Chapter 2
Background

In this chapter we provide the background and terminology that are necessary for the understanding of association rule hiding. Specifically, in Section 2.1, we present the theory behind association rule mining and introduce the notion of the positive and the negative borders of the frequent itemsets. Following that, Section 2.2 explicitly states the goals of association rule hiding methodologies, discusses the different types of solutions that association rule hiding algorithms can produce, as well as it delivers the formal problem statement for association rule hiding and its popular variant, frequent itemset hiding.

2.1 Terminology and Preliminaries

Association rule mining is the process of discovering sets of items (also known as *itemsets*) that frequently co-occur in a transactional database so as to produce significant association rules that hold for the data. Each association rule is defined as an implication of the form $I \Rightarrow J$, where I, J are frequently occurring itemsets in the transactional database, for which $I \cap J = \varnothing$ (i.e., I and J are disjoint). The itemset $I \cup J$ that leads to the generation of an association rule is called *generating* itemset. An association rule consists of two parts: the *Left Hand Side* (LHS) or antecedent, which is the part on the left of the arrow of the rule (here I), and the *Right Hand Side* (RHS) or consequent, which is the part on the right of the arrow of the rule (here J). Two metrics, known as *support* and *confidence*, are incorporated to the task of association rule mining to drive the generation of association rules and expose only those rules that are expected to be of interest to the data owner. In particular, the measure of support eliminates rules that are not adequately backed up by the transactions of the dataset and thus are expected to be uninteresting, i.e. occurring simply by chance. On the other hand, confidence measures the strength of the relation between the itemsets of the rule as it quantifies the reliability of the inference made by the rule [68]. A low value of confidence in a rule $I \Rightarrow J$ shows that it is rather rare for itemset J to be present in transactions that contain itemset I.

A. Gkoulalas-Divanis and V.S. Verykios, *Association Rule Hiding for Data Mining*,
Advances in Database Systems 41, DOI 10.1007/978-1-4419-6569-1_2,
© Springer Science+Business Media, LLC 2010

Table 2.1: Notation table.

Notation	Description
$\mathcal{I} = \{i_1, i_2, \ldots, i_M\}$	Universe of literals (items) with cardinality M
$\|R\|$	Cardinality of a set R
$I, J, \ldots$	Itemsets produced from $\mathcal{I}$
k–itemset	An itemset of length (equiv. size) k
$T_n = (tid, I)$	A transaction with unique identifier tid and itemset I
$D = \{T_1, T_2, \ldots, T_N\}$	A database consisting of N transactions
T_{nm}	The m-th item of the n-th transaction in a database
$I \Rightarrow J$	An association rule between itemsets I and J
D_I	The supporting transactions of itemset I in database D
$\wp(\ )$	The powerset of a set of literals
$\mathcal{P} = \wp(\mathcal{I})$	The set of all possible itemsets (patterns) extracted from $\mathcal{I}$
$\sup(I, D)$, $\sup(I)$	Support of itemset I in database D
$\text{freq}(I, D)$, $\text{freq}(I)$	Frequency of itemset I in database D
$msup, mfreq, mconf$	Minimum support/frequency/confidence threshold
$\mathcal{F}_D$	Set of all frequent itemsets in database D
$\mathcal{B}d^+(\mathcal{F}_D), \mathcal{B}d^+$	Positive border of $\mathcal{F}_D$
$\mathcal{B}d^-(\mathcal{F}_D), \mathcal{B}d^-$	Negative border of $\mathcal{F}_D$
$\mathcal{B}d(\mathcal{F}_D)$	Border of $\mathcal{F}_D$
$\mathcal{S}$	Set of sensitive itemsets (patterns)
$\mathcal{R}$	Set of mined association rules
$\mathcal{R}_\mathcal{S}$	Set of sensitive association rules from $\mathcal{R}$

Association rule mining, introduced by Agrawal, et al. [5, 7] in 1993, operates by first mining all the itemsets that are frequent in the database and then by using these itemsets to derive association rules that are strong enough to be considered as interesting. The process of frequent itemset mining is defined as follows[1]: Let $\mathcal{I} = \{i_1, i_2, \ldots, i_M\}$ be a finite set of literals, called *items*, where M denotes the cardinality of the set. Any subset $I \subseteq \mathcal{I}$ is called an *itemset* over $\mathcal{I}$. A k–itemset is an itemset of *length* (equiv. of *size*) k, i.e. an itemset consisting of k items. A transaction T_n over $\mathcal{I}$ is a pair $T_n = (tid, I)$, where I is the itemset and tid is a unique identifier, used to distinguish among transactions that correspond to the same itemset. A transactional database $D = \{T_1, T_2, \ldots, T_N\}$ over $\mathcal{I}$ is a $N \times M$ table consisting of N transactions over $\mathcal{I}$ carrying different identifiers, where entry $T_{nm} = 1$ if and only if the m-th item ($m \in [1, M]$) appears in the n-th transaction ($n \in [1, N]$). Otherwise, $T_{nm} = 0$. A transaction $T = (tid, J)$ is said to *support* an itemset I over $\mathcal{I}$, if $I \subseteq J$. Let D_I denote the supporting transactions of itemset I in database D. Furthermore, let S be a set of items. Notation $\wp(S)$ denotes the powerset of S, which is the set of all subsets of S. Given the universe of all items $\mathcal{I}$ in D, we use notation $\mathcal{P} = \wp(\mathcal{I})$ to refer to all possible itemsets that can be produced from $\mathcal{I}$.

Given an itemset I over $\mathcal{I}$ in D, $\sup(I, D)$ denotes the number of transactions $T \in D$ that support I and $\text{freq}(I, D)$ denotes the fraction of transactions in D that

[1] The reader can refer to the work of Agrawal, et al. [5, 7] for a more detailed presentation of the association rule mining framework as well as for a set of computationally efficient algorithms for the derivation of the association rules. Moreover, [68] provides very good coverage of this topic.

support I [2]. An itemset I is called *large* or *frequent* in database D, if and only if, its frequency in D is at least equal to a minimum frequency threshold *mfreq*, set by the owner of the data. Equivalently, I is large in D, if and only, $\sup(I,D) \geq msup$, where $msup = mfreq \times N$. The set of frequent itemsets for a database D is denoted as $\mathcal{F}_D = \{I \subseteq \mathcal{I} : \text{freq}(I,D)$. All the itemsets having a frequency lower than *mfreq* (equivalently a support lower than *msup*) are called *infrequent* and their set is $\wp(\mathcal{I}) - \mathcal{F}_D$.

The second step in the process of association rule mining typically involves the identification of all the significant association rules that hold among the derived frequent itemsets. An association rule $I \Rightarrow J$ is *significant* if it holds in database D with a *confidence* that is higher than a minimum confidence threshold *mconf* set by the owner of the data, i.e. when $\frac{sup(I \cup J,D)}{sup(I,D)} \geq mconf$. The support of this rule in D is equal to that of its generating itemset, i.e. is equal to $sup(I \cup J,D)$. An example of association rule mining (borrowed from [72]) is shown on Fig. 2.1.

tid	*Itemset*
T_1	abc
T_2	abc
T_3	abc
T_4	ab
T_5	a
T_6	ac

Itemset	*Support*
a	6
b	4
c	4
ab	4
ac	4
bc	3
abc	3

Rules	*Confidence*	*Support*
$b \Rightarrow a$	100%	4
$b \Rightarrow c$	75%	3
$c \Rightarrow a$	100%	4
$c \Rightarrow b$	75%	3
$b \Rightarrow ac$	75%	3
$c \Rightarrow ab$	75%	3
$ab \Rightarrow c$	75%	3
$ac \Rightarrow b$	75%	3
$bc \Rightarrow a$	100%	3

Fig. 2.1: Database D along with its itemsets and related association rules.

Border Theory

The theory of the borders of the frequent itemsets is also very important in our discussion as both border-based and exact hiding methodologies rely on a process inspired from this theory. Let $\mathcal{F}_D$ be the set of all frequent itemsets in D, and $\mathcal{P} = \wp(\mathcal{I})$ be the set of all patterns in the lattice of D (e.g., see Fig. 2.2). The *positive border* of $\mathcal{F}_D$, denoted as $\mathcal{B}d^+(\mathcal{F}_D)$, consists of all the maximally frequent patterns in $\mathcal{P}$, i.e. all the patterns in $\mathcal{F}_D$, whose all proper supersets are infrequent. Formally, $\mathcal{B}d^+(\mathcal{F}_D) = \{I \in \mathcal{F}_D |$ for all $J \in \mathcal{P}$ with $I \subset J$ we have that $J \notin \mathcal{F}_D\}$. Respectively, the *negative border* of $\mathcal{F}_D$, denoted as $\mathcal{B}d^-(\mathcal{F}_D)$, consists of all the minimally infrequent patterns in $\mathcal{P}$, i.e. all the patterns in $\mathcal{P} \backslash \mathcal{F}_D$, whose all proper subsets are frequent. Formally, $\mathcal{B}d^-(\mathcal{F}_D) = \{I \in \mathcal{P} \backslash \mathcal{F}_\mathcal{D} |$ for all $J \subset I$ we have that

[2] We will use $\sup(I)$ and $\text{freq}(I)$ instead of $\sup(I,D)$ and $\text{freq}(I,D)$, respectively, for notational convenience, when database D is obvious in our context.

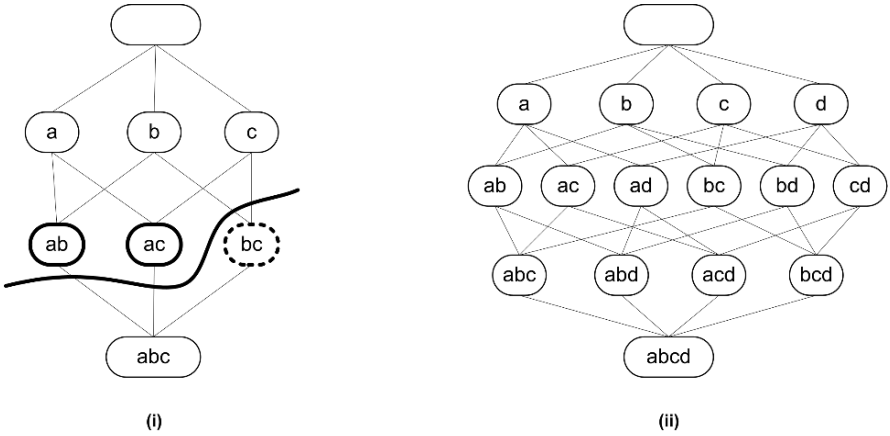


Fig. 2.2: Examples of two lattices for a database with (i) $\mathcal{I} = \{a,b,c\}$, and (ii) $\mathcal{I} = \{a,b,c,d\}$. In the first lattice we also demonstrate the positive and the negative borders when the database D is the same as in Fig. 2.1 and *msup*=4.

$J \in \mathcal{F}_D\}$. Finally, the *border* of $\mathcal{F}_D$, denoted as $\mathcal{B}d(\mathcal{F}_D)$, is the union of these two sets: $\mathcal{B}d(\mathcal{F}_D) = \mathcal{B}d^+(\mathcal{F}_D) \cup \mathcal{B}d^-(\mathcal{F}_D)$ [3]. For example, assuming that *msup*$= 4$ in the database of Fig. 2.1, we have that $\mathcal{B}d^+(\mathcal{F}_D) = \{ab,ac\}$, $\mathcal{B}d^-(\mathcal{F}_D) = \{bc\}$ and $\mathcal{B}d(\mathcal{F}_D) = \{ab,ac,bc\}$. These borders are presented in Fig. 2.2(i). Borders allow for a condense representation of the itemsets' lattice, identifying the key itemsets which separate all frequent patterns from their infrequent counterparts. More details on the theory of borders as well as its underlying concepts can be found in the work of Mannila & Toivonen [46].

2.2 Problem Formulation and Statement

Having presented the necessary background for association rule mining, in this section we formally set out the problem of association rule hiding. First, in Section 2.2.1, we highlight the goals of association rule hiding algorithms, rank these goals in terms of importance of being satisfied, as well as discuss the side-effects that are introduced when each of them is left unsatisfied in the sanitized database. Following that, in Section 2.2.2, we deliver the formal problem statement.

[3] For notational convenience, we will use $\mathcal{B}d^+$ and $\mathcal{B}d^-$ to refer to the positive and the negative border of a set of itemsets, respectively, when the set of frequent itemsets is obvious in our context.

2.2.1 Goals of Association Rule Hiding Methodologies

Association rule hiding methodologies aim at sanitizing the original database in a way that at least one of the following goals is accomplished:

1. No rule that is considered as sensitive from the owner's perspective and can be mined from the original database at pre-specified thresholds of confidence and support, can be also revealed from the sanitized database, when this database is mined at the same or at higher thresholds
2. All the nonsensitive rules that appear when mining the original database at pre-specified thresholds of confidence and support can be successfully mined from the sanitized database at the same thresholds or higher, *and*
3. No rule that was not derived from the original database when the database was mined at pre-specified thresholds of confidence and support, can be derived from its sanitized counterpart when it is mined at the same or at higher thresholds.

The first goal requires that all the sensitive rules disappear from the sanitized database, when the database is mined under the same thresholds of support and confidence as the original database, or at higher thresholds. A hiding solution that achieves the first goal is termed *feasible* as it accomplishes the hiding task.

The second and the third goals involve the nonsensitive rules that may be lost or generated as a side-effect of the employed sanitization process. Specifically, the second goal simply states that there should be no *lost rules* in the sanitized database, meaning that all the nonsensitive rules that were mined from the original database should also be mined from its sanitized counterpart at the same (or higher) levels of confidence and support. The third goal, on the other hand, states that no *false rules* (also known as *ghost rules*) should be produced when mining the sanitized database at the same (or higher) levels of confidence and support. A false (ghost) rule is an association rule that was not among the ones mined from the original database and thus it constitutes an artifact that was generated by the hiding process.

Based on these three goals, the sanitization process of a hiding algorithm has to be accomplished in a way that *minimally affects the original database, preserves the general patterns and trends, and achieves to conceal all the sensitive association rules*. A solution that addresses all these three goals (i.e., is feasible and introduces no side-effects) is called *exact*. Exact hiding solutions that cause the least possible distortion (modification) to the original database are called *ideal* or *optimal*. Lastly, non-exact but feasible solutions are called *approximate*.

As a final remark, we should point out that association rule hiding methodologies usually differ in the way they rank the aforementioned goals (especially the second and the third goal) in terms of importance of having them satisfied. With respect to the first goal, it is interesting to notice that for *any* database and *any* set of sensitive association rules there exists a feasible hiding solution, i.e. a solution that effectively hides all the sensitive association rules in the database. This means that the first goal can always be accomplished irrespective of the specific properties of the database or the peculiarities of the hiding problem. The most trivial way to identify a feasible

hiding solution in a database is to select one item from the generating itemset of each sensitive rule and delete it from all transactions of the database.

2.2.2 Problem Statement

Having presented the goals of association rule hiding methodologies, we now proceed to present the problem statement. Association rule hiding has been widely researched along two principal directions (henceforth referred as *variants*). The first variant involves approaches that aim at hiding specific association rules among those mined from the original database. The second variant, on the other hand, collects methodologies that aim at hiding specific frequent itemsets from those found when applying frequent itemset mining to the original database. The two variants of the problem are very similar in nature. Indeed, concealing the sensitive association rules by hiding their generating itemsets is a common strategy that is adopted by the majority of researchers. By ensuring that the itemsets that lead to the generation of a sensitive rule become insignificant in the disclosed database, the data owner can be certain that his or her sensitive knowledge is adequately protected from untrusted third parties. In what follows, we lay out the formal statement for each variant of the problem by introducing the problem statement both in the context of association rule mining and that of frequent itemset mining.

Variant 1: Hiding sensitive itemsets

We assume that we are provided with a database $\mathcal{D}_\mathcal{O}$, consisting of N transactions, and a threshold *mfreq* set by the owner of the data. After performing frequent itemset mining in $\mathcal{D}_\mathcal{O}$ with *mfreq*, we yield a set of frequent patterns, denoted as $\mathcal{F}_{\mathcal{D}_\mathcal{O}}$, among which a subset $\mathcal{S}$ contains patterns which are considered to be *sensitive* from the owner's perspective.

Given the set of sensitive itemsets $\mathcal{S}$, the goal of frequent itemset hiding methodologies is to construct a new, sanitized database $\mathcal{D}$ from $\mathcal{D}_\mathcal{O}$, which achieves to protect the sensitive itemsets $\mathcal{S}$ from disclosure, while minimally affecting the nonsensitive itemsets existing in $\mathcal{F}_{\mathcal{D}_\mathcal{O}}$ (i.e., the itemsets in $\mathcal{F}_{\mathcal{D}_\mathcal{O}} - \mathcal{S}$). The hiding of a sensitive itemset corresponds to a lowering of its statistical significance, depicted in terms of support, in the resulting database. To hide a sensitive itemset, the privacy preserving algorithm has to modify the original database $\mathcal{D}_\mathcal{O}$ in such a way that when the sanitized database $\mathcal{D}$ is mined at the same (or a higher) level of support, the frequent itemsets that are discovered are all nonsensitive.

Variant 2: Hiding sensitive association rules

We assume that we are provided with a database $\mathcal{D}_{\mathcal{O}}$, consisting of N transactions, and thresholds *mfreq* and *mconf* set by the owner of the data. After performing association rule mining in $\mathcal{D}_{\mathcal{O}}$ using thresholds *mfreq* and *mconf*, we yield a set of association rules, denoted as $\mathcal{R}$, among which a subset $\mathcal{R}_{\mathcal{S}}$ of $\mathcal{R}$ contains rules which are considered to be *sensitive* from the owner's perspective.

Given the set of sensitive association rules $\mathcal{R}_{\mathcal{S}}$, the goal of association rule hiding methodologies is to construct a new, sanitized database $\mathcal{D}$ from $\mathcal{D}_{\mathcal{O}}$, which achieves to protect the sensitive association rules $\mathcal{R}_{\mathcal{S}}$ from disclosure, while minimally affecting the nonsensitive rules existing in $\mathcal{R}$ (i.e., those in $\mathcal{R} - \mathcal{R}_{\mathcal{S}}$). The hiding of a sensitive association rule corresponds to a lowering of its significance, depicted in terms of support or confidence, in the resulting database. To hide a sensitive rule, the privacy preserving algorithm modifies the original database $\mathcal{D}_{\mathcal{O}}$ in such a way that when the sanitized database $\mathcal{D}$ is mined at the same (or a higher) levels of confidence and support, the association rules that are discovered are all nonsensitive.