TRANSACTIONS ON DATA PRIVACY 15 (2022) 177–192

Improving Data Utilization of K-anonymity through Clustering Optimization

Hewen Wang*, **Jingsha He***, **Nafei Zhu***

*Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China.

E-mail: hewenwang@emails.bjut.edu.cn, jhe@bjut.edu.cn, znf@bjut.edu.cn

Received 19 July 2021; received in revised form 1 October 2022; accepted 19 December 2022

Abstract. K-anonymity privacy protection model demonstrates good performance in privacy protection and, has been widely applied in such scenarios as data publishing, location-based services, and social networks. With the aim of ensuring k-anonymity to conform to the requirements of privacy protection with improved data utilization, this study proposes a k-anonymity algorithm based on central point clustering, so as to improve the quality of clustering through optimizing the selection of cluster centroids, leading to the improvement in effectiveness and efficiency of k-anonymity. After clustering, the quasi-identifier attributes are aligned for classification and generalization, which is evaluated using appropriate information loss metrics. To measure the distance between records and between records and clusters, this study also establishes a definition of such distance that is positively correlated to the amount of information that is lost by combining the characteristics of the depth and width of the generalization hierarchy, in an effort to improve of the utility of the algorithm. The experimental results show that the proposed algorithm not only meets the basic anonymity requirements, but also improves data utilization compared with some prevailing algorithms.

Keywords. Clustering-based k-anonymity, Data privacy, Privacy protection, Information security, Microaggregation

1 Introduction

With the rapid development of Internet-related information sharing technology and the continuous expansion of the scope of data sharing among applications, how to effectively reduce the risk of user-sensitive information disclosure and maximize data utilization has become one of the challenges of privacy protection in the process of data publishing. Some institutions and organizations usually preprocess sensitive data before they publish original data sheets, such as deleting names, ID numbers, and other identifiers that point to specific individuals. However, attackers can still manage to initiate *linking attacks*, so as to achieve accurate identification of an individual in the data set through the combination of relevant information outside the released data set and the information inside. Sweeney from Carnegie Mellon University in the United States pointed out that by connecting with the published relational table, individual privacy can be inferred with a higher probability [1].

K-anonymity [2, 3] is a relatively classic approach to privacy protection. It was first proposed by Sweeney et al. and has been mainly used to prevent privacy leakage from *linking*

attacks. The core idea of k-anonymity is to ensure that each record in the data table to be published is at least indistinguishable from other $k-1$ ($k>1$) records on the quasi-identifier set. In other words, each quasi-identifier sequence value appears at least k times to ensure that the probability of a *linking attack* on the published data table does not exceed $1/k$. In addition, some studies indicated that the computational complexity of finding the optimal solution for k-anonymity is *NP-hard* [4]. *Generalization* and *suppression* [5] are two commonly used techniques to implement the k-anonymity model in applications. Generalization replaces the original value with a more generalized value, while suppression is a special form of generalization by deleting the original value or replacing the original value with "$*$". Both methods will cause information loss of the original data, and overgeneralization will greatly increase the amount of information loss and reduce the data utility. Therefore, how to maintain the balance between maximizing data utilization and protecting data privacy is still an important topic in the research of privacy protection technology, which is also the main objective of this study.

In addition to generalization and suppression techniques, cluster-based methods can also achieve k-anonymity. In 2006, Aggrawal et al. [6] first combined the idea of clustering algorithms with the k-anonymity problem of data. Since then, studies based on such a combination have been continuously conducted. Similarly, the main objective of this study, as mentioned above, is to solve the problem of balancing between the data privacy and the data utility of k-anonymity based on the idea of clustering. In view of this objective, this study proposes a K-anonymity Algorithm Based on Center Point Clustering (i.e., KACPC). Moreover, this study also (1) reestablishes the definition and formula of the distance between records and between records and clusters (i.e., equivalence class), (2) more practically measures the pros and cons of anonymous results, and (3) proposes a new method for selecting the initial cluster centroid and the subsequent cluster centroids during the clustering process. The experimental results show that, compared with Improved K-anonymity Algorithm based on Clustering (i.e., IKAC) [7], One-pass K-means Algorithm (i.e., OKA) [8], MDAV-generic algorithm [9] and Greedy k-anonymity Algorithm Based on Clustering Partition (i.e., GAA-CP) [1], KACPC can greatly improve data utility and simultaneously meet the requirements of data privacy.

The remainder of this study is organized as follows: Section 2 outlines the research work related to the k-anonymity model; Section 3 introduces the basic concepts related to the k-anonymity model, and presents new metrics of distance; Section 4 proposes KACPC. Section 5 mainly compares KACPC with other k-anonymity algorithms based on clustering and analyzes the experimental results; Section 6 summarizes this study as the end.

2 Related Work

As a simple and effective approach to achieving anonymity, the k-anonymity privacy protection model is of great significance and value not only in theoretical research but also in practical application. Therefore, scholars worldwide have continuously carried out research work related to the model. The MinGen [5] algorithm proposed by Sweeney in 2002 manages to find the optimal solution in the search space of global generalization, but the algorithm needs to traverse the entire generalization space when the optimal generalization operation is selected, which leads to high time complexity and low practical applicability. Then, Wang et al. [10] proposed an anonymous method of bottom-up progressive generalization in 2004, but this method is only applicable to categorical attributes. The K-optimize algorithm proposed by Bayardo et al. [11] in 2005 adopts a tree-search strategy that utilizes

both cost-based pruning and dynamic search rearrangement to successfully find the optimal solution in the search space of global generalization, though the values of all attributes need to be sorted in advance. Babu et al. [12] proposed an improved greedy heuristic algorithm to achieve k-anonymity in 2013. This algorithm mainly adopts a heuristic search method to balance the time complexity of calculation and the amount of information loss. However, the global recoding technology incorporated in the algorithm can easily lead to overgeneralization. In 2020, Liang et al. [13] proposed the optimal segmentation theory based on the standards of disclosure risk measures, and built a global optimal model for data privacy protection, that is, (d, q)-division, in which data precision and security are controlled by the parameters d and q, though the setting of the two parameters d and q increases the burden of the model users.

Among the many prevailing k-anonymity techniques, clustering is a mature and widely used data analysis method. Similar to clustering, many microaggregation algorithms are based on a certain clustering idea [14], though they can not determine the number of clusters to be generated in advance, as clustering algorithms can. Additionally, microaggregation integrates a series of statistical disclosure control techniques that were originally designed for quantitative (numeric) data [15] and then gradually expand to categorical data [9, 16]. To the best of our knowledge, the Maximum Distance to Average Vector (i.e., MDAV) algorithm [17] is a microaggregation algorithm with superior performance. In order to improve the execution efficiency of MDAV, Rodríguez-Hoyos et al. [18] proposed five strategies to simplify the internal operation of the MDAV algorithm, but the MDAV algorithm can only handle numerical data. Domingo-Ferrer and Torra [9] introduced a distance quantization method for categorical data based on earlier MDAV algorithms, and an MDAV-generic algorithm suitable for multiple data types. Nevertheless, the MDAV-generic algorithm is based on microaggregation techniques, so it fails to consider the generalization tree while measuring the distance for categorical attributes, that is, the distance between the two values of a categorical attribute, which is 0 if the values are equal, or 1 if they are not. In the clustering process, the MDAV-generic algorithm only uses 0 or 1 to quantify the distance for categorical attributes, thereby degrading the quality of clustering.

The k-anonymity algorithm based on clustering is a kind of clustering algorithm that inherits the constraints of k-anonymity models. Since Aggrawal et al. [6] incorporated the clustering method into the k-anonymity privacy protection model, numerous relevant studies and clustering-based k-anonymity methods have emerged in recent years. In 2006, Li et al. [19] proposed K-Anonymization by Clustering in Attribute (i.e., KACA) hierarchies, which randomly selects a cluster that does not meet the requirement of anonymity each time and merges it with the closest cluster. The iterative process continues until all clusters achieve anonymity requirement. However, the KACA algorithm needs to predefine the generalization hierarchies of quasi-identifier attributes and can not distinguish between numerical attributes and categorical attributes, which can also easily lead to overgeneralization. In 2007, Chiu et al. [20] proposed a Weighted Feature C-Means Clustering (i.e., WF-C-means) algorithm to build the k-anonymity model. However, the class-merging mechanism of the algorithm does not consider the merging of two equivalence classes that do not meet the anonymity requirement. In 2016, Bhaladhare et al. [21] proposed two methods based on systematic clustering algorithms to successfully achieve k-anonymity. In 2018, Zheng et al. [7] proposed the IKAC algorithm which guarantees the quality of anonymous data by ensuring the minimum intra-cluster distance and the maximum inter-cluster distance. However, the IKAC algorithm randomly selects the initial cluster centroid, which leads to the randomness of anonymous results. Besides, the IKAC algorithm only considers the number of leaf nodes from the horizontal dimension of the attribute generalization

hierarchy, and ignores the factors in the vertical dimension of the hierarchy.

3 Basic Concepts

In order to facilitate the subsequent discussion of this study, this section introduces some basic concepts related to the k-anonymity privacy protection model and some definitions and equations used to measure the amount of information loss after anonymization.

3.1 K-anonymity Related Concepts

Generally, the attributes in the original data table to be released can usually be classified into the following four categories according to their functions: explicit identifier (i.e., EI) attributes, quasi-identifier (i.e., QI) attributes, sensitive (i.e., S) attributes and other attributes.

Definition 1. (**EI attributes**). Attributes of data tables that denote unique identity information, such as name, ID number, social security card number, etc. In general, these attributes are to be deleted directly when the data tables that contain them are to be published.

Definition 2. (**QI attributes**). Attributes of data tables that can be used to accurately infer information of individuals, such as age, country, gender, etc., when they are linked to external data. In general, the determination of quasi-identifier attributes depends on application scenarios, and they usually exist in the form of collections of quasi-identifiers. To a certain extent, attackers can launch *linking attacks* based on quasi-identifiers to obtain private information.

Definition 3. (**S attributes**). Attributes of data tables that contain sensitive private information of individuals, such as salary, phone number, diseases, etc.

Definition 4. (**Other attributes**). Attributes of data tables that can be disclosed. In general, these attributes are ignored to facilitate the discussion.

Definition 5. (**Cluster**). A cluster is also known as an equivalence class. Records with the same value on the quasi-identifier sequence constitute a cluster, which means that each value of the quasi-identifier sequence in k-anonymity is a cluster. Assuming that a table satisfying k-anonymity contains n records, then the table contains at most m clusters, $m = \lfloor n/k \rfloor$.

Definition 6. (**Information Loss Measures**). Since *precision* [5] is considered to be a metric for the data utility and the information loss of anonymous tables, this study will inherit this metric to conduct information loss measures. Assuming that there are t records in the data table $T(N_1, \cdots, N_m, C_1, \cdots, C_n)$, the number of numeric attributes and categorical attributes are m and n respectively, and T' is the k-anonymity result of T and contains $\omega = \lfloor t/k \rfloor$ clusters, and each cluster E_r ($r \in [1, \omega]$) has at least k records. The information loss caused by the generalization of a cluster is defined as follows.

$$Eloss(E_r) = \sum_{i=1}^{m} |E_r| \times \frac{\hat{E}_r[N_i] - \check{E}_r[N_i]}{\hat{T}[N_i] - \check{T}[N_i]} + \sum_{j=1}^{n} |E_r| \times \frac{h(E_r[C_j]) - 1}{h(C_j) - 1}, \tag{1}$$

where $|E_r|$ represents the number of records contained in the cluster E_r. $\hat{E}_r\,[N_i]$ and $\check{E}_r\,[N_i]$ represent the maximum and the minimum values in the cluster E_r for a numeric attribute

N_i, respectively, $\hat{T}[N_i]$ and $\check{T}[N_i]$ represent the maximum and the minimum values in the entire data table, respectively. $h(E_r[C_j])$ represents the height of the attribute value of E_r on the attribute C_j corresponding to the generalization hierarchy for a categorical attribute C_j, and $h(C_j)$ represents the height of the generalization hierarchy of the attribute C_j. When the amount of information loss is measured, the initial height of all leaf nodes in the generalization hierarchy is 1. Thus, the data precision corresponding to the total information loss generated by the data table T' can be calculated to reflect the utility of anonymous data as follows:

$$Pre(T') = 1 - \frac{\sum_{r=1}^{\omega} Eloss(E_r)}{t(n+m)} \tag{2}$$

3.2 Proposed Distance Metrics

At the heart of every clustering problem are the distance functions that measure the dissimilarities among data points and the cost function of which the minimum value is expected to be searched for [22]. Choosing the appropriate distance function and the cost function is very important for clustering problem. Generally, the determination of the distance function is based on the calculated data type, while that of the cost function is based on the specific scenario to which the clustering problem is applied. To some extent, the clustering division is consistent with the equivalence class division in k-anonymity. One of the key procedures to integrating the clustering method into k-anonymity is to define an appropriate distance function that measures the similarity between records: Records that are closer are more similar to each other. In consideration of the respective characteristics of numerical attributes and categorical attributes, we will use different distance functions to quantify their respective information loss.

Definition 7. (**Distance Between Numeric Data**) Assuming that in data table T, the interval size of a numeric attribute N is D, for any two attribute values v_i and v_j, where v_i, $v_j \in N$, the distance between v_i and v_j is defined as follows:

$$DistN(v_i, v_j) = \frac{|v_i - v_j|}{D} \tag{3}$$

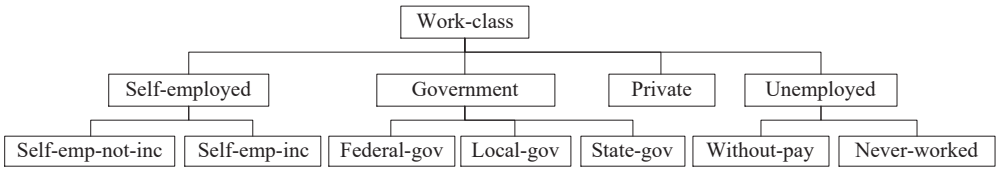


Figure 1: Generalization hierarchy of categorical attribute

Each categorical attribute corresponds to a specific generalization hierarchy, which is different from the single characteristic of the numeric attribute. For the generalization hierarchy of categorical attributes, it has both depth and width with regard to its characteristics. When the nodes in the generalization hierarchy are generalized upward, the information loss can be measured from the perspective of the horizontal dimension by calculating the

proportion of the number of leaf nodes contained in the generalized node to the total number of leaf nodes. And it can also be measured by calculating the height proportion of the generalized node on the generalization hierarchy from the perspective of the vertical dimension. Taking the above factors into consideration, we will combine width in the horizontal dimension and depth in the vertical dimension of the generalization hierarchy, which is a first time attempt to formulate a new definition of the distance between two categorical attribute values.

Definition 8. (**Distance Between Categorical Data**) Assuming that in data table T, T_C is a generalization hierarchy of categorical attribute C, for any two attribute values v_i and v_j, where $v_i, v_j \in C$, $\wedge(v_i, v_j)$ is the nearest common parent node of v_i and v_j on T_C. Then, the distance between v_i and v_j is the product of the distance between v_i and $\wedge(v_i, v_j)$ and the distance between v_j and $\wedge(v_i, v_j)$ as shown in Equation (5).

$$Dist(v_i, \wedge(v_i, v_j)) = (H(\wedge(v_i, v_j)) \times (H(\wedge(v_i, v_j)) - H(v_i)))^{\frac{H(\wedge(v_i, v_j))}{H(T_C)}} \times \frac{node(\wedge(v_i, v_j))}{node(T_C)} \quad (4)$$

$$DistC(v_i, v_j) = Dist(v_i, \wedge(v_i, v_j)) \times Dist(v_j, \wedge(v_i, v_j)), \quad (5)$$

where $H(X)$ represents the height of node X on T_C, $H(T_C)$ represents the height of T_C, (in this study, the initial height of the generalization hierarchy is 1, but the height of the leaf node in the calculation of the distance is based on its height in the generalization hierarchy), $node(X)$ represents the number of the leaf nodes contained in node X on T_C, and $node(T_C)$ represents the total number of leaf nodes of T_C. An example to the generalization hierarchy of a categorical attribute is shown in Figure 1, in which the value of $H(Private)$ is 2 and $H(Local - gov)$ is 1.

Equation (5) unifies the distance determination method for the same values of the attribute and different values of the attribute for the first time. More importantly, simple addition is no longer used when the distance between two leaf nodes in Equation (5) is measured, so as to improve the sensitivity of the distance when any two leaf nodes are generalized upward. In addition, when Equation (4) is used to calculate the distance between node v_i and the generalized node $\wedge(v_i, v_j)$, the height $H(\wedge(v_i, v_j))$ of the generalization and the height difference $H(\wedge(v_i, v_j)) - H(v_i)$ before and after the generalization are considered. In order to combine the impacts of the width and the height of the generalization hierarchy on the distance, $H(\wedge(v_i, v_j))/H(T_C)$ is used to weaken the product of the height in the previous part of Equation (4). Therefore, the distance between two leaf nodes is positively correlated with the generalization height difference, the height of the smallest common parent node, and the number of leaf nodes contained in the smallest common parent node. Taking the leaf nodes $Private$, $Local - gov$, and $State - gov$ in Figure 1 as an example to demonstrate the above idea, by using Equation (5), $DistC(Private, Local - gov) = 18$ and $DistC(State - gov, Local - gov) = 0.4628$. It can be seen that the generalization height, the height difference before and after generalization, and the number of leaf nodes all affect the distance between any two leaf nodes.

Definition 9. (**Distance Between Records**) The distance between two records is the accumulation of the distance of their respective attributes. Assuming that in data table T, $QIs = \{N_1, \ldots, N_m, C_1, \ldots, C_n\}$ is the set of its quasi-identifier attributes, where N_i ($i \in [1, m]$) and C_j ($j \in [1, n]$) represent numeric attributes and categorical attributes respectively, the definition of the distance between any two records r_α and r_β is given in Equation (6).

$$DistR(r_\alpha, r_\beta) = \sum_{i=1}^{m} DistN(r_\alpha[N_i], r_\beta[N_i]) + \sum_{j=1}^{n} DistC(r_\alpha[C_j], r_\beta[C_j]), \quad (6)$$

where $r\,[N]$ represents the attribute value of the record r on the attribute N. In particular, the algorithm proposed in this study will use the distance between the record and the cluster centroid to represent the distance between them.

4 K-anonymity Algorithm based on Center Point Clustering

In the clustering problem, the selection of the cluster centroid and the metrics of distance are two crucial factors. Since the metrics of distance have already been mentioned above, this section will focus on the centroid selection. The design of clustering methods can affect the security and utility of anonymized data. Arava et al. [23] pointed out that the challenge of clustering is to find the best seed values for collecting allied records which can be anonymized at the same level in order to reduce information loss. Therefore, the quality of clustering depends on the choice of centroid, and this section will introduce a new method for the selection of the cluster centroids. In addition, the KACPC algorithm will also be presented in this section.

4.1 Center Point Clustering

Initial cluster centroid selection is normally random, therefore the anonymous results of the data will be rendered unrepeatable and random. In contrast, the KACPC algorithm selects the appropriate initial cluster centroid according to the frequency of each QI attribute value and the distance between any two such values as defined above. Each QI attribute column in the table can be regarded as a dimension. First, the attribute value with the highest frequency of occurrence in each dimension will be selected to form a sequential value. Second, Equation (6) is applied to calculate the distance from all records to the sequential value. Finally, the nearest record to the sequential value is selected as the initial cluster centroid. This method can move the position of the initial cluster centroid to the center of each dimension as much as possible while eliminating randomness.

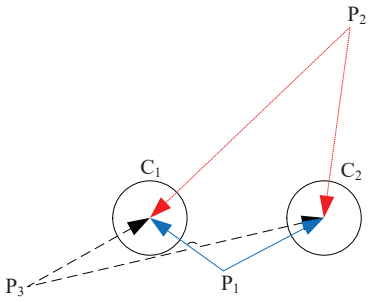


Figure 2: Selection of the cluster centroid

Different algorithms have their own selection mechanisms for the subsequent cluster centroids. As shown in Figure 2, C_1 and C_2 are two sequentially generated clusters, and P_1,

P_2 and P_3 are candidate cluster centroids. According to the description of the k-member algorithm [22], P_3 will be selected as the centroid of the next cluster when it is the farthest from the centroid of the previous cluster C_2. As for IKAC algorithm [7], the information of the positions of all previously generated clusters will be considered. The candidate cluster centroid with the maximum average distance from all previous clusters will be selected, which proves to be P_2 in Figure 2. In order to maximize the distance between the clusters, the above two methods tend to choose outliers as the cluster centroids, which reduces the quality of clustering and greatly increases the amount of information loss. Therefore, the KACPC adopts a different approach and selects the candidate centroid that has the minimum average distance from all previous clusters as the next cluster centroid, which proves to be P_1 (see Figure 2). By the selection mechanism of the initial and the subsequent cluster centroids of the KACPC, the clustering process can gradually spread from the center of the table space to the edge, so that the probability that the candidate centroid at the edge becomes the cluster centroid is reduced and the clustering quality is ensured.

4.2 Description of the KACPC Algorithm

This section proposes the KACPC algorithm which proves to be able to achieve k-anonymity. The core of KACPC is to use the idea of clustering to divide the records to be released into multiple clusters according to the principle of minimum information loss, when each cluster contains at least k records. Then the divided clusters are generalized according to the predetermined rules. The algorithm mainly includes the following steps. Given data table T with n records, select a record from T as the centroid ω_0 of the first cluster c_1 based on the method of selecting the initial cluster centroid in Section 4.1, then select $k-1$ records that are the closest to ω_0 from all unassigned records and merge them into c_1, so as to form the first cluster c_1 that satisfies k-anonymity. Afterwards, for each selection of the remaining $m = \lfloor n/k \rfloor - 1$ centroids ω_i ($i \in [1, m]$), the impact of the location of the previous $i-1$ clusters on the selection of the cluster centroid will be considered: calculate the total distance from each unassigned record in table T to all existing cluster centroids ω_j ($j \in [1, i-1]$) in turn and select the record with the minimum average distance as the centroid of the i-th cluster, then repeat this process until each cluster meets the requirements of k-anonymity. Finally, for the remaining $n - k \times \lfloor n/k \rfloor$ unassigned records, the KACPC sequentially inserts each unassigned record into the nearest cluster to reduce the amount of information loss.

After a complete clustering process, the number of records in each cluster meets the basic requirement of k-anonymity. The divided data set needs to be generalized with clusters as the basic unit in accordance with the predefined rules. For the generalization of numerical attributes, if the mean value of the numerical attributes in a cluster is used to represent the anonymous result, although the statistical significance of the data is guaranteed, it may produce data that does not exist in the original data set, thereby increasing the amount of information loss. Therefore, the KACPC algorithm adopts the interval between the maximum and the minimum values in a cluster to represent the anonymous result of the cluster. For the generalization of categorical attributes, the nearest common parent node of all attribute values in a cluster is used to represent the anonymous result of the cluster. In brief, the KACPC eliminates the influence of the random selection of the cluster centroid on the anonymous results, and takes into account the information of the locations of previously generated clusters for the selection of the subsequent cluster centroids, the latter of which prevents outliers from becoming the cluster centroids and ensures the quality of clustering.

4.3 Time Complexity Analysis

Algorithm 1: k-anonymity algorithm based on central point clustering

Input: Data table T with n records, anonymous parameter k
Output: a set of clusters and each cluster contains at least k records

1 $result = \emptyset; i = 0;$
2 $dist = 0;$ //A two-dimensional matrix storing distances, with rows representing clusters and columns representing records;
3 **while** $len(T) \geq k$ **do**
4 $i = i + 1;$
5 **if** $i == 1$ **then**
6 Find the attribute value with the highest frequency in each attribute column in table T and form it into a sequence value r_temp;
7 **for** $each\ record\ in\ T$ **do**
8 Calculate the distance between r_temp and $record$ according to Equation (6);
9 Select the $record$ with the smallest distance to r_temp as the initial cluster $centroid$;
10 **else**
11 Sum the first $i-1$ rows in $dist$ according to the records in each column;
12 Select the unassigned record with the smallest summed value as the cluster $centroid$;
13 $c_i = \{centroid\};$
14 $T = T - \{centroid\};$
15 **for** $each\ record\ in\ len(T)$ **do**
16 Calculate the distance between $record$ and $centroid$ according to Equation (6) and save it in the column corresponding to the i-th row in $dist$;
17 Select $k-1$ unallocated records $\left\{\bigcup_{i=1}^{k-1} r_i\right\}$ corresponding to the smallest value in $dist(i,:)$;
18 $c_i = c_i \cup \left\{\bigcup_{i=1}^{k-1} r_i\right\};$
19 $T = T - \left\{\bigcup_{i=1}^{k-1} r_i\right\};$
20 $result = result \cup \{c_i\};$
21 **while** $len(T) \neq 0$ **do**
22 Get a record r from T, find the closest cluster c_j from r to the centroid of all clusters in the $dist$ matrix;
23 $T = T - \{r\};$
24 $c_j = c_j \cup r;$
25 Update $result$;
26 return $result$;

The time complexity analysis is conducted according to the steps of Algorithm 1, a pesudo brief code of the KACPC. To begin with, Step 3 is a nested loop of which the outer loop is mainly used to divide the data set, so that the number of executions of the outer loop is calculated as the number of clusters divided by data set m, where $m = \lfloor n/k \rfloor$. The selection of the initial cluster centroid is described by Steps 6 to 9, which are performed only once,

and this step requires the execution of the distance evaluation operation up to $n-1$ times and that of the distance comparison operation up to $n-2$ times, which can be completed in a period of $O(n)$. Step 11 is for the selection of the subsequent cluster centroids, which needs to consider the previously generated clusters. With the development of clustering, the number of clusters to be considered is increasing. As the outer loop iteration proceeds, the number of operations performed in Step 11 increases from $(n-k)$ to $(\lfloor n/k \rfloor - 1) \times (n - (\lfloor n/k \rfloor - 1)\, k)$. It can be calculated that the time complexity of Step 11 is $O\left(n^2\right)$ (see Equation (7)). Afterwards, the inner loop of Step 15 is mainly used to calculate the distance between records and the current cluster centroid for subsequent clustering. The upper limit of the number of operations of Step 15 is the number of records in the data set except for the cluster centroid, that is, $n-1$. As the outer loop iteration proceeds, the number of operations performed in Step 15 decreases from $(n-1)$ to $(n-1-(\lfloor n/k \rfloor - 1)\, k)$. It is easy to see from Equation (8) that step 15 can be completed in $O\left(n^2\right)$ time. The time complexity of Step 17 in algorithm flow is $O\left(n^2\right)$ which can be inferred based on Equation (9), and the loop body of Step 21 can be completed in a period of $O(n)$. In summary, the total time complexity of the KACPC is $O\left(n^2\right)$.

$$T(select_{centroid}) = (n-k) + 2 \times (n-2k) + \cdots + (\lfloor n/k \rfloor - 1) \times (n - (\lfloor n/k \rfloor - 1)k) \quad (7)$$

$$T(distance) = (n-1) + (n-1-k) + \cdots + (n-1-(\lfloor n/k \rfloor - 1)k) \quad (8)$$

$$T(select_{record}) = (n-1) + (n-2) + \cdots + (n - \lfloor n/k \rfloor\, k) \quad (9)$$

5 Experiment and Analysis

In this section we will explain a comprehensive experimental study that we have conducted, in which we compare several algorithms that we have introduced before with the KACPC and analyze the results. The algorithms that we have used for comparison are listed below:

- The MDAV-generic algorithm is a universal variant of the MDAV method, the latter of which is developed from the multivariate fixed-size microaggregation [15]. This algorithm can act on any type of data attributes and generate reproducible results. It is therefore regarded as one of the most representative k-anonymous algorithms.
- IKAC uses generalization to implement k-anonymity based on the idea of clustering. IKAC gathers k records at a time, though other algorithms [1, 19] may also gather multiple records at a time. This algorithm also considers the location information of the previously generated clusters when selecting the cluster centroid. The feature of IKAC is randomness, and each iteration can only determine the location of the centroid of one cluster.
- OKA also uses generalization to achieve k-anonymity based on the idea of clustering. The clustering method adopted by OKA is to obtain all the cluster centroids at one time, and then add records to the corresponding clusters. A similar clustering method was also used by another study [20], but the number of clusters may be reduced during the cluster merging stage [20].

- GAA-CP is an anonymous algorithm based on greedy clustering and partitioning [1]. This algorithm gathers multiple records at once and divides them into equivalence classes, and the anonymous results are made random.

5.1 Experimental Setup

The experimental setup is performed on a personal computer equipped with Intel(R) Core (TM) i7-8550U CPU @1.80GHz 1.99GHz processor, 8.0GB RAM, and environment variables in Windows 10, while all the algorithms are executed in MATLAB R2016a.

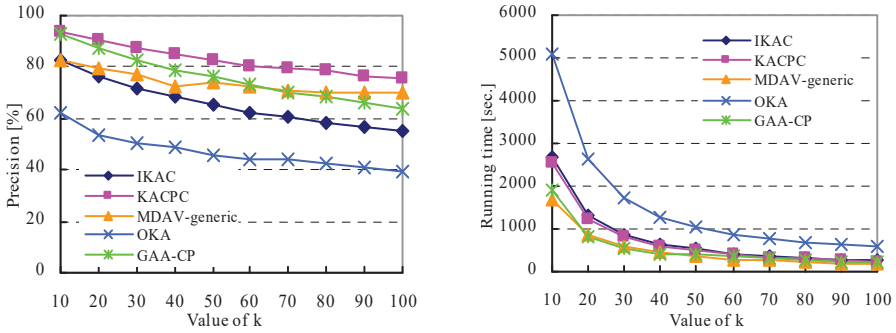
The Adult dataset from the UCI machine learning database is adopted as a benchmark for evaluating k-anonymity performance in this field. Before the start of experiment, the dataset is preprocessed to delete records with missing values. The preprocessed dataset contains a total of 30162 records that covers only 8 attributes. Each record contains a set of sensitive attributes and a set of QI attributes, namely, $\{Occupation, Salary\ class\}$ and $\{Age, workclass, Education, Marital\_status, Race, Sex\}$, respectively. Except for Age, which is a numeric attribute, all other attributes are categorical. Moreover, since IKAC, OKA and GAA-CP all randomly select the initial cluster centroid, their anonymous results of data cannot be reproduced and have randomness. In order to reduce the experimental deviation, each group of the experiments will be run independently for 10 times, and the average value is taken as the statistical result of the experiment.

5.2 Comparison & Analysis

In order to study the features of KACPC of data utility, execution efficiency and anonymity, we have designed three types of experiments to compare between the KACPC, the MDAV-generic, IKAC, OKA and GAA-CP algorithms. For the purpose of unifying the evaluation criteria and measuring the data utility better, we use Equation (2) to evaluate data utility, since the precision of the generated anonymous results can reflect the data utility of the algorithm. In terms of anonymity, once the specific quasi-identifier and the parameter k suitable for the disclosure scenario are determined, there is no need to worry about the disclosure risk anymore. This is the advantage of using k-anonymity.

5.2.1 Effect of k

First, we fix the data size n and the number of attributes in the data set and modify the values of k for experimentation. We have studied the effect of the k value on the compared algorithms, based on the Adult dataset with a dimensionality $d = 6$. Figure 3 presents our results, with the range of k from 10 to 100. These results reconfirm the superiority of our algorithm in retaining data utility. It is not difficult to observe from Figure 3(a) that, except for the MDAV-generic algorithm, the data precision of the other algorithms all show a downward trend as the value of k increases. This is because an increasing value of k will lead to an increase in the number of records in the equivalence class. The more records that are generalized, the higher the degree of generalization, the higher the amount of information loss, and the lower the overall data precision will be. Remarkably, it can be seen from Figure 3(a) that our method has higher data precision than other algorithms. The precision divergence between our method and IKAC is broadened as k increases (with an improvement by up to 20.7%). Additionally, the precision divergence between our method and OKA is relatively stable as k grows (improved by 30%−40%). In terms of runtime, these methods show a consistent downward trend as k grows (see Figure 3(b)). The divergence

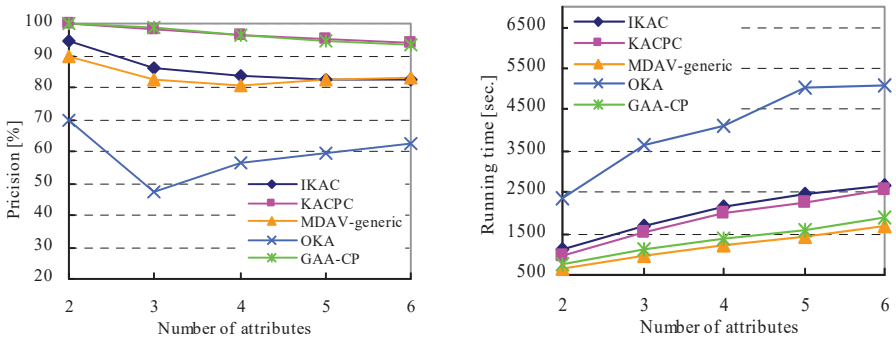


(a) Precision $vs.k$ (b) Runtime $vs.k$

Figure 3: Effect of k

in runtime of these methods gradually shrinks as k increases. Obviously, since the total number of records is fixed, the increase in the value of k will cause the number of the divided equivalence classes to decrease. Thus, the runtime of the algorithm will show a downward trend.

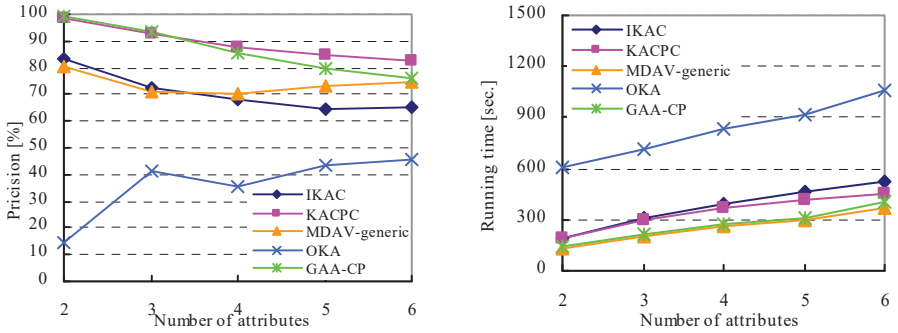
5.2.2 Effect of dimensionality of attributes



(a) Precision $vs.d$ (b) Runtime $vs.d$

Figure 4: Effect of dimensionality ($k = 10$)

Next, we study the effect of the dimensionality on each of the competing methods. We fix the value of k and the size of the data set and change the dimensionality of the attributes. We adopt the Adult dataset, and examine the performance of our method as a function of the dimensionality of selected attributes d, letting d range from 2 to 6, and setting $k = 10$, $k = 50$ and $n = 30162$. Figure 4 and Figure 5 show the experimental results. We observe that, KACPC, IKAC and GAA-CP methods show a consistent downward trend as attribute dimensionality d grows, while the MDAV-generic and the OKA methods exhibit unstable behavior (see Figure 4(a) and Figure 5(a)). Owing to the differences in the clustering mechanisms used in the algorithms as well as in the amount of information loss caused by the numerical attribute and the categorical attribute, data precision would fluctuate along with

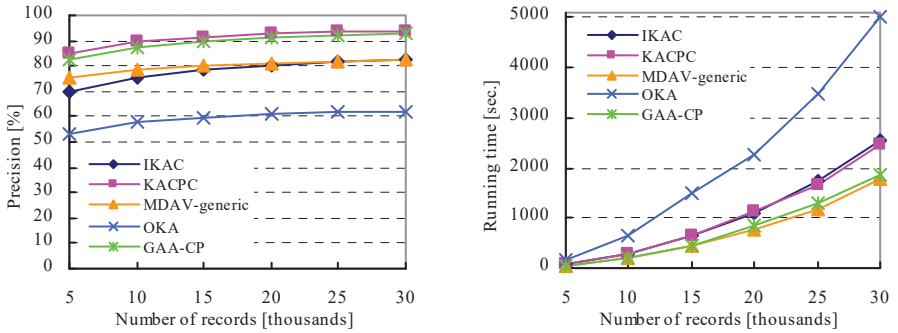


(a) Precision $vs.d$ (b) Runtime $vs.d$

Figure 5: Effect of dimensionality ($k = 50$)

the increase in the number of attributes. Obviously, our method is still the best at retaining data precision compared with the methods in previous work. In terms of runtime, these methods show a consistent upward trend as dimensionality d grows (see Figure 4(b) and Figure 5(b)). The reason for this phenomenon is that if the size of data and the value of k are fixed, the computational overhead of the algorithm will increase along with the increase in the number of attributes, thereby reducing the overall efficiency of the algorithm. In general, the execution efficiency of OKA is the worst among these methods, while the execution efficiency of the MDAV-generic algorithm is the highest. Additionally, our method show better performance than IKAC in execution efficiency.

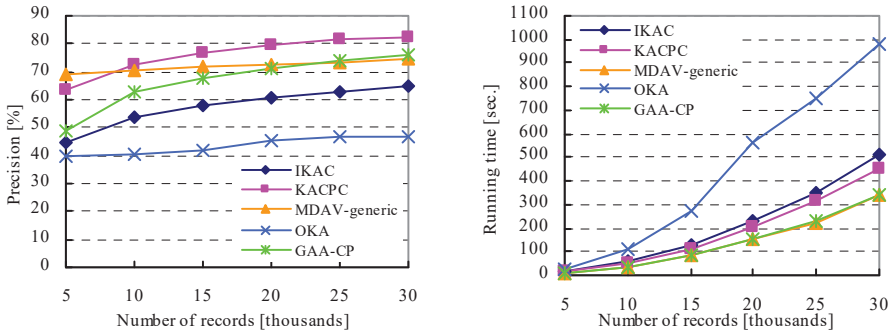
5.2.3 Effect of size



(a) Precision $vs.n$ (b) Runtime $vs.n$

Figure 6: Effect of data set size n ($k = 10$)

Last, we research the expansibility of the compared methods as the data set size grows. we obtain data sets of linearly increasing size, ranging from $5k$ to $30k$ records, from the Adult dataset, with full dimensionality $d = 6$. We present precision, runtime in Figure 6 and Figure 7, for k values set at $k = 10$, $k = 50$. Remarkably, the precision of the data generated by our method is still the best among these methods. The precision divergence between



(a) Precision $vs.n$ (b) Runtime $vs.n$

Figure 7: Effect of data set size n ($k = 50$)

MDAV-generic and IKAC gradually shrinks as the data set size grows. On the whole, the data precision of these algorithms basically shows an upward trend as the amount of data increases (see Figure 6(a) and Figure 7(a)). The reason for this phenomenon is that increasing the amount of data would also increase the probability of having more similar records, which reduces the degree of generalization of equivalence classes. Thus, the overall data precision will be improved to varying degrees. In terms of runtime, the runtime results present a uniform pattern. All methods show a consistent upward trend as data set size grows (see Figure 6(b) and Figure 7(b)). The reason for this phenomenon is that as the amount of data increases, the amount of computation of the algorithm also increases. The runtime divergence between these methods is widened as data set size grows. Similarly, the execution efficiency of OKA is the worst among these methods, while the execution efficiency of the MDAV-generic algorithm is the highest.

6 Conclusion

This study has explored the characteristics of distance in clustering for different types of data(numerical, categorical), and has proposed a new measurement formula and definition of distance in the clustering process. With efforts to promote privacy protection in data publishing, a novel algorithm is established, which can achieve k-anonymity and simultaneously reduce information loss for different types of data. Furthermore, in consideration of the quality of clustering, the proposed KACPC algorithm optimizes the selection of the cluster centroid to eliminate the randomness of the generated anonymous results. In addition, the time complexity of the proposed KACPC algorithm is also analyzed. Moreover, multiple sets of experiments have been carried out to analyze the KACPC algorithm and compare it with four previous algorithms [1,7–9]. The experimental results show that, in general, the MDAV-generic algorithm is the best among all the algorithms in terms of execution efficiency, while OKA is the worst. And the KACPC algorithm has better performance than the IKAC algorithm in execution efficiency. More importantly, the KACPC algorithm has the best performance than the other four algorithms in data precision on the Adult dataset. In other words, compared with the other four k-anonymity algorithms based on clustering, the KACPC algorithm can greatly improve the utility of data, reduce the amount of information loss, and provide more information for subsequent data mining

and processing needs while ensuring privacy requirements.

In the future, we will continue working on improving the efficiency of the anonymity mechanism while maintaining the utility of anonymized data. Meanwhile, we will also reconsider the proposed method with respect to the personalization of k-anonymity and the ability of k-anonymity algorithms based on clustering to deal with high-dimensional data.

Acknowledgements

The work in this study has been supported in part by National Key Research and Development Project (No. 2019QY(Y)0601).

References

[1] JIANG, H.W., ZENG, G.S., MA, H.Y. (2017). Clustering-anonymity method for privacy preservation of table data-publishing. *Journal of Software* 28:2, 341–351.

[2] SAMARATI, P., SWEENEY, L. (1998). Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression. *Technical report*, SRI International.

[3] SWEENEY, L. (2002). K-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10:5, 557–570.

[4] AGGARWAL, G., FEDER, T., KENTHAPADI, K., MOTWANI, R., PANIGRAHY, R., THOMAS, D., ZHU, A. (2008). Anonymizing tables. In *Eiter, T., Libkin, L., eds., Database Theory, vol. 3363 of Lecture Notes in Computer Science*, pages 246–258. Springer.

[5] SWEENEY, L. (2002). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10:5, 571–588.

[6] AGGARWAL, G., FEDER, T., KENTHAPADI, K., KHULLER, S., PANIGRAHY, R., THOMAS, D., ZHU, A. (2006). Achieving anonymity via clustering. In *Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'06*, pages 153–162. ACM.

[7] ZHENG, W., WANG, Z., LV, T., MA, Y., JIA, C. (2018). K-anonymity algorithm based on improved clustering. In *Vaidya, J., Li, J., eds., International Conference on Algorithms and Architectures for Parallel Processing - ICA3PP 2018, vol. 11335 of Lecture Notes in Computer Science*, pages 462–476. Springer.

[8] LIN, J.L., WEI, M.C. (2008). An efficient clustering method for k-anonymization. In *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society, PAIS'08*, pages 46–50. ACM.

[9] DOMINGO-FERRER, J., TORRA, V. (2005). Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery* 11:2, 195–212.

[10] WANG, K., YU, P.S., CHAKRABORTY, S. (2004). Bottom-up generalization: A data mining solution to privacy protection. In *Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM'04*, pages 249–256. IEEE.

[11] BAYARDO, R.J., AGRAWAL, R. (2005). Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering, ICDE'05*, pages 217–228. IEEE.

[12] BABU, K.S., REDDY, N., KUMAR, N., ELLIOT, M., JENA, S.K. (2013). Achieving k-anonymity using improved greedy heuristics for very large relational databases. *Transactions on Data Privacy* 6:1, 1–17.

[13] LIANG, X., GUO, Y., GUO, Y. (2020). A global optimal model for protecting privacy. *Wireless Personal Communications* 112:3, 1451–1478.

[14] HAN, J.M., CEN, T.T., YU, H.Q. (2008). Research in microaggregation algorithms for k-anonymization. *Acta Electronica Sinica* 36:10, 2021–2029.

[15] DOMINGO-FERRER, J., MATEO-SANZ, J.M. (2002). Practical data-oriented microaggregation for statisti-cal disclosure control. *IEEE Transactions on Knowledge and Data Engineering* 14:1, 189–201.

[16] TORRA, V. (2004). Microaggregation for categorical variables: A median based approach. In *Domingo-Ferrer, J., Torra, V., eds., International Workshop on Privacy in Statistical Databases - PSD 2004, vol. 3050 of Lecture Notes in Computer Science*, pages 162–174. Springer.

[17] HUNDEPOOL, A., DE WETERING, A.V., RAMASWAMY, R., FRANCONI, L., GAPO-BIANCHI, A., DE WOLF, P.-P., DOMINGO-FERRER, J., TORRA, V., BRAND, R., GIESSING, S. (2003). μ-ARGUS version 3.2 software and user's manual. *Statistics Netherlands, Voorburg Netherlands*. http://neon.vb.cbs.nl/casc.

[18] RODRÍGUEZ-HOYOS, A., ESTRADA-JIMÉNEZ, J., REBOLLO-MONEDERO, D., MEZHER, A.M., PARRA-ARNAU, J., FORNÉ, J. (2020). The fast maximum distance to average vector (F-MDAV): An algorithm for K-anonymous microaggregation in big data. *Engineering Applications of Artificial Intelligence* 90.

[19] LI, J., WONG, R.C.W., FU, A.W.C., PEI, J. (2006). Achieving k-anonymity by clustering in attribute hierarchical structures. In *Tjoa, A.M., Trujillo, J., eds., International Conference on Data Warehousing and Knowledge Discovery - DaWaK 2006, vol. 4081 of Lecture Notes in Computer Science*, pages 405–416. Springer.

[20] CHIU, C.C., TSAI, C.Y. (2007). A k-anonymity clustering method for effective data privacy preservation. In *Alhajj, R., Gao, H., Li, J., Li, X., Zaïane, O.R., eds., International Conference on Advanced Data Mining and Applications - ADMA 2007, vol. 4632 of Lecture Notes in Computer Science*, pages 89–99. Springer.

[21] BHALADHARE, P.R., JINWALA, D.C. (2016). Novel approaches for privacy preserving data mining in k-anonymity model. *Journal of Information Science and Engineering* 32:1, 63–78.

[22] BYUN, J.W., KAMRA, A., BERTINO, E., LI, N. (2007). Efficient k-anonymization using clustering techniques. In *Kotagiri, R., Krishna, P.R., Mohania, M., Nantajeewarawat, E., eds., International Conference on Database Systems for Advanced Applications - DASFAA 2007, vol. 4443 of Lecture Notes in Computer Science*, pages 188–200. Springer.

[23] ARAVA, K., LINGAMGUNTA, S. (2020). Adaptive k-anonymity approach for privacy preserving in cloud. *Arabian Journal for Science and Engineering* 45:4, 2425–2432.