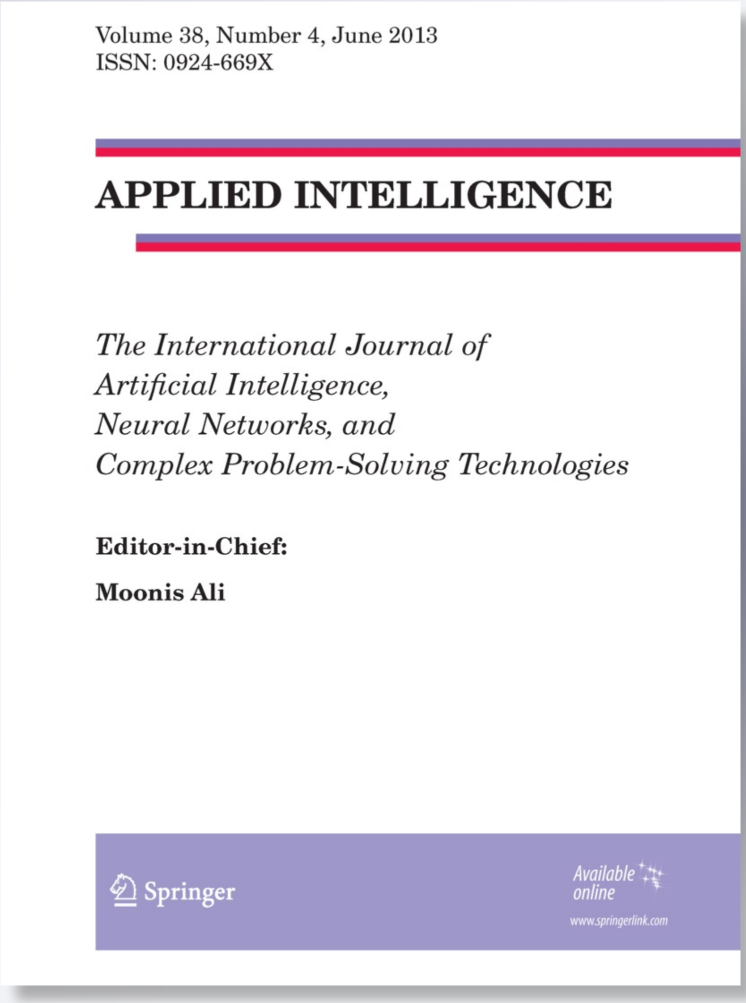
Using TF-IDF to hide sensitive itemsets

Tzung-Pei Hong, Chun-Wei Lin, Kuo-Tung Yang & Shyue-Liang Wang

Applied Intelligence
The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies

ISSN 0924-669X
Volume 38
Number 4

Appl Intell (2013) 38:502-510
DOI 10.1007/s10489-012-0377-5



Volume 38, Number 4, June 2013
ISSN: 0924-669X

APPLIED INTELLIGENCE

The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies

Editor-in-Chief:

Moonis Ali

Springer

Available online
www.springerlink.com

Springer

Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com”.

Springer

Author's personal copy

Appl Intell (2013) 38:502–510
DOI 10.1007/s10489-012-0377-5

Using TF-IDF to hide sensitive itemsets

Tzung-Pei Hong · Chun-Wei Lin · Kuo-Tung Yang · Shyue-Liang Wang

Published online: 26 August 2012
© Springer Science+Business Media, LLC 2012

Abstract Data mining technology helps extract usable knowledge from large data sets. The process of data collection and data dissemination may, however, result in an inherent risk of privacy threats. Some sensitive or private information about individuals, businesses and organizations needs to be suppressed before it is shared or published. The privacy-preserving data mining (PPDM) has thus become an important issue in recent years. In this paper, we propose an algorithm called SIF-IDF for modifying original databases in order to hide sensitive itemsets. It is a greedy approach based on the concept borrowed from the Term Frequency and Inverse Document Frequency (TF-IDF) in text mining. The above concept is used to evaluate the similarity degrees between the items in transactions and the desired sensitive itemsets and then selects appropriate items in some transactions to hide. The proposed algorithm can easily make good trade-offs between privacy preserving and execution time. Experimental results also show the performance of the proposed approach.

Keywords Privacy preserving · Data mining · TF-IDF · Greedy approach · Data sanitization

1 Introduction

In recent years, the privacy-preserving data mining (PPDM) has become an important issue due to the quick proliferation of electronic data in governments, corporations and non-profit organizations. Such data may implicitly contain confidential information and lead to privacy threats if they are misused. As the data mining technology rapidly progresses, getting users' privacy information through data mining technology has become easier. Privacy information includes some confidential information, such as social security numbers, address information, credit card numbers, credit ratings and customer purchasing behavior, among others. Besides, the range of privacy information may be extended to businesses as well. Based on business purposes, some shared information among companies may be extracted and analyzed by other partners, which may not only increase the benefits of the companies but also cause threats to sensitive data. This has led to increasing concerns about the privacy of the underlying data and the implicit knowledge on the data.

Verykios et al. [21] thus proposed a data sanitization process to hide sensitive knowledge by item addition or deletion. Their main concept was to reduce the supports of sensitive items such that the sensitive knowledge including

T.-P. Hong · K.-T. Yang
Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan

T.-P. Hong
e-mail: tphong@nuk.edu.tw

K.-T. Yang
e-mail: kdyang@nuk.edu.tw

T.-P. Hong
Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan

C.-W. Lin (✉)
Innovative Information Industry Research Center (IIIRC), School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, HIT Campus of Shenzhen University Town, Xili, Shenzhen 518055, P.R. China
e-mail: jerrylin@ieee.org

S.-L. Wang
Department of Information Management, National University of Kaohsiung, Kaohsiung, Taiwan
e-mail: slwang@nuk.edu.tw

Author's personal copy

the items might not be exposed. Verykios et al. then presented a taxonomy and reviewed some related approaches for privacy-preserving issues in association rules [20]. He generally classified the privacy issues into two categories, which were data hiding and knowledge hiding [19]. Data hiding concerned protection of underlying private data, but knowledge hiding focused on preserving high-level knowledge. Atallah et al. demonstrated that the problem was NP-hard [7]. Since then, many techniques have then been proposed to modify or transform data such that the data privacy can be preserved. For example, Samarati and Sweeney proposed the k-anonymity method for achieving the purpose of preserving privacy information [15]. Although the k-anonymity technology could successfully inhibit sensitive information, an appropriate balance between privacy and knowledge might not be guaranteed.

Zhu et al. then discussed what kind of public information type was suitable for not revealing sensitive data and insinuated that the k-anonymity technique might still have security problems [26, 27]. Wu et al. proposed a greedy approach to hide sensitive rules based on the designed FCET data structure for preserving the maximal frequent itemsets to efficiently speed up the progress of sanitization [24].

In text mining, the technique of *term frequency-inverse document frequency* (*TF-IDF*) [16] is usually used to evaluate how relevant a word in a corpus is to a document. It may be thought of as a statistical measure. The importance of a word to a document depends on its appearing number in the document, but is offset by the frequencies of the words in the other documents. In this paper, a novel greedy-based approach called *sensitive items frequency-inverse database frequency* (*SIF-IDF*) is proposed to evaluate the degree of transactions associated with given sensitive itemsets. It uses the concept and modifies the *TF-IDF* to design for reducing the frequencies of sensitive itemsets for data sanitization. Based on the greedy *SIF-IDF* algorithm, the user-specific sensitive itemsets can be completely hidden with reduced side effects. The proposed algorithm can thus easily make good trade-offs between privacy preserving and execution time. Experimental results also show the performance of the proposed approach.

The rest of this paper is organized as follows. Some related works are described in Sect. 2. The proposed *SIF-IDF* algorithm is stated in Sect. 3. An illustrative example is given in Sect. 4. Experimental results are then shown in Sect. 5. Conclusions are given in Sect. 6.

2 Review of related works

In this section, we review some related researches about this paper. Section 2.1 describes the data mining process. Section 2.2 introduces the general concept of data sanitization, which can be further classified as anonymity, blocking and encryption.

2.1 Data mining process

Years of effort in data mining have produced a variety of efficient techniques. Depending on the type of databases processed, the mining approaches may be classified as finding association rules [1, 4], classification rules [18], clustering rules [11, 14, 17], sequential patterns [2, 22], among others. Among them, association rules mining is the most commonly seen in data mining, such that the presence of certain items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal et al. proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [1, 4, 5]. They divided the mining process into two phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the count of an itemset appearing in the transactions was larger than a pre-defined threshold value (called the minimum support), the itemset was considered a large itemset. Itemsets containing only one item were processed first. Large itemsets containing only single items were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called the minimum confidence) were output as association rules.

2.2 Data sanitization

Years of effort in data mining have produced a variety of efficient techniques, which have also caused the problems of security and privacy threats [3, 10]. The research of privacy-preserving data mining (PPDM) has thus become a critical issue. PPDM is usually performed to hide sensitive information. Dasseni et al. proposed a hiding algorithm based on the hamming-distance approach to reduce the confidence or support values of association rules [8]. Three heuristic hiding approaches were thus proposed to increase the supports of antecedent parts, to decrease the supports of consequent parts, and to decrease the support of either the antecedent or the consequent parts, respectively. When the supports or the confidences of sensitive association rules were below user-specific minimum support thresholds, they could thus be hidden.

Oliveira and Zaïane [12] then introduced the multiple-rule hiding approach to efficiently hide sensitive itemsets. It required only two database scans no matter the number of sensitive itemsets. In the first database scan, the index

Author's personal copy

file was created to efficiently find sensitive itemsets within transactions. Three algorithms called MinFIA, MaxFIA and IGA were then used in the second database scan to remove minimal individual items. Amiri then proposed three heuristic approaches to hide multiple sensitive rules [6]. The first approach was called Aggregate, which computed the union of the supporting transactions for all sensitive itemsets and expelled the transaction that supports the most sensitive and the least non-sensitive itemsets. The second one was called Disaggregate, which aimed at removing individual items from transactions, rather than removing whole transactions. The third approach, called Hybrid, was a combination of the previous two. It uses the Aggregate approach to identify sensitive transactions and adopts the Disaggregate approach to selectively delete items from these transactions, until the sensitive knowledge has been hidden. Pontikakis et al. [13] then proposed two heuristics approaches based on data distortion. The first approach named priority-based distortion algorithm (PDA) was designed to reduce the confidences of sensitive rules by trying to decrease consequent items. The second approach called weight-based sorting distortion algorithm (WDA) was then proposed to prioritize selection of sanitized transactions. It used the priority values to weight the transactions based on effective data structures. Wang et al. proposed two algorithms called DCIS (Decrease Confidence by Increase Support) and DCDS (Decrease Confidence by Decrease Support), to automatically hide the collaborative recommendation association rules without premining and selection of hidden rules [23].

The optimal sanitization of databases is, in general, regarded as an NP-hard problem. Atallah et al. [7] proved that selecting which data to modify or sanitize was also NP-hard. Their proof was based on the reduction from the NP-hard problem of hitting-sets [9]. The hitting-set problem was first proven NP-hard. The PPDM problem was then reduced to the hitting-set problem in polynomial time. In this case, the PPDM problem could be said an NP-hard problem as well and could not be solved in polynomial time for now. That paper provided a solid theoretical background to explain that PPDM was a difficult issue.

3 The proposed SIF-IDF approach for data sanitization

In the problem of PPDM, some basic concepts are borrowed from association rule mining. Thus before exploring the PPDM's issue, we need to know the definition of association rule mining. Agrawal extended and formalized the problem as follows [4]. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction $T \subseteq D$ consists of a set of items, such that $T \subseteq I$. Each transaction T has a unique identifier, called it TID. A set of items $X \subset I$ is called an itemset. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$. Usually, Y consists of only a single item.

We say an association rule $X \Rightarrow Y$ holds in a database D if the following two factors are satisfied. The first one is the support condition, which is defined as at least s % of the transactions in D contain $X \cup Y$. It can be thought of as a measure of the frequency of a rule, and is expressed by $\frac{|X \cup Y|}{N} \geq s$, where N is the number of transactions in D. The second factor is the confidence condition, which is defined as at least c % of transactions with the itemset X also contains Y. It is thus a measure of the strength of the rule, and is expressed by $\frac{|X \cup Y|}{|X|} \geq c$.

In privacy-preserving data mining, users may pre-specify a set of sensitive itemsets $H = \{\{h_1\}, \{h_2\}, \dots, \{h_i\}\}$, which may be mined out from a database but is sensitive. We aim at preventing these sensitive itemsets being disclosed, and a solution is to reduce the frequencies of the sensitive itemsets from D. Let the modified database be denoted D'. Thus, each sensitive itemset will not have enough support to be frequent in D'. This kind of approaches can be thought of as support-based ones, and have to satisfy the constraint of $|h_i|/N' < s$, where N' is the number of transactions in D' and $|h_i|$ is number of occurrences of the sensitive itemset h_i. In addition to hiding the sensitive itemsets from being mined, some other goals have been set when the original database is sanitized. For example, all of the non-sensitive rules should be successfully mined from the sanitized database D'. Besides, the rules that are not found in the original database D should not be generated from the sanitized database D'.

In this paper, the sensitive itemsets to be hidden must be pre-defined by users. For achieving this purpose, new items or transactions may be inserted or old items or transactions may be deleted or modified. Here, the deletion of items in PPDM is thus used for hiding sensitive itemsets or knowledge which reduces the support of the rules below the user specified security threshold. It uses and modifies the concept of *TF-IDF* [16] in text mining to evaluate the degrees of transactions associated with given sensitive itemsets. The measure for the *sensitive items frequency-inverse database frequency* (*SIF-IDF*) value of a transaction T_i is defined as follows:

$$\mathit{SIF\text{-}IDF}(T_i) = \sum_{j=1}^{n} \left(\frac{|si_{ij}|}{|T_i|} \times \sum_{k=1}^{p} \log \frac{|n|}{|f_k - \mathit{MRC}_k|} \right),$$

where $|si_{ij}|$ is the number of sensitive items contained in the j-th sensitive itemset in T_i, and $|T_i|$ is the number of items in transaction T_i, $|n|$ is the number of records in a database, $|f_k|$ is the frequency count of each item, and $|\mathit{MRC}_k|$ is the maximum reduced count of each item.

The above formula consists of two components. One is the *sensitive items frequency* (*SIF*) and the other is the *inverse database frequency* (*IDF*). The *sensitive items frequency* (*SIF*) value is measured for each sensitive itemset si_j

Author's personal copy

in a transaction T_i. It is calculated as the number ($|si_{ij}|$) of sensitive items in T_i which are included in an assigned sensitive itemset si_j divided by the number of all the items in T_i. On the contrary, the *inverse database frequency* (*IDF*) value shows the influence degree of the sensitive itemsets within a transaction by considering the whole database. In this paper, the *SIF-IDF* value of each transaction is calculated and is used to measure whether a transaction has a large number of sensitive items but with less influence to other transactions. The transactions with high *SIF-IDF* values are considered to be processed with high probabilities for sanitization.

The proposed approach first calculates the *maximum reduced count* (*MRC*) of each item in the database. In doing this, the *reduced count* value (RC_{kj}) of each item i_k is first calculated for each sensitive itemset si_j as $f_j - s \times n + 1$ if si_j includes i_k and as 0 otherwise, where f_j is the occurrence frequency of the sensitive itemset si_j in the database, s is the minimum support threshold, and n is the number of transactions in the database, $1 \le j \le m$, $1 \le k \le p$.

The *IDF* value of each item is then calculated as the number of transactions in the database divided by the occurrence frequency of a processed item minus its *MRC* value. The *IDF* value for each sensitive itemset is then estimated as the summation of the items contained in the itemset. That is, the *SIF-IDF* value of each transaction is the summation of the *SIF* values of the sensitive itemsets appearing in a transaction multiplied by its corresponding *IDF* value. The transactions are then sorted in a descending order of their *SIF-IDF* values. The order is used as the processing order of the transactions for the proposed algorithm. In data sanitization, an item with a higher occurrence frequency in the sensitive itemsets may be considered to have a larger influence than the ones with a lower occurrence frequency. The sensitive items in the processed transactions are then deleted according to the ordering of their occurrence frequencies. This procedure is repeated until the set of sensitive itemsets becomes **null**, which indicates all the supports of the sensitive itemsets are under the user-specific minimum support threshold. The proposed SIF-IDF algorithm is then described in details as follows.

The proposed algorithm

INPUT: A transaction dataset $D = \{T_1, T_2, \ldots, T_i, \ldots, T_n\}$ with a set of p items $I = \{i_1, i_2, \ldots, i_k, \ldots, i_p\}$, a user-specific minimum support threshold s, and a set of m user-specified sensitive itemsets $S = \{si_1, si_2, \ldots, si_j, \ldots, si_m\}$.

OUTPUT: A sanitized database with no sensitive rules mined out.

STEP 1: Find the transactions with sensitive itemsets in the database D.

STEP 2: Calculate the sensitive items frequency (SIF_{ij}) value of each sensitive itemset si_j in each transaction T_i as:

$$SIF_{ij} = \frac{|si_{ij}|}{|T_i|},$$

where $|si_{ij}|$ is the number of sensitive items in T_i which appears in si_j, and $|T_i|$ is the number of items in T_i.

STEP 3: Calculate the value of the *inverse database frequency* (*IDF*) of each sensitive itemset in each transaction by the following substeps.

Substep 3-1: Calculate the reduced count value (RC_{kj}) of each item i_k for each sensitive itemset si_j as $f_j - s \times n + 1$ if si_j includes i_k and as 0 otherwise, where f_j is the occurrence frequency of the sensitive itemset si_j in the database, s is the minimum support threshold, and n is the number of transactions in the database, $1 \le j \le m$, $1 \le k \le p$.

Substep 3-2: Calculate the maximum reduced count value (MRC_k) of each item i_k as:

$$MRC_k = \max_{j=1}^{m} RC_{kj}.$$

Substep 3-3: Calculate the inverse database frequency (IDF_k) value of each items i_k as follows:

$$IDF_k = \log \frac{|n|}{|f_k - MRC_k|},$$

where f_k is the occurrence count of item i_k in the database.

Substep 3-4: Sum the *IDF* values of all sensitive items within sensitive itemsets and calculate the *SIF-IDF* value for each transaction as follows:

$$\mathit{SIF\text{-}IDF}(T_i) = \sum_{i=1}^{n} \left(\frac{|si_{ij}|}{|T_i|} \times \sum_{k=1}^{p} \log \frac{|n|}{|f_k - MRC_k|} \right).$$

STEP 4: Find the transaction (T_b) which has the best SIF-IDF value.

STEP 5: Process the transaction T_b to prune appropriate items by the following substeps.

Substep 5-1: Sort the items in a descending order of their occurrence frequencies within the sensitive itemsets.

Substep 5-2: Find the first sensitive item ($item_o$) in T_b according to the sorted order obtained in Substep 5-1.

Substep 5-3: Delete the item ($item_o$) from the transaction.

STEP 6: Update the occurrence frequencies of the sensitive itemsets.

STEP 7: Repeat STEPS 2 to 6 until the set of sensitive itemsets is **null**, which indicates that the supports of all the sensitive itemsets are below the user-specific minimum support threshold s.

Author's personal copy

Table 1 A database example with 10 transactions

TID	Item
T_1	a, b, c, d, f, g, h
T_2	a, b, d, e
T_3	b, c, d, f, g, h
T_4	a, b, c, f, h
T_5	c, d, e, g, i
T_6	a, c, f, i
T_7	b, c, d, e, f, g
T_8	c, d, f, h, i
T_9	a, d, e, f, i
T_{10}	a, c, e, f, h

Table 2 The *SIF* values of each sensitive itemset in each transaction

TID	Item	SIF_{cfh}	SIF_{af}	SIF_c
T_1	a, b, c, d, f, g, h	3/7	2/7	1/7
T_2	a, b, d, e	0/4	1/4	0/4
T_3	b, c, d, f, g, h	3/6	1/6	1/6
T_4	a, b, c, f, h	3/5	2/5	1/5
T_5	c, d, e, g, i	1/5	0/5	1/5
T_6	a, c, f, i	2/4	2/4	1/4
T_7	b, c, d, e, f, g	2/6	1/6	1/6
T_8	c, d, f, h, i	3/5	1/5	1/5
T_9	a, d, e, f, i	1/5	1/5	0/5
T_{10}	a, c, e, f, h	3/5	2/5	1/5

Table 3 The *MRC* values of all the items

Item	RC_{cfh}	RC_{af}	RC_c	*MRC*
a	–	2	–	2
b	–	–	–	0
c	2	–	5	5
d	–	–	–	0
e	–	–	–	0
f	2	2	–	2
h	2	–	–	2
i	–	–	–	0

Table 4 The *IDF* value of each item

Item	Count	*MRC*	*IDF*	Item
a	6	2	0.398	a
b	5	0	0.301	b
c	8	5	0.523	c
d	7	0	0.155	d
e	5	0	0.301	e
f	8	2	0.222	f
h	5	2	0.523	h
i	4	0	0.398	i

4 An illustrative example

In this section, an example is given to demonstrate the proposed *sensitive items frequency-inverse database frequency* (*SIF-IDF*) algorithm for privacy preserving data mining (PPDM). Assume a database shown in Table 1 is used as the example. It consists of 10 transactions and 9 items, denoted a to i.

Assume the set of user-specific sensitive itemsets S is $\{cfh, af, c\}$. Also assume the user-specified minimum support threshold is set at 40 %, which indicates that the minimum count is 0.4×10, which is 4. The proposed approach proceeds as follows to hide the sensitive itemsets in order of avoiding being mined from the database.

STEP 1: The transactions with sensitive itemsets in the database are found and kept. In this example, all the 10 transactions contain at least one sensitive itemset. All of them are then kept for later processing.

STEP 2: The *sensitive items frequency* (*SIF*) value of each sensitive itemset in each transaction is calculated. Take the first transaction as an example to illustrate the step. The first transaction includes the following seven items: $\{a, b, c, d, f, g, h\}$. The given sensitive itemsets include $\{cfh, af, c\}$. The appearing sensitive items in the first transaction for the sensitive itemset $\{cfh\}$ are c, f, h, and the number is 3. Similarly, the numbers of the appearing sensitive items in the first transaction for the sensitive itemsets $\{af\}$ and $\{c\}$ are 2 and 1, respectively. Thus, the *SIF* values of each sensitive itemset in the first transaction are calculated as 3/7, 2/7 and 1/7, respectively. The *SIF* values of each sensitive itemset in the other transactions could be found in a similar way. The results are shown in Table 2.

STEP 3: The *inverse database frequency* (*IDF*) value of each sensitive itemset in each transaction is calculated. In this step, the *reduced count* (*RC*) of each item for each sensitive itemset is first calculated and the maximum of the *RC* values of each item is found as the *MRC* value. Take item a as an example. The *MRC* value of an item a is calculated as $\max\{0, 5 - 0.4 \times 10 + 1, 0\}$, which is $\max\{0, 2, 0\} = \{2\}$. The *MRC* values of the other items can be found in the same way. The results are shown in Table 3.

The *IDF* value of each item is then calculated. Take item a as an example. The occurrence count of item a is 6 and the *MRC* value is 2. Its *IDF* value is then calculated as $\log(10/(6-2))$, which is 0.398. The *IDF* values of all the items are shown in Table 4.

The *IDF* value of each sensitive itemset in each transaction is then calculated. Take the first transaction for the first sensitive itemset $\{cfh\}$ as an example to illustrate the process. The *IDF* value of $\{cfh\}$ in the first transaction is the

Author's personal copy

Table 5 The *IDF* value of each sensitive itemset in each transaction

TID	IDF_{cfh}	IDF_{af}	IDF_c
T_1	1.268	0.62	0.523
T_2	0	0.398	0
T_3	1.268	0.222	0.523
T_4	1.268	0.62	0.523
T_5	0.523	0	0.523
T_6	0.745	0.62	0.523
T_7	0.523	0.222	0.523
T_8	1.268	0.222	0.523
T_9	0.222	0.62	0
T_{10}	1.268	0.62	0.523

Table 6 The *SIF-IDF* values for all the transactions

TID	SIF_1	IDF_1	SIF_2	IDF_2	SIF_3	IDF_3	SIF-IDF
T_1	3/7	1.268	2/7	0.62	1/7	0.523	0.795
T_2	0/4	0	1/4	0.398	0/4	0	0.099
T_3	3/6	1.268	1/6	0.222	1/6	0.523	0.758
T_4	3/5	1.268	2/5	0.62	1/5	0.523	1.113
T_5	1/5	0.523	0/5	0	1/5	0.523	0.209
T_6	2/4	0.745	2/4	0.62	1/4	0.523	0.813
T_7	2/6	0.523	1/6	0.222	1/6	0.523	0.299
T_8	3/5	1.268	1/5	0.222	1/5	0.523	0.901
T_9	1/5	0.222	1/5	0.62	0/5	0	0.168
T_{10}	3/5	1.268	2/5	0.62	1/5	0.523	1.113

Table 7 The sorted transactions according to the *SIF-IDF* values

TID	Item	SIF-IDF
T_4	a, b, c, f, h	1.113
T_{10}	a, c, e, f, h	1.113
T_8	c, d, f, h, i	0.596
T_6	a, c, f, i	0.813
T_1	a, b, c, d, f, g, h	0.795
T_3	b, c, d, f, g, h	0.758
T_7	b, c, d, e, f, g	0.299
T_5	c, d, e, g, i	0.209
T_9	a, d, e, f, i	0.168
T_2	a, b, d, e	0.099

Table 8 The final sanitized result in the database

TID	Item
T_1	a, b, d, f, g, h
T_2	a, b, d, e
T_3	b, c, d, g, h
T_4	a, b, h
T_5	c, d, e, g, i
T_6	a, f, i
T_7	b, c, d, e, f, g
T_8	d, f, h, i
T_9	a, d, e, f, i
T_{10}	a, e, h

sum of the *IDF* values of the three items c, f and h, which is $0.523 + 0.222 + 0.523$ and is equal to 1.268. All the results after this step are shown in Table 5.

The *SIF-IDF* value of a sensitive itemset in each transaction is then calculated as the *SIF* value of the sensitive itemset multiplied by its *IDF* value in the transaction. Take the first transaction as an example to illustrate the process. The *SIF* value of the sensitive itemset $\{cfh\}$ in the first transaction is 3/7 as shown in Table 2, and its *IDF* value is 1.268 as shown in Table 5. The *SIF-IDF* value of $\{cfh\}$ is then calculated as $\frac{3}{7} \times 1.268$, which is 0.5434. The other *SIF-IDF* values for the two sensitive itemsets $\{af\}$ and $\{c\}$ are calculated as 0.1771 and 0.0747, respectively. That is, the *SIF-IDF* value of the first transaction is summed as $0.5434 + 0.1771 + 0.0747$, which is 0.795. The other transactions are processed in the same way. After that, the results are shown in Table 6.

STEP 4: The transactions in Tables 4–6 are sorted in the descending order of their *SIF-IDF* values. The results are then shown in Table 7.

STEP 5: The transactions are processed in the above descending order to prune appropriate items. In this example, the set of sensitive itemsets is $\{cfh, af, c\}$. The occurrence frequencies of the items within the sensitive itemsets are $\{a:1, c:2, f:2, h:1\}$. The items are then sorted in the descending order of their frequencies as $\{c:2, f:2, a:1, h:1\}$, which will be used as the deletion sequence. From Table 7, transaction 4 has the best *SIF-IDF* value among 10 transactions. It is thus selected to be processed. The item c in transaction 4 is then first selected to be deleted.

STEP 6: After an item c is deleted from the fourth transaction, the new occurrence frequencies of the sensitive itemsets in the transactions are symmetric updated. The sensitive itemsets with their occurrence frequencies are then updated from $\{cfh:5, af:5, c:8\}$ to $\{cfh:4, af:5, c:7\}$.

STEP 7: STEPs 2 to 6 are then repeated until the supports of all the sensitive itemsets are below the minimum count. The results of the final sanitized database in the example are shown in Table 8.

5 Experimental results

Experiments were made to show the performance of the proposed approaches. They were performed on a Pentium IV 2 GHz CPU with 512 MB RAM based on the Mandriva

Author's personal copy

Table 9 The parameters of the two databases

Database	# of transactions	# of items	Maximum transaction size	Average transaction size
Web-View1	59,602	497	267	2.5
Web-View2	77,512	3,340	161	5.0

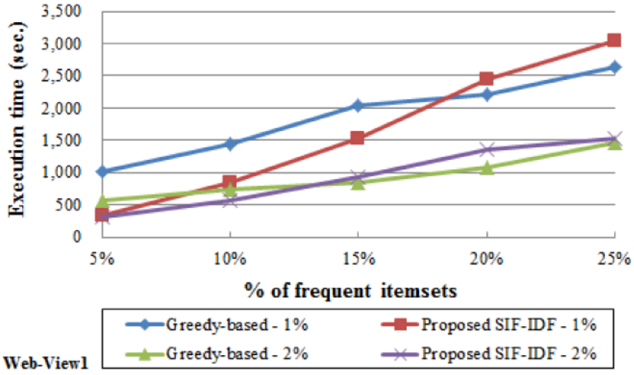


Fig. 1 The comparison of execution time for the two different minimum support thresholds in Web-View1

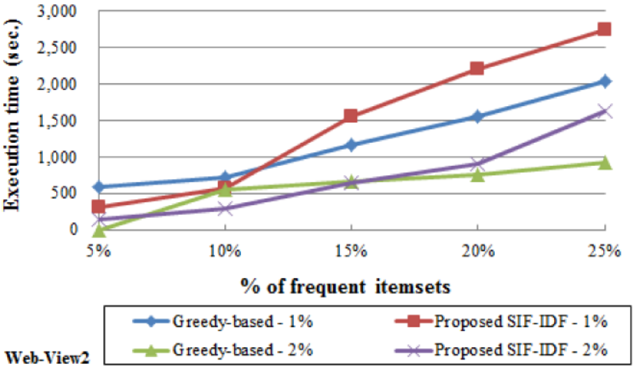


Fig. 2 The comparison of execution time for the two different minimum support thresholds in Web-View2

platform. The parameters of the two databases [25] used in the experiments were shown in Table 9. A greedy-based approach for hiding sensitive itemsets [24] was also executed for comparison in both the execution time and the number of deleted transactions. It defined four different criteria for decreasing or increasing the supports of sensitive itemsets for data sanitization.

In the experiments, the minimum support thresholds were set at 1 % and 2 %. The numbers of sensitive itemsets were randomly generated according to percentages of the numbers of frequent itemsets. The execution time for the two different minimum support thresholds in the two different databases was respectively shown in Figs. 1 and 2.

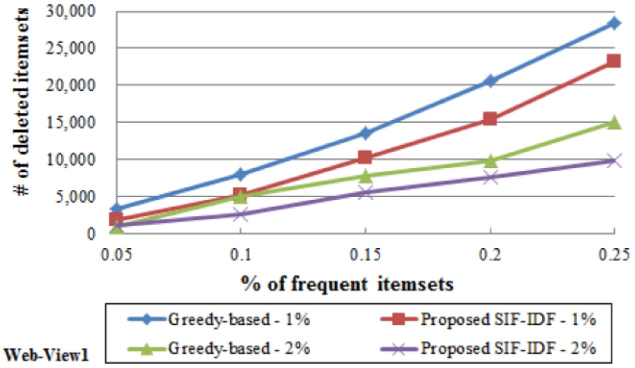


Fig. 3 The comparison of the numbers of deleted itemsets for the two different minimum support thresholds in Web-View1

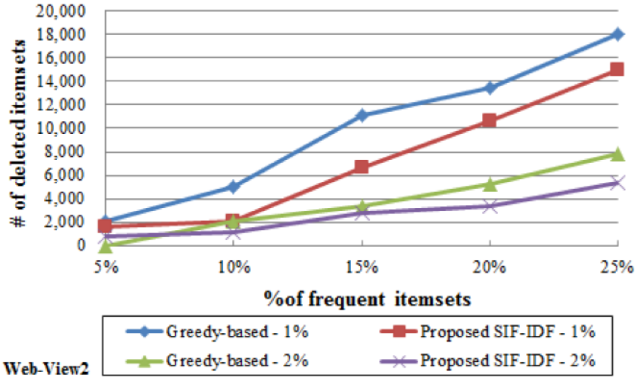


Fig. 4 The comparison of the numbers of deleted itemsets for the two different minimum support thresholds in Web-View2

Also, the number of deleted itemsets in the two different databases were also respectively evaluated and shown in Figs. 3 and 4.

It could be observed from Figs. 1 and 2 that the execution time for the 2 % minimum support threshold was less that for 1 % in both the algorithms because a smaller threshold would cause more frequent itemsets. Besides, since the greedy-based algorithm [24] could process multiple transactions for inserting or deleting itemsets at the same time, it had better performance than our proposed SIF-IDF algorithm when the percentage of sensitive itemsets was set high (more sensitive itemsets needed to be processed). The proposed SIF-IDF algorithm, however, needed to delete a smaller number of transactions than the greedy-based approach did, as shown in Figs. 3 and 4. At last, the three different side effects (artificial rules, missing rules, and hiding failures) were also evaluated and the results were shown in Figs. 5 and 6.

In Figs. 5 and 6, α, β and γ represented the numbers of hiding failures, missing rules and artificial rules, respectively. From Figs. 5 and 6, it was obvious to see that the proposed algorithm was efficient without generating any hiding failure. That is, all the sensitive itemsets could be completely

Author's personal copy

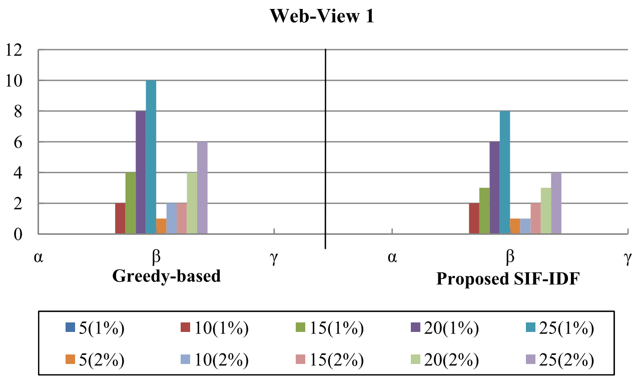


Fig. 5 The three side effects for the two different minimum support thresholds inWeb-View1 database

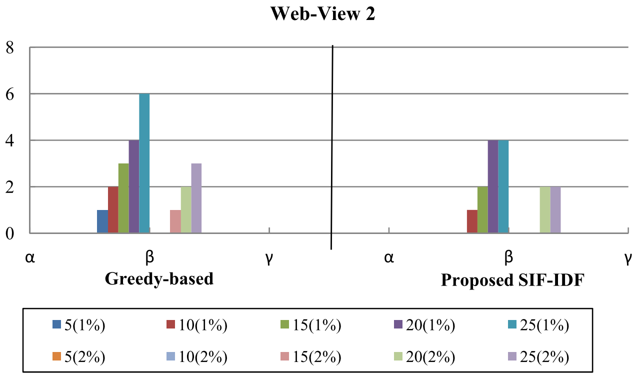


Fig. 6 The three side effects for the two different minimum support thresholds in Web-View2 database

hidden through the proposed algorithm. Besides, there was no artificial rule generated. The number of missing rules was larger than those of the other two kinds because the approach deleted some itemsets, thus causing some rules removed as well. Besides, the proposed SIF-IDF algorithm had less missing rules than the greedy-based algorithm.

6 Conclusions

Removing itemsets or transactions are commonly seen in privacy-preserving data mining (PPDM). In this paper, a greedy *SIF-IDF* algorithm is proposed for removing sensitive itemsets from transactions. It first evaluates the similarity between sensitive itemsets and transactions for minimizing the side effects, which is borrowed from the *TF-IDF* algorithm in information retrieval. It then calculates the *SIF-IDF* values of all transactions and sorts them in a descending order as the processing priority. The frequencies of items within sensitive itemsets are also calculated as the deletion priority in the processed transactions. The procedure is repeated until all the given sensitive itemsets are hidden. Based on the user-specified sensitive itemsets in the experiments, the proposed *SIF-IDF* algorithm can process all specified sensitive itemsets without little side effect in the two databases. In the future, we will try to combine other intelligent techniques to further improve the performance of the proposed approach.

References

1. Agrawal R, Srikant R (1994) Fast algorithm for mining association rules. In: The international conference on very large data bases, pp 487–499
2. Agrawal R, Srikant R (1995) Mining sequential patterns. In: The international conference on data engineering, pp 3–14
3. Agrawal R, Srikant R (2000) Privacy-preserving data mining. In: ACM SIGMOD international conference on management of data, pp 439–450
4. Agrawal R, Imielinski T, Sawmi A (1993) Mining association rules between sets of items in large databases. In: ACM SIGMOD international conference on management of data, pp 207–216
5. Agrawal R, Srikant R, Vu Q (1997) Mining association rules with item constraints. In: The international conference on knowledge discovery in databases and data mining, pp 67–73
6. Amiri A (2007) Dare to share: Protecting sensitive knowledge with data sanitization. Decis Support Syst 43(1):181–191
7. Atallah M, Bertino E, Elmagarmid A, Ibrahim M, Verykios VS (1999) Disclosure limitation of sensitive rules. In: IEEE knowledge and data engineering exchange workshop, pp 45–52
8. Dasseni E, Verykios VS, Elmagarmid AK, Bertino E (2001) Hiding association rules by using confidence and support. In: The international workshop on information hiding, pp 369–383
9. Garey MR, Johnson DS (1979) Computers and intractability: A guide to the theory of NP-completeness. W. H. Freeman, New York
10. Leary DEO (1991) Knowledge discovery as a threat to database security. In: Knowledge discovery in databases, pp 507–516
11. Liu F, Lu Z, Lu S (2001) Mining association rules using clustering. Intell Data Anal 5:309–326
12. Oliveira SRM, Zaïane OR (2002) Privacy preserving frequent itemset mining. In: IEEE international conference on privacy, security and data mining, pp 43–54
13. Pontikakis ED, Tsitsonis AA, Verykios VS (2004) An experimental study of distortion-based techniques for association rule hiding. In: The conference on database security, pp 325–339
14. Popović B, Janev M, Pekar D, Jakovljević N, Gnjatović M, Secujskǐ M, Delic V (2012) A novel split-and-merge algorithm for hierarchical clustering of Gaussian mixture models. Appl Intell. doi:10.1007/s10489-011-0333-9
15. Samarati P, Sweeney L (1998) Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report
16. Salton G, Fox EA, Wu H (1983) Extended boolean information retrieval. Commun ACM 26(2):1022–1036
17. Tsai C-F, Yeh H-F, Chang J-F, Liu N-H (2010) PHD: an efficient data clustering scheme using partition space technique for knowledge discovery in large databases. Appl Intell 33:39–53
18. Verma B, Hassan SZ (2011) Hybrid ensemble approach for classification. Appl Intell 34:258–278
19. Verykios VS, Gkoulalas-Divanis A (2008) Privacy-preserving data mining models and algorithms, Chap 11, pp 267–289
20. Verykios VS, Gkoulalas-Divanis A (2008) A survey of association rule hiding methods for privacy. In: The Kluwer international series on advances in database systems, vol 34, pp 267–289

Author's personal copy

21. Verykios VS, Elmagarmid A, Bertino E, Saygin Y, Dasseni E (2004) Association rule hiding. IEEE Trans Knowl Data Eng 16(4):434–447
22. Wang CY, Hong TP, Tseng SS (2002) Maintenance of discovered sequential patterns for record deletion. Intell Data Anal 6:399–410
23. Wang SL, Patel D, Jafari A, Hong TP (2007) Hiding collaborative recommendation association rules. Appl Intell 27(1):67–77
24. Wu CM, Huang YF, Chen JY (2009) Privacy preserving association rules by using greedy approach. In: WRI world congress on computer science and information engineering, pp 61–65
25. Zheng Z, Kohavi R, Mason L (2001) Real world performance of association rule algorithms. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 401–406
26. Zhu Z, Du WL (2010) K-anonymous association rule hiding. In: ACM symposium on information, computer and communications security, pp 305–309
27. Zhu Z, Wang G, Du W (2009) Deriving private information from association rule mining results. In: IEEE international conference on data engineering, pp 18–29