into a larger value range, so that it can no longer uniquely represent a record in the dataset [2]. For example, “male” and “female” can be generalized into “gender unknown”, age “25” can be generalized to an age domain “20~29”. Generalization is more suitable for typed data. For continuous data, it has no method to determine the reasonable domain of identifiers, which will lose more information. Whereas, microaggregation (Fig.1) classifies at least *k* nearest records of original data into one equivalence class, and replaces the identifiers of these records

FFIT 2022, October 28-30, Shenzhen, People's Republic of China

DOI 10.4108/eai.28-10-2022.2328460

with the identifiers of centroid. It is more suitable to aggregate continuous data, because algorithms require calculating distances of records [3].

According to the size of each equivalence class, heuristic microaggregation algorithms can be divided into two types: fixed-size heuristic and variable-size heuristic. The most famous fixed-size heuristic is Maximum Distance to Average Vector (MDAV) [4]. However, being a fixed-size heuristic, it lacks flexibility for adapting the group size to the distribution of the records. Variable-size Maximum Distance to Average Vector (V-MDAV) adds a step that allows MDAV to adapt to the dataset distribution and generates variable-size groups [5]. In order to further improve the efficiency of algorithm, Song, Ma and Tian et al. [6] proposed a bottom-up microaggregation algorithm. Before searching for equivalence classes, they split datasets into several parts to reduce the search size. Experiments show that this method is effective. But the bottom-up steps are highly dependent on random sampling. If inappropriate samples are chosen, it will result in low quality clustering results.



	quasi-identifier		sensitive attribute
Company	Surface	No.emp	Turnover
A	790	55	3212334
B	710	44	2283579
C	720	33	1923784
D	610	14	60376
E	320	23	88512
F	330	32	1035767

Original data T

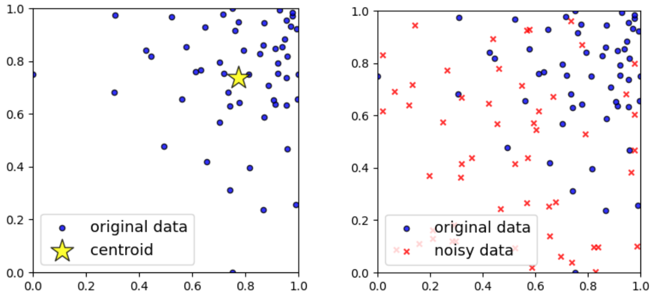
	Surface	No.emp	Turnover
	740	44	3212334
mean	740	44	2283579
	740	44	1923784
	320	23	60376
mean	320	23	88512
	320	23	1035767

Anonymous data T' $(k = 3)$

Fig.1. Microaggregation.

Microaggregation has some disadvantages when it anonymizes continuous attributes. Firstly, as shown in Fig.1, the quasi-identifiers of records near centroid just change a little bit after microaggregation. It could result in more frequent link attacks on these records, such as Company E, with a high disclosure risk. Secondly, anonymous tables T' cannot resist differential attacks. Assuming that an attacker knows $k - 1$ records in an equivalence class, the remaining record can easily be guessed. For example, in Fig.1, if an attacker knows information about Company A and B, columns (Surface, No.emp) of lines 1 and 2 in T' probably be guessed through sensitive attribute (Turnover). Then, the quasi-identifiers of line 3 will be inferred from the mean value and company C may be linked successfully.

In view of differential attacks, differential privacy (DP) method can be used in microaggregation algorithms to enhance privacy protection. Dwork, Kenthapadi and Mcsherry et al. [7] randomly selected records in the original data through a sampling algorithm satisfying DP. They proved that these secure k-anonymous tables could resist differential attacks. In addition, Song, Ma and Tian et al. [6] proposed a random k-anonymous method to solve the above two disadvantages. As shown in Fig.2b, this method makes anonymous records appear randomly anywhere within the equivalence class range. Attackers cannot obtain the quasi-identifiers of the unknown individuals through differential reasoning. At the same time, records near the centroid may move away after being anonymous, so that link attacks are successfully resisted. Experimental results showed that this method could keep useful macro trends of microdata. However, squared errors of random k-anonymous is always larger than the one of microaggregation (Fig.2a). This is because the data after adding noise may be much different from the original data. But in the traditional method, the sum of squared distances from the centroid to all records is minimum. Data availability versus security is a difficult trade-off.



(a) Microaggregation (b) Random *k*-anonymity

Fig.2. Two ways to anonymize equivalence class.

This paper mainly makes the following contributions.

- A top-down method for finding equivalence classes is proposed, which is more efficient in experiments. It is suitable for continuous data.
- A new microaggregation method, called random aggregation, is proposed. It adds noises on the centroids to satisfy DP, and also protects the records close to the centroid of equivalence classes.
- An algorithm is proposed to implement random aggregation. A scale factor ρ is considered to balance data availability and disclosure risk.

The rest of the paper is organized as follows. Section 2 contains background on *k*-anonymity, microaggregation, DP and random *k*-anonymity. In section 3, random aggregation is proposed. In order to achieve *k*-anonymous efficiently, a top-down partition method is adopted as well. Section 4 shows experimental results of two datasets. And conclusions are given out in section 5.

2 BACKGROUND

2.1 Notations

Table 1 gives the key notations used in this paper.

Table 1 Key Notations

Notations	**Definition**
T	The original data.
T'	The *k*-anonymous data.
T_i	The *i*-th partition of T.

M	A matrix of distances containing the distances between any two records.
E	Equivalence classes of T.
E_i	Equivalence classes of T_i.
k	A positive integer.
c	A positive integer.
ρ	A float in the range [0, 1].
A_n	The n-th attribute of records.
x_i	The i-th record in data.

2.2 k-Anonymity

k-Anonymity is a kind of privacy protection method proposed by Samarati and Sweeney [2]. The following is a brief introduction of its definition.

Definition 1 (k-anonymity): Let $T(A_1, A_2, \ldots, A_n)$ be a table and QI be the quasi-identifiers associated with it. T is said to satisfy *k*-anonymity with respect to QI, if and only if each sequence of values in $T[QI]$ appears at least with k times.

k-Anonymity protects personal privacy to a certain extent, but at the same time reduces data availability. Therefore, the researches on *k*-anonymity mainly focus on improving data availability while protecting private information. Recent studies evaluated *k*-anonymity algorithms by availability and security indicators [5, 6, 8, 10].

Considering a microdata set *T* with *p* numeric attributes and *n* records, each record x_i is represented as a vector in a *p*-dimensional space. The mean of all records is $\bar{x}$. For a given positive integer *k*, an anonymization method partitions *T* into *g* classes, where each class contains at least *k* records to satisfy *k*-anonymity. x_i' is the ith record of anonymous data *T'*. On this basis, Equation (1) is usually used to represent information loss (IL) [5, 8, 10]. SSE in (2) is the sum of squared errors. And SST in (3) is the total sum of squares (sum of squared Euclidean distances from all records to the centroid of original data), which is fixed regardless of how *T* is partitioned.

$$IL = SSE/SST \cdot 100, \tag{1}$$

$$SSE = \sum_{i=1}^{n} \|x_i - x_i'\|_2 , \tag{2}$$

$$SST = \sum_{i=1}^{n} \|x_i - \bar{x}\|_2 \ . \tag{3}$$

k-Anonymity and its variants can resist many attacks. For example, *l*-diversity model can resist homogenization attack [9], and *t*-closeness model can resist skew attack [10]. However, all *k*-anonymous microdata need to face up to the link attack. Attackers would link an anonymous record to the most similar individual. One way to measure the disclosure risk is to calculate the percentage of records in anonymous data *T'* which are successfully linked. For example, (740, 44) is the first noisy record in Fig.1, its first and second nearest records in *T* are (710, 44) and (720, 33). So it is not linked to the original record (790, 55). However, the second noisy record is successfully linked, because its original record is (710, 44). As shown in (4), distance linked disclosure (DLD) risk is the ratio of successfully linked records to total records [11].

$$DLD = \frac{linked\ records}{total\ records} \tag{4}$$

2.3 Microaggregation

The definitions about microaggregation were given in [3] as follows.

Definition 2 (k-partition): Let $T(A_1, A_2, \dots, A_n)$ be a table and QI be the quasi-identifiers associated with it. T is partitioned into g classes, and n_i is the size of the i-th class. If $\forall i, n_i > k$, this partition is called *k*-partition.

Definition 3 (aggregation): Let $T(A_1, A_2, \dots, A_n)$ be a table and QI be the quasi-identifiers associated with it. T is *k*-partitioned into g classes with respect to QI, and c_i is the centroid of the i-th class. For $\forall i = 1, \dots, g$, replacing all elements of the i-th class with c_i is called aggregation.

Microaggregation consists of two steps: *k*-partition and aggregation. Optimal microaggregation is based on optimal *k*-partition, which requires maximum homogeneity within the classes after partition. Finding optimal *k*-partition is NP-hard, so heuristic algorithms are often used [5]. Some studies have proved that the optimal size of classes should be [*k*, 2*k*-1] [12].

2.4 Differential Privacy

Differential privacy (DP) was proposed by Dwork [13], without making assumptions about the background knowledge of the attacker which could be proved mathematically. The definition was given as follows.

Definition 4 (differential privacy): A randomized function K gives ε-differential privacy if

$$e^{-\varepsilon} \leq \frac{\Pr[K(D_1) = S]}{\Pr[K(D_2) = S]} \leq e^{\varepsilon}, \tag{5}$$

for both datasets D_1 and D_2 differing on at most one element, and all $S \in Range(K)$.

The notation $\Pr[K(D) = S]$ represents the probability that the function K takes as input a dataset D and outputs the result S. DP makes it impossible for an attacker to identify the source dataset for S. The record that D_1 and D_2 differs on is protected.

2.5 Random k-Anonymity

Random *k*-anonymity was proposed by Song, Ma and Tian et al. [6]. The definition was given as follows.

Definition 5 (random k-anonymity): q is a random query on table T, the probability that $q(T)$ is generated by $x_1, x_2, \dots, x_{k'}$ is equal, where $x_i \in T, k' \geq k$. It is said that T satisfies random *k*-anonymity for query q.

To implement this definition, they *k*-partitioned table T into g classes firstly. But k records no longer shared the same quasi-identifiers. Their goal was that k records were indistinguishable. For each equivalence class, they added random noise to x_i. Consider x_i in class g has p numeric attributes. For $j = 1, \dots, p$, the noise n_j is uniform random numbers on $[-\frac{a_j - b_j}{2}, \frac{a_j - b_j}{2}]$, where a_j and b_j is the maximum and minimum of the j-th attribute in class g. Anonymous attributes $x_{ij}{}' = x_{ij} + n_j$. In order to keep $x_{ij}{}'$ within $[a_j, b_j]$, $x_{ij}{}'$ is further processed by (6). The information entropy between the anonymous table and the original table is low, but IL in (1) will be large.

$$x'_{ij} = \begin{cases} x_{ij}{}' - (a_j - b_j), & if\ x_{ij}{}' > a_j \\ x_{ij}{}' + (a_j - b_j), & if\ x_{ij}{}' < b_j \end{cases} \tag{6}$$

Compared with Definition 1, k records may have different quasi-identifiers, which means that every record can have its own unique identifier. The only requirement is the probability that each record corresponding to the query results is equal to each other, so that Definition 4 is satisfied. Definition 1 is only one of the methods in the implementation of Definition 5. If the one that satisfies Definition 1, it certainly satisfies Definition 5. Definition 5 is an extension of Definition 1 on DP.

3 A NEW METHOD FOR RANDOM K-ANONYMITY

3.1 Normalization of Data

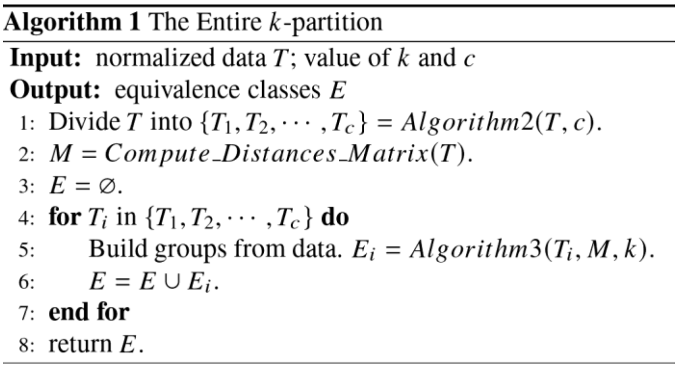
To avoid attribute bias, microdata should be normalized before applying microaggregation methods. In this paper, the following normalization formulae is used in order to adjust values measured on different scales to a notionally common scale [8]:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{7}$$

This unity-based normalization carried out in order to bring all values into the range [0, 1].

3.2 A Top-Down Partition Method

A new method for random k-anonymity is proposed in this paper. There are two steps to achieve k-anonymity. Firstly, the original data will be k-partitioned as Algorithm 1. And secondly, the quasi-identifiers of records in each equivalence class will be replaced with noisy centroid as Algorithm 4.



```
Algorithm 1 The Entire $k$-partition
Input: normalized data $T$; value of $k$ and $c$
Output: equivalence classes $E$
Divide $T$ into $\{T_1, T_2, \cdots, T_c\} = Algorithm2(T, c)$.
$M = Compute\_Distances\_Matrix(T)$.
$E = \varnothing$.
for $T_i$ in $\{T_1, T_2, \cdots, T_c\}$ do
    Build groups from data. $E_i = Algorithm3(T_i, M, k)$.
    $E = E \cup E_i$.
end for
return $E$.
```

Algorithm 2 Divide T into subclasses

```
Input: normalized data $T$; the number of clusters $c$
Output: $\{T_1, T_2, \cdots, T_c\}$ is the partition of $T$
A random sample is selected as the first clustering centroid $C_1$.
for $j$ in $\{2, \cdots, c\}$ do
    Calculate distances between records and centroids.
      $D(x_i) = min\{\|x_i - C_1\|, \cdots, \|x_i - C_{j-1}\|\}, \forall x_i \in T$.
    According to $D(x_i)$, the next cluster centroid $C_j$
      is selected by roulette wheel selection.
end for
while the cluster centroids do not change or the number of
  iterations exceeds a certain value do
    for each $T_i$ in $\{T_1, T_2, \cdots, T_c\}$ do
        Calculate distances between each record and each
          cluster center.
        Classify the record to the nearest cluster centroid.
    end for
    Update $\{T_1, T_2, \cdots, T_c\}$, and the new cluster centroid is
      the mean value of records in each cluster.
end while
return $\{T_1, T_2, \cdots, T_c\}$.
```

Algorithm 3 Build groups from data

```
Input: data $T$; distances matrix $M$; the value of $k$
Output: equivalence classes $E^*$
$E = \{T_i\}$, $E^* = \varnothing$.
while $E \neq \varnothing$ do
    Randomly remove $E_n$ from $E$.
    if $|E_n| \geq 2k$ then
        $(i, j)$ is the pair in $E_n$ that $M_{ij}$ is maximum.
        $E_1$ of $k$ records is formed by selecting $k - 1$ records
          closest to $i$, and $E_2$ of $k$ records is formed by
          selecting $k - 1$ records closest to $j$.
        Assign remaining records to the nearest group.
        for $E_x$ in $\{E_1, E_2\}$ do
            if $|E_x| < 2k$ then
                Add $E_x$ to $E^*$.
            else
                Add $E_x$ to $E$.
            end if
        end for
    else
        Add $E_n$ to $E^*$.
    end if
end while
return $E^*$.
```

Algorithm 1 uses two methods to find equivalence classes faster. One is dividing the original dataset into smaller datasets (line 1), and the other is building a matrix of distances in advance to avoid repetitive computing (line 2). Some studies [5, 6] have proved that these methods obtain high quality clustering results faster. Algorithm 2 is the specific steps for finding subclasses, which is based on the principle of c-means++. The first clustering centroid C_1 is selected randomly. Then, the minimum distances $D(x_i)$ between remaining records and centroids are calculated. On this basis, the next cluster centroid is selected by roulette wheel selection. Once all centroids are selected, all records are assigned to the closest centroid. And the new centroid is the mean value of records in each cluster. Then the partition is updated with this strategy until the cluster centroids do not change or the number of iterations exceeds a certain value. Compared with the c-means method, it has fewer errors and less iterations [14].

Algorithm 3 is a top-down method to find equivalence classes for each subclass. It starts by treating all records as an equivalence class. Next, select the farthest pair in the class. Once the pair is found, two groups of k records are formed by selecting $k - 1$ records closest to the pair. And the remaining records are assigned to the nearest group. The method will apply this strategy to each group until the size of group is [k, 2k-1]. Compared with the bottom-up method [6], Algorithm 3 is significantly less dependent on random numbers. Fixed distribution of dataset results in fixed partition.

3.3 Random ρ-Aggregation for DP

Random aggregation is a new method proposed in this paper to anonymize equivalence classes. Its definition is as follows.

Definition 6 (random aggregation): Let $T(A_1, A_2, \dots, A_n)$ be a table and QI be the quasi-identifiers associated with it. T is k-partitioned into g classes with respect to QI, and c_i is the centroid of the i-th class. n_i is a series of random numbers. For $\forall i = 1, \dots, g$, replacing all elements of the i-th class with $c_i + n_i$ is called random aggregation.

Obviously, Definition 3 is a special case of Definition 6 when $n_i = 0$. Algorithm 4 gives a method to implement random aggregation. Before replacing the records with centroids, there is an additional step (line 3~5) to add random noises. Consider a, b, m are respectively the maximum, minimum and mean of E_i. The noises are uniformly chosen from $[\rho(b-m), \rho(a-m)]$ at random. ρ is a scale factor that controls the scope of noisy records, just like Fig.3. If $\rho = 0$, the following method is the same as Definition 3 (Fig.2a). If $\rho = 1$, noisy records will be uniformly distributed between intervals $[a, b]$, which is the same as Fig.2b.

At the same time, Algorithm 4 satisfies the definition of random k-anonymity in Definition 5. The following is a brief proof [6].

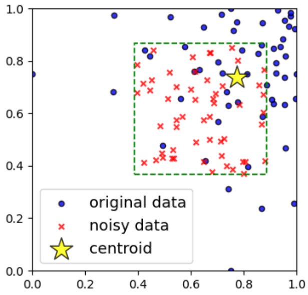
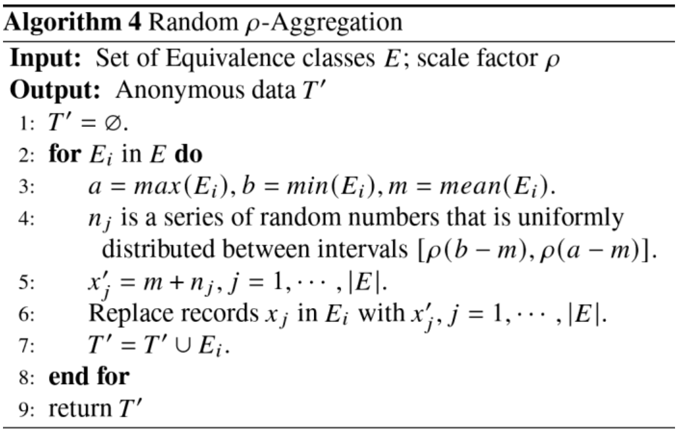


Fig.3. Random ρ-Aggregation (ρ = 0.5).

Proof: $x_j{}'$ is a noisy record in the output T' of Algorithm 4. x_j is the corresponding original record in T, and x_i is another record. n_j is the random number uniformly distributed in $[\rho(b-m), \rho(a-m)]$. Δx is the difference between $x_j{}'$ and x_i, also in the interval.

$$\Pr(x_j') = \Pr(n_j = x_j' - x_j) = \Pr(\Delta x = x_j' - x_i) \tag{8}$$

There are at least k records that have the same probability in one equivalence class, so Algorithm 4 satisfies the Definition 5 and DP. Even if an attacker knows $k - 1$ records in an equivalence class, the remaining record cannot be guessed. On the other hand, noisy records protect the original records near the centroid, which are no longer easy to link successfully. To sum up, the larger ρ, the lower disclosure risk.



Algorithm 4 Random ρ-Aggregation

```
Input: Set of Equivalence classes $E$; scale factor $\rho$
Output: Anonymous data $T'$
$T' = \varnothing$.
for $E_i$ in $E$ do
    $a = max(E_i), b = min(E_i), m = mean(E_i)$.
    $n_j$ is a series of random numbers that is uniformly
      distributed between intervals $[\rho(b - m), \rho(a - m)]$.
    $x'_j = m + n_j, j = 1, \cdots, |E|$.
    Replace records $x_j$ in $E_i$ with $x'_j, j = 1, \cdots, |E|$.
    $T' = T' \cup E_i$.
end for
return $T'$
```

3.4 Analysis

In Algorithm 1, the time complexity of computing distances matrix is $O(n^2)$. And the complexity of Algorithm 2 (c-means++) is $O(c \cdot n)$, which c is the number of clusters and n is the size of data [14]. The original dataset T is divided into $\{T_1, T_2, \dots, T_c\}$. Each class has n/c records on average. The complexity of line 5 (Algorithm 3) is $O(n^2/c^2)$. So, the total complexity of k-partition method (Algorithm 1) is $O(n^2) + O(c \cdot n) + c \cdot O(n^2/c^2) = O(n^2 + n^2/c + c \cdot n)$. Since c is much smaller than n, the complexity of Algorithm 1 is $O(n^2)$. On the other hand, the time complexity of random aggregation (Algorithm 4) is $c \cdot O(n/c) = O(n)$.

Dividing the original dataset into c clusters can speed up k-partition. However, too many clusters may reduce the quality of equivalence classes. How to determine the best values of c is not straightforward and due to space limitations it will not be discussed in this paper.

4 EXPERIMENTAL RESULTS

4.1 Experiments Design

This paper carries on 3 experiments to test the performance of methods. Experiment 1 uses the k-partition method proposed in this paper and the bottom-up method proposed in latest reference [6] to microaggregate the datasets. They are compared in terms of running time and cluster similarity. Experiment 2 compares the information loss (IL) of random k-anonymity, microaggregation and random ρ-aggregation on two datasets. Experiment 3 compares their distance linked disclosure (DLD) risk in the same way.

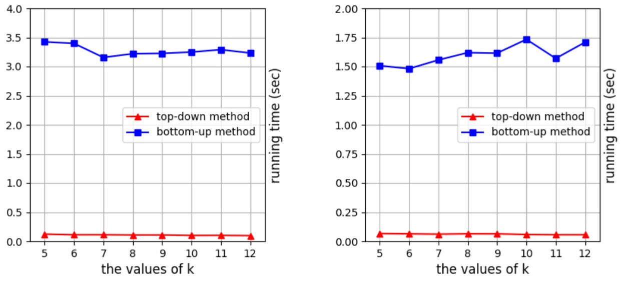
The following two datasets from the related website of UCI machine learning (https://archive-beta.ics.uci.edu) are used in this section:

- *Breast Cancer.* It consists of 699 records with 13 numeric attributes.
- *Diabetes.* It consists of 769 records with 8 numeric attributes.

All experiments are conducted on a laptop with Intel(R) Core (TM) i7-1165G7 @ 2.80GHz, 16.0 GB RAM, Windows 10 64-bit and Python 3.7.

4.2 Efficiency of k-Partition

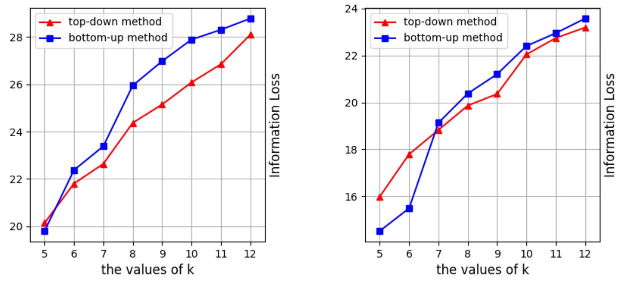
The top-down method and the bottom-up method are used to microaggregate the datasets in this experiment. As can be seen from Fig.4, the top-down method takes much less time than the bottom-up method in both datasets. On the other hand, IL in (1) is used to represent the cluster similarity of equivalence classes. As shown in Fig.5, the top-down method has less IL in most cases. This is because the bottom-up method frequently makes random choice, and a bad choice may produce a poor-quality result. In a few cases of Fig.5b, the top-down method has more IL because of the distribution of dataset and the value of *k*. But on the whole it is more efficient than the bottom-up method.



(a) Breast Cancer

(b) Diabetes

Fig.4. Running Time (sec)



(a) Breast Cancer

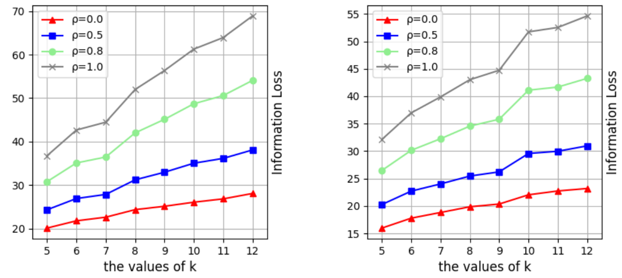
(b) Diabetes

Fig.5.IL Comparison of Two *k*-Partition Methods

4.3 IL of Random ρ-Aggregation

Microaggregation, random ρ-aggregation and random *k*-anonymity are used to anonymize datasets in this experiment. In fact, microaggregation and random *k*-anonymity are special cases of random ρ-aggregation when ρ=0.0 and ρ=1.0. Fig.6 shows the IL of random ρ-aggregation with different values of ρ. With the same value of *k*, the larger ρ, the more information loss. The difference among them is obvious, because the noisy records are widely distributed with large ρ.

With the same value of ρ, the larger k, the more information loss. The trend is very smooth, because a bigger equivalence class will cause more errors in the aggregation process.



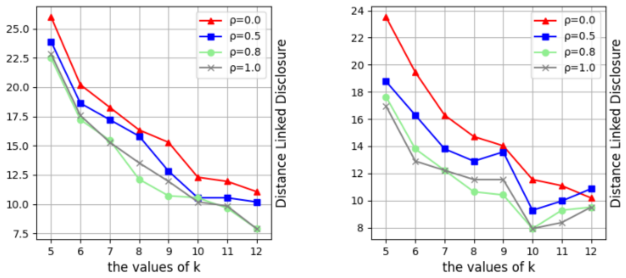
(a) Breast Cancer

(b) Diabetes

Fig.6.IL Comparison of different ρ

4.4 DLD of Random ρ-Aggregation

Experiment 3 tests the DLD risk by (4) in the case of ρ=0.0, 0.5, 0.8 and 1.0. Fig.7 shows that microaggregation (ρ=0.0) is poorer at resisting link attack, and a larger ρ has lower DLD. Sometimes random ρ-aggregation (ρ=0.8) has lower DLD than ρ=1.0. It is related to the distribution of datasets. Overall, a larger k generates a smaller DLD risk, because the attacker needs to try linking more records. In Fig.7b, there are upward trends when k=9 and k=11. This is because the randomly distributed noisy records are close to the original records accidently.



(a) Breast Cancer

(b) Diabetes

Fig.7.DLD Risk of different ρ

4.5 Discussion

The above experimental results and analyses show that random aggregation method has the advantage of high efficiency and less information loss to provide privacy protection. The algorithm proposed in this paper balances data availability and disclosure risk by using the ρ scale factor. In general, a larger ρ will result in greater information loss and lower disclosure risk.

But in a few cases, random noises may have a negative effect on the disclosure risk. For example, in the case of k=8 and k=9, ρ=0.8 performs lower disclosure risk than ρ=1.0 in two datasets. So it is necessary to adjust ρ several times for best results.

5 CONCLUSION

Random aggregation, a new method for multivariate microaggregation has been presented. Random aggregation adds random noises to the centroids of equivalence classes, which reduces the risks of differential attacks and link attacks. At the same time, a top-down method is proposed to k-partition microdata in this paper. The original dataset is divided into several smaller datasets, and the equivalence classes are formed by splitting datasets repeatedly until the size of each class is [k, 2k-1]. Experimental results show that the top-down method consumes less time to form better equivalence classes.

REFERENCES

[1] W. Stallings, "Handling of Personal Information and Deidentified, Aggregated, and Pseudonymized Information Under the California Consumer Privacy Act," in IEEE Security & Privacy, vol. 18, no. 1, pp. 61-64, Jan.-Feb. 2020, doi: 10.1109/MSEC.2019.2953324.

[2] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in Proc. PODS, Jun. 1998, p. 188. doi: 10.1145/275487.275508.

[3] W. Lixia and H. Jianmin, "Utility evaluation of K-anonymous data by microaggregation," 2009 ISECS International Colloquium on Computing, Communication, Control, and Management, 2009, pp. 381-384, doi: 10.1109/CCCM.2009.5270417.

[4] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," in IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 1, pp. 189-201, Jan.-Feb. 2002, doi: 10.1109/69.979982.

[5] Solanas, Am Agusti . "V-MDAV: A Multivariate Microaggregation With Variable Group Size." Seventh Compstat Symposium of the Iasc.

[6] F. Song, T. Ma, Y. Tian and M. Al-Rodhaan, "A New Method of Privacy Protection: Random k-Anonymous," in *IEEE Access,* vol. 7, pp. 75434-75445, 2019, doi: 10.1109/ACCESS.2019.2919165.

[7] C. Dwork, K. Kenthapadi, F. Mcsherry, I. Mironov and M. Naor, "Our Data, Ourselves: Privacy Via Distributed Noise Generation." International Conference on Advances in Cryptology-eurocrypt (2006).

[8] M. E. Kabir, A. N. Mahmood, H. Wang and A. K. Mustafa, "Microaggregation Sorting Framework for K-Anonymity Statistical Disclosure Control in Cloud Computing," in IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 408-417, 1 April-June 2020, doi: 10.1109/TCC.2015.2469649.

[9] H. Zhu, S. Tian and K. Lü, "Privacy-Preserving Data Publication with Features of Independent ℓ-Diversity," in The Computer Journal, vol. 58, no. 4, pp. 549-571, April 2015, doi: 10.1093/comjnl/bxu102.

[10] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez and S. Martínez, "t-closeness through microaggregation: Strict privacy with enhanced utility preservation," 2016 IEEE 32nd International Conference on Data Engineering (ICDE), 2016, pp. 1464-1465, doi: 10.1109/ICDE.2016.7498376.

[11] J. Domingo-Ferrer, J. M. Mateo-Sanz and V. Torra, "Comparing SDC Methods for Microdata on the Basis of Information Loss and Disclosure." Proceedings of Etk Ntts Luxemburg Eurostat (2004).

[12] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression". International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002, Vol 10, Issue 5, pp. 571-588.

[13] C. Dwork, "Differential privacy, " Lect. Notes Comput. Sci., vol. 26, no. 2, pp. 1-12, 2006. doi: 10.1007/11787006_1.

[14] O. Chovancova, L. Piatrikova and A. Dudas, "Improving Fuzzy C-Means Algorithm using Particle Swarm Optimization and Fuzzy C-Means++," 2019 International Conference on Information and Digital Technologies (IDT), 2019, pp. 173-179, doi: 10.1109/DT.2019.8813.