Knowl Inf Syst (2016) 47:625–645
DOI 10.1007/s10115-015-0862-3

REGULAR PAPER

# A transversal hypergraph approach for the frequent itemset hiding problem

**Elias C. Stavropoulos**[1,2] · **Vassilios S. Verykios**[1,3] · **Vasileios Kagklis**[1,4]

Received: 9 December 2014 / Revised: 20 March 2015 / Accepted: 6 July 2015 /
Published online: 17 July 2015


**Abstract** We propose a methodology for hiding all sensitive frequent itemsets in a transaction database. Our methodology relies on a novel technique that enumerates the minimal transversals of a hypergraph in order to induce the ideal border between frequent and sensitive itemsets. The ideal border is then utilized to formulate an integer linear program (ILP) that answers whether a feasible sanitized database that attains the ideal border, exists. The solution of the program identifies the set of transactions that need to be modified (sanitized) so that the hiding can be achieved with the maximum accuracy. If no solution exists, we modify the ILP by relaxing the constraints needed to be satisfied so that the sanitized database preserves the privacy with guarantee but with minimum effect in data quality. Experimental evaluation of the proposed approach on a number of real datasets has shown that the produced sanitized databases exhibit higher accuracy when compared with the solutions of other well-known approaches.

**Keywords** Privacy-preserving data mining · Hiding frequent itemsets · Transversal hypergraph generation

✉ Elias C. Stavropoulos
estavrop@teiwest.gr

Vassilios S. Verykios
verykios@eap.gr

Vasileios Kagklis
kagklis@ceid.upatras.gr

1 Educational Content, Methodology and Technology Laboratory, Hellenic Open University, 278, Patron Claus Str., 263 35 Patras, Greece

2 Business Administration Department, Technological Educational Institute of Western Greece, 263 34 Patras, Greece

3 School of Science and Technology, Hellenic Open University, 263 35 Patras, Greece

4 Computer Engineering and Informatics Department, University of Patras, 265 04 Patras, Greece

## 1 Introduction

*Privacy-Preserving Data Mining* [1,3] is a new research area that combines methods of data mining and data privacy to perform data mining while preserving their privacy. The constant and incremental need for the continuous flow and accumulation of data that comprises a driving force for the application of data mining, most of the times affects in a negative way the privacy of the subjects recorded in the data. More specifically, it is well known that data about human subjects may reveal their identity, while business-related data may compromise trade secrets, like a forthcoming company merge. The main goal of privacy-preserving data mining is therefore to facilitate the mining of the data, while prohibiting the leakage of sensitive information. It is universally accepted that simplistic approaches to the preservation of privacy, like the removal of identifiers and sensitive attributes, are inadequate in accomplishing the desired effects, since there is an ample number of counter measures like the use of quasi-identifiers and inference-based approaches that under certain circumstances can easily breach the privacy of the data [27,54,55].

Privacy-preserving data mining techniques can be classified as *input privacy* and *output privacy* approaches. Input privacy techniques [3,23,51] are exploring approaches to protect the sensitive data by some kind of transformation so that no explicit information is visible, by ensuring at the same time, the maximum utility of the transformed data for data mining purposes. Output privacy approaches [15,18,41] are inclined in the masking or in the covering up (camouflage) of the induced patterns, so that sensitive knowledge is stripped off from the raw data. Among the output privacy techniques, the so-called *knowledge hiding* [9] techniques, also known as *data sanitization*, are used to transform the input data in such a way that specific knowledge is removed, or hidden, from the data. Depending on the type of knowledge that we need to hide, there are different forms of algorithms used for knowledge hiding. In the context of the frequent itemset mining [2], the one investigated in this paper is known as a *frequent itemset hiding* algorithm, since its goal is to hide *frequent itemsets*, i.e., itemsets such that the number of their occurrences in the database is at least a predefined minimum support threshold, that are considered sensitive, and have to be hidden.

The *frequent itemset hiding problem*, which was introduced in [4] and has been shown to be NP-hard, investigates the sanitization of a database from a set of sensitive frequent itemsets, in such a way that (a) these sensitive itemsets cannot be mined from the sanitized database under any support threshold less than the minimum support threshold originally used for the mining of the itemsets in the initial database, (b) the quality of the sanitized data, with respect to its utility, is maximized and (c) the non-sensitive frequent itemsets maintain their frequent status in the sanitized database as well.

A wide variety of approximate workarounds have been proposed for this problem. Heuristic techniques, such as those proposed in [49,53], attempt to exploit border theory, so as to achieve hiding while introducing minimum side effects. Linear-programming techniques [30,40,45,47] formulate the problem as a linear program where its solution indicates which transactions or which specific items should be sanitized. Although heuristic techniques are easier to understand and more scalable than their ILP-based counterpart techniques, their results are less reliable about providing minimum side effects. On the other hand, ILP-based techniques can be more difficult to understand because of the nature of the formulation of the problem. However, ILP-based techniques are more reliable and found the best possible solution that the specific formulation can give. Nevertheless, the difficulty is inherent to the problem and no approximation guarantees with respect to the optimal solution exist for none of the above techniques. For example, in a transaction database that supports the frequent

itemsets *abc* and *bc* with a support count of 2, if *bc* is sensitive, any attempt to hide it would result in hiding *abc*, too; thus, no ideal solution exists. There is a strong connection of data mining tasks with the Kolmogorov Complexity of the data that measures the randomness of strings based on their information content (see, for example, [24]).

Recently, emphasis has been put on the *inverse frequent itemset mining problem*, where given a collection of itemsets, the computation of a transactional database such that these itemsets are frequent, is asked. Complexity issues on variants of the problem were firstly studied by Mielikäinen [48], then by Calders [16,17] and lately by Guzzo et al. [35,36]. The relation of the problem to the *probabilistic satisfiability* [29] and the *transversal hypergraph generation* [7,26] are also discussed there, and heuristic solutions for the problem are given.

For the frequent itemset hiding problem, a significant contribution could be an approach where given the set of frequent itemsets and a set of sensitive itemsets, it induces the ideal border between these two sets. In this paper, our aim is to propose an efficient technique that computes this ideal border and next utilizes it in order to enhance and evolve an existing integer linear-programming-based technique that is used for hiding sensitive itemsets. The solution of the program identifies the set of transactions that need to be sanitized, so that the hiding can be achieved with the maximum accuracy. If no solution exists, we relax the constraints needed to be satisfied, so that the resulting sanitized database would preserve privacy accuracy with the minimum impact on data quality. We establish the effectiveness of our approach by experimentally evaluating it on a number of real datasets and comparing it with other well-known approaches.

The rest of the paper is organized as follows. In Sect. 2, we briefly discuss previous work related to the frequent itemset hiding problem. In Sect. 3, we state the problem and give all the necessary background information. The relation of the problem with the transversal hypergraph generation is also given there. In Sect. 4, we formulate a method for enumerating the ideal borders of the sanitized database, based on the hypergraph theory. We present our proposed methodology in Sect. 5, and we experimentally evaluate and compare it with previous related methods in the next section. Finally, in Sect. 7, conclusions are given.

## 2 Related work

Sun and Yu [53] introduce a greedy border-based approach for hiding sensitive frequent itemsets. They propose a heuristic algorithm that takes advantage of the housekeeping between maximally non-sensitive and minimally sensitive frequent itemsets, giving an accurate and efficient hiding solution. Moustakides et al. [49] rely on the border revision theory presented in [53] to build an algorithm that implements the "maxmin" criterion. The algorithm hides sensitive itemsets by eliminating items so that the side effects to the minimum support itemsets in the positive border are minimized. The results are of similar accuracy, but are produced in a much more efficient way than the results in the original border-based algorithm.

Menon et al. [47] were the first to introduce an integer linear programming formulation of the frequent itemset hiding problem. The solution of the ILP indicates which transactions need to be sanitized in order to conceal the sensitive itemsets. The sanitization process is examined separately and independently of the linear programming solution. The formulation of this method completely ignores the impact of the hiding process upon the non-sensitive frequent itemsets.

Gkoulalas-Divanis and Verykios in [30] use binary integer programming combined with the border revision theory in order to develop two techniques for optimally solving the hiding problem. The first technique, known as the *Inline* approach, introduces binary variables into

the original database, whereas the second one, known as *Hybrid*, extends the original database with synthetically generated transactions. The aim of both approaches is to alter specific items in the database, or in its extension, respectively, so as to control the support of sensitive and non-sensitive frequent itemsets.

Leloglu et al. [45] improve the formulation of Menon's et al. [47] technique. The authors calculate a coefficient for each transaction. The coefficient of a transaction is the number of non-sensitive frequent itemsets that would stop being supported by the transaction, if the item contained by most of the sensitive itemsets, supported by this transaction, is removed. These coefficients are used to augment the objective function of the formulation presented in [47], in order to determine which transactions should be sanitized.

Finally, Kagklis et al. [40] introduce a more complex heuristic coefficient formulation than the one presented in [45]. The intuition behind this technique is to identify which non-sensitive frequent itemsets will most likely become infrequent, known as endangered itemsets, before applying the hiding process. Thus, a higher cost is assigned to transactions supporting endangered itemsets, so that these transactions will not be sanitized.

## 3 Problem formulation

Let $\mathcal{I} = \{i_1, i_2, i_3, \ldots, i_n\}$ be a set of distinct literals called *items*, and $\mathcal{P}(\mathcal{I})$ be the powerset of $\mathcal{I}$. A *k-itemset* $X$ is a nonempty subset of $\mathcal{I}$ with cardinality, or *length*, $k$. A *transaction* $T$ over $\mathcal{I}$ is a 2-tuple $T = \langle \text{tid}, t \rangle$, where tid is the identifier of $T$ and $t$ is an itemset. A transaction $T$ *supports* an itemset $X$ if $X \subseteq t$. A transaction database $\mathcal{D}$ is a set of transactions. We denote by $|\mathcal{D}|$ the size of $\mathcal{D}$, i.e., the number of transactions in $\mathcal{D}$. The *support count* $\text{supc}_{\mathcal{D}}(X)$ of $X$ in $\mathcal{D}$ is the number of transactions supporting $X$ and the *support* $\text{sup}_{\mathcal{D}}(X)$ is the number of transactions supporting $X$, over the total number of transactions in $\mathcal{D}$. A sample of a transaction database $\mathcal{D}_0$ of size $|\mathcal{D}_0| = 6$ is demonstrated in Table 1. Here, $\text{supc}_{\mathcal{D}_0}(bcd) = 1$ and $\text{sup}_{\mathcal{D}_0}(bcd) = 1/6$ while $\text{supc}_{\mathcal{D}_0}(ad) = 3$ and $\text{sup}_{\mathcal{D}_0}(ad) = 1/2$.

Given a user-specified support threshold $\sigma$, we call an itemset $X$ *$\sigma$-frequent*, or *frequent*, in $\mathcal{D}$ iff $\text{sup}_{\mathcal{D}}(X) \geq \sigma$, otherwise $X$ is *$\sigma$-infrequent*. Let $\mathcal{F}^{\sigma}_{\mathcal{D}} \subseteq \mathcal{P}(\mathcal{I})$ be the family of all $\sigma$-frequent itemsets in $\mathcal{D}$. The generation of all frequent itemsets is a significant task emerging in many areas of AI and Databases. Frequent itemsets may fulfill the interests of users, and mining them is important. According to the anti-monotonic Apriori property, subsets of $\sigma$-frequent itemsets are $\sigma$-frequent too, while supersets of $\sigma$-infrequent itemsets are also $\sigma$-infrequent. Even this property, frequent itemsets may be exponentially many and the process of enumerating them might be unaffordable.

A condensed representation of $\mathcal{F}^{\sigma}_{\mathcal{D}}$ can be obtain by its *borders*. The *positive border* $\mathcal{B}d^{+}(\mathcal{F}^{\sigma}_{\mathcal{D}})$ and the *negative border* $\mathcal{B}d^{-}(\mathcal{F}^{\sigma}_{\mathcal{D}})$ of $\mathcal{F}^{\sigma}_{\mathcal{D}}$ are defined, respectively, as

**Table 1** A sample of a transaction database $\mathcal{D}_0$

| Tid | Itemsets |
|---|---|
| 1 | *abcd* |
| 2 | *abc* |
| 3 | *abd* |
| 4 | *acd* |
| 5 | *cd* |
| 6 | *ac* |

**Table 2** Notation used in the paper

| Notation | Explanation |
|---|---|
| $\sup_D(X)$ | The support of itemset $X$ in database $\mathcal{D}$ |
| $\sigma$ | The user-specified support threshold |
| $\mathcal{F}_D^\sigma$, or $\mathcal{F}$ | The set of frequent itemsets of $\mathcal{D}$ |
| $\mathcal{S}$ | The set of sensitive frequent itemsets |
| $\mathcal{B}d^+, \mathcal{B}d^-$ | The positive and negative borders |
| $\mathcal{H}^c$ | The complement of hypergraph $\mathcal{H}$ |
| $Tr(\mathcal{H})$ | The transversal hypergraph of hypergraph $\mathcal{H}$ |
| $\mathcal{H}^+, \mathcal{H}^-$ | The hypergraphs corresponding to $\mathcal{B}d^+$ and $\mathcal{B}d^-$ |
| $\mathcal{H}_\mathcal{S}$ | The hypergraph corresponding to $\mathcal{S}$ |
| $\widetilde{F}$ | The ideal set of non-sensitive frequent itemsets |

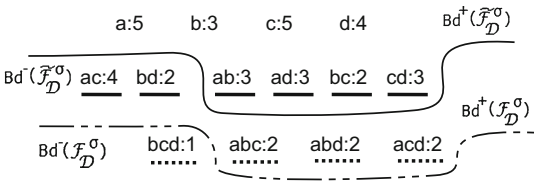


**Fig. 1** The lattice of itemsets in the sample database $\mathcal{D}_0$ along with its borders (*dashed line*), for $\sigma = 1/3$. The support count for every itemset is depicted and the boundary itemsets are underlined. The *borders* of the sanitized database are also shown (*solid line*)

$$\mathcal{B}d^+(\mathcal{F}_\mathcal{D}^\sigma) = \left\{X \in \mathcal{F}_\mathcal{D}^\sigma \,|\forall\, Y, X \subset Y \Rightarrow Y \notin \mathcal{F}_\mathcal{D}^\sigma\right\}, \quad \text{and}$$
$$\mathcal{B}d^-(\mathcal{F}_\mathcal{D}^\sigma) = \left\{X \in \mathcal{P}(\mathcal{I})\backslash\mathcal{F}_\mathcal{D}^\sigma \,|\,\forall\, Y, Y \subset X \Rightarrow Y \in \mathcal{F}_\mathcal{D}^\sigma\right\}.$$

The positive border is the family of all *maximal* frequent itemsets (i.e., those itemsets that are $\sigma$-frequent but no proper superset of them is frequent), and the negative border is the family of all *minimal* infrequent itemsets (i.e., those itemsets that are infrequent but no proper subset of them is infrequent). For the rest of the paper, we denote $\mathcal{F}_\mathcal{D}^\sigma$ with $\mathcal{F}$, for simplicity. Moreover, the notation used throughout the paper is summarized in Table 2.

In Fig. 1, we give with dashed line the positive border $\mathcal{B}d^+(\mathcal{F}) = \{abc, abd, acd\}$ and the negative border $\mathcal{B}d^-(\mathcal{F}) = \{bcd\}$ of the sample database $\mathcal{D}_0$ of Table 1, for $\sigma = 1/3$. For the shake of completeness, we include the whole lattice of the itemsets.

Let $\mathcal{S} \subseteq \mathcal{F}$ be a set of frequent itemsets that are sensitive and need to be hidden in order to protect their privacy. To hide this knowledge, one has to apply a sanitization process over the original transactional database $\mathcal{D}$ and derive a new database $\mathcal{D}'$ such that the support of every itemset in $\mathcal{D}'$ is below the support threshold $\sigma$. A formal definition of the problem is as follows:

**Definition 3.1** (Frequent Itemset Hiding Problem) Given a transaction database $\mathcal{D}$ over a set of items $\mathcal{I} = \{i_1, i_2, i_3, \ldots, i_n\}$, a support threshold $\sigma$, and a set of sensitive frequent itemsets $\mathcal{S}$, transform $\mathcal{D}$ into $\mathcal{D}'$ such that $sup_{\mathcal{D}'}(X) < \sigma$ for every $X \in \mathcal{D}'$.

### 3.1 Hypergraphs and minimal transversals

It was shown in [46] that positive and negative borders are closely related to minimal transversals of hypergraphs [7]. A *hypergraph* $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is a finite collection $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_m\}$ of sets (the *hyperedges*) over a finite set $\mathcal{V} = \{v_1, \ldots, v_n\}$ (the *nodes*), such that $\mathcal{E}_i \neq \emptyset$ $(i = 1, \ldots, m)$ and $\cup_{i=1}^{m} \mathcal{E}_i = \mathcal{V}$. The *complement* of $\mathcal{H}$ is defined as $\mathcal{H}^c = (\mathcal{V}, \mathcal{E}^c)$ where $\mathcal{E}^c = \{\mathcal{V} \backslash E \mid E \in \mathcal{E}\}$. A hypergraph $\mathcal{H}$ is *simple*, if for every pair $\mathcal{E}_i, \mathcal{E}_j \in \mathcal{E}$, $\mathcal{E}_j \subseteq \mathcal{E}_i \Rightarrow j = i$. A *transversal* of $\mathcal{H}$ is a set $\mathcal{T} \subseteq \mathcal{V}$ such that $\mathcal{T} \cap \mathcal{E}_i \neq \emptyset, \forall \mathcal{E}_i \in \mathcal{E}$, i.e., it *hits* all hyperedges of $\mathcal{H}$. A transversal $\mathcal{T}$ is *minimal*, if no proper subset of $\mathcal{T}$ is a transversal of $\mathcal{H}$. All minimal transversals of $\mathcal{H}$ form the *transversal hypergraph*, $Tr(\mathcal{H})$, of $\mathcal{H}$.

The transversal hypergraph generation (THG) is one of the most challenging problems on hypergraphs, with a vast number of practical applications, especially in Databases and Knowledge Discovery (for an exposition, see [20,21,33]). Its complexity, an open issue for the last four decades, strongly depends on the decision version (TH), where given two hypergraphs $\mathcal{H}$ and $\mathcal{G}$, we are asked whether $\mathcal{G} = Tr(\mathcal{H})$ holds. In general, TH is in co–NP, while it was shown in [22,42] that it can be solved with limited non-determinism and it was placed in co-NP[$\log^2 n$], the subclass of co-NP where only the first $O(\log^2 n)$ steps are non-deterministic, where $n$ is the combined size of the input hypergraphs. (For computational complexity issues and limited non-determinism, see, for example [28,31]). A recent result solves TH in quadratic logspace [32].

While several polynomial time cases for TH exist [12,20,21], for the general case the only remarkable result still remains the algorithm of Fredman and Khachiyan [26] that solves TH in subexponential time $n^{o(\log n)}$. This algorithm can be used as an oracle for solving THG in *incremental output-subexponential time* [34]. This is the best provable upper time bound yet, while several other approaches have been published in recent years [5,11,19,43,50] (see [37,56] for a worst-case analysis of these approaches).

### 3.2 The transversal hypergraph model

Consider now a bijective function $\mathcal{R}$ defined on $\mathcal{I}$, that *represents as sets* the elements of any subfamily of itemsets in $\mathcal{D}$. For example, $\mathcal{R}(\{acd, ac\}) = \{\{a, c, d\}, \{a, c\}\}$. Both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are polynomially computable. Function $\mathcal{R}$ just disguises itemsets of a transaction database into hyperedges of a hypergraph.

By setting $\mathcal{H}^+ = \mathcal{R}(\mathcal{B}d^+(\mathcal{F}))$ and $\mathcal{H}^- = \mathcal{R}(\mathcal{B}d^-(\mathcal{F}))$, the following fact is due to [46]:

$$\mathcal{H}^- = Tr((\mathcal{H}^+)^c) \tag{1}$$

Thus, the minimal $\sigma$-infrequent itemsets are exactly the minimal transversals of the complements of the maximal $\sigma$-frequent itemsets. Complexity issues on the generation of itemsets are discussed in [13]. By applying the *duality property* $Tr(Tr(\mathcal{H})) = \mathcal{H}$ that holds on hypergraphs [7], we obtain that

$$\mathcal{H}^+ = (Tr(\mathcal{H}^-))^c \tag{2}$$

By representing itemsets of a transaction database as hyperedges, we can move back and forth between both areas and gain from the good properties of hypergraphs. Given $\mathcal{B}d^-(\mathcal{F})$, we can apply Eq. (2) to enumerate $\mathcal{B}d^+(\mathcal{F})$. For example, for the sample database $\mathcal{D}_0$, the negative border $\mathcal{B}d^-(\mathcal{F}) = \{bcd\}$ corresponds to a hypergraph $\mathcal{H}^- = \{\{b, c, d\}\}$ with one single hyperedge. It is easy to see that the transversals of $\mathcal{H}^-$ are the singletons $\{b\}$, $\{c\}$, and $\{d\}$, since a node of hyperedge $\{b, c, d\}$ suffices to hit her. Thus, $Tr(\mathcal{H}^-) = \{\{b\}, \{c\}, \{d\}\}$ and the complementary hypergraph is $\mathcal{H}^+ = \{\{a, c, d\}, \{a, b, d\}, \{a, b, c\}\}$, that corresponds to the pos-

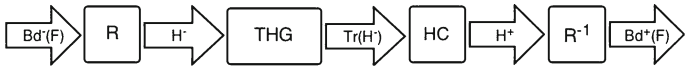


**Fig. 2** A transversal hypergraph-based process for enumerating $\mathcal{B}d^+(\mathcal{F})$, given $\mathcal{B}d^-(\mathcal{F})$. Functions R and $\mathsf{R}^{-1}$ convert transactions into hyperedges, and vice versa, while HC outputs the complement of the input hypergraph. Procedure THG is a transversal hypergraph generator

itive border $\mathcal{B}d^+(\mathcal{F}) = \{acd, abd, abc\}$. The whole process is visualized in Fig. 2. We shall apply this technique for computing the ideal border of the sanitized database, in the sequel.

See now that $\mathcal{F}$ and $\mathcal{S}$ determine the ideal set $\widetilde{\mathcal{F}}$ of non-sensitive frequent itemsets based on the Apriori property [2,33]: $\widetilde{\mathcal{F}} = \mathcal{F} \backslash \{X \in \mathcal{F} \mid \exists\, s \in \mathcal{S} : s \subseteq X\}$. Moreover, if, for example, $ab$ and $abc$ belong to $\mathcal{S}$, then it suffices $ab$ to be hidden, since based on the antimonotonicity property of the frequent itemsets, $abc$ will also be hidden during the process. Thus, it suffices to deal only with the set $Min(\mathcal{S})$ of the *minimal*, with respect to the above property, itemsets of $\mathcal{S}$ and transfer them to $\mathcal{B}d^-(\widetilde{\mathcal{F}})$.

To demonstrate the *border revision process*, consider the sample database $\mathcal{D}_0$ of Table 1 and let $\mathcal{S} = \{ac, bd, abc, acd\}$. The expected ideal borders, shown in Fig. 1 with solid line, are $\mathcal{B}d^+(\widetilde{\mathcal{F}}) = \{ab, ad, bc, cd\}$ and $\mathcal{B}d^-(\widetilde{\mathcal{F}}) = \{ac, bd\}$. In this example, the set of minimal itemsets of $\mathcal{S}$ is identical with the ideal negative border, even though $Min(\mathcal{S}) \subseteq \mathcal{B}d^-(\widetilde{\mathcal{F}})$, in general.

Due to the large number of frequent itemsets in $\widetilde{\mathcal{F}}$ for real datasets, it would be unrealistic to design a process that protects all of them from hiding. Instead, one can deal only with the borders of $\widetilde{\mathcal{F}}$. Sun and Yu [52] provide a sketch of an algorithm of high time and space complexity for computing $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ and $\mathcal{B}d^-(\widetilde{\mathcal{F}})$, since they firstly create the lattice of frequent itemsets $\mathcal{F}$ and next apply expensive set-oriented computations on it to determine the ideal borders. We focus only on $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ and hide all sensitive itemsets in $\mathcal{S}$ retaining as much as possible the frequent itemsets in $\mathcal{B}d^+(\widetilde{\mathcal{F}})$. Our aim is to come up with a sanitized database $\mathcal{D}'$ where its set of frequent itemsets $\mathcal{F}'$ approximates the ideal one $\widetilde{\mathcal{F}}$ with the maximum accuracy. The quality of the solution is measured by the *side effects* and the *information loss* of the sanitization procedure. We present our approach in the next section.

## 4 Enumeration of ideal borders

Given $\mathcal{D}, \sigma$, and $\mathcal{S}$, we want to determine the set $\widetilde{\mathcal{F}}$ of non-sensitive $\sigma$-frequent itemsets of the ideal sanitized database. As it is already discussed, we focus on the enumeration of $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ and we operate on the hypergraphs corresponding to $\mathcal{B}d^-(\mathcal{F})$ and the minimal itemsets in $\mathcal{S}$. Before we proceed, we state two useful, polynomial time computable, operations on hypergraphs [7]. For two hypergraphs $\mathcal{H} = \{\mathcal{E}_1, \ldots, \mathcal{E}_m\}$ and $\mathcal{G} = \{\mathcal{E}'_1, \ldots, \mathcal{E}'_{m'}\}$ (we skip mentioning the set of nodes $\mathcal{V}$), $Min(\mathcal{H}) = \{\mathcal{E}_i \in \mathcal{E} \mid \nexists\, \mathcal{E}_j \in \mathcal{E}, i \neq j : \mathcal{E}_j \subseteq \mathcal{E}_i\}$ and $\mathcal{H} \cup \mathcal{G} = \{\mathcal{E}_1, \ldots, \mathcal{E}_m, \mathcal{E}'_1, \ldots, \mathcal{E}'_{m'}\}$. Obviously, if $\mathcal{H}$ is simple, then $Min(\mathcal{H}) = \mathcal{H}$.

Let $\widetilde{\mathcal{H}}^+ = \mathcal{R}(\mathcal{B}d^+(\widetilde{\mathcal{F}}))$, $\widetilde{\mathcal{H}}^- = \mathcal{R}(\mathcal{B}d^-(\widetilde{\mathcal{F}}))$, and $\mathcal{H}_S = \mathcal{R}(\mathcal{S})$. It is easy to see that the set of minimal sensitive itemsets in $\mathcal{S}$ are exactly the minimal hyperedges of $\mathcal{H}_S$, i.e., $\mathcal{R}^{-1}(Min(\mathcal{H}_S))$. In this way, we actually have *a simple algorithm for computing the minimal sensitive itemsets* in $\mathcal{S}$; Convert $\mathcal{S}$ to $\mathcal{H}_S$, compute $Min(\mathcal{H}_S)$ and inversely convert $Min(\mathcal{H}_S)$ to obtain the set of minimal sensitive itemsets to be hidden. Next, we define $\mathcal{B}d^-(\widetilde{\mathcal{F}})$, in terms of hypergraphs.

**Proposition 4.1** $\widetilde{\mathcal{H}}^- = Min(\mathcal{H}^- \cup \mathcal{H}_S)$.

**Algorithm 1**: An algorithm for enumerating the ideal positive border of the sanitized transaction database.

**Input**: $\mathcal{S}$, $\mathcal{B}d^-(\mathcal{F})$
Set $\mathcal{H}^- = \mathcal{R}(\mathcal{B}d^-(\mathcal{F}))$ and $\mathcal{H}_S = \mathcal{R}(\mathcal{S})$ ;
Compute hypergraph $\widetilde{\mathcal{H}}^- = Min(\mathcal{H}^- \cup \mathcal{H}_S)$ ;
Call THG on input $\widetilde{\mathcal{H}}^-$ to generate $Tr(\widetilde{\mathcal{H}}^-)$ ;
Compute $\widetilde{\mathcal{H}}^+ = (Tr(\widetilde{\mathcal{H}}^-))^c$;
Set $\mathcal{B}d^+(\widetilde{\mathcal{F}}) = \mathcal{R}^{-1}(\widetilde{\mathcal{H}}^+)$ ;
**return** $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ ;

*Proof* Since sensitive itemsets in $\mathcal{S}$ have to be hidden, they must be added to the ideal negative border of the sanitized database, along with the itemsets of the negative border of the original database. $Min$ operator removes redundancy, since any superset of a $\sigma$-infrequent itemset is also $\sigma$-infrequent. □

**Theorem 4.1** $\widetilde{\mathcal{H}}^+ = (Tr(\widetilde{\mathcal{H}}^-))^c$.

*Proof* It follows from the definitions of $\widetilde{\mathcal{H}}^+$ and $\widetilde{\mathcal{H}}^-$ and Eq. (2). □

Continuing with our sample database $\mathcal{D}_0$, we have that $\mathcal{H}_S = \{\{a, c\}, \{b, d\}, \{a, b, c\}, \{a, c, d\}\}$, $\mathcal{H}^- \cup \mathcal{H}_S = \{\{b, c, d\}, \{a, c\}, \{b, d\}, \{a, b, c\}, \{a, c, d\}\}$, and $\widetilde{\mathcal{H}}^- = \{\{a, c\}, \{b, d\}\}$. We easily obtain $Tr(\widetilde{\mathcal{H}}^-) = \{\{a, b\}, \{a, d\}, \{b, c\}, \{c, d\}\}$, and $\widetilde{\mathcal{H}}^+ = \{\{c, d\}, \{b, c\}, \{a, d\}, \{a, b\}\}$. Hence, $\mathcal{R}^{-1}(\widetilde{\mathcal{H}}^+) = \{cd, bc, ad, ab\}$ (cf. the ideal positive border in Fig. 1 with solid line).

Theorem 4.1 allows the computation of $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ without looking at $\mathcal{D}$. It only assumes the knowledge of $\mathcal{S}$ and $\mathcal{B}d^-(\mathcal{F})$. The latter can be computed by any frequent itemset mining algorithm, like the Apriori algorithm [2,33]. All our ideas are incorporated in Algorithm 1. Procedure THG can be replaced by any algorithm for solving the Transversal Hypergraph Generation problem (see [37,50] and references in them). Time complexity of Algorithm 1 depends on the size of $\mathcal{B}d^-(\mathcal{F})$ (that, in the worst case may be exponential), and on the execution time of THG. Its correctness is a consequence of Theorem 4.1.

## 5 The enhanced integer linear programming methodology

We will demonstrate now the effectiveness of the ideal border enumeration Algorithm 1 by applying it to extend and improve the integer linear program formulation presented by Menon et al. in [47] for the solution of the frequent itemset hiding problem. The aim is to sanitize the transactions of $\mathcal{D}$, in a minimum way, such that all sensitive frequent itemsets to lose their support and become infrequent. In the ILP formulation 1, the objective function (3), defined on variables $x_i$, $i = 1, \ldots, |\mathcal{D}|$, minimizes the number of transactions sanitized, where $x_i = 1$ if transaction $T_i$ is sanitized, and $x_i = 0$, otherwise (as constraint (5) imposes). Finally, constraint (4) indicates that more than $\text{supc}_{\mathcal{D}}(X_j) - \sigma$ transactions supporting each itemset $X_j$ have to be sanitized so that all sensitive itemsets to be hidden (parameters $a_{ij}$ take value 1 if $T_i$ supports sensitive itemset $X_j$, or 0 otherwise). Constraint (4) defines a *constraint matrix* which is the transpose of the $a_{ij}$ matrix and relates sensitive itemsets to their supporting transactions. A solution of ILP indicates the transactions that need to be sanitized. The *intelligent sanitization procedure* of [47] aims on removing the fewest number of items from a transaction until all sensitive itemsets are supported less than the predefined

support threshold (see Menon et al. [47] for a detailed description of the so called *Maximum Accuracy* method).

$$min \sum_{\forall i:T_i \in \mathcal{D}} x_i, \tag{3}$$

$$s.t. \sum_{\forall i:T_i \in \mathcal{D}} a_{ij} x_i \geq \text{supc}_{\mathcal{D}}(X_j) - \sigma + 1, \forall X_j \in \mathcal{S}, \tag{4}$$

$$x_i \in \{0, 1\}, \forall i : T_i \in \mathcal{D}. \tag{5}$$

We next formulate the ILP of [47] for our sample database $\mathcal{D}_0$. For example, the sensitive itemset $ac \in \mathcal{S}$ is supported by the 1st, the 2nd, the 4th, and the 6th transaction, so for $ac$ to be hidden, more than $4 - \sigma = 2$ transactions have to be sanitized. This corresponds to constraint $x_1 + x_2 + x_4 + x_6 \geqslant 3$ in the constraint matrix. The same reasoning holds for the rest of sensitive itemsets to be hidden (see ILP 2).

$$\begin{aligned} min \;\; & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ s.t. \;\; & x_1 + x_2 + x_4 + x_6 \geqslant 3 \\ & x_1 + x_3 \geqslant 1 \\ & x_1 + x_2 \geqslant 1 \\ & x_1 + x_4 \geqslant 1 \\ & x_1, \ldots, x_6 \in \{0, 1\}. \end{aligned}$$

A solution of ILP 2 is $(x_1, \ldots, x_6) = (1, 1, 0, 1, 0, 0)$ that corresponds to the sanitized database $\mathcal{D}'_0 = \{cd, cd, abd, cd, cd, ac\}$ if the intelligent sanitization procedure is applied. Recall that the ideal positive border of $\mathcal{D}_0$ is $\mathcal{B}d^+(\widetilde{\mathcal{F}}) = \{cd, bc, ad, ab\}$. Compared to that, it is easy to verify that only the itemset $cd$ belongs to the positive border of $\mathcal{D}'_0$ while the rest are not; thus, an information loss of 75 % has occurred, indicating that the efficiency of the method is different than the expected one.

### 5.1 The positive border-based ILP formulation

The approach of Menon et al. in [47] completely lacks the idea of using the notion of the borders to lead the hiding process so that (a) to automatically account for the frequent itemsets that are hidden in the process as supersets of the sensitive itemsets and (b) to minimize the side effects, caused by the accidental hiding of non-sensitive frequent itemsets. Using Proposition 4.1, we are able to concentrate only on $Min(\mathcal{S})$ instead of $\mathcal{S}$, and make the formulation of the ILP more compact. Moreover, we can account for the side effects caused by the hiding process on the non-sensitive frequent itemsets and increase the data quality of the sanitized database by incorporating into the ILP the appropriate constraints that ensures that all itemsets in $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ need to remain into the positive border of the sanitized database.

$$min \sum_{\forall i:T_i \in \mathcal{D}} x_i, \tag{6}$$

$$s.t. \sum_{\forall i:T_i \in \mathcal{D}} a_{ij} x_i \geq \text{supc}_{\mathcal{D}}(X_j) - \sigma + 1, \forall X_j \in Min(\mathcal{S}), \tag{7}$$

$$\sum_{\forall i:T_i \in \mathcal{D}} a_{ij} x_i \leq \text{supc}_{\mathcal{D}}(X_j) - \sigma, \forall X_j \in \mathcal{B}d^+(\widetilde{\mathcal{F}}), \tag{8}$$

$$x_i \in \{0, 1\}, \forall i : T_i \in \mathcal{D}. \tag{9}$$

The enhanced positive border-based ILP formulation 3 aims to enforce the hiding of the itemsets in $Min(\mathcal{S})$, while at the same time, it takes into consideration the way this process affects the itemsets in $\mathcal{B}d^+(\widetilde{\mathcal{F}})$. In this way, the number of inequalities in the linear program is equal to $|Min(\mathcal{S})| + |\mathcal{B}d^+(\widetilde{\mathcal{F}})|$. For an itemset in $Min(\mathcal{S})$, say $ac$, the inequality in the linear program implies the minimum number of transactions that should not support $ac$ after the hiding so that the support of $ac$ becomes lower than the minimum support threshold. For an itemset in $\mathcal{B}d^+(\widetilde{\mathcal{F}})$, say $bc$, we simply require its support to be minimized so that the hiding of the sensitive itemsets is not blocked. For the utility of the sanitized database to be maximized, the objective function must ensure that the number of the binary variables $x_i$, that will be set to 1, must be minimized.

The ILP 4 corresponds to the sample database $\mathcal{D}_0$. The first two inequalities correspond to itemsets in $Min(\mathcal{S})$ ($ac$ and $bc$, resp.), while the last four ones correspond to itemsets in $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ ($ab$, $ad$, $bc$, and $cd$, resp.). The right-hand side of the inequalities corresponding to $Min(\mathcal{S})$ implies the differences between their current support counts and their minimum support counts, while those of $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ correspond to their actual support counts. For example, itemset $ab$ is supported by the first three transactions of $\mathcal{D}_0$. Thus, to retain $ab$ being in $\mathcal{B}d^+(\widetilde{\mathcal{F}})$, inequality $x_1 + x_2 + x_3 \leqslant 3 - \sigma = 1$ must hold.

$$
\begin{aligned}
min \quad & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
s.t. \quad & x_1 + x_2 + x_4 + x_6 \geqslant 3 \\
& x_1 + x_3 \geqslant 1 \\
& x_1 + x_2 + x_3 \leqslant 1 \\
& x_1 + x_3 + x_4 \leqslant 1 \\
& x_1 + x_2 \leqslant 0 \\
& x_1 + x_4 + x_5 \leqslant 1 \\
& x_1, \ldots, x_6 \in \{0, 1\}.
\end{aligned}
$$

Unfortunately, the above ILP is infeasible, pointing out that no ideal sanitized database exists. For the sensitive itemsets to be hidden, itemsets from the ideal positive border should also become infrequent, and thus, some loss of information should occur. Some constraints have to be relaxed and a trade-off between quality and accuracy seems to be unavoidable.

### 5.2 The elastic positive border-based ILP formulation

The infeasibility of the positive border-based ILP formulation 3 comes from the addition of the set of constraints that correspond to the ideal positive border $\mathcal{B}d^+(\widetilde{\mathcal{F}})$. To tackle this inconvenient situation, we relax our requirements by adding $|\mathcal{B}d^+(\widetilde{\mathcal{F}})|$ in total new slack variables, and obtain the ILP formulation 5.

$$
min \quad \sum_{\forall i: T_i \in \mathcal{D}} x_i + M \cdot \sum_{\forall j: X_j \in \mathcal{B}d^+(\widetilde{\mathcal{F}})} s_j, \tag{10}
$$

$$
s.t. \quad \sum_{\forall i: T_i \in \mathcal{D}} a_{ij} x_i \geq \text{supc}_{\mathcal{D}}(X_j) - \sigma + 1, \ \forall X_j \in Min(\mathcal{S}), \tag{11}
$$

$$
\sum_{\forall i: T_i \in \mathcal{D}} a_{ij} x_i - s_j \leq \text{supc}_{\mathcal{D}}(X_j) - \sigma, \ \forall X_j \in \mathcal{B}d^+(\widetilde{\mathcal{F}}), \tag{12}
$$

$$
x_i \in \{0, 1\}, \forall i : T_i \in \mathcal{D}, \tag{13}
$$

$$
x_j \in \{0, 1\}, \forall j : X_j \in \mathcal{B}d^+(\widetilde{\mathcal{F}}). \tag{14}
$$

By setting a slack variable $s_j$ equal to 1, the corresponding constraint may be satisfied but this would impact a penalty of value $M$ for the objective function. However, the ILP may become feasible and the corresponding sanitized database preserves the privacy with guaranty but with the minimum side effects on data quality.

$$\begin{aligned} min \;\; & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + M(s_1 + s_2 + s_3 + s_4) \\ s.t. \;\; & x_1 + x_2 + x_4 + x_6 \geqslant 3 \\ & x_1 + x_3 \geqslant 1 \\ & x_1 + x_2 + x_3 - s_1 \leqslant 1 \\ & x_1 + x_3 + x_4 - s_2 \leqslant 1 \\ & x_1 + x_2 - s_3 \leqslant 0 \\ & x_1 + x_4 + x_5 - s_4 \leqslant 1 \\ & x_1, \ldots, x_6, s_1, \ldots, s_4 \in \{0, 1\}. \end{aligned}$$

To continue with our sample database $\mathcal{D}_0$, the slack variables $s_1, \ldots, s_4$ are multiplied with $M$ and added to the objective function, each of them corresponding to an itemset in $\mathcal{B}d^+(\widetilde{\mathcal{F}})$. Moreover, each of them, say $s_1$ for example, is subtracted from the first constraint so that to enable this constraint to be satisfied (by setting $s_1 = 1$) but with an impact of $M$ for the objective function (see ILP 6).

$$\begin{aligned} min \;\; & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + M(s_1 + s_2 + s_3 + s_4) \\ s.t. \;\; & x_1 + x_2 + x_4 + x_6 \geqslant 3 \\ & x_1 + x_3 \geqslant 1 \\ & x_1 + x_2 + x_3 - s_1 \leqslant 1 \\ & x_1 + x_3 + x_4 - s_2 \leqslant 1 \\ & x_1 + x_2 - s_3 \leqslant 0 \\ & x_1 + x_4 + x_5 - s_4 \leqslant 1 \\ & x_1, \ldots, x_6, s_1, \ldots, s_4 \in \{0, 1\}. \end{aligned}$$

A solution of the elastic ILP 6 is $(x_1, \ldots, x_6, s_1, \ldots, s_4) = (1, 0, 0, 1, 0, 1, 1, 0, 1, 1)$, resulting to the sanitized database $\mathcal{D}_0'' = \{ad, abc, abd, cd, cd, c\}$. If compared with the database that correspond to the ideal positive border $\mathcal{B}d^+(\widetilde{\mathcal{F}}) = \{cd, bc, ad, ab\}$, it is easy to see that only $bc$ does not belong to the positive border of $\mathcal{D}_0''$ while the rest do; thus only an information loss of 25 % has occurred.

We conclude that the proposed ILP formulation 5 is more efficient than the ILP formulation 1 since it results in a sanitized database that is closer to the ideal one; this holds for our sample database but also for a number of test instances experimentally evaluated in Sect. 6. The ideal situation would be if the positive border-based ILP was feasible; then it would result in a sanitized database with positive border identical to the revised one returned by Algorithm 1. We shall extensively evaluate both methods on real datasets in the next section.

## 6 Experimental evaluation

In this section, we evaluate the proposed border revision technique based on the Elastic Positive Border-based ILP formulation 5 (referred as EPB-MA). We compare the proposed

**Table 3** Characteristics of real datasets

| Dataset | Number of transactions | Number of items | Aver. trans. length | Support threshold |
|---|---|---|---|---|
| chess | 3196 | 75 | 37 | 2557 |
| mushroom | 8124 | 119 | 23 | 1625 |
| BMS1 | 59,602 | 497 | 2.5 | 51 |
| BMS2 | 77,512 | 3340 | 5.6 | 39 |
| retail | 88,162 | 16,470 | 10.3 | 44; 88 |
| kosarak | 990,002 | 41,270 | 8.1 | 4950 |

**Table 4** Characteristics of synthetic datasets

| Dataset | Number of transactions | Number of items | Aver. trans. length | Support threshold |
|---|---|---|---|---|
| T10I4D100K | 100,000 | 870 | 10.10 | 100 |
| T40I10D100K | 100,000 | 942 | 39.60 | 1100 |
| D500k-I20K | 500,000 | 20,313 | 11.31 | 400 |
| D500K-I60K | 500,000 | 59,983 | 11.31 | 400 |
| 1M | 1,000,000 | 2398 | 4.58 | 800 |

method with previous related ones, already discussed in Sect. 2: the Max-Accuracy technique [47] (referred as MA), the Coefficient-Based Max-Accuracy technique [45] (referred as CBMA), the Max–Min 1 & 2 techniques [49] (referred as MM1 and MM2 respectively), and the Heuristic Coefficient-Based Approach [40] (referred as HCBA).

To establish the effectiveness of our approach, we evaluate the methods on real and synthetic datasets. Real datasets along with *T10I4D100K* and *T40I10D100K* synthetic datasets are available in the frequent itemset mining dataset repository [25]. The rest of the synthetic datasets were generated using the IBM Basket Data Generator [39]. We utilized datasets with a variety of characteristics in terms of the number of transactions, number of items and average transaction length. The chess and mushroom datasets were prepared by Roberto Bayardo [6]. The BMS1 and BMS2 datasets were used for the KDD Cup 2000 [44]. The retail dataset is a market Basket Dataset supplied by an anonymous Belgian retail supermarket store [14]. Finally, the kosarak dataset [8] contains anonymized click-stream data of a Hungarian online news portal. The synthetic datasets were generated so as to study the results given by datasets with different number of items (*D500K-I20K*, *D500K-I60K*) and different average transaction length (*T10I4D100K*, *T40I10D100K*). We summarize the details of the real and synthetic datasets in Tables 3 and 4, respectively.

To evaluate our proposed method, we do not deal with ILP formulation 3, since if it is feasible, its solutions can be obtained by solving ILP formulation 5, where all slack variables $s_j$ are equal to 0. Tables 5 and 6 display the percentage of dropped constraints of ILP formulation 5 for real and synthetic datasets, respectively. Nonzero values imply that the initial border-based ILP formulation 3 was infeasible, and thus the elastic positive border-based ILP formulation 3 was used. Observe that the denser the dataset is, the larger the percentage of the constraints dropped is.

All techniques were implemented in Python. The PyFIM extension module of Christian Borgelt [10] was used to efficiently mine the set of frequent itemsets. All experiments were performed on a personal computer with an Intel Core i5 at 3.2 GHz processor, under Windows 7. The integer linear programs were solved using IBM ILOG CPLEX 12.6 [38].

**Table 5** Percentage of dropped constraints of ILP formulation 3 for real datasets

| Dataset ($\sigma$) | Sensitive itemsets | Dropped constraints |
|---|---|---|
| chess (2557) | 10 | 97.6 |
| | 20 | 98.8 |
| | 50 | 99.7 |
| mushroom (1625) | 10 | 89.1 |
| | 20 | 83.2 |
| | 50 | 97.8 |
| BMS1 (51) | 10 | 9.4 |
| | 20 | 13.3 |
| | 50 | 32.6 |
| BMS2 (39) | 10 | 13.7 |
| | 20 | 32.2 |
| | 50 | 60 |
| retail (44) | 10 | 0.5 |
| | 20 | 1.5 |
| | 50 | 5.4 |
| retail (88) | 10 | 0.2 |
| | 20 | 2.6 |
| | 50 | 11.6 |
| kosarak (4950) | 10 | 29.9 |
| | 20 | 64.1 |
| | 50 | 61 |

**Table 6** Percentage of dropped constraints of ILP formulation 3 for synthetic datasets

| Dataset ($\sigma$) | Sensitive itemsets | Dropped constraints |
|---|---|---|
| T10I4D100K (100) | 10 | 0.4 |
| | 20 | 0.8 |
| | 50 | 2.6 |
| T40I10D100K (1100) | 10 | 7.9 |
| | 20 | 5.9 |
| | 50 | 8.4 |
| D500K-I20K (400) | 10 | 0.3 |
| | 20 | 0.8 |
| | 50 | 1.6 |
| D500K-I60K (400) | 10 | 2.4 |
| | 20 | 4.7 |
| | 50 | 15.1 |
| 1M (800) | 10 | 1.3 |
| | 20 | 2.2 |
| | 50 | 3.4 |

**Table 7** Side effects for real and synthetic datasets

| Dataset ($\sigma$) | Sensitive itemsets | Side Effects | | | | | |
|---|---|---|---|---|---|---|---|
| | | MA | EPB-MA | CBMA | MM1 | MM2 | HCBA |
| chess (2557) | 10 | 4318 | 3589 | 3504 | 5398 | 5214 | 3237 |
| | 20 | 4793 | 3666 | 3692 | 5534 | 5183 | 3206 |
| | 50 | 4632 | 4205 | 4288 | 5299 | 5127 | 4069 |
| mushroom(1625) | 10 | 30,848 | 17,931 | 16,768 | 22,073 | 31,497 | 16,391 |
| | 20 | 28,664 | 19,017 | 19,726 | 26,745 | 28,791 | 20,984 |
| | 50 | 34,729 | 27,992 | 22,797 | 28,715 | 32,170 | 24,016 |
| BMS1 (51) | 10 | 1555 | 462 | 388 | 1307 | 1035 | 352 |
| | 20 | 3216 | 681 | 562 | 2121 | 1635 | 444 |
| | 50 | 4808 | 1453 | 1425 | 3162 | 2784 | 1286 |
| BMS2 (39) | 10 | 17,274 | 11,335 | 8369 | 23,922 | 14,786 | 8703 |
| | 20 | 49,743 | 23,898 | 18,105 | 39,390 | 40,191 | 17,564 |
| | 50 | 58,194 | 43,090 | 33,291 | 55,492 | 55,374 | 32,131 |
| retail (44) | 10 | 135 | 50 | 50 | 127 | 55 | 33 |
| | 20 | 320 | 150 | 154 | 201 | 143 | 80 |
| | 50 | 720 | 319 | 296 | 501 | 226 | 115 |
| retail (88) | 10 | 74 | 15 | 28 | 71 | 11 | 5 |
| | 20 | 200 | 63 | 79 | 126 | 56 | 21 |
| | 50 | 395 | 206 | 198 | 272 | 179 | 108 |
| kosarak (4950) | 10 | 285 | 61 | 66 | 225 | 103 | 43 |
| | 20 | 534 | 215 | 201 | 440 | 375 | 133 |
| | 50 | 542 | 262 | 256 | 502 | 364 | 212 |
| T10I4D100K (100) | 10 | 24 | 0 | 2 | 142 | 22 | 65 |
| | 20 | 172 | 3 | 44 | 527 | 114 | 63 |
| | 50 | 261 | 54 | 63 | 753 | 162 | 488 |
| T40I10D100K (1100) | 10 | 2486 | 707 | 811 | 5044 | 947 | 838 |
| | 20 | 2929 | 1038 | 975 | 6137 | 2601 | 1375 |
| | 50 | 4716 | 1925 | 1539 | 6646 | 4256 | 2197 |
| D500K-I20K (400) | 10 | 406 | 356 | 354 | 625 | 346 | 774 |
| | 20 | 82 | 33 | 54 | 751 | 528 | 434 |
| | 50 | 253 | 103 | 112 | 1289 | 733 | 771 |
| D500K-I60K (400) | 10 | 52 | 37 | 44 | 366 | 249 | 200 |
| | 20 | 240 | 63 | 61 | 564 | 145 | 687 |
| | 50 | 588 | 363 | 401 | 1160 | 535 | 543 |
| 1M (800) | 10 | 210 | 49 | 61 | 484 | 153 | 689 |
| | 20 | 722 | 113 | 187 | 1064 | 365 | 1684 |
| | 50 | 1109 | 146 | 283 | 1300 | 517 | 2535 |

Our main evaluation metric is the information loss on the revised positive border. We measure the percentage of the itemsets in the revised positive border $\mathcal{B}d^+(\widetilde{\mathcal{F}})$ that belong to the positive border of the sanitized database $\mathcal{D}'$, after the intelligent sanitization algorithm is applied:

**Table 8** Information loss for real and synthetic datasets

| Dataset ($\sigma$) | Sensitive itemsets | Information loss (%) | | | | | |
|---|---|---|---|---|---|---|---|
| | | MA | EPB-MA | CBMA | MM1 | MM2 | HCBA |
| chess (2557) | 10 | 57.1 | 47.5 | 46.6 | 71 | 68.7 | 42.7 |
| | 20 | 68.7 | 52.8 | 53.1 | 79.1 | 74.2 | 45.9 |
| | 50 | 67.1 | 61.2 | 62.2 | 76.2 | 73.9 | 59 |
| mushroom (1625) | 10 | 57.9 | 34.6 | 31.2 | 42.4 | 58.7 | 30.4 |
| | 20 | 57.1 | 39 | 39.3 | 53.7 | 57.4 | 41.5 |
| | 50 | 81.6 | 66.3 | 55.1 | 68.1 | 76 | 57.2 |
| BMS1 (51) | 10 | 10.4 | 3.1 | 2.7 | 8.8 | 7 | 2.4 |
| | 20 | 21.8 | 4.7 | 3.9 | 14.5 | 11.1 | 3.1 |
| | 50 | 34.8 | 10.6 | 10.4 | 22.7 | 19.7 | 9.3 |
| BMS2 (39) | 10 | 15.4 | 11.8 | 9.7 | 20.7 | 14.6 | 9.9 |
| | 20 | 42 | 23.4 | 19.2 | 34.4 | 36.2 | 18.5 |
| | 50 | 55.7 | 43.4 | 35.7 | 52.9 | 53 | 34.1 |
| retail (44) | 10 | 0.6 | 0.3 | 0.2 | 0.6 | 0.3 | 0.2 |
| | 20 | 1.5 | 1 | 0.8 | 1.1 | 0.8 | 0.5 |
| | 50 | 3.5 | 2.2 | 1.8 | 2.6 | 1.4 | 1 |
| retail (88) | 10 | 1.1 | 0.7 | 0.5 | 1 | 0.4 | 0.3 |
| | 20 | 2.7 | 1.5 | 1.3 | 1.8 | 1.1 | 0.7 |
| | 50 | 5.4 | 3.8 | 3.4 | 4.4 | 3.3 | 2.3 |
| kosarak (4,950) | 10 | 14.1 | 5.2 | 4.7 | 11.8 | 6.4 | 3.6 |
| | 20 | 27.3 | 14 | 13 | 23.2 | 20.4 | 9.6 |
| | 50 | 28.5 | 16.8 | 16 | 27.2 | 20.8 | 13.7 |
| T10I4D100K (100) | 10 | 1.3 | 1.2 | 1.2 | 1.5 | 1.3 | 1.3 |
| | 20 | 3.1 | 2.7 | 2.8 | 4 | 3.4 | 2.8 |
| | 50 | 6.1 | 5.6 | 5.5 | 7.1 | 5.8 | 6.3 |
| T40I10D100K(1100) | 10 | 9.2 | 4 | 4.2 | 16.8 | 5 | 4.3 |
| | 20 | 11.1 | 5.5 | 5.2 | 20.6 | 10.2 | 6.4 |
| | 50 | 17.1 | 8.6 | 7.4 | 23.1 | 15.8 | 9.3 |
| D500K-I20K(400) | 10 | 3.1 | 2.7 | 2.7 | 4.5 | 2.9 | 5.3 |
| | 20 | 1.6 | 1.3 | 1.4 | 5.9 | 4.4 | 3.7 |
| | 50 | 3.2 | 2.2 | 2.2 | 9.8 | 6.3 | 6.3 |
| D500K-I60K (400) | 10 | 1.4 | 1.2 | 1.3 | 5 | 3.6 | 3.1 |
| | 20 | 5.1 | 3 | 3 | 8.9 | 3.9 | 10.1 |
| | 50 | 9.9 | 7.3 | 7.7 | 17 | 9.7 | 9.4 |
| 1M (800) | 10 | 1.6 | 1.2 | 1.2 | 2.3 | 1.7 | 2.6 |
| | 20 | 3.3 | 1.9 | 2 | 4.1 | 2.5 | 5.3 |
| | 50 | 4.8 | 2.6 | 2.8 | 5.5 | 3.8 | 8 |

$$IL_{\mathcal{B}d}(\mathcal{B}d^+(\widetilde{\mathcal{F}}), \mathcal{B}d^+(\mathcal{F}')) = \frac{|X \in \mathcal{B}d^+(\widetilde{\mathcal{F}}) : X \in \mathcal{B}d^+(\mathcal{F}')|}{|\mathcal{B}d^+(\widetilde{\mathcal{F}})|}. \quad (15)$$

A modified version of the information loss, as used in [40], is the ratio between the sum of the absolute differences in the frequencies of the ideal frequent itemsets in the original

**Table 9** Information loss on the revised positive border for real and synthetic datasets

| Dataset ($\sigma$) | Sensitive itemsets | Information loss on positive border (%) | | | | | |
|---|---|---|---|---|---|---|---|
| | | MA | EPB-MA | CBMA | MM1 | MM2 | HCBA |
| chess (2557) | 10 | 99 | 94 | 96 | 100 | 100 | 89 |
| | 20 | 100 | 98 | 99 | 100 | 100 | 82 |
| | 50 | 100 | 99 | 100 | 100 | 100 | 99 |
| mushroom (1625) | 10 | 85 | 80 | 74 | 77 | 78 | 49 |
| | 20 | 69 | 58 | 53 | 65 | 73 | 51 |
| | 50 | 98 | 97 | 96 | 92 | 98 | 94 |
| BMS1 (51) | 10 | 19 | 3 | 4 | 16 | 13 | 3 |
| | 20 | 40 | 5 | 6 | 29 | 21 | 4 |
| | 50 | 57 | 16 | 15 | 43 | 39 | 15 |
| BMS2 (39) | 10 | 18 | 7 | 11 | 13 | 13 | 13 |
| | 20 | 44 | 18 | 27 | 29 | 36 | 22 |
| | 50 | 52 | 43 | 47 | 49 | 50 | 44 |
| retail (44) | 10 | 2 | 1 | 1 | 2 | 1 | 0 |
| | 20 | 4 | 2 | 2 | 2 | 2 | 1 |
| | 50 | 8 | 4 | 3 | 6 | 3 | 1 |
| retail (88) | 10 | 2 | 0 | 1 | 2 | 0 | 0 |
| | 20 | 6 | 2 | 2 | 3 | 2 | 1 |
| | 50 | 11 | 6 | 5 | 7 | 5 | 3 |
| kosarak (4,950) | 10 | 55 | 15 | 15 | 38 | 21 | 9 |
| | 20 | 77 | 41 | 39 | 60 | 49 | 26 |
| | 50 | 70 | 40 | 38 | 66 | 47 | 30 |
| T10I4D100K (100) | 10 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 20 | 1 | 0 | 0 | 1 | 1 | 1 |
| | 50 | 3 | 1 | 1 | 4 | 2 | 2 |
| T40I10D100K(1100) | 10 | 7 | 4 | 5 | 4 | 5 | 4 |
| | 20 | 4 | 3 | 3 | 4 | 4 | 3 |
| | 50 | 6 | 5 | 5 | 6 | 5 | 4 |
| D500K-I20K (400) | 10 | 0 | 0 | 0 | 1 | 0 | 1 |
| | 20 | 0 | 0 | 0 | 1 | 1 | 1 |
| | 50 | 1 | 1 | 1 | 2 | 1 | 1 |
| D500K-I60K (400) | 10 | 3 | 2 | 2 | 5 | 3 | 4 |
| | 20 | 9 | 3 | 4 | 1.1 | 3 | 10 |
| | 50 | 19 | 9 | 13 | 2 | 12 | 13 |
| 1M (800) | 10 | 2 | 1 | 1 | 3 | 1 | 2 |
| | 20 | 4 | 1 | 2 | 4 | 3 | 4 |
| | 50 | 5 | 2 | 2 | 6 | 4 | 6 |

database $\mathcal{D}$ and the sanitized database $\mathcal{D}'$:

$$IL(\widetilde{\mathcal{D}}, \mathcal{D}') = \frac{\sum_{X \in \widetilde{\mathcal{F}}} |supc_{\mathcal{D}}(X) - supc_{\mathcal{D}'}(X)|}{\sum_{X \in \widetilde{\mathcal{F}}} supc_{\mathcal{D}}(X)}. \quad (16)$$

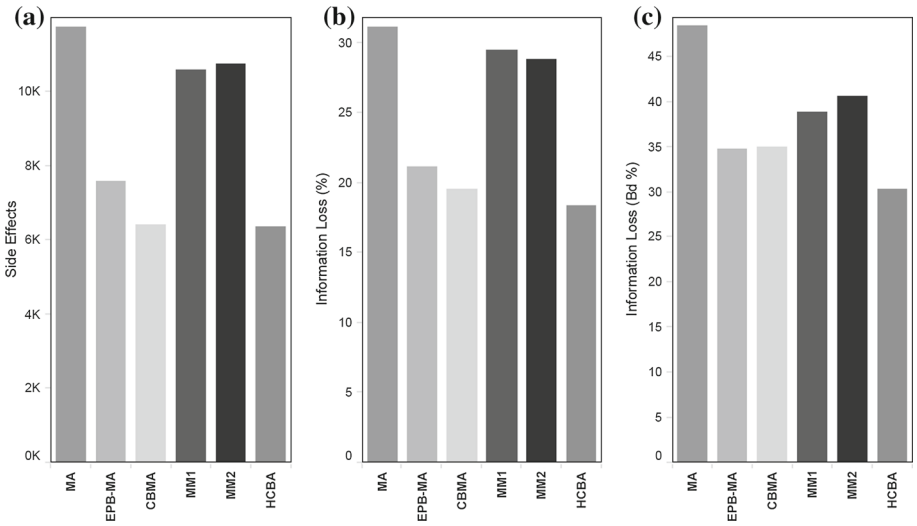


**Fig. 3** Mean value of the metrics for the real datasets, **a** side effects, **b** information loss, **c** information loss on $\mathcal{B}d^{+}(\widetilde{\mathcal{F}})$

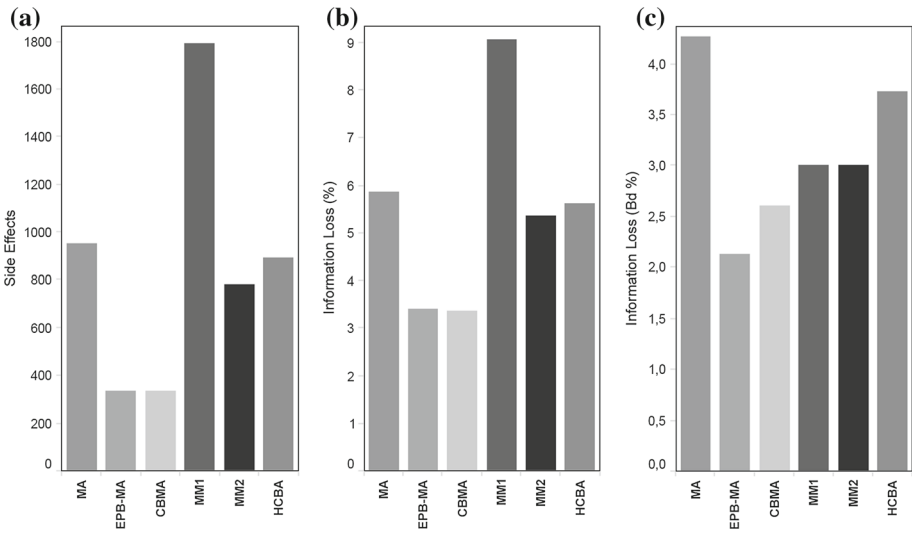


**Fig. 4** Mean value of the metrics for the synthetic datasets, **a** side effects, **b** information loss, **c** information loss on $\mathcal{B}d^{+}(\widetilde{\mathcal{F}})$

Lastly, we measure the difference in the number of frequent itemsets of the revised ideal database and the number of frequent itemsets of the sanitized database. This difference equals to number of the side effects introduced by the sanitization process:

$$SE(\widetilde{\mathcal{F}}, \mathcal{F}') = \frac{|\widetilde{\mathcal{F}}| - |\mathcal{F}'|}{|\widetilde{\mathcal{F}}|}. \tag{17}$$

We summarize our experimental results in Tables 7, 8 and 9. We show the side effects and the information loss for hiding 10, 20 and 50 sensitive itemsets. We observe that in

all test cases, the elastic positive border based method (EPB-MA) outperformed most the other methods. The chess and mushroom datasets are rather dense and not representatives of real transactional databases. Even for such instances, EPM-MA achieves good results. The mean value of the metrics for each method, presented in Figs. 3 and 4, shows clearly that the proposed method is among the best ones.

## 7 Conclusions

In this paper, we proposed a methodology for hiding sensitive information in a transactional database. The frequent itemset hiding problem was modeled by using integer linear programming, based on the formulation of Menon et al. [47]. We enhanced the ILP formulation by taking into account the ideal positive border of the sanitized database. To compute it, we utilized a transversal hypergraph generation algorithm. We experimentally evaluated the proposed method on a number of datasets, using a variety of performance metrics, and compared it this previews related work. Compared with the work in [47], the proposed method decreased the side effects and the information loss introduced by the hiding process, while she outperformed the most of the rest ones.

**Acknowledgments** The authors wish to thank the anonymous referees for their valuable comments that improved the final presentation of the paper.

## References

1. Aggarwal CC, Yu PS (eds) (2008) Privacy-preserving data mining: models and algorithms. Advances in database systems. Springer, New York
2. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th international conference on very large data bases (VLDB'94), pp 487–499
3. Agrawal R, Srikant R (2000) Privacy-preserving data mining. In: Proceedings of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD 2000), pp 439–450
4. Atallah M, Bertino E, Elmagarmid A, Ibrahim M, Verykios V (1999) Disclosure limitation of sensitive rules. In: Proceedings of the knowledge and data engineering exchange (KDEX'99), pp 45–52
5. Bailey J, Manoukian T, Ramamohanarao K (2003) A fast algorithm for computing hypergraph transversals and its application in mining emerging patterns. In: Proceedings of the 3rd IEEE international conference on data mining (ICDM 2003), pp 485–488. IEEE computer Society, Dec 2003
6. Bayardo R (1998) Efficiently mining long patterns from databases. In: Proceedings of the 1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), pp 85–93
7. Berge C (1989) Hypergraphs: combinatorics of finite sets, volume 45 of North Holland mathematical library. Elsevier Science Publishers B.V., Amsterdam
8. Bodon F (2003) A fast APRIORI implementation. In: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations (FIMI'03), vol 90, pp 56–65
9. Bonchi F, Ferrari E (2011) Privacy-aware knowledge discovery: novel applications and new techniques. Chapman & Hall/CRC data mining and knowledge discovery series. CRC Press INC
10. Borgelt C (2012) Frequent item set mining. Wiley Interdiscip Rev: Data Min Knowl Discov 2(6):437–456
11. Boros E, Elbassioni K, Gurvich V, Khachiyan L (2003) An efficient implementation of a quasi-polynomial algorithm for generating hypergraph transversals. In: Proceedings of the 11th annual European symposium on algorithms (ESA 2003), vol 2432 of LNCS, 556–567
12. Boros E, Elbassioni K, Makino K (2008) On Berge multiplication for monotone boolean dualization. In: Proceedings of the 35th international colloquium on automata, languages and programming (ICALP 2008), volume 5125 of LNCS, 48–59
13. Boros E, Gurvich V, Khachiyan L, Makino K (2003) On maximal frequent and minimal infrequent sets in binary matrices. Ann Math Artif Intell 39(3):211–221
14. Brijs T, Swinnen G, Vanhoof K, Wets G (1999) Using association rules for product assortment decisions: a case study. In: proceedings of the 5th ACM-SIGKDD international conference on knowledge discovery and data mining (KDD'99), pp 254–260

15. Bu S, Lakshmanan LVS, Ng RT, Ramesh G (2007) Preservation of patterns and input–output privacy. In: Proceedings of the IEEE 23rd international conference on data engineering (ICDE 2007), pp 696–705
16. Calders T (2004) Computational complexity on itemset frequency satisfiability. In: Proceedings of symposium on principles of database systems 2004 (PODS'04), pp 143–154
17. Calders T (2008) Itemset frequency satisfiability: complexity and axiomatization. Theor Comput Sci 394(1–2):84–111
18. Clifton C (1999) Protecting against data mining through samples. In: Proceedings of the 13th international conference on database security (DBSec'99), pp 193–207
19. Dong G, Li J (2005) Mining border descriptions of emerging patterns from dataset pairs. Knowl Info Syst 8(2):178–202
20. Eiter T, Gottlob G (1995) Identifying the minimal transversals of a hypergraph and related problems. SIAM J Comput 24(6):1278–1304
21. Eiter T, Gottlob G (2002) Hypergraph transversal computation and related problems in Logic and AI. In: Proceedings of European conference on logic in AI (JELIA 2002), vol 2424 of LNCS/LNAI, pp 549–564
22. Eiter T, Gottlob G, Makino K (2003) New results on monotone dualization and generating hypergraph transversals. SIAM J Comput 32(2):514–537
23. Evfimievski AV, Srikant R, Agrawal R, Gehrke J (2004) Privacy preserving mining of association rules. Info Syst 29(4):343–364
24. Faloutsos C, Megalooikonomou V (2007) On data mining, compression, and Kolmogorov complexity. Data Min Knowl Discov 15(1):3–20
25. Frequent itemset mining dataset repository. http://fimi.ua.ac.be/data/
26. Fredman ML, Khachiyan L (1996) On the complexity of dualization of monotone disjunctive normal forms. J Algorithm 21:618–628
27. Fung BCM, Wang K, Chen R, Yu PS (2010) Privacy-preserving data publishing: a survey of recent developments. ACM Comput Surv 42(4):571–588
28. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman and Company, San Francisco
29. Georgakopoulos G, Kavvadias D, Papadimitriou CH (1988) Probabilistic satisfiability. J Complex 4:1–11
30. Gkoulalas-Divanis A, Verykios VS (2009) Hiding sensitive knowledge without side effects. Knowl Info Syst 20(3):263–299
31. Goldsmith J, Levy MA, Mundhenk M (1996) Limited nondeterminism. ACM SIGACT News 27(2):20–29
32. Gottlob G (2013) Deciding monotone duality and identifying frequent itemsets in quadratic logspace. Technical report arxiv:1212.1881v3 [cs.DC]
33. Gunopulos D, Khardon R, Mannila H, Saluja S, Sharma HTR (2003) Discovering all most specific sentences. ACM Trans Database Syst 28(2):140–174
34. Gurvich V, Khachiyan L (1999) On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions. Discret Appl Math 96–97:363–373
35. Guzzo A, Moccia L, Saccà D, Serra E (2013) Solving inverse frequent itemset mining with infrequency constraints via large-scale linear programs. ACM Trans Knowl Discov Data 7(4), Article 18, 1–39
36. Guzzo A, Saccà D, Serra E (2009) An effective approach to inverse frequent set mining. In: Proceedings of the 9th IEEE international conference on data mining (ICDM'09), pp 806–811
37. Hagen M (2009) Lower bounds for three algorithms for transversal hypergraph generation. Discret Appl Math 157:1460–1469
38. IBM ILOG CPLEX user's manual v12.6
39. IBM Basket Data Generator. http://sourceforge.net/projects/ibmquestdatagen/
40. Kagklis V, Verykios VS, Tzimas G, Tsakalidis AK (2014) An integer linear programming scheme to sanitize sensitive frequent itemsets. In: Proceedings of 2014 IEEE international conference on tools with AI (ICTAI 2014), 2014. To appear
41. Kantarcioglu M, Jin J, Clifton C (2004) When do data mining results violate privacy? In: Proceedings of the 10th ACM-SIGKDD international conference on knowledge discovery and data mining (KDD'04), pp 599–604
42. Kavvadias DJ, Stavropoulos EC (2003) Monotone Boolean dualization is in co-NP[$\log^2 n$]. Info Process Lett 85(1):1–6
43. Kavvadias DJ, Stavropoulos EC (2005) An efficient algorithm for the transversal hypergraph generation. J Graph Algorithms Appl 9(2):239–264
44. Kohavi R, Brodley C, Frasca B, Mason L, Zheng Z (2000) KDD-Cup 2000 organizers' report: peeling the onion. SIGKDD explorations, 2(2):86–98. http://www.ecn.purdue.edu/KDDCUP
45. Leloglu E, Ayav T, Ergenc B (2014) Coefficient-based exact approach for frequent itemset hiding. In: eKNOW2014: The 6th international conference on information, process, and knowledge management, pp 124–130

46. Mannila H, Toivonen H (1997) Levelwise search and borders of theories in knowledge discovery. Data Min Knowl Discov 1:241–258
47. Menon S, Sarkar S, Mukherjee S (2005) Maximizing accuracy of shared databases when concealing sensitive patterns. Info Syst Res 16(3):256–270
48. Mielikäinen T (2003) On inverse frequent set mining problems. In: Proceedings of the 2nd workshop on privacy preserving data mining (PPDM'03), pp 18–33
49. Moustakides GV, Verykios VS (2008) A maxmin approach for hiding frequent itemsets. Data Knowl Eng 65(1):75–89
50. Murakami K, Uno T (2011) Efficient algorithms for dualizing large-scale hypergraphs. Technical report arxiv:1102.3813v2 [cs.DC]
51. Rizvi S, Haritsa JR (2002) Maintaining data privacy in association rule mining. In: Proceedings of the 28th international conference on very large data bases (VLDB'02), pp 682–693
52. Sun X, Yu P (2005) A border–based approach for hiding sensitive frequent itemsets. In: Proceedings of 5th IEEE internationa conference on data mining (ICDM 2005), pp 426–433
53. Sun X, Yu PS (2007) Hiding sensitive frequent itemsets by a border-based approach. J Comput Sci Eng 1(1):74–94
54. Sweeney L (2002) Achieving k-anonymity privacy protection using generalization and suppression. Int J Uncertain Fuzziness Knowl Based Syst 10(5):571–588
55. Sweeney L (2002) k-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl Based Syst 10(5):557–570
56. Takata K (2007) A worst-case analysis of the sequential method to list the minimal hitting sets of a hypergraph. SIAM J Discret Math 21(4):936–946

**E. C. Stavropoulos** received his B.Sc. degree from the Department of Mathematics and his Ph.D. degree in theoretical computer science from the Computer Engineering and Informatics Department of the University of Patras, Greece. He is an Adjunct Professor at the Technological Educational Institute of West Greece, a Tutor at the Hellenic Open University, and a researcher at the HOU e-Comet Lab (http://eeyem.eap.gr/en), the Educational Content, Methodology and Technology Laboratory of the Hellenic Open University. His research interests include knowledge discovery and privacy preserving data mining, information retrieval, web services, and open and distance learning methodology and technology.

**V. S. Verykios** received his Diploma Degree in Computer Engineering from the University of Patras in 1992, and his M.Sc. and Ph.D. Degrees from Purdue University, Indiana, USA, in 1997 and 1999, respectively. From 1999 to 2001, he was member of the Faculty of Information Systems Division in the College of Information Science and Technology at Drexel University, Pennsylvania, USA, as a tenure track Assistant Professor. From 2001 to 2005, he held various research positions, and from 2005 to 2011, he was Assistant Professor in the Department of Computer and Communication Engineering at the University of Thessaly in Volos, Greece. Since January of 2011, he is an Associate Professor in the School of Science and Technology, while he serves as the Director of the Graduate Program on Information Systems, and since May of 2014, he is the Director of the HOU e-Comet Lab (http://eeyem.eap.gr/en) at the Hellenic Open University. His main research interests include data management and data mining. He has published over 100 papers, while he has more than 5000 citations. He has co-authored a monograph on "Association Rule Hiding for Data Mining" by Springer. He has also served on the program committees of several international scientific events as KDD, ICDM, ECML/PKDD, and CIKM.

**V. Kagklis** received his Diploma and MSc Degrees from the Computer Engineering and Informatics Department, University of Patras, Greece in 2013 and 2015, respectively. He is currently a developer and a research fellow within the Educational Content, Methodology and Technology Laboratory, Hellenic Open University, Greece. His research interests include Knowledge Discovery and Machine Learning Algorithms, Sentiment Analysis, and Privacy Preserving Data Mining.