Clustering Based K-anonymity Algorithm for Privacy Preservation

Sang Ni[1], Mengbo Xie[1], Quan Qian[1,2]
(Corresponding author: Quan Qian)

School of Computer Engineering & Science, Shanghai University[1]
Materials Genome Institute of Shanghai University[2]
No. 99, Shangda Road, Shanghai, China
(Email: qqian@shu.edu.cn)

(Received July 31, 2016; revised and accepted Nov. 5, 2016 & Jan. 15, 2017)

Abstract

K-anonymity is an effective model for protecting privacy while publishing data, which can be implemented by different ways. Among them, local generalization are popular because of its low information loss. But such algorithms are generally computation expensive making it difficult to perform well in the case of large amount of data. In order to solve this problem, this paper proposes a clustering based K-anonymity algorithm and optimizes it with parallelization. The experimental result shows that the algorithm performs better in information loss and performance compared with the existing KACA and Incognito algorithms.

Keywords: Clustering based K-anonymity; Information Loss; Privacy Preservation

1 Introduction

Along with the development of computer network, distributed computing, data mining and big data, huge amounts of data can be collected and analyzed efficiently. But when we explore the potential value of large amounts of data, privacy and privacy protection would be the focusing point. According to Manish Sharma [21], the secondary use of data is a source of privacy disclosure, which is the use of data for some purpose other than the purpose for which the data was collected initially. Jisha also noted that privacy is being violated mainly through three types of attack, such as linking attack, homogeneity attack and background knowledge attack [16]. Therefore, it is a very important issue to pay equal attention to data secondary using, data misusing, data mining and privacy preservation.

Privacy preservation is tightly associated with database security. Database security is usually achieved by means of access control, security management and database encryption [24]. Access control is a selective policy for restricting unauthorized users to access a resource through the user permissions. Security management refers to what kind of security management mechanisms are used to distribute database management authorities. Centralized control and decentralized control are 2 typical modes. Database encryption mainly includes three aspects: record encryption, database structure encryption and hardware encryption. These measures can protect the security of the database to a certain extent, for example, the direct disclosure of sensitive information such as, identification card number, home address, health information etc. But they are unable to prevent those indirect accesses to private data through federation reasoning. In [5], it shows that through joining voter registration table and medical information table (individual identification is hidden), by attributes of Zip code, Sex, Date of Birth, etc., more than 85% of American citizens can be uniquely identified. In addition, encryption and access control, to some extent, limits the sharing of data.

For such reasons, data anonymity is an effective means to achieve privacy preservation. The basic idea is to transform some part of the original data, for instance, through generalization, compression, etc., and let the transformed data cannot be combined with other information to reason about any personal privacy information. Specifically, the implementation of privacy preservation mainly concentrates on two aspects: (1) How to ensure that the data been used without privacy disclosure? (2) How to make the data to be better utilized? Therefore, a better trade off between privacy preservation and data utilization is a problem that the academia and industry need to be solved urgently.

K-anonymity was first proposed in 1998 by Sweeney et al. [23]. K-anonymity depends on anonymizing the original data set to satisfy the anonymization requirements, which can be used for data publishing. The common anonymization techniques are generalization and hidden. The basic idea of K-anonymity is anonymizing the publishing data to meet the requirement that at least K

tuples cannot be distinguished by each other. Namely, for each tuple there exists at least K tuples with equal value of quasi-identifiers. Researchers have proved that the complexity of K-anonymity is NP-hard [20].

Currently, there are many algorithms to implement K-anonymity [17]. From the point of generalization, can be categorized as recoding mode (global recoding, local recoding), data grouping strategy (classification, clustering, and Apriori algorithm). From the perspective of the data characteristics, they are static data set and dynamic data set (incremental data, stream data, and uncertain data).

The rest of the paper is organized as follows: Sections 2 and 3 discuss the related work and some prerequisite knowledge of k-anonymity. Our main contributions, the clustering based k-anonymity algorithm *GCCG* and its parallel optimization are described in Sections 4 and 5. The experimental results are shown in Section 6. Section 7 provides some final conclusions and directions for future work.

2 Related Work

Privacy preservation was firstly concerned in the field of statistics and then extended to various areas. The main research directions are as shown in Table 1.

Table 1: Main research directions of privacy preservation [17]

Research direction	Relevant techniques
General privacy protection technique	Data perturbation, randomization, data exchange, data encryption, etc.
Privacy protection technology for data mining	association rules mining, classification, clustering etc.
Privacy preservation based data publishing principles	K-anonymity, M-invariant, T-closeness, etc.

2.1 Different Kinds Of Anonymity Algorithms

Anonymity methods mainly include generalization, classification and clustering etc. [17].

Generalization and suppression based anonymity. The main idea of generalization based anonymity is to increase equivalence class size of the table by reducing the data precision of the quasi-identifier attributes. Generally, quasi-identifier can be divided into two kinds: numeric attribute and category attribute. Numeric attributes are usually generalized to interval, for instance, the age 16 can be generalized to interval $[10-20]$. And for category attributes, the original concrete values will be replaced by more general ones, according to a priori established VGH(value generalization hierarchies). For example, a nationality attribute whose value is "China", then it can be generalized to "Asia". Suppression can be viewed as an extreme form of generalization, in which the generalized attributes cannot be further generalized [6]. Kameya et al. proposed a cell-suppression based k-anonymization method which aims to preserve the MAR(Missing at random) condition uses the Kullback-Leibler (KL) divergence as a utility measure [28]. Besides, He et al. proposed a novel linking-based anonymity model, which can resist the attack incurred by homogeneous generalization [7].

Dividing and grouping based anonymity. According to the different ways of dividing, it can be divided into micro aggregation, condensation, anatomy and permutation, etc.

- Micro aggregation divides each class according to the data similarity, making the tuple size of a class at least k, and then using the class centroid to generalize all the data of a class. Mortazavi et al. proposed a Fast Data-oriented micro aggregation algorithm (FDM)in [18] that efficiently anonymizes large multivariate numerical datasets for multiple successive values of k and proposed a disclosure-aware aggregation model in [19], where published values are computed in a given distance from the original ones to obtain a more protected and useful published dataset.
- Condensation[1, 2, 3] is a new kind of method similar to micro aggregation, which was proposed by Aggarwal et al. in 2004. The basic idea is to divide the original data into different groups, and then process each group with condensation technique. Condensated data will be reconstructed by general reconstruction algorithm, which would not reveal any privacy information of the original tuples.
- Anatomy method was firstly put forward by Xiao et al. [25]. It publishes the sensitive attributes and quasi-identifier attributes separately to reduce the correlation degree between them. Permutation depends on disturbing the order of sensitive attributes after grouping to reduce the correlation between quasi-identifiers and numeric sensitive attributes. In [30], Yu et at. proposed a novel anonymization method based on anatomy and reconstruction in LBS privacy preservation.

Clustering based anonymity. Anonymity can also be implemented by clustering, which is the most commonly used method. The basic idea is to produce at

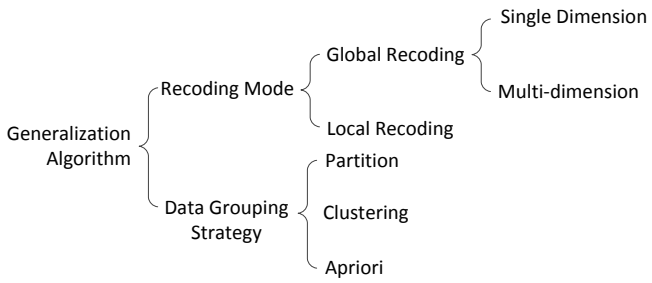

Figure 1: Generalization based anonymity algorithm [5]

least k records of a class as the equivalence class. The tuples in a same class need to be as similar as possible to make the information loss minimum after generalization. Liu et al. proposed a privacy-preserving data publishing method, namely MNSACM, which uses the ideas of clustering and Multi-Sensitive Bucketization (MSB) to publish microdata with multiple numerical sensitive attributes [14]. Bhaladhare et al. proposed two approaches for minimizing the disclosure risk and preserving the privacy by using systematic clustering algorithm [4]. Xin et al. proposed a trajectory privacy preserving method based on the adaptive clustering, and designed a 2-stage clustering method for trajectory k-anonymity [26].

2.2 Generalization Based Anonymity

Currently, there are many algorithms to implement k-anonymity, and most of them use the generalization and suppression as shown in Figure 1.

From the perspective of generalization methods, it can be divided into 2 categories: global generalization and local generalization.

Global generalization. This kind of algorithms allows the whole domain of identifier attributes mapped to a generalization domain, that is to say, a value in a table would only have one generalization value. In general, global generalization algorithm is simple and efficient. But it needs to set generalization level in advance and has problems of over generalization causing high information loss. Representative global generalization algorithms include: *u-Argus* algorithm [25], Datafly [22], Incognito [10], etc.

u-Argus algorithm proposed by De.Wall et al. [8], which the published data includes all tuples and all properties of the initial data, except very few data will be lost. But if there exists many attribute combinations, it can not provide enough protection for released data. Datafly proposed by Sweeney uses suppression and heuristic generalization, which is efficient but has much distortion. LeFevre proposed Incognito, which adopts the Generalization Graph for data generalization with the bottom-up approach. It prunes redundant branches of the graph to narrow the search scope. However, the information distortion of Incognito is relatively high.

Local generalization. This kind of algorithm usually maps attribute value to generalization value based on the grouping , that is to say, even the same attribute values can be generalized to different values if they are in the different groups. Grouping data usually adopts some heuristic principles, such as division, clustering and so on. Information loss of this kind of algorithm is less than the global generalization algorithms, but its complexity usually is higher. In the case of a large amount of data, the performance is a problem to be concerned. Representative algorithms are: GA [9], Mondrian Multidimensional algorithm [11], $KACA$ algorithm [12] as follows.

Iyengar proposed GA (Genetic algorithm), which can meet the requirements of K-anonymity, but when processing large amount of data, it will spend a few hours. Mondrian multidimensional algorithm proposed by $Le.Fevre$ can partition continuous attributes but not for discrete attributes. Li put forward $KACA$ algorithm, through merging the nearest equivalence class to form a bigger cluster. Although, $KACA$ has low information loss, the performance is poor because of massive distance computations.

To sum up, it concludes that, generally, the global generalization algorithms are efficient, but the information quality is low. On the contrary, local generalization algorithms can greatly improve the information quality, but it is often inefficient. So, in this paper our motivation is to propose an efficient local generalization algorithm which has great information quality and great performance simultaneously.

3 Prerequisite Knowledge

To use K-anonymity, supposing the original data are stored in database with the form of structured table. We assume that data publishers have the raw data table T, each row in the table is corresponding to a specific entity, such as student id, name, gender, birth place, etc. As shown in Table 2, each row in the table is so called a tuple.

Definition 1. *Quasi-identifier. Quasi-identifier is some form of attributes combination, which can determine some individuals in table T by joining some external information.*

Theoretically, in a table, all attributes except identifiers, can be quasi-identifiers.

Definition 2. *Sensitive identifier. Sensitive identifiers are those attributes concerning sensitive privacy information, such as salary, health information, etc.*

Table 2: An example of data to be anonymized

Education	Workclass	Race	Sex	Age
Bachelors	State-gov	White	Male	39
Bachelors	Self-emp-not-inc	White	Male	50
HS-grad	Private	White	Male	38
11th	Private	Black	Male	53
Bachelors	Private	Black	Female	28
Masters	Private	White	Female	37
9th	Private	Black	Female	49
HS-grad	Self-emp-not-inc	White	Male	52

Since attribute sensitivity is context-dependent, so it is not invariable and should be configured manually according to actual situations. In this paper, taking Table 2 as an example, we set attribute *workclass* as a sensitive identifier.

Definition 3. *Equivalence class. The Equivalence class of table T on attribute $A_i, \cdots, A_j$ is the set of tuples that all values of these attributes are identical.*

For example, in Table 1, the top 2 rows: $\{Bachelors, State-gov, White, Male, 39\}$ and $\{Bachelors, Self-emp-not-inc, White, Male, 50\}$ are the same equivalence class on attribute $\{Sex, Education, Race\}$.

Definition 4. *K-anonymity Property. Generate several equivalence classes on quasi-identifiers. If each size of the equivalence class is no less than K, we can say the equivalence class partition has k-anonymity property.*

That is to say, according to the quasi-identifiers, each record has at least $(k-1)$ other same records to make them unable to be identifier by each other. Therefore, we can say the Table 2 is a 2-anonymity result of Table 1.

K-anonymity adopts generalization and suppression to preserve privacy. So there will be a certain degree of information loss inevitably. In order to describe quantitatively, it is necessary to introduce a corresponding measurement for information loss. There are many different measurement models for information loss, including Prec, DM, NE, etc. In this paper we use the measurement method in [27]. In order to compare the result under different amount of data, the sum of the original formula is modified as the mean value. Such a change won't change the relationship of size between results.

Definition 5. *Information loss. Supposing that a numeric attribute in a tuple, the original value x is generalized to $[x_{min}, x_{max}]$, where x_{min} is the minimum of the equivalence class and x_{max} is the maximum of the equivalence class. Max and Min is the maximum and minimum value of the attribute in the whole domain. Then the information loss (IL) of the tuple on the numeric attribute is defined as Equation (1) .*

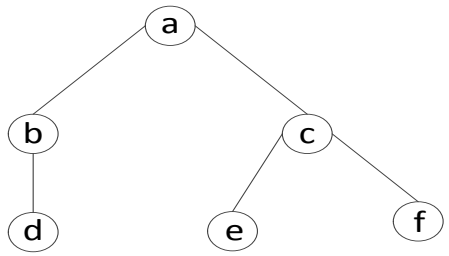

Figure 2: An classification tree example for a category attribute

$$IL = \frac{x_{max} - x_{min}}{Max - Min}. \quad (1)$$

For a category attribute, we usually need to build a classification tree at first. As shown in Figure 2, it is a classification tree example for a category attribute. Supposing that the value of a tuple is generalized from e to c. Then the information loss of the tuple on the attribute is defined as Equation (2).

$$IL = \frac{size(c)}{Size}. \quad (2)$$

"size(c)" is the number of its descendant leaf nodes and $Size$ is the number of all leaf nodes. Therefore, in Figure 2, the information loss is 2/3, in that the total number of leaf nodes is 3, and node c's descendant leaf is 2. For all the attributes of a tuple, its information loss is defined as Equation (3), where m is the number of all attributes.

$$IL = \frac{\sum_{i=1}^{m} IL_i}{m}. \quad (3)$$

Finally, the average IL of all the tuples is the information loss of the whole data set after generalization.

4 GCCG: Clustering Based K-anonymity

The key point of K-anonymity is to produce a number of equivalence classes whose size is at least k and each equivalence class has the same form on quasi-identifiers. This idea is very similar to clustering. Each equivalence class can be regarded as a cluster, and at the same time, the centroid of a cluster can be seem as a generalization form of a equivalence class. Next, we will take the data set in Table 2 as a 2-anonymity example to explain the GCCG algorithm in detail.

Table 3: An example of anonymized data after k-anonymity (k=2)

Education	Workclass	Race	Sex	Age
Bachelors	State-gov	White	Male	[39 - 50]
Bachelors	Self-emp-not-inc	White	Male	[39 - 50]
HS-grad	Private	White	Male	[38 - 52]
HS-grad	Self-emp-not-inc	White	Male	[38 - 52]
Low	Private	Black	*	[49 - 53]
Low	Private	Black	*	[49 - 53]
High	Private	*	Female	[28 - 37]
High	Private	*	Female	[28 - 37]

4.1 Algorithm Overview

There are four main steps of our clustering based K-anonymity algorithm: *Grading*, *Centering*, *Clustering*, and *Generalization*, abbreviated by *GCCG*. The pseudo code of *GCCG* is as Algorithm 1, and we will explain each step in detail.

Algorithm 1 GCCG Algorithm

1: Input: Dataset D (with n records), Anonymity constant K, Classification tree for each attribute.
2: Output: D's k-anonymity result
3: Begin
4: **for** $i = 1$ to n **do**
5: Grade each tuple by the cluster centroid;
6: **end for**
7: Sort D by the centroid grading score;
8: **for** $i = 1$ to $n/k - 1$ **do**
9: Choose the first tuple to be the clustering centroid;
10: Choose the centroid and the nearest $(k-1)$ tuples to make up a new equivalence class;
11: Remove the k tuples from D;
12: **end for**
13: Make the rest tuples to be the last equivalence class;
14: **for** each equivalence class **do**
15: Generalize the tuples using the class centroid;
16: **end for**
17: End

4.2 Grading Tuples Using Cluster Centroid

Since the cluster centroid will be used for equivalence class generalization. So, the clustering quality will affect anonymity greatly. If the tuples in a cluster are much scattered, then we need a more general value to generalize them, thus resulting in much information loss. So in this paper, we propose an evaluation method to find an appropriate clustering centroid. The method can be divided into two kinds according to different attributes: category or numeric.

Table 4: Original data table

Education	Workclass	Race	Sex	Age
Bachelors	State-gov	White	Male	39
Bachelors	Self-emp-not-inc	White	Male	50
HS-grad	Private	White	Male	38
11th	Private	Black	Male	53
Bachelors	Private	Black	Female	28
Masters	Private	White	Female	37
9th	Private	Black	Female	49
HS-grad	Self-emp-not-inc	White	Male	52
Masters	Private	White	Female	31
Bachelors	Private	White	Male	42

For a category attribute, the score is the ratio of the count of the attribute value to the whole attribute values in dataset. Assuming that the ratio is P1, then P1 is regared as the attribute score of the tuple. That means more frequent a value, more possibility for it to be a center. For a numeric attribute, we take the proportion of the ratio of the value to K as the score. Finally, the sum of all attributes is the score of the tuple on center grading. Table 5 is result of sorted data set of Table 4 by the score.

4.3 Tuples Distance Definition And Calculation

After selecting the cluster centroid, distance computation among tuples is another key problem for clustering. Not as KACA (another typical clustering based generalization algorithm for k-anonymity), it uses iterative generalization and its efficiency is low [29]. So, in our algorithm, we just calculate the distances among tuples, which simplifies the distance calculation. In this paper, the distance

Table 5: Dataset sorted by score

Education	Workclass	Race	Sex	Age	Score
Bachelors	State-gov	White	Male	39	1.8
Bachelors	Self-emp-not-inc	White	Male	50	1.8
Bachelors	Private	White	Male	42	1.8
HS-grad	Private	White	Male	38	1.6
HS-grad	Self-emp-not-inc	White	Male	52	1.6
Masters	Private	White	Female	37	1.4
Masters	Private	White	Female	31	1.4
Bachelors	Private	Black	Female	28	1.2
11th	Private	Black	Male	53	1.1
9th	Private	Black	Female	49	0.9

between tuples is defined as follows:

For a numeric attribute A, supposing a_1, a_2 are the values of two tuples, then the distance on attribute A is:

$$dist = \frac{|a_1 - a_2|}{range}. \quad (4)$$

Where "range" is the difference between the maximum and minimum value on attribute A.

For a category attribute B, supposing b_1, b_2 are values of two tuples, then the distance on attribute B is:

$$dist = \frac{Parent\_depth(b_1, b_2)}{Depth}. \quad (5)$$

Where "Parent_depth(...)" is the subtree depth whose root is the nearest common ancestor of b_1 and b_2. "Depth" is the depth of the entire classification tree. For example, in Figure 2, the distance between e and f is 1/2.

According the distance definition above, Table 6 is the result after clustering. Each two rows is regarded as an equivalence class.

4.4 Generalization Procedure

Since *GCCG* algorithm is based on clustering, we generalize the tuples after being clustered. That is, use the cluster centroid to generalize them. Specific ways are as follows:

For a numeric attribute A, we use $[a_{min}, a_{max}]$ to generalize. a_{min} is the minimum value of A in the equivalence class and a_{max} is the maximum value. For a category attribute, we use the value of the nearest common ancestor in the classification tree to generalize. For both type of attributes, "*" is the most generic form, that is to say, all attribute information is removed. According to the method, the generalized result of Table 6 is shown as Table 7.

Table 6: Dataset afer clustering

Education	Workclass	Race	Sex	Age
Bachelors	State-gov	White	Male	39
Bachelors	Private	White	Male	42
Bachelors	Self-emp-not-inc	White	Male	50
HS-grad	Self-emp-not-inc	White	Male	52
HS-grad	Private	White	Male	38
Masters	Private	White	Female	37
Masters	Private	White	Female	31
Bachelors	Private	Black	Female	28
11th	Private	Black	Male	53
9th	Private	Black	Female	49

5 GCCG Parallel Optimization

To enhance the performance of *GCCG* algorithm, *GCCG* parallelization is necessary. Observing the whole algorithm, we can find that most of the operations are focusing on centroid selection and distance computation. So, if this part of operations can be parallelized, the performance of the whole algorithm will be improved.

In this paper, we use distances to divide the data set into a few small sub-datasets. That is, based on the original center selection method, the original data set is divided into n clusters (sub-datasets). Then all the sub-datasets do anonymity at the same time by GCCG with multithreading. During the clustering based data partition, we choose similar tuples into a same sub-dataset that reduces the information loss. Moreover, as the original times of distance computing is an arithmetic progression, after data partition, the distance computation

Table 7: Dataset after being generalized

Education	Workclass	Race	Sex	Age
Bachelors	State-gov	White	Male	[39 - 42]
Bachelors	Private	White	Male	[39 - 42]
*	Self-emp-not-inc	White	Male	[50 - 52]
*	Self-emp-not-inc	White	Male	[50 - 52]
*	Private	White	*	[37 - 38]
*	Private	White	*	[37 - 38]
High	Private	*	Female	[28 - 31]
High	Private	*	Female	[28 - 31]
Low	Private	Black	*	[49 - 53]
Low	Private	Black	*	[49 - 53]

Table 8: Result using parallel anonymity

Education	Workclass	Race	Sex	Age
Bachelors	State-gov	White	Male	[39 - 42]
Bachelors	Private	White	Male	[39 - 42]
*	Self-emp-not-inc	White	*	[37 - 50]
*	Private	White	*	[37 - 50]
*	Private	White	*	[37 - 50]
Senior high	Self-emp-not-inc	*	Male	[52 - 53]
Senior high	Private	*	Male	[52 - 53]
*	Private	*	Female	[28 - 49]
*	Private	*	Female	[28 - 49]
*	Private	*	Female	[28 - 49]

Algorithm 2 Dataset partition in parallel mode

```
Input: Dataset D (n records), anonymity constant k, parallel constant c;
Output: sub-dataset D'[c];
Begin
subsize=n/c;
while exists dataset's size > subsize do
  for each dataset with size more than subsize do
    Choose the first tuple to be the cluster center;
    Choose the nearest size/2-1 tuples to make up a sub-dataset with the new center;
    Make the rest tuples to be another sub-dataset;
  end for
end while
End
```

complexity can be reduced by the square of the number of sub-datasets. Table 8 is the result using 2-threads to do the anonymity for the data in Table 4.

6 Experimental Evaluation

The hardware used in the experiment is: Intel Xeon E5504 @ 2.00 GHz, 4G DDR3 Memory. Program implementation is Java7 and use Java7 Fork/Join multi-threading framework to do parallel programming [15]. Database: MySQL 5.6.18. The experimental data are Adult Database from UCI Machine Learning Repository [13]. After preprocessing, the dataset contains 30,661 tuples and 5 attributes. Table 9 provides a brief description of the dataset including 4 quasi- identifiers and 1 sensitive attribute.

6.1 Information Loss

The experiments in this paper are all implemented by JAVA under the same hardware environment. We choose Incognito and KACA as the algorithm to compare.

Figure 3 describes the information loss of the three algorithms when K changes from 3 to 10. It shows that the two local generalization algorithms have lower information loss than the global generalization algorithm. More-

Table 9: Adult dataset description

Attribute	Distinct values	Type	CTree Height
Age	72	Numeric	5
Sex	2	Category	2
Workclass	8	Sensitive	-
Race	5	Category	2
Education	16	Category	3

Table 10: Average size of equivalence classes

K	Incognito	KACA	GCCG
3	851.69	3.08	3.00
4	851.69	4.14	4.00
5	1533.05	5.12	5.00
6	1533.05	6.09	6.00
7	1533.05	7.24	7.00
8	1533.05	8.20	8.00
9	1533.05	9.22	9.00
10	1533.05	10.30	10.00

over, concerning the information loss, *GCCG* performs the best, about one third of the Incognito.

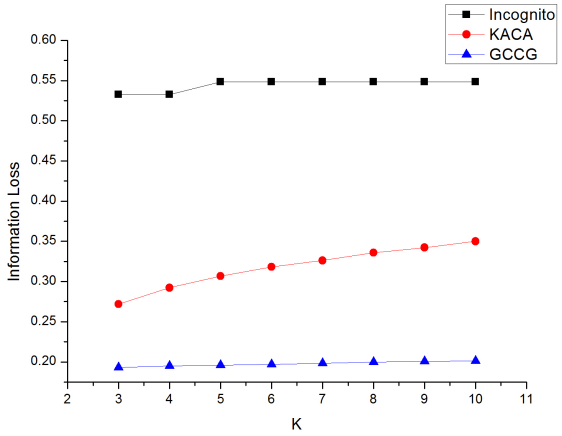

Figure 3: Information loss comparisons among different generalizations

Table 10 describes the average size of the equivalence classes of the three algorithms when K changes from 3 to 10. The result shows that the information loss is related to the size of equivalence class. When increasing the size of equivalence class, the information loss will increase appropriately. Therefore, controlling the size of the equivalences class is a effective way to control the information loss.

6.2 Execution Performance

Figure 4 describes the running time of the three algorithms when K changes from 3 to 10. It says that, as a global generalization algorithm, Incognito performs the best and $KACA$ consumes much time because of its frequent distance computation and cluster merging. $GCCG$, also belongs to local generalization algorithm, performs about 10 times faster than that of $KACA$ and is much close to Incognito.

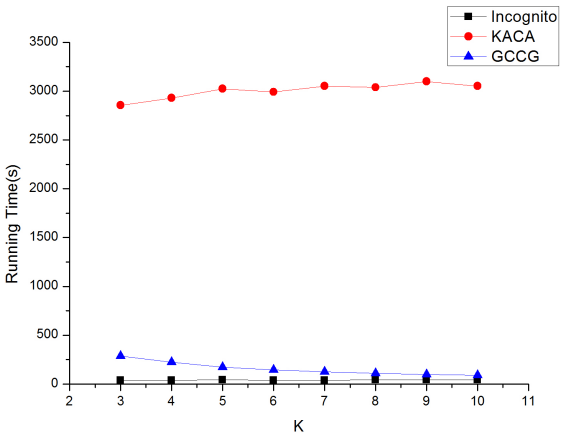

Figure 4: Execution performance comparison among different algorithms

6.3 Comparison Between Serial and Parallel Algorithm

Figure 5 shows the running time of serial $GCCG$ and parallel $GCCG$ when K changes from 3 to 10. From Figure 5, it shows that the performance of 2-threads parallelization improved about 3.5 times faster than that of the serial one, and 4-threads improved about 10.5 times. Moreover, be different from the serial algorithm, the running time of the parallel algorithm is not relevant to K, that is to say, the performance keeps relatively stable.

Figure 6 describes the information loss of serial $GCCG$ and parallel $GCCG$ when K changes from 3 to 10. From the result we can see that parallel $GCCG$ will have more information loss. But in the case of large amount of data, the growth is quite few.

7 Conclusions

In this paper, we proposed an clustering based local generalization algorithm $GCCG$ for K-anonymity. Experimental results show that $GCCG$ has lower information loss and better performance compared with the classical local generalization algorithms,$KACA$. More specifically, comparing with classical local generalization algorithm $KACA$, the information loss of $GCCG$ is about half of $KACA$, but with 10 times of performance improvement. Comparing with global generalization algo-

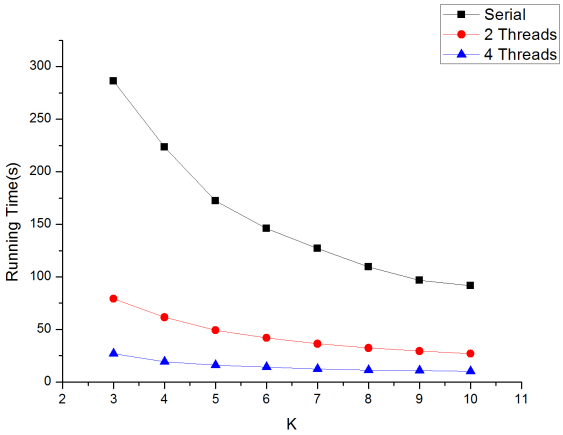

Figure 5: Performance comparison between serial and parallel *GCCG*

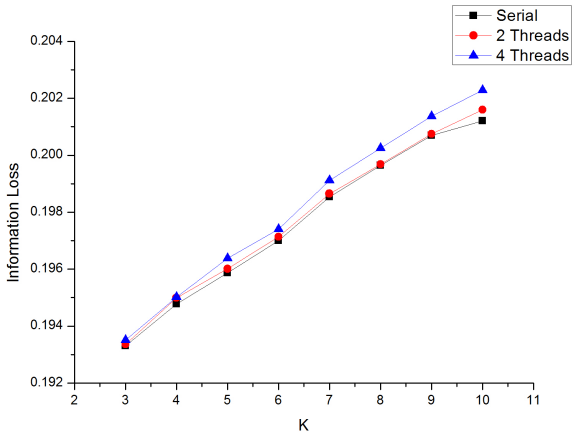

Figure 6: Information loss comparison between serial and parallel *GCCG*

rithm *Incognito*, the information loss of *GCCG* is about one third but with almost equal performance. Besides that, the parallel *GCCG* shows great performance improvement, when using 4-threads parallelization, there are 10 times acceleration with little information loss in the case of a large amount of data.

With the arrival of the era of big data, compared with the traditional data model, it is more likely to become the target of network attacks. Due to the system fault, hacker intrusion, internal leakage and other reasons, data leakage may occur at any time, resulting in unquantifiable losses. Therefore, aiming at big data oriented privacy protection issues deserves a enough attention with broad prospects.

Acknowledgments

This work is partially sponsored by National Key Research and Development Program of China(2016YFB0700504), Shanghai Municipal Science and Technology Commission(15DZ2260301), Natural Science Foundation of Shanghai(16ZR1411200). The authors gratefully appreciate the anonymous reviewers for their valuable comments.

References

[1] C. C. Aggarwal and P. S. Yu, "A condensation approach to privacy preserving data mining," in *Advances in Database Technology (EDBT'04)*, pp. 183–199, Heraklion, Crete, Greece, Mar. 2004.

[2] C. C. Aggarwal and P. S. Yu, "A framework for condensation-based anonymization of string data," *Data Mining & Knowledge Discovery*, vol. 16, no. 3, pp. 251–275, 2008.

[3] C. C. Aggarwal and P. S. Yu, "On static and dynamic methods for condensation-based privacy-preserving data mining," *ACM Transactions on Database Systems*, vol. 33, no. 1, pp. 41–79, 2008.

[4] P. R. Bhaladhare and D. C. Jinwala, "Novel approaches for privacy preserving data mining in k-anonymity model," *Journal of Information Science and Engineering*, vol. 32, no. 1, pp. 63–78, 2016.

[5] C. Clifton, M. Kantarcioglu, and J. Vaidya, "Defining privacy for data mining," in *National Science Foundation Workshop on Next Generation Data Mining*, pp. 126–133, Baltimore, MD, Nov. 2002.

[6] K. Dhivya and L. Prabhu, "Privacy preserving updates using generalization-based and suppression-based k-anonymity," *International Journal of Emerging Technology in Computer Science & Electronics*, vol. 8, no. 1, pp. 98–103, 2014.

[7] X. M. He, X. Y. Wang, D. Li, and Y. N. Hao, "Semi-homogenous generalization: Improving homogenous generalization for privacy preservation in cloud computing," *Journal of Computer Science and Technology*, vol. 31, no. 6, pp. 1124–1135, 2016.

[8] A. Hundepool and L. Willenborg, "Argus,software for statistical disclosure control," in *Proceedings of 13th Symposium on Computational Statistics*, pp. 341–345, Bristol, Great Britain, Aug. 1998.

[9] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pp. 279–288, Edmonton, AB, Canada, July 2002.

[10] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (Sigmod'05)*, pp. 49–60, Baltimore, Maryland, USA, June 2005.

[11] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *22nd International Conference on Data Engineering (ICDE'06)*, pp. 25, Atlanta, Georgia, USA, Apr. 2006.

[12] J. Y. Li, R. C. Wong, A. W. Fu, and J. Pei, "Achieving k-anonymity by clustering in attribute

hierarchical structures," in *8th International Conference on Data Warehousing and Knowledge Discovery*, pp. 405–416, Krakow, Poland, Sept. 2006.

[13] M. Lichman, *UCI Machine Learning Repository*, 2013. (`http://archive.ics.uci.edu/ml`)

[14] Q. H. Liu, H. Shen, and Y. p. Sang, "Privacy-preserving data publishing for multiple numerical sensitive attributes," *Tsinghua Science and Technology*, vol. 20, no. 3, pp. 246–254, 2015.

[15] K. L. Nitin and S. Sangeetha, *Java 7 Fork/Join Framework*, 2012. (`http://www.developer.com/java-7-forkjoin-framework.html`)

[16] J. J. Panackal, A. S. Pillai, and V. N. Krishnachandran, "Disclosure risk of individuals: a k-anonymity study on health care data related to indian population," in *International Conference on Data Science & Engineering (ICDSE'14)*, pp. 200–205, Kochi, India, Aug. 2014.

[17] X. M. Ren, "Research for privacy protection method based on k-anonymity(in chinese)," Master Thesis, Harbin Engineering University, 2012.

[18] M. Reza and J. Saeed, "Fast data-oriented microaggregation algorithm for large numerical datasets," *Knowledge-Based Systems*, vol. 67, pp. 195–205, 2014.

[19] M. Reza and J. Saeed, "Enhancing aggregation phase of microaggregation methods for interval disclosure risk minimization," *Data Mining and Knowledge Discovery*, vol. 30, no. 3, pp. 605–639, 2016.

[20] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, p. 188, Seattle, WA, USA, June 1998.

[21] M. Sharma, A. Chaudhary, M. Mathuria, and S. Chaudhary, "A review study on the privacy preserving data mining techniques and approaches," *International journal of computer science and telecommunications*, vol. 4, no. 9, pp. 42–46, 2013.

[22] L. Sweeney, "Computational disclosure control - a primer on data privacy protection," Ph.d Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, May 2001.

[23] P. F. Wu and Y. Q. Zhang, "K-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[24] P. F. Wu and Y. Q. Zhang, "Summary of database security," *Computer Engineering (in Chinese)*, vol. 32, no. 12, pp. 85–88, 2006.

[25] X. K. Xiao and Y. F. Tao, "Anatomy: Simple and effective privacy preservation," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06)*, pp. 139–150, Seoul, Korea, Sept. 2006.

[26] Y. Xin, Z. Q. Xie, and J. Yang, "The privacy preserving method for dynamic trajectory releasing based on adaptive clustering," *Information Sciences*, vol. 378, pp. 131–143, 2017.

[27] J. Xu, "Data anonymization based on the data availability(in chinese)," Master Thesis, Fudan University, Shanghai, China, June 2008.

[28] K. Yoshitaka and H. Kentaro, "Bottom-up cell suppression that preserves the missing-at-random condition," in *International Conference on Trust and Privacy in Digital Business*, pp. 65–78, Porto, Portugal, Sept. 2016.

[29] J. Yu, J. M. Han, and J. M. Chen, "Topdown-kaca: an efficient local-recoding algorithm for k-anonymity," in *The 2009 IEEE International Conference on Granular Computing (GrC'09)*, pp. 727–732, Nanchang, China, Aug. 2009.

[30] L. Yu, J. M. Han, Y. U. Juan, J. Jia, and H. B. Zhan, "A novel anonymization method based on anatomy and reconstruction in lbs privacy preservation," *Advances in Differential Equations*, vol. 19, no. 11, pp. 544–553, 2014.

Biography

Sang Ni is a master degree student in the school of computer science, Shanghai University. His research interests include cloud computing, big data analysis, computer and network security.

Mengbo Xie is a master degree student in the school of computer science, Shanghai University. His research interests include privacy protection, big data analysis, computer and network security.

Quan Qian is a Professor in Shanghai University, China. His main research interests concerns computer network and network security, especially in cloud computing, big data analysis and wide scale distributed network environments. He received his computer science Ph.D. degree from University of Science and Technology of China (USTC) in 2003 and conducted postdoc research in USTC from 2003 to 2005. After that, he joined Shanghai University and now he is the lab director of network and information security.