2013 IEEE 14th International Conference on Mobile Data Management

KAMP: Preserving k-anonymity for Combinations of Patterns

Chia-Hao Hsu
Department of Electrical Engineering
National Chung Hsing University
Taichung, Taiwan, R.O.C.
Email:w100064001@mail.nchu.edu.tw

Hsiao-Ping Tsai
Department of Electrical Engineering
National Chung Hsing University
Taichung, Taiwan, R.O.C.
Email: hptsai@nchu.edu.tw

***Abstract*—As huge data are increasingly generated and accumulated, outsourcing data for storage, management, and knowledge discovery is becoming a paradigm. However, given that a lot of sensitive information and valuable knowledge are hidden in data, the outsourcing of data is vulnerable to privacy crises and leads to demands for generalization or suppressing techniques to protect data from re-identification attacks. Differing from previous works that aim at satisfying the k-anonymity on individual patterns, we propose the k-anonymity of multi-pattern (KAMP) problem to protect data from re-identifying users by using a combination of patterns and also propose the KAMP-p1 algorithm to generalize and suppress data. To study the effectiveness of the proposed algorithm, we conduct experiments on a synthetic and a small real dataset. The experimental results show that KAMP-p1 algorithm can satisfy k-anonymity while preserving many patterns in order to retain useful knowledge for decision making.**

I. Introduction

Nowadays, with the advances of sensing technologies, wireless communication techniques and the prevalence of intelligent mobile devices, a tremendous number of applications, such as monitoring, tracking, and entertainments, are being developed and great amounts of data are generated and accumulated, which demands a cost effective approach to maintain and manage data. For this reason, outsourcing data is becoming a paradigm. Furthermore, to improve quality of life, efficiency of administrators, security and safety of the public, governments and academia also increasingly demand the sharing of public data for knowledge discovery in order to devise policies and political strategies. However, since sensitive information, confidential messages, valuable business secrets, and unwilling to be revealed details can be hidden in data, many privacy preserving issues emerge which stagger the progress of the release of data. To trade off privacy issues and public/business profits, a lot of effort has been devoted to hiding sensitive information, such as patient identity in patient records or the buyer information in purchasing transactions [18][19]. To protect data from re-identification attacks, one emerging protection model is k-anonymity [11][12][17] which has been recently proposed to provide a compromising data protection by imposing uncertainty. Specifically, a k-anonymity technique removes sensitive information from data such that for each person the information in the released data cannot be distinguished from at least k-1 other individuals in the same release.

In the last few decades, many data mining techniques have been proposed and applied to increase business profits or to improve human life. Among the many mining techniques, pattern discovery involves the discovery of important characteristics from data, such as a motif in DNA sequences [1][4][5], consumer purchasing behavior in transaction data [3], and movement regularity in the location sequences [6][7]. In general, the discovered patterns are more representative and contain more knowledge than the original data. What is worth noticing is that in many cases, with some easily obtained patterns, it is not difficult to re-identify a person from released data, e.g., if a man is used to purchase new-born diapers, it can be easy to re-identify the buyer of a transaction with diapers because everybody knows which family has a newborn baby in a small town. And what is worse is that the association of a few patterns divulges more clues to recognize identities. For example, with some prior knowledge about the address of a worker, the commute path of his report for duty can be predicted. Although the departure pattern may be shared by other village residents and the arrival pattern can be shared by many colleagues in the same factory, by combining the departure pattern with the arrival pattern, the owner of a complete route can be vividly portrayed. What can be imagined is that as more data become published or outsourced, more clues, such as the buying pattern, movement patterns of departure and arrival, can be accessed, thus even worsening the multi-pattern re-identification problem. Consequently, more people will find themselves enmired in a security and privacy crisis sooner or later.

Similar to the above examples, many mining results violate privacy issues [2]. Therefore, instead of providing k-anonymity to data, a lot of studies focus on anonymizing mining results data [16][13][14][15][21]. Some consider satisfying k-anonymity for individual patterns and a few works further handle the inference problem across large itemsets [14][15]. However, few consider the multi-pattern re-identification problem.

To provide k-anonymity, our idea is to divide patterns into common patterns and special patterns, with a special pattern is a pattern that is not shared by more than k people, and then only remove or blur special patterns before data release. Furthermore, to prevent re-identification attacks by using the combination of patterns, we first formulate the k-anonymity of the multi-pattern (KAMP) problem and discuss the generalization and suppression approaches for transactional data;

978-0-7695-4973-6/13 $26.00 © 2013 IEEE
DOI 10.1109/MDM.2013.74

and based on these, we introduce our KAMP-p1 algorithm that solves the KAMP problem in a greedy manner.

To show the effectiveness of our KAMP-p1 algorithm, we propose the average pattern distance PD as the evaluation metric and conduct experiments with a real dataset, BMS-WebView-1, and a synthetic dataset generated by the IBM generator, IBM-SynData, in order to study the impact of several important parameters, including the minimal support count σ, the minimal anonymity degree threshold k, and the maximal size threshold M, on information loss in terms of the average pattern distance PD. Our experimental results first show that re-identification by using a combination of patterns is especially severe when the data inherently contains more patterns as well as more pattern combinations (small σ, large M), or when the privacy preserving requirement is more precise (big k). They also show that our KAMP-p1 algorithm can effectively suppress or generalize sensitive and individual patterns combinations to protect privacy while retaining pattern-level information for its usability.

The rest of this paper is organized as follows. Section II first shows the related works and then formulates the KAMP problem as well as defines parameters and the evaluation metric. In Section III, we discuss the generalization and suppression operations and propose our KAMP-p1 algorithm. The performance study is described in Section IV and the paper concludes in Section V.

II. Preliminary

A. Related Works

Pattern Discovery is to discover important, representative, or comprehensible features hidden in sequence data. These features may be the motifs that are a short distinctive subsequence shared by a number of related gene or protein sequences [1][4][5] or repeating movement subsequence that represent regular movement behavior in moving objects' trajectories [6][7]. Other examples include periodic events in Web system logs [8], similar recurring phrases in word sequences [9], and repeating themes in music [10]. These features are usually easier to understand and more meaningful to illustrate than the original sequences. For example, a motif of a protein sequence is an ordered list of amino acids which presents structural characteristics and can determine a high level functionality and often plays the role of a signature in the course of evolution; motifs thus play a key functional or structural role in analyzing the diversity or closeness of various species. Unfortunately, they also involve many privacy issues, such as species and diseases. Similarly, the movement patterns in trajectory sequences are the repeating subsequences that can be utilized to foretell the locations in the future. Thus, a lot of research is aimed at discovering useful movement patterns for the development of intelligent applications. However, movement patterns also lead to security and safety problems. There are several other mining consequences that also yield these kinds of side effects and have thus received greater attention in an effort to protect mining results instead of the original data by hiding patterns that are associated with someone or a small community [19][18].

k-anonymity is a new emerging privacy preserving model [11][17]. Friedman et al. [12] proposed an algorithm that combines mining and anonymization in a single process for directly building a k-anonymous decision tree from a private table. In [16], the authors assume the maximum knowledge of an adversary is at most m items in a specific transaction and proposed an efficient generalization algorithm to provide k^m anonymity for raw transactional datasets with itemsets containing less than or equal to m items. In [14] [15], the researchers figure out the inference problem across large itemsets and derive inference channel detection methods to generate anonymized association rules. They also proposed information loss measurement metrics for performance evaluation. In [21], the authors suggested hiding infrequent, and thus potentially sensible, subsequences before disclosing the sequential data. Their approach utilizes a prefix tree in infrequent subsequence pruning and guarantees that the disclosed data are k-anonymous.

B. Problem Formulation

Let $I = \{i_1, i_2, \ldots, i_o\}$ be a set of items and U denotes a set of users. A dataset D is a set of transactions, where a transaction t is a tuple of user identity and a set of items on I, i.e., $t =< u, j >, u \in U, j \subseteq I$.

Definition 1 (Support Count): An itemset $I_j, I_j \subseteq I$, is contained by a transaction $t =< u, j >$ if $I_j \subseteq j$. The number of transactions in D that contains I_j is defined as the support count of I_j, denoted by $sup(D, I_j)$. And the number of transactions of a user u that contains I_j is $sup(D, u, I_j)$.

Definition 2 (Pattern): For a given dataset D and a minimal support count σ, a pattern p is a frequent itemset with the support count larger than or equal to the minimal support count, i.e., $sup(D, p) \geq \sigma$. All of the patterns of D are denoted by $FP(D)$, i.e., $FP(D) = \{p | \forall p \subseteq I, sup(D, p) \geq \sigma\}$.

A pattern p is contained by a user if at least one of his transactions in D contains p, i.e., $\exists t_i =< u_i, I_i > \in D$ such that $u_i = u$ and $p \subseteq I_i$. All of the patterns of user u in D is denoted by $FP(D, u) = \{p | p \in FP(D)$ and $\exists t_i =< u_i, I_i > \in D$ such that $u_i = u$ and $p \subseteq I_i\}$.

Definition 3 (Maximal Pattern): A maximal pattern is a pattern of a user u in D that it's not a subset of any other pattern in $FP(D, u)$. All of the maximal patterns of u are denoted by $MP(D, u)$.

Definition 4 (m-pattern Set): An m-pattern set is a set of m maximal patterns. All m-pattern sets of u are all of the m-combinations of the maximal patterns of u in D, denoted by $CP(D, u, m)$.

For example, while Colin has 3 maximal patterns, i.e., $MP(D_0, Colin) = \{A, B, C\}$, $CP(D_0, Colin, 2) = \{\{A, B\}, \{A, C\}, \{B, C\}\}$.

Definition 5 (Combination of Patterns, CP): For a maximal size threshold M, $CP(D, u)$ is all combinations of the maximal patterns of u in D with a size less and equal than M, i.e., $CP(D, u) = \bigcup_{m=1}^{M} CP(D, u, m)$. In addition, the union of all users' m-pattern sets, $m \leq M$, is denoted by $CP(D)$, i.e., $CP(D) = \bigcup_{u \in U} CP(D, u)$.

Following the previous example, given $M = 3$, $CP(Colin) = \{\{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$.

Definition 6 (Anonymity Degree, AD): For a pattern set cp, we say that cp is contained by u if $\forall$ pattern p in cp, there exists a maximal pattern p' in $MP(D,u)$ such that p is a subset of p'. The anonymity degree (AD) of pattern set cp is defined as the number of users that contains cp, i.e., $AD(D,cp) = |\{u | u \in U \, and \, \forall p \in cp, \exists p' \in MP(D,u)$, such that $p \subseteq p'\}|$.

Definition 7 (k-anonymized): For a minimal anonymity degree k, an m-pattern set mp is k-anonymized if its anonymity degree is larger than or equal to k, i.e., $AD(D,mp) \geq k$; otherwise, it is un-k-anonymized.

Problem Formulation: With the above definitions, we formulate the k-anonymity of multi-pattern (KAMP) problem below. Given a transaction dataset D, a minimal support count σ, a minimal anonymity degree threshold k, and the maximal size threshold M, the KAMP problem is to generate the transaction dataset D' such that the patterns of D' retain most information as that of D and any m combination of patterns of D', $m \leq M$, is k-anonymized, i.e., $\forall u \in U, \forall cp \in CP(D',u), AD(D',cp) \geq k$.

C. Evaluation Metric

Since the difference of a user's patterns before and after anonymization directly relates to the amount of modifications that were done on the original transaction dataset, we define the pattern distance regarding individual users as the measure of information loss and take the average pattern distance as the evaluation metric. Specifically, let P_i and P_i' denote the maximal pattern set of user i before and after anonymization, respectively. The distance of a maximal pattern p with respect to a maximal pattern set P is

$$distance(p,P) = min_{p' \in P}(p \cup p' - p \cap p').$$

Then, the pattern distance $PD(i)$ of user i is

$$PD(i) = \sum_{p \in P_i} distance(p, P_i') + \sum_{p' \in P_i'} distance(p', P_i).$$

For example, if $P_0 = \{A = \{i_0\}, B = \{i_1, i_2\}, C = \{i_2, i_3\}\}$ and $P_0' = \{A, B, D = \{i_4, i_5\}\}$, then $PD(0) = (0+0+2)+(0+0+3) = 5$. And the average pattern distance PD is defined as

$$PD = \frac{\sum_{i=1}^{|U|} PD(i)}{|U|}.$$

According to the definition of PD, we can see that if every user's patterns remain unchanged, $PD = 0$; otherwise, as more patterns are modified for anonymization, PD increases.

III. The Proposed Algorithm

To solve the KAMP problem, we propose a framework comprising two phases. In phase 1, we regenerate the patterns of D with the goal of keeping the maximum amount of pattern-level information. And in phase 2, we modify D according to the regenerated patterns with the goal of preserving the transaction data and doing as little modification as possible. Due to the page limitations of this paper, we focus on the regeneration of the patterns to meet k-anonymity requirements. To satisfy the k-anonymity problem while retaining the maximum of pattern-level information in order to retain its practical usefulness, we first introduce two operations, including generalization and suppression, that are used to regenerate every user's patterns.

Generalization: For an un-k-anonymized m-pattern set mp, i.e., $AD(mp) < k$, the generalization operation adds patterns to other $K - AD(mp)$ users so as to complete mp's anonymity degree to k. Basically, the simplest way to increase the anonymity degree is to find $K - AD(mp)$ users and add deficient items to their transactions to make the $K - AD(mp)$ users contain mp. Though the approach can complete the anonymity degree of mp, the addition operations may lead to changes of the users' patterns and cause information loss in terms of pattern distance. Further, the pattern changes may result in more combinations of patterns to check for their anonymity degrees and the side effect can further destroy pattern-level information. Therefore, the key to generalization is to choose the $K - AD(mp)$ users properly in order to minimize the average pattern distance. For this reason, we adapt the strategy of avoiding the creation of new pattern combinations that are not in $CP(D)$ so as to minimize the side effects. More specifically, let $PS(S)$ denote the set of all subsets of S, i.e., the power set of S, and we search for candidate users, where a user is recognized as a candidate if

$$\forall cp' \in \{cp \in PS(mp) \times CP(D,u), |cp| \leq M\}, cp' \in CP(D).$$

For all candidates, we sort them by the number of patterns that individual candidates lack, i.e., $|mp - FP(D,u)|$, in descending order and choose the smallest $K - AD(mp)$ users to make them contain mp. By this approach, we prioritize the users with similar patterns so as to minimize modifications for the generation operation. On the other hand, if the candidates are insufficient for our needs, we create a virtual user and assign cp to him to increase the anonymity degree until $AD(cp) \geq k$.

Suppression: For an unqualified m-pattern set, mp, the suppression operation is to remove patterns in mp from $AD(mp)$ users that contain mp in order to decrease its anonymity degree to zero. Specifically, to suppress an mp, at least a pattern $p \in mp$ should be removed from all transactions of $AD(mp)$ users. Note that removing a pattern p from the $AD(mp)$ users is a complex operation. To avoid a vicious spiral of suppression, the removing operation should evade harming the anonymity degree of those already k-anonymized m-pattern sets. Moreover, in case other users lose patterns because of the removing operation, we should not decrease the support counts of patterns other than p to lower than σ. In addition, we should also avoid removing a pattern that results in producing originally in-existing patterns. In summary, the key to the suppression operation is to properly choose which pattern to remove from which $AD(mp)$ users so as to minimize the average pattern distance. Let U_{mp} denote the $AD(mp)$ users and $T(U_{mp}, p)$ denote all of the transactions of the users in U_{mp} that contain pattern p. In order to choose U_{mp} to remove a pattern p, $p \in mp$, from $T(U_{mp}, p)$, our strategy contains three rules:

1) Not harming the anonymity degree of the k-anonymized m pattern set of the $AD(mp)$ users. Specifically, let $KM(U_{mp})$ denote the k-anonymized m-pattern sets and $AD(U_{mp}, cp)$ denote the number of users in U_{mp} that contains a pattern set cp; the first rule is

$$\forall cp \in KM(U_{mp}), AD(cp) - AD(U_{mp}, cp) \geq k.$$

2) Not suppressing other patterns contained by any user in U_{mp}. Specifically, let $FP_{U_{mp}}$ denote the patterns contained by at least one the transactions in $T(U_{mp}, p)$; the 2nd rule is

$$\forall p' \in (FP_{U_{mp}} - p), sup(D, p') - |T(U_{mp}, p)| \geq \sigma.$$

3) Producing no new patterns. Let $T'(U_{mp}, p)$ denote the remnants of $T(U_{mp}, p)$ after removing p. The 3rd rule is

$$\forall t \in T'(U_{mp}, p), \forall p \subseteq t, p \in FP(D).$$

As shown in Algorithm 1, our KAMP-p1 algorithm adapts a greedy strategy, starting from $m = 1$ to M, to generalize or suppress the m-pattern sets of all users since combinations of fewer patterns are prone to k-anonymized (Lines 2-19). As in Lines 5 and 6, C^k is a set of the k-anonymized pattern combinations with size $\leq m$ while C_m^* is the set of un-k-anonymized pattern combinations with the size equal to m. In Line 7, the un-k-anonymized m-pattern sets are sorted by their anonymity degrees in descending order; from Line 8 to Line 18 each of them are generalized or suppressed depending on whether it is suppressible according to not only the three suppression rules but also the cost of suppression and generation in terms of the number of transactions to be modified. The generation cost of m-pattern set cp is $k - AD(cp)$ while the suppression cost is $|T(U_{mp}, p)|$. As in Lines 9-11, we check whether U_{mp} exists such that removing p from $T(U_{mp}, p)$ satisfies the suppression rules, where p is an element of the m-pattern set, cp. In line 12, we compare the cost of generation with that of suppression and choose the operation with a lower cost. Note that while multiple pairs of U_{mp} and p exist, we choose the pair with the minimal cost as the later half of Line 12. If no such pair exists, the suppression cost is ∞. Next, either the suppression operation (Line 16) or the generation operation (Line 13) is carried out to update the maximal pattern table, MP. Finally, after all un-k-anonymized m-pattern sets are blurred or hidden, MP is returned for phase 2 processing.

Algorithm 1 KAMP-p1

Input: A maximal pattern table MP, a minimal support count σ, an anonymity threshold k, and maximal size of m-pattern sets M
Output: A k-anonymized pattern table MP

1: $C^k = \emptyset$
2: **for** $m = 1$ to M **do**
3: C_m = get union of all user's m-pattern sets from MP
4: **if** $C_m == \emptyset$ **then exit for**
5: $C^k = C^k \cup \{cp | cp \in C_m$ and $AD(cp) \geq k\}$
6: $C_m^* = \{cp | cp \in C_m$ and $AD(cp) < k\}$
7: sort C_m^* by anonymity degree in descending order
8: **for** each $cp \in C_m^*$ **do**
9: **for** each $p \in cp$ **do**
10: **if** p is suppressible **then** add p to Ps
11: **end for**
12: **if** $Ps == \emptyset$ or $gene\_cost(cp) \leq \min_{p \in P_s} supp\_cost(cp, p)$ **then**
13: $generalize(MP, cp)$
14: add cp to C^k
15: **else**
16: $suppress(MP, cp, p)$
17: **end if**
18: **end for**
19: **end for**
20: **return** MP

user	maximal patterns
u_1	D E F
u_2	A B C
u_3	A B D
u_4	B D E F
u_5	B D E
u_6	E F
u_7	A B C
u_8	D E F
u_9	A C
u_{10}	B D E
u_{11}	B E
u_{12}	A D
u_{13}	D E F
u_{14}	A D

(a)

user	maximal patterns
u_1	D E F
u_2	A B C
u_3	A B D
u_4	~~**B**~~ D E F
u_5	B D E
u_6	E F
u_7	A B C
u_8	D E F
u_9	A **B** C
u_{10}	B D E
u_{11}	B E
u_{12}	A D
u_{13}	D E F
u_{14}	A D

(b)

user	maximal patterns
u_1	D E F
u_2	A B C
u_3	A ~~**B**~~ D
u_4	D E F
u_5	B D E
u_6	E F
u_7	A B C
u_8	D E F
u_9	A B C
u_{10}	B D E
u_{11}	B **D** E
u_{12}	A D
u_{13}	D E F
u_{14}	A D

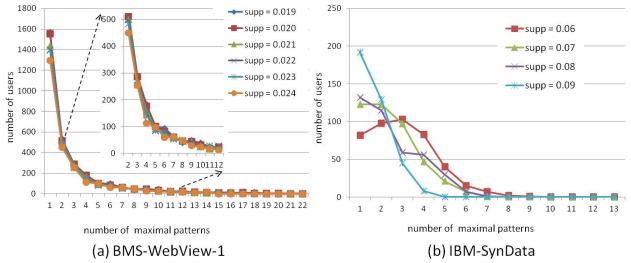
(c)

Fig. 1. Examples suppression and generalization operations; (a)maximal pattern table, (b)result of $m = 2$, and (c)final result.

For example, assuming $k = 3$, $M = 3$, and $\sigma = 2$, Fig. 1(a) shows a maximal pattern table of 14 users, where each capital letter represents a maximal pattern. First, while $m = 1$, since all 1-pattern sets are k-anonymous already, the process continues to $m = 2$. As $m = 2$, only $\{B, C\}$ and $\{B, F\}$ are un-k-anonymized 2-pattern sets, where $AD(\{B, C\}) = 2$ and $AD(\{B, F\}) = 1$. Assume $supp(u_2, B) = 1$ and $supp(u_7, B) = 1$, the suppression cost $supp\_cost(\{B, C\}, B) = supp(u_2, B) + supp(u_7, B) = 2$. Since $AD(C) = 3$ and suppressing C from u_2 and u_7 violates suppression rule 1 (not harming those already k-anonymized m-pattern sets), $supp\_cost(\{B, C\}, C) = \infty$. Thus, $P_s = \{B\}$. On the other hand, the generalization cost is $gene\_cost(\{B, C\}) = k - AD(\{B, C\}) = 1$. Therefore, generalization is chosen. To generalize $\{B, C\}$, we examine the users one by one for the candidates. For example, given that adding $\{B, C\}$ to u_1 creates $\{C, D\}$ and $\{C, E\}$ that are not in $CP(D)$, u_1 is not a candidate. After examining u_1, $u_3 - u_6$, and $u_8 - u_{14}$, only u_9 is a candidate and thus $\{B, C\}$ is added to u_9 to make $AD(\{B, C\}) = 3$. As for $\{B, F\}$, assume $supp(u_4, B) = 1$, and $supp(u4, F) = 3$, since both B and F are suppressible, i.e., $P_s = \{B, F\}$, the minimal suppression cost is $min_{p \in \{B,F\}} supp\_cost(\{B, F\}, p) = supp(u_4, B) = 1$. Compared with $gene\_cost(\{B, F\}) = 2$, we suppress B from u_4. Fig. 1(b) shows the results for $m = 2$. As $m = 3$, $\{B, D, E\}$ and $\{A, B, D\}$ are un-k-anonymized. For $\{B, D, E\}$, since all of B, D, and E are not suppressible, we generalize it by adding D to u_{11}. For $\{A, B, D\}$, since $AD(\{A, D\}) = 3$, removing any $\{A, D\}$ decreases the anonymity degree, which violates suppression rule 1. In this case, only B is suppressible and thus we remove B from u_3. The final result is shown in Fig. 1(c); note that only 4 users' maximal patterns are influenced in this example.

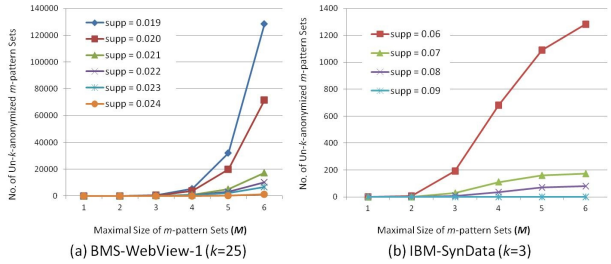
IV. Experimental Results

To show the effectiveness of our algorithm, we conduct experiments with a real dataset, BMS-WebView-1 [20], as well as a synthetic dataset generated by the IBM generator [22], abbreviated as IBM-SynData, to study the impact of parameters M, k, and σ on the average pattern distance PD. BMS-WebView-1 is a Web log down-loaded from the KDD-Cup 2000 home page. It contains 59602 transactions and 497 distinct items. The maximal and minimal transaction sizes



(a) BMS-WebView-1 (b) IBM-SynData

Fig. 2. Histograms of the maximal pattern number of individual users in (a) BMS-WebView-1 and (b) IBM-SynData.



(a) BMS-WebView-1 (k=25) (b) IBM-SynData (k=3)

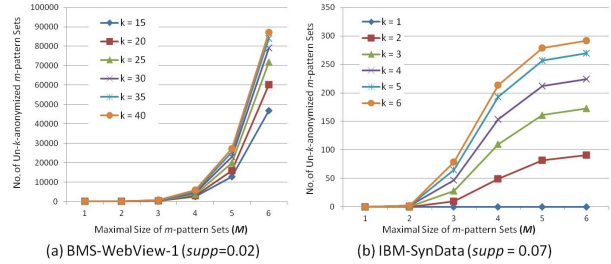
Fig. 3. No. of un-k-anonymized m-pattern sets in (a) BMS-WebView-1 and (b) IBM-SynData as M and σ vary.

are 267 and 1 respectively, and the average is 2.5 items per transaction. Since it lacks the user identity information, we cluster similar transactions into 5000 groups by using Jaccard coefficient [1] and assign each group of transactions to a user, such that the average number of transactions per user is about 11.9. With min-support 0.02, each user is associated with at most 20 maximal patterns, at least one pattern, and 3 maximal patterns on average. As for the synthetic dataset, it contains 5000 transactions on 1000 distinct items and the average transaction size is 5. Moreover, the number of patterns is 1000 and the average pattern size is 3. The correlation between consecutive patterns is 0.25. Similar to the BMS-WebView-1 dataset, we cluster similar transactions into 500 groups using Jaccard coefficient and the average number of transactions per user is 10. Fig. 2 shows the histograms of maximal pattern number of individual users in BMS-WebView-1 and IBM-SynData. In BMS-WebView-1, as the support varies from 0.019 to 0.024, the number of maximal patterns of individual users decreases; most users have two maximal patterns and a few users have more than 10 maximal patterns. In IBM-SynData, the trend is apparent as the minimal support varies from 0.06 to 0.09 and almost all users contain less than 6 maximal patterns.

A. Experiment 1: Un-k-anonymized m-pattern Sets

In the first experiment, we study the order of severity of the KAMP problem. Fig. 3 presents the number of un-k-anonymized m-pattern sets in BMS-WebView-1 and IBM-SynData as the minimal support varies from 0.019 to 0.024 and 0.06 to 0.09, respectively. In both datasets, as M increases, there are more un-k-anonymized m-pattern sets. The trend is more obvious when the minimal support is small since users tend to contain more maximal patterns, and as a result there will be more un-k-anonymized pattern combinations. The results suggest that the risk of re-identification by using a combination of patterns is more serious. Fig. 4 depicts the un-k-anonymized situations of BMS-WebView-1 and IBM-SynData as k varies from 15 to 40 and 1 to 6, respectively. As a larger k indicates a stricter anonymity requirement, there are more un-k-anonymized m-pattern sets as k is larger. Also, as shown in Fig. 3 and Fig. 4, the tendency of IBM-SynData is to slow down as M approaches 6 because almost all users contain less than 6 maximal patterns, thus indicating that the number of pattern combinations does not augment rapidly as M increases.

[1] $Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$, where A and B are two itemsets.



(a) BMS-WebView-1 ($supp$=0.02) (b) IBM-SynData ($supp$ = 0.07)

Fig. 4. No. of un-k-anonymized m-pattern sets in (a) BMS-WebView-1 and (b) IBM-SynData as M and k vary.

B. Experiment 2: Effectiveness of the KAMP-p1 Algorithm (Impact of M, k, and σ on PD)

In the 2nd experiment, we show the effectiveness of the KAMP-p1 algorithm in terms of PD. Note that PD indicates the pattern-level information loss, with a larger PD representing a greater loss. Fig. 5 shows the curves of PD versus M as $supp$ varies from 0.019 to 0.024 for BMS-WebView-1 and from 0.06 to 0.09 for IBM-SynData. It can be seen that as the minimal support becomes larger, PD becomes smaller. This is because with a larger minimal support, the discovered patterns are more general and shared by more users, i.e., fewer sensitive and private m-pattern sets to generalize or suppress, thus requiring fewer modifications to achieve k-anonymity. Fig. 6 depicts the curves of PD versus M as k varies from 15, 20, ..., 40 and from 1 to 6 for BMS-WebView-1 and IBM-SynData respectively. As k increases, PD becomes larger, which means that a stricter privacy requirement necessitates a greater effort to provide protection (i.e., perform more modifications), which in turn means that the pattern-level information loss becomes larger. Note that in both Fig. 5(a) and Fig. 6(a) the curves' turning points are near $m = 3$ and PD is almost unchanged after $m = 5$. This is because most users contain 3 maximal patterns. In addition, as the number of m-pattern sets increases as m increases, PD becomes larger with m at the beginning. Then, sensitive m-pattern sets with $m = 3$ are k-anonymized at $m = 3$, and conveniently those un-k-anonymized m-pattern sets disappear, thus complying with the generation and suppression operations performed for $m \leq 3$. Fig. 5(b) and Fig. 6(b) reveal similar trends in IBM-SynData. Compared with Fig. 2, the turning points are unhurried around 4 because in IBM-SynData several users contain 4 to 6 maximal patterns as $supp = 0.07$.

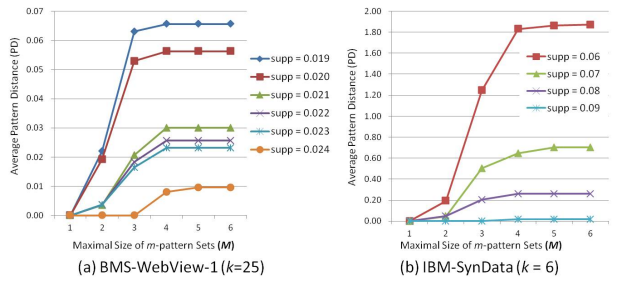


Fig. 5. Pattern-level information loss (PD) of (a) BMS-WebView-1 and (b) IBM-SynData for various σ.

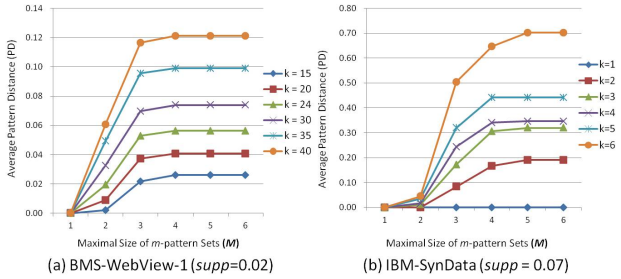


Fig. 6. Pattern-level information loss (PD) of (a) BMS-WebView-1 and (b) IBM-SynData for different privacy preserving requirement (k).

V. Conclusion and Future Works

In this work, we study the problem of k-anonymity of multi-pattern (KAMP) to protect data from re-identifying users by using the combination of patterns. To solve the KAMP problem, we derive one generalization rule and three suppression rules for the generalization and suppression operations and propose the KAMP-p1 algorithm that adapts a greedy strategy to blur and hide sensitive m-pattern sets. For the performance study, we conduct experiments with a real dataset, BMS-WebView-1, and a synthetic dataset, IBM-SynData, generated by the IBM generator. Our experimental results show that as the data inherently contains more patterns as well as more pattern combinations, the minimal support is smaller, or the privacy preserving requirement is more precise, it is prone to having a larger number of sensitive and individual pattern combinations, which means that the risk of re-identification by using a combination of patterns is more serious. In addition, the experimental results also indicate that our KAMP-p1 algorithm can effectively suppress or generalize sensitive and individual patterns combinations to protect privacy while retaining pattern-level information for its practicality.

Future research directions include studying the computing complexity of the KAMP-p1 algorithm and designing an efficient algorithm to overcome the bottleneck. Furthermore, the completion of the whole framework and the conducting of more comprehensive experiments for the performance study are still required.

Acknowledgment

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC101-2220-E-005-007.

References

[1] E. Eskin and P.A. Pevzner, "Finding composite regulatory patterns in DNA sequences," Bioinformatics, Vol. 18, No. l, pp. 354-363, 2002.

[2] M Kantarcoglu, J Jin, and C Clifton, "When do Data Mining Results Violate Privacy?" in Proc. of Int'l Conf. on Knowledge Discovery and Data Mining, pp. 599-604, 2004.

[3] D. Burdick, M. Calimlim, and J. Gehrke, "MAFIA: a maximal frequent itemset algorithm for transactional databases," In Proc. of the 2001 Int'l Conf. on Data Engineering, pp. 443452, 2001.

[4] P. G. Ferreira and P. J. Azevedo, "Protein Sequence Classification Through Relevant Sequence Mining and Bayes Classifiers," Progress in Artificial Intelligence, Vol. 3808, pp. 236-247, 2005.

[5] K. D. MacIsaac and E. Fraenkel, "Practical Strategies for Discovering Regulatory DNA Sequence Motifs," PLoS Comput Biol, Vol. 2, No. 4, pp. e36, 2006.

[6] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in Proc. 13th ACM SIGKDD Intl Conf. on Knowledge Discovery and Data Mining, pp. 330-339, 2007.

[7] H. P. Tsai, D. N. Yang and M. S. Chen, "Mining Group Movement Patterns for Tracking Moving Objects Efficiently," IEEE Trans. on Knowledge and Data Engineering, Vol.23, No.2, pp. 266-281, 2011.

[8] K. Saxena and R. Shukla, "Significant Interval and Frequent Pattern Discovery in Web Log Data," Intl J. Computer Science Issues(IJCSI), Vol. 7, Issue. 1, No. 3, pp. 29, 2010.

[9] A. Don, E. Zheleva, M. Gregory, S. Tarkan, L. Auvil, T. Clement,B. Shneiderman1, and C. Plaisant, "Discovering interesting usage patterns in text collections:Integrating text mining with visualization," In Proc. 6th ACM conf. on information and knowledge management, pp. 213-222, 2007.

[10] C.-C. Liu, J.-L. Hsu, and Arbee L. P. Chen, "Efficient Theme and Non-Trivial Repeating Pattern Discovering in Music Databases," in Proc. Intl Conf. Data Engineering, pp. 14-21, 1999.

[11] L. Sweeney, "k-anonymity: a model for protecting privacy. Int'l J. on Uncertainty," Fuzziness and Knowledge-based Systems, 10 (5), pp. 557-570, 2002.

[12] A. Friedman, R. Wolff, and A. Schuster, "Providing k-anonymity in data mining, Providing k-anonymity in data mining," VLDB J., Vol. 17, No. 4, pp. 789-804, 2008.

[13] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, "Blocking Anonymity Threats Raised by Frequent Itemset Mining," in Proc. of the Fifth IEEE Int'l Conf. on Data Mining, pp. 561-564, 2005.

[14] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, "k-anonymous patterns," in Proc. of the 9th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 10–21, 2005.

[15] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, "Anonymity Preserving Pattern Discovery," Int'l J. on VLDB, pp. 703-727, 2008.

[16] M. Terrovitis, N. Mamoulis, and P. Kalnis, "Privacy-preserving anonymization of set-valued data," in Proc. VLDB Endow., pp. 115-125, 2008.

[17] V. Ciriani and S. De and Capitani Vimercati and S. Foresti and P. Samarati, "Chapter 1 K-ANONYMOUS DATA MINING: A SURVEY," Privacy-preserving data mining, pp. 105136, Springer, 2008.

[18] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu, " Privacy-Preserving Data Publishing: A Survey of Recent Developments," ACM Computing Surveys, Vol. 42, No. 4, pp. 14:1-53, 2010.

[19] Charu C. Aggarwal,Philip S. Yu, "Privacy-preserving Data Mining: Models and Algorithms," Springer, 2008.

[20] BMS-WebView-1, the KDD-Cup2000, http://www.sigkdd.org/kddcup/index.php?section=2000&method=data.

[21] R.G. Pensa, A. Monreale, F. Pinelli, D. Pedreschi, "Pattern-Preserving k-Anonymization of Sequences and its Application to Mobility Data Mining," in Int. Workshop on Privacy in Location-Based Applications, 2008.

[22] IBM Quest, http://www.almaden.ibm.com/cs/quest/syndata.html