De-identification for Transaction Data Secure against Re-identification Risk Based on Payment Records

Satoshi Ito[1], Reo Harada[1], Hiroaki Kikuchi[1]

Meiji University Graduate School, Tokyo, 164-8525 Japan,
mmhm@meiji.ac.jp, kikn@meiji.ac.jp

Abstract. De-identification is a process used to prevent individuals from being identified using personal data, including personal identification information. In conventional de-identification studies, re-identification is a process used to identify individuals from *static* data where there is one record specified for each individual. In contrast, in this paper, we employ *dynamic* data, for example, trajectory data and online payment records. In particular, we consider the open competition data from the 2016 Privacy Workshop Cup (PWSCup) held in Japan consisting of purchasing history data. Throughout the analysis, we find that attackers can re-identify individuals with a high degree of accuracy from the de-identified purchase history data based on a feature of the set of goods. To address this re-identification risk, we propose a new method to de-identify history data by adding dummy records under a restriction. We evaluate the performance of our proposed method and compare it with the performance of the PWSCup participants as an experiment in data privacy.

1 Introduction

De-identification is a process to prevent individuals from being identified from datasets containing personally identifiable information (PII). The selection of de-identification methods to be chosen carefully to reduce the risks of re-identification in the given de-identified data. Companies should confirm that the re-identification risks have been reduced sufficiently before transferring big data to their business partners. In Japan, the Act on the Protection of Personal Information fully came into effect in 2015, in which a new notion called "*Anonymously Processed Information*[1]" was introduced [1]. Due to this revision, a data controller is allowed to provide various services using a data containing PII, free from the risk of re-identification through the acceptable handling of data.

However, the most of the common de-identification techniques assumes the datasets are well *structured*, i.e., data is represented logically in the form of a table. Hence, the dataset to be applied de-identification techniques is limited

[1] Japanese version of de-identified information with slightly changes to common anonymized data.

within the small fraction of big data. For example, ISO/IEC 20889 does not apply to complex datasets, e.g., free-form text, images, audio, or video[2]. Nevertheless, the diversity of datasets dealt in industries increases year by year.

In order to enrich the range of techniques for de-identification can happen to the whole dataset significant sufficiently, in this paper, we study a transaction data consisting of multiple records per individuals at the time of events. Especially, the payment histories data are widely stored in many business applications and are at risk to be identified for promoting targeted advertisement. With the motivation of development of secure de-identification techniques to be applied to more complex data, the data competition, Privacy Workshop Cup (PWSCup) [3], was held in 2015. Even if the de-identification of purchasing history was performed confidentially, a motivated adversary can happen to the whole dataset and successfully re-identify individuals based on some features of payments. We noted that the customers in data must have some characteristics on purchasing goods and the sets of purchasing goods are significant sufficiently to re-identify individuals. In this paper, we provide some re-identification algorithms exploiting the feature of payment and evaluate the risk of record linkage in these algorithms.

De-identification resilient against the re-identification threats is not easy. A simplest method for de-identification is suppression of records so that no two records are indistinguishable. Record suppression loses utility of data extremely. Second way is adding some dummy records so as to hide the purchasing characteristics of customers. However, naive addition may suffer the utility reduction of data when we add too many dummy records. To balance the trade off between security and utility, we shall carefully classify the set of customers in dataset into some smaller clusters in which customers share common purchasing characteristics and less dummy records are required. However, the conventional clustering algorithms such as k-means method suffers from the following problems: (1)(*Monopoly of cluster*) A few huge clusters are occupied the most of records. For instance of PWS Cup 2016, the purchase history data contains 38,087 records of unique 2,781 goods and results in the large cluster with common payment pattern of the most frequent goods. Such a big cluster be suffered the reduction and needs to add many dummy records. (2)(*Too-many Minorities*) Many small (mostly, size of 1) clusters are produced and most of them are free of change of dummy record. However, the singletons are easy to re-identify. Typical transaction data has similar property, i.e., with many records of small unique diversity. Hence it could be skewed and be suffered the reduction of utility of many noisy dummy records. Therefore, we need to develop a new clustering algorithm customized so that all cluster sizes are balanced. As well as clustering, we should identify the optimal number of clusters in the perspectives of utility (with minimizing dummy records) and security (with sufficient number of individuals in a cluster).

In this paper, we address the unbalanced issue of clustering in the following ways: (1) A clustering method replacing Term (good) Frequency–Inverse Document (individual) Frequency (TF-IDF) weights by the frequencies of purchasing goods. With TF-IDF weight, the rare items are weighted higher than common items and hence the monopoly cluster can be weakened. (2) A new algorithm for clustering prevents from being uniquely identified with restricted size of clusters by the minimum clustering size. As far as the threshold size, every cluster grows to a certain size that prevents from identified uniquely. We evaluate the proposed algorithm empirically in open datasets as well as theoretical bounds in terms of number of clusters.

The remainder of the paper is organized as follows. In Section 2, we show the characteristics of the purchase history data and the re-identification risks that are revealed in PWSCup 2016. In Section 3, we propose a method to de-identify data. In Section 4, we describe some experimental results. We introduce some references in Section 5 and Section 6 concludes the paper.

2 Characteristics of Purchase History Data and Re-identification Risks

2.1 Purchase History Dataset

The Online Retail Data Set [5] comprises the actual purchase history data observed in one year for an online retail shop in the UK and is published at the UCI Machine Leaning Repository [4]. This dataset has been used in many studies [3].

In this paper, we define the fundamental quantities of the dataset as follows.

Definition 1 *Let $U = \{u_1, \ldots, u_n\}$ be a set of customers in the dataset. Let $U' = \{u'_1, \ldots, u'_n\}$ be the set of customers in the de-identified data. Let $I(U) = \{g_1, \ldots, g_\ell\}$ be a set of goods purchased by all customers Let $I(u_i)$ be a subset of $I(U)$ purchased by customer u_i. Let b be the mean number of goods that a customer purchases in a year.*

We quantify a degree of similarity between customer u_i and u_j in terms of the sets of purchased goods as the Jaccard coefficient as follows.

Definition 2 *Let μ be the mean of the Jaccard coefficients between every two customers defined as $\mu = 1/\binom{n}{2} \sum_{i \neq j \in U} J(u_i, u_j)$, where $J()$ is defined by $J(u_i, u_j) = |I(u_i) \cap I(u_j)|/|I(u_i) \cup I(u_j)|$. Let h be the mean number of goods that are purchased by every two customers u_i and u_j, i.e., $h = |I(u_i) \cap I(u_j)|$.*

Given the dataset statistics, we estimate the mean Jaccard coefficient in the following way.

Proposition 1 *Let b and μ be the mean number of goods that a customer purchases in a year and the mean size of the intersection of the two sets of goods purchased by distinct customers. Let h be the mean number of goods that are purchased by every two customers u_i and u_j, i.e., $h = |I(u_i) \cap I(u_j)|$. Then, the mean Jaccard coefficient is $h = 2b\mu/(1 + \mu)$.*

Proof: We are able to transform μ

$$\mu = \frac{E(|I(u_i) \cap I(u_j)|)}{E(|I(u_i)|) + E(|I(u_j)|) - E(|I(u_i) \cap I(u_j)|)} = \frac{h}{2b - h}. \tag{1}$$

By solving for μ, we have the proposition. □

The transaction data contains 400 users ($n = 400$), 38,087 transactions ($m = 38,087$), and 2,781 goods. From observation of these data, we found that a customer purchases $b = 65$ goods on average and the mean Jaccard coefficient is 0.03. Using these values, we estimate $h = 4$ out of 65 goods also purchased by other customers. The maximum value of the Jaccard coefficient between two customers is 0.41 and the mean value is $\mu = 0.03$. This means that the most similar pair of customers has a similarity of only 41%. In other words, the sets of purchased goods are quite distinct and there is great diversity in customers.

2.2 Record Linkage Risk from the Jaccard Coefficient

The Jaccard coefficient is a critical quantity for records given the threat of relinking it with the de-identified data. This is because a motivated attacker who happens to observe the set of goods that the target customer purchased on the retail site can easily distinguish the customer's records by examining the Jaccard coefficients of all candidate customers.

To prevent the attacker from identifying customers, we need to modify somehow the dataset so that the attacker can single out no one set. For example, the participants de-identify data by adding noise, deleting records, and adding dummy records. We define the quantities related to this process.

Definition 3 *Let m and Δm be the total number of records in the dataset and the difference in the number of records through de-identification, respectively. The resulting number of records through de-identifying is $m' = m + \Delta m$.*

The purchase history data is dynamic data consisting of some transactions records over time. We argue that dynamic data is more vulnerable than static data with a very high re-identification risk because of its observation over the long term. For example, the Online Retail Dataset has re-identification risk via the purchased goods set for one year.

To model the malicious behavior of the attacker, we propose a re-identification method using the characteristics of the purchased goods set of customers in Algorithm 1. In the method, we assume that an attacker has access to all of the transaction records in the original data. Given the de-identified data, the attacker will then attempt to re-identify the victim customer who has the most similar pair to the target customer using the Jaccard coefficient. Note that the calculation amount of our algorithm is $\mathcal{O}(n^2)$.

Algorithm 1 Re-identification Using the Jaccard Coefficient

```
Input: $M, T, M', T'$
Step 1.
    Let $M, T$ be the data and $M', T'$ be the de-identified data. Let $I(u_i), I(u'_i)$ $(i = 1, \ldots, n)$ be a set of purchased goods of customer $u_i$ in $T$ and $u'_i$ in $T'$.
Step 2.
    Let $i^*_j = \underset{i \in \{1,\ldots,n\}}{\arg\max}\ J(I(u'_j), I(u_i))$ $(j = 1, \ldots, n')$ be the index of the customer in $T'$ who is the nearest to $u_i$.
Output: $Q = (i^*_1, i^*_2, \ldots, i^*_n)$
```

3 Our Proposal on De-identification

3.1 How to Prevent Data Being Distinguished by the Jaccard Coefficient

The challenge is to prevent data being distinguished by the Jaccard coefficient. We pursued this by mixing the records of purchased goods so that no customer could be re-identified with the Jaccard coefficient using three methods. (1)Altering some existing records ($m' = m$). (2) Deleting some existing records ($m' < m$). (3) Adding some dummy records ($m' > m$). Methods 1 and 2 (altering and deleting records) may lose accuracy in the data. In contrast, Method 3 (adding some dummy records) preserves the existing purchase histories.

In this paper, we study a method to add some fake records that do not spoil the utility of the data. In Figure 1, we illustrate how our algorithm works. Table (a) is the original transaction data T of three attributes, user IDs, record IDs, and the good IDs of five records. We detail the list of purchased goods for each customer in Table (b). In this case, we mix up three customers u_1, u_2, and u_3 by adding some dummy records randomly chosen from the set of goods. Finally, we provide the de-identified data in Table (d), shown as $I(u'_1) = I(u'_2) = I(u'_3) = I(u_1) \cup I(u_2) \cup I(u_3) = \{g_1, g_2, g_3, g_4, g_5\}$.

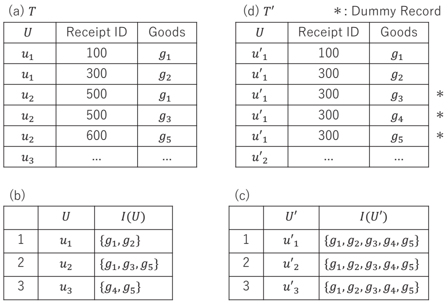
As shown, there is a trade-off between the number of dummy records and the utility of the de-identified data. Simply put, if we attempt to unify all customers, the number of dummy records will be huge and the data useless. Therefore, we need to minimize the amount of dummy data by carefully classifying the set of customers into some small clusters sharing similar purchasing characteristics.

The simplest way to cluster similar customers is to begin with representative c customers, and then extend other customers to the closest cluster, letting c be the number of clusters, $X = x_1, \ldots, x_c$ the set of clusters, and $s_i = |x_i|$ the size of a cluster. Note that cluster x_i is that set of customers partitioning the whole set of customers U, i.e., $\bigcup_{i=1}^{c} x_i = U$. The number of dummy records is calculated as $\varDelta m = \sum_{u \in x} |I(x)| - |I(u)|$.

3.2 TF-IDF Distances between Records

Generally, purchase history data contain many goods that are distributed "long-tailed," whereby a few customers occupy most records and so a simple clustering method involves a large number of dummy records. When we make the distribution of the cluster sizes resulting in the simple clustering method (k-means method) with the Jaccard coefficient as the distance between two customers, the largest cluster size is 294, which is excessively large, while the remaining 45 clusters have just one element. This suggests the cluster sizes are greatly biased.

To address the *monopoly behavior* of clusters, we propose replacing the simple Jaccard coefficient by the TF-IDF value of the set of goods. That is, we use the frequency of the term (good) times the inverse number of documents (customers) that contain the term (good) to weight the clustering of customers. Consequently, we obtain the improved method of clustering using the TF-IDF weight in Algorithm 2.



(a) T

U	Receipt ID	Goods
u_1	100	g_1
u_1	300	g_2
u_2	500	g_1
u_2	500	g_3
u_2	600	g_5
u_3	...	...

(d) T' $*$: Dummy Record

U	Receipt ID	Goods	
u'_1	100	g_1	
u'_1	300	g_2	
u'_1	300	g_3	$*$
u'_1	300	g_4	$*$
u'_1	300	g_5	$*$
u'_2	...	...	

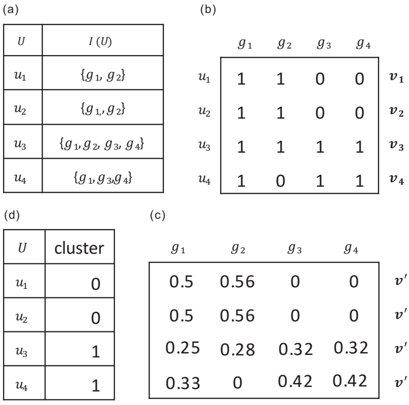
(b)

	U	$I(U)$
1	u_1	$\{g_1, g_2\}$
2	u_2	$\{g_1, g_3, g_5\}$
3	u_3	$\{g_4, g_5\}$

(c)

	U'	$I(U')$
1	u'_1	$\{g_1, g_2, g_3, g_4, g_5\}$
2	u'_2	$\{g_1, g_2, g_3, g_4, g_5\}$
3	u'_3	$\{g_1, g_2, g_3, g_4, g_5\}$

Fig. 1: How to Add Dummy Records to Data



(a)

U	$I(U)$
u_1	$\{g_1, g_2\}$
u_2	$\{g_1, g_2\}$
u_3	$\{g_1, g_2, g_3, g_4\}$
u_4	$\{g_1, g_3, g_4\}$

(b)

$$\begin{array}{c|cccc|c} & g_1 & g_2 & g_3 & g_4 & \\ \hline u_1 & 1 & 1 & 0 & 0 & \boldsymbol{v}_1 \\ u_2 & 1 & 1 & 0 & 0 & \boldsymbol{v}_2 \\ u_3 & 1 & 1 & 1 & 1 & \boldsymbol{v}_3 \\ u_4 & 1 & 0 & 1 & 1 & \boldsymbol{v}_4 \end{array}$$

(d)

U	cluster
u_1	0
u_2	0
u_3	1
u_4	1

(c)

$$\begin{array}{cccc|c} g_1 & g_2 & g_3 & g_4 & \\ \hline 0.5 & 0.56 & 0 & 0 & \boldsymbol{v}'_1 \\ 0.5 & 0.56 & 0 & 0 & \boldsymbol{v}'_2 \\ 0.25 & 0.28 & 0.32 & 0.32 & \boldsymbol{v}'_3 \\ 0.33 & 0 & 0.42 & 0.42 & \boldsymbol{v}'_4 \end{array}$$

Fig. 2: Example of Clustering of Customers via TF-IDF

Figure 2 depicts how the algorithm works for an example of four customers. Suppose we classify customers $U = \{u_1, u_2, u_3, u_4\}$ into two clusters $X = \{x_1, x_2\}$. Table (a) details the list of the purchased goods sets for the four customers, characterized by a binary matrix of purchased goods in (b). We replace the binary matrix by the matrix of TF-IDF weights of goods shown in (c). For example, the characteristics value of goods g_1 of u_1 is 0.5 because TF $= 1/2$ and IDF $= 1$. Finally, we have the resulting clusters $x_1 = \{u_1, u_2\}$ and $x_2 = \{u_3, u_4\}$ based on the cosine similarity between the two customers, as shown in (d). Note that the size of the clusters is evenly balanced because of the similarities in the TF-IDF values.

Algorithm 2 Weighting of Purchased Goods via TF-IDF

```
Input: $u_i \in U, I(u_i), c$
Step 1. Let $\boldsymbol{v}_i = (f_{i1}, f_{i2}, \ldots, f_{i\ell})$ be a characteristics vector of dimension $\ell$ of $u_i$ where
    $$f_{ij} = \begin{cases} 1 & \text{if} \quad I(u_i) \ni g_j \\ 0 & \text{otherwise.} \end{cases}$$
Step 2. Let $D_j = \{u_i \in U \big| I(u_i) \ni g_j\}$ be a set of customers who purchased a good $g_j$. Let $f'_{ij} = (f_{ij} / \sum_{k=1}^{\ell} f_{ik})(\log \frac{n}{|D_j|} + 1)$ be a weight of $f_{ij}$ via TF-IDF and $\boldsymbol{v}'_i = (f'_{i1}, f'_{i2}, \ldots, f'_{i\ell})$ be a characteristics vector of $u_i$.
Step 3. Classify the customers $U$ into clusters via $k$-means and the cosine similarity between the characteristics vectors $\boldsymbol{v}'$.
Output: $X = \{x_1, x_2, \ldots, x_c\}$
```

3.3 Method 1 : De-identification Method Based on *k*-means Clustering

We propose two de-identification methods in this paper.

Method 1 using weighting goods in the TF-IDF performs a clustering of the k-means method via cosine similarity, and adds some dummy records so that the sets of purchased goods in the cluster are indistinguishable. Figure 3 plots the distribution of cluster sizes when the number of clusters is specified as $c = 50$. Letting x_{max} and x_{min} be the largest and smallest clusters, respectively, we observe that $|x_{max}|$ is 32 and $|x_{min}|$ is 1. Obviously, there is still skewness in the distribution and it also suffers from reduction in accuracy caused by adding too many dummy records for customers belonging to a large cluster.

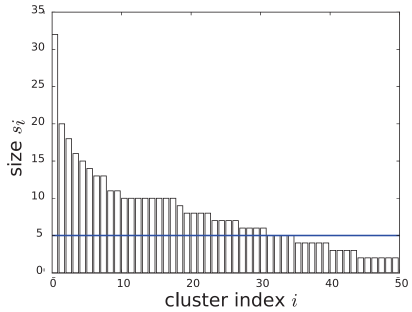


Fig. 3: Distribution of Cluster Size via Method 1 ($c = 50$)

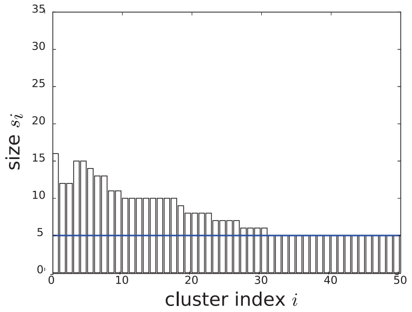


Fig. 4: Distribution of Cluster Size via Method 2 ($s_{min} = 5, c = 50$)

Algorithm 3 Algorithm to Balance Method 1

```
Input: $s_{min}, c, M, T$
  Clustering via Method 1
  Set of clusters: $X = \{x_1, x_2, \ldots, x_c\}$
  for $x$ in $\{x_i \in X \big| |x_i| < s_{min}\}$ do
    Maximum cluster: $x_{max} \in X$
    while $|x'| < s_{min}$ do
      $u_j = \underset{u_j \in x_{max}, u_i \in X}{\arg\max} J(I(u_i), I(u_j)), \quad x'_{max} \leftarrow x_{max} - \{u_j\}, \quad x' \leftarrow x \cup \{u_j\}$
    end while
  end for
  Add some dummy records in a way like Section 3.1.
Output: $M', T', P$
```

3.4 Method 2 : Balanced De-identification Method

To address the unbalanced issue, we propose a second de-identification method with the restriction of the smallest cluster size. In Method 2, we restrict the cluster sizes so that these are not below the lower limit of s_{min}, which corresponds to quantity k of k-anonymity.

Algorithm 3 shows the modified method. We move a customer whose cluster is the largest cluster x_{max} to the cluster with a size less than s_{min}. We repeat the moving operation until all cluster sizes are larger than s_{min}. The minimum threshold value s_{min} is specified depending on the number of clusters c and will be in the range of $\{2, 3, \ldots, \lfloor n/c \rfloor\}$. Figure 4 illustrates the distribution of cluster sizes in Method 2 when the minimum threshold is $s_{min} = 5$ and the number of clusters is $c = 50$. Compared with the clustering result in Figure 4, the maximum cluster size falls from 32 (Figure 3) to 16 (Figure 4) and the sizes of all clusters are satisfied as they are all more than s_{min}.

4 Experiments and Evaluation of Our Method

4.1 The Relationship between the Utility and the Number of Dummy Records

The utility of the de-identified data greatly depends on the number of dummy records Δm. Table 1 provides the relationship between some known utility metrics used in PWSCup 2016 and Δm. We identify a strong negative correlation between Δm and utility metrics (U1-cMAE, U2-cMAE, and U3-RFM). U1 and U2 are metrics that evaluate utility of de-identified data with mean absolute error (MAE) between cross tabulations of the original data and de-identified data. U3 is a metrics that evaluate utility of de-identified data with RFM (Recently, Frequency, Monetary) analysis that is a method to analyze customers. This implies that the utility of the de-identified data decreases as Δm increases. When the cluster size c increases, Δm decreases, and accordingly, the rate of re-identification increases because the correlation coefficient between Δm and c is –0.8454.

Table 1: Correlation Coefficients between Δm and Utility metrics

	Δm	$U1$	$U2$	$U3$	jaccard	Reid	c
Δm	1.0000						
$U1$-cMAE	0.9798	1.0000					
$U2$-cMAE	0.9798	1.0000	1.0000				
$U3$-rfm	0.9547	0.9876	0.9876	1.0000			
jaccard	-0.8586	-0.9327	-0.9327	-0.9494	1.0000		
Reid	-0.8489	-0.9247	-0.9247	-0.9432	0.9996	1.0000	
c	-0.8454	-0.9220	-0.9220	-0.9406	0.9994	0.9999	1.0000

4.2 Theoretical value of Δm

We are interested in estimating the theoretical value of Δm in the method to add dummy records given a_i of a customer u in a cluster x as follows. (1) a_1: The number of goods purchased by only customer. (2) a_2: The number of goods purchased by customer u and another customer. (3) a_3: The number of goods purchased by the customer u and two other customers. Note that a_i is the mean within cluster x. We translate h and b as follows via a_i.

$$h = a_2 + \sum_{i=1}^{s-2} \binom{s-2}{i} a_{i+2}, \quad b = a_1 + \sum_{i=1}^{s-1} \binom{s-1}{i} a_{i+1} \tag{2}$$

Let us calculate the number of dummy records that we should add to this cluster. First, because a_1 goods are purchased by only u_1, we should add a_1 dummy records for each of u_2 and u_3. Second, because a_2 goods are purchased by u_2 and u_3, we should add a_2 dummy records for u_1. We repeat this operation for u_2 and u_3. As a result, we have added $3(2a_1 + a_2)$ dummy records to this cluster in total. The number of dummy records added to a cluster x of size s is calculated as $\Delta m(x, s) = \sum_{i=1}^{s}(s-i)\binom{s}{i}a_i$ Therefore, we calculate the expected value of the number of dummy records added to the data as follows.

$$\begin{aligned} E(\Delta m) = c\Delta m(x,s) &= -\frac{hn^3}{2c^2} + (b + \frac{h}{2})\frac{n^2}{c} - bn \qquad (3) \\ &\geq (b + \frac{h}{2})\frac{n^2}{c} \qquad (4) \end{aligned}$$

We generalize this using parameter b, μ, n, and the number of clusters c. Note that we assume that a_i is at least zero ($a_i \geq 0$) and the cluster size s is fixed for all clusters ($s = n/c$).

4.3 Utility and Security

Table 2 shows the relationship between Δm and s_{min}. Note that Δm is minimized when s_{min} is $\lfloor \frac{n}{c} \rfloor$ in each c. We observe that the Jaccard coefficients

Table 2: Relationship between s_{min} and Δm

	$c = 50$			$c = 100$			$c = 125$		
	Δm	jaccard	Reid	Δm	jaccard	Reid	Δm	jaccard	Reid
Method 1	182897	0.1728	0.1235	128568	0.3060	0.2488	97581	0.3692	0.3120
$s_{min} = 2$	183902	0.1729	0.1223	99228	0.3061	0.2475	60492	0.3687	0.3105
$s_{min} = 3$	175449	0.1726	0.1222	68357	0.3041	0.2480	*46101	0.3667	0.3102
$s_{min} = 4$	162474	0.1723	0.1218	*59374	0.3044	0.2465			
$s_{min} = 8$	*125798	0.1681	0.1218						

are distributed across a small range and the standard deviation of the Jaccard coefficient is smaller than 0.01.

Figure 5 shows the distribution of Δm with respect to c. In the experiment, we investigate the purchase history data of 400 customers with the threshold value s_{min} specified as $\lfloor \frac{n}{c} \rfloor$. In a comparison of Methods 1 and 2, Method 2 has only about 53% of the Δm of Method 1. The solid line in Figure 5 plots the theoretical estimation of Δm calculated in Eq. (3)

We show the actual rate of re-identified records of each c in the column labeled Reid in Table 2. For each of the de-identified data, we apply Algorithm 1, being the Jaccard re-identification method described in Section 2, for some c. The Jaccard re-identification method successfully identifies at least one customer in each cluster who purchased goods most frequently. We observe no difference between the two methods. Therefore, we have the simplest consequence that the expected rate of de-identified data to be re-identified using either Method 1 or 2 is calculated as $E(Reid) = c/n$.

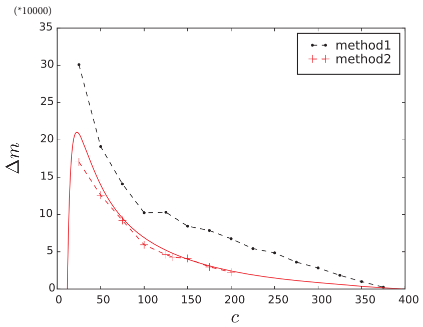


Fig. 5: Comparison of Utility of Method 1 and Method 2

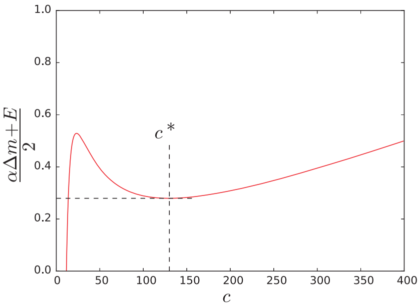


Fig. 6: Best Number of Clusters c^* for Method 2

4.4 Optimum Number of Clusters

When we de-identify data, we reduce the utility of the data but enhance its security. However, these metrics depend on the use case and structure of the target data and it is difficult to evaluate data comprehensively. In this paper, we evaluate data comprehensively via the metrics $(\alpha E(\Delta m) + E(Reid))/2 \cdots$ (5) referring to the metrics $(\text{Utility} + \text{Security})/2$ used in PWSCup 2016 and calculate the best number of clusters c^*. Let α be a coefficient to normalize Δm to the range of $0 \leq E(\Delta m) \leq 1$. Figure 6 illustrates the best number of clusters c^*. When we de-identified the data for $n = 400$, $b = 65$, and $\mu = 0.03$ via Method 2, the best number of clusters c^* is 130. Let us consider the best number of clusters c^* for the number of customers n. Substituting equation (4) for Δm in equation (5), we obtain $c^* = \sqrt{\alpha(b + \frac{h}{2})n^3}$ from the minimum value of equation (5). Note that α is a parameter depending on n.

5 Related Works

Technical Specification ISO/TS 25237 [6] defines anonymization as "a process that removes the association between the identifying data and the data subject." The ISO definition classifies anonymization techniques into masking and de-identification. Many anonymization algorithms have been proposed to preserve privacy while retaining the utility of the data that have been *anonymized*. That is, the data are made less specific so that a particular individual cannot be identified. Anonymization algorithms employ various operations, including *suppression* of attributes or records, *generalization* of values, replacing values with *pseudonyms*, *perturbation* with random noise, sampling, rounding, swapping, top/bottom coding, and microaggregation [7, 8].

Koot et al. proposed a method to quantify anonymity via an approximation of the uniqueness probability using a measure of the Kullback–Leibler distance [9]. Monreale et al. proposed a framework for the anonymization of semantic trajectory data, called c-safety [10]. Based on this framework, Basu et al. presented an empirical risk model for privacy based on k-anonymous data release [11]. Their experiment using car trajectory data gathered in the Italian cities of Pisa and Florence allowed the empirical evaluation of the protection of anonymization of real-world data. Stokes et al. defined n-confusion [12], which is a generalization of k-anonymity. In 2017, Torra presented a general introduction to data privacy [13]. Li and Lai proposed a definition of a new δ-privacy model that requires that no adversary could improve more than δ privacy degree [14].

6 Conclusions

We revealed the risk of data to be re-identified via the characteristics of purchasing goods of customers and proposed the de-identification method by minimizing additional dummy records to be add the datasets. In our proposal method, the set of customers are classified into some clusters based on the characteristics of purchasing goods weighted as the TF-IDF. We have demonstrated our proposed algorithm reduces the number of dummy records as far as restricted size of clusters. We estimated the expected value of the number of dummy records in a simple mathematical model and identified the optimal number of clusters that minimizes the mean re-identification rate under the balanced utility metrics.

Our future studies include the evaluation of accuracy of clustering and effectiveness in case of other datasets. We will try to use other de-identification methods like deleting and adding noise to improve our de-identification method.

References

1. "Report by the Personal Information Protection Commission Secretariat: Anonymously Processed Information –Towords Balanced Promotion of Personal Data Utilization and Consumer Trust–", Personal Information Protection Commission Secretariat, 2017.
2. "Privacy enhancing data de-identification terminology and classification of techniques", ISO/IEC 20889.
3. H. Kikuchi, T. Yamaguchi, K. Hamada, Y. Yamaoka, H. Oguri and J. Sakuma, "What is the Best Anonymization Method? – a Study from the Data Anonymization Competition Pwscup 2015", Data Privacy Management Security Assurance (DPM2016), LNCS 9963, pp. 230 - 237. (2016)
4. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/index.php, refered in December 17, 2018.
5. Online Retail Data Set, https://archive.ics.uci.edu/ml/datasets/online+retail, refered in December 17, 2018.
6. "Health informatics – Pseudonymization", ISO Technical Specification ISO/TS 25237.
7. Information Commissioner's Office (ICO), Anonymisation: managing data protection risk code of practice. (2012)
8. C.C. Aggarwal and P.S. Yu., "A General Survey of Privacy-Preserving Data Mining, Models and Algorithms", *Privacy-preserving Data Mining*, Springer, pp.11–52. (2008)
9. Koot, M. R., Mandjes, M., van't Noordende, G., and de Laat, C., "Efficient probabilistic estimation of quasi-identifier uniqueness", In Proceedings of ICT OPEN 2011, 14–15, pp.119–126. (2011)
10. A Monreale, R Trasarti, D Pedreschi, C Renso and V Bogorny, "C-safety: a framework for the anonymization of semantic trajectories", Transactions on Data Privacy, Vol. 4(2), pp.73–101. (2011)
11. A. Basu, A. Monreale, R. Trasarti, J. C. Corena, F. Giannotti, D. Pedreschi, S. Kiyomoto, Y. Miyake and T. Yanagihara, "A risk model for privacy in trajectory data", Journal of Trust Management, pp.2–9. (2015)
12. Klara Stokes, Vicen Torra, n-confusion: a generalization of k-anonymity, EDBT/ICDT Workshops 2012: pp.211–215. (2012)
13. V. Torra, "Data Privacy: Foundations, New Developments and the Big Data Challenge", Studies in Big Data 28, Springer. (2017)
14. Zhizhou Li, Ten H. Lai, δ-privacy: Bounding Privacy Leaks in Privacy Preserving Data Mining, DPM/CBT 2017, LNCS 10436, pp. 124-142, Springer. (2017)