The International Journal on Very Large Data Bases manuscript No.
(will be inserted by the editor)

Anonymity Preserving Pattern Discovery

Maurizio Atzori[1,2], **Francesco Bonchi**[1], **Fosca Giannotti**[1], **Dino Pedreschi**[2]

[1] Pisa KDD Laboratory
ISTI - CNR, Area della Ricerca di Pisa
Via Giuseppe Moruzzi, 1 - 56124 Pisa, Italy

[2] Pisa KDD Laboratory
Computer Science Department, University of Pisa
Largo B.Pontecorvo, 3 - 56127 Pisa, Italy

Received: date / Revised version: date

Abstract It is generally believed that data mining results do not violate the *anonymity* of the individuals recorded in the source database. In fact, data mining models and patterns, in order to ensure a required statistical significance, represent a large number of individuals and thus conceal individual identities: this is the case of the minimum support threshold in *frequent pattern* mining. In this paper we show that this belief is ill-founded. By shifting the concept of *k-anonymity* [42, 44] from the source data to the extracted patterns, we formally characterize the notion of a threat to anonymity in the context of pattern discovery, and provide a methodology to efficiently and effectively identify all possible such threats that arise from the disclosure of the set of extracted patterns. On this basis, we obtain a formal notion of privacy protection that allows the disclosure of the extracted knowledge while protecting the anonymity of the individuals in the source database. Moreover, in order to handle the cases where the threats to anonymity cannot be avoided, we study how to eliminate such threats by means of pattern (not data!) distortion performed in a controlled way.

Keywords: *Knowledge Discovery, Privacy Preserving Data Mining, Frequent Pattern Mining, Individual Privacy, Anonymity.*

1 Introduction

Improving trust in the knowledge society is a key requirement for its development. Privacy-awareness, if addressed at a technical level and acknowledged by regulations and social norms, may foster social acceptance and dissemination of new emerging knowledge-based applications. This is true of data mining, which is aimed at learning patterns, models and trends that hold across a collection of data. While the potential benefits of data mining are clear, it is also clear that the analysis of personal sensitive data arouses concerns about citizen's privacy, confidentiality and freedom. Obtaining the potential benefits of data mining with a privacy-aware technology would enable a wider social acceptance of a multitude of new services and applications based on the knowledge discovery process.

The awareness that privacy protection in data mining is a crucial issue has driven the attention of many researchers in the last few years, and consequently *Privacy Preserving Data Mining*, i.e., the study of data mining side-effects on privacy, has rapidly become a hot and lively research area, which receives an increasing attention from the research community [47]. However, despite such efforts, we agree with [13] that a common understanding of what is meant by "privacy" is still missing. As a consequence, there is a proliferation of many completely different approaches of privacy preserving data mining, but all sharing the same generic goal: producing valid mining models without disclosing "private" information.

As highlighted in [30], the approaches pursued so far leave a privacy question open: *do the data mining results themselves violate privacy?* Put in other words, do the disclosure of extracted patterns open up the risk of privacy breaches that may reveal sensitive information? In this paper we study when data mining results represent a threat to privacy. In particular, we concentrate on *individual privacy*, in the strict sense of *non-identifiability*, as prescribed by the European Union regulations on privacy, as well as US rules on protected health information (HIPAA rules). The medical domain is indeed a prototypical application instance for the framework we develop in this paper.

Correspondence to: francesco.bonchi@isti.cnr.it

Example 1 (Medical Knowledge Discovery) Medical informatics has become an integral part of successful medical institution. Many modern hospitals and health care institutions are now well equipped with monitoring and other data-collection devices, and data is gathered and shared in inter – and intra – hospital information systems. This increase in the volume of medical data available, has created the right premises for the birth of *Medical Data Mining*, i.e., the application of data analysis techniques to extract useful knowledge for supporting decision making in medicine [28]. Because medical data are collected on human subjects, there is an enormous ethical and legal tradition designed to prevent the abuse and misuse of medical data. The strength of the privacy requirements together with the incredible benefits that the whole society can achieve from it, make Medical Data Mining a challenging and unique research field; while the kind of privacy required (i.e., the anonymity of the patients in a survey), make it a perfect prototypical application instance for our framework.

In concrete, consider a medical institution where the usual hospital activity is coupled with medical research activity. Since physicians are the data collectors and holders, and they already know everything about their patients, they have unrestricted access to the collected information. Therefore, they can perform real mining on all available information using traditional mining tools – not necessarily the privacy preserving ones. This way they maximize the outcome of the knowledge discovery process, without any concern about privacy of the patients which are recorded in the data. But the anonymity of individuals patients becomes a key issue when the physicians want to share their discoveries (e.g., association rules holding in the data) with their scientific community.

In this article we concentrate on *frequent pattern mining*, and study anonymity in this setting. Note that frequent patterns are very basilar structures, that can be used as basic bricks to build more complex mining models such as association rules, classification or clustering models. The key question is: can anonymity be guaranteed when a collection of frequent patterns resulting from a data mining computation is disclosed?

1.1 Data Mining Results Can Violate Anonymity

At a first sight, it may seem that data mining results do not violate the anonymity of the individuals recorded in the source database. In fact, data mining models and patterns, in order to ensure a required statistical significance, represent a large number of individuals and thus conceal individual identities. In the following we show that this belief is ill-founded, using *association rules* [3] as prototypical example. The idea of mining association rules originates from the analysis of *market-basket* data where we are interested in finding rules describing customers behavior in buying products. In particular, association rules seek for sets of products which are associated, in the sense that they are bought together quite frequently.

An association rule is an expression $X \Rightarrow Y$ where X and Y are two disjoint sets of items. The association rule is said to be *valid* if:

1. the *support* of the itemset $X \cup Y$, i.e., the number of transactions in the database in which the set $X \cup Y$ appears, is greater than a given threshold;
2. the *confidence* (or accuracy) of the rule, defined as the conditional probability $P(Y \mid X)$, i.e., the support of $X \cup Y$ over the support of Y, is greater than a given threshold.

Since association rules in order to be valid must be common to a large number of individuals, i.e., their support must be larger than a given threshold, we might be tempted to conclude that, if such a threshold is large enough, we can always safely disclose the extracted association rules.

The next example shows that this is not true.

Example 2 Consider the following association rule:

$$a_1 \wedge a_2 \wedge a_3 \Rightarrow a_4 \quad [sup = 80,\ conf = 98.7\%]$$

where *sup* and $conf$ are the usual interestingness measures of *support* and *confidence* as defined above. Since the given rule holds for a number of individuals (80), which seems large enough to protect individual privacy, one could conclude that the given rule can be safely disclosed. But, is this all the information contained in such a rule? Indeed, one can easily derive the support of the premise of the rule:

$$sup(\{a_1, a_2, a_3\}) = \frac{sup(\{a_1, a_2, a_3, a_4\})}{conf} \approx \frac{80}{0.987} \approx 81.05$$

Given that the pattern $a_1 \wedge a_2 \wedge a_3 \wedge a_4$ holds for 80 individuals, and that the pattern $a_1 \wedge a_2 \wedge a_3$ holds for 81 individuals, we can infer that in our database there is just one individual for which the pattern $a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4$ holds. Later in the article we will show how the knowledge inferred can be used to link the identity of this individual to some sensitive information.

It is worth noting that this problem is very general: the given rule could be, instead of an association, a classification rule, or the path from the root to the leaf in a decision tree, and the same reasoning would still hold. Moreover, it is straightforward to note that, unluckily, the more accurate is a rule, the more unsafe it may be w.r.t. anonymity.

In this article we say that the two itemsets $\{a_1, a_2, a_3\}$ and $\{a_1, a_2, a_3, a_4\}$ represent an *inference channel*, for the anonymity of the individual corresponding to the pattern $a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4$. This is a trivial kind of inference channel, but in general, much more complex kinds of inference channels exist, as studied in the rest of this paper.

1.2 Article Contribution and Organization

In this article we study the anonymity problem in the very general setting of patterns which are *boolean formulas over a binary database*.

One might guess that each frequent pattern is related to a relatively large group of individuals in the source database, and therefore non-identifiability is guaranteed. Unfortunately, as shown in Example 2, this is not the case: a malicious adversary can reason on the collection of frequent patterns, and deduce new patterns from the available ones, precisely identifying individuals or small groups. In other words it is not sufficient to constrain each pattern to be anonymous in itself: we show in this paper how also the combination of patterns allows to infer new derived patterns, which may violate anonymity and hence break the non-identifiability requirement. This is the driving idea of our work.

To cope with this phenomenon we introduce a notion of anonymity which applies to patterns – rather than data – and study two natural problems: (*i*) how to check whether or not a collection of patterns guarantees anonymity, and (*ii*) if this is not the case, how to *sanitize* the collection of patterns, i.e., how to transform the output collection in such a way that anonymity is preserved, along with the quality of the disclosed patterns.

In Section 2 we justify originality and adequacy of our approach within the - rather young - state of the art of Privacy Preserving Data Mining. In Section 3 we briefly recall the works on k-anonymity on databases, and discuss the benefits of shifting such concepts from the data to the extracted patterns. Such discussion leads to the formal definition of *k-anonymous patterns* in Section 4. Here we characterize k-anonymous patterns, and we study the possible channels of *inference* in a collection of patterns that may be exploited by a malicious adversary to threat anonymity of the individuals recorded in the source data. The formal definition of the anonymity preservation problem as a problem of logical inference, is one of the major contributions of this article. Next, the two practical problems are addressed: how to detect the inference channels in a collection of patterns, and how to block them. In Section 5 is defined a first naïve algorithm to detect such potential threats which yields a methodology to check whether the mining results may be disclosed without any risk of violating anonymity. In Section 6 we introduce a condensed representation of the set of inference channels. A condensed representation is a subset of the original collection of patterns which contains the same information. The condensed representation we define has a twofold benefit, since it helps both the detecting and the blocking task. On one hand it reduces the computational cost of the detection task, yielding to an improved algorithm presented in Section 6. On the other hand, exploiting the condensed representation we avoid redundant sanitization and thus our blocking algorithms (developed in Section 7), geared on the condensed representation, introduce less distortion. In Section 8 we report the experimental analysis that we have conducted in order to assess the distortion introduced by our sanitization strategies; to measure time needed by our sanitization framework and to compare empirically the differences between k-anonymizing the data and k-anonymizing the patterns. Finally, in Section 9 we describe some on-going and future works and we conclude.

The results described in this article represents a preliminary step along a path that is crucial, both from the ethical point of view and that of social acceptance – data mining solutions that are not fully trustworthy will find insuperable obstacles to their deployment. On the other hand, demonstrably trustworthy solutions may open up tremendous opportunities for new knowledge-based applications of public utility and large societal and economic impact.

2 Related Work

As stated in the Introduction, many different approaches to Privacy Preserving Data Mining have emerged in the last few years. In the section we review the main approaches followed so far and we collocate our proposal within this rather young state-of-the-art.

The *Intensional Knowledge Hiding* approach, also known as *sanitization*, is aimed at hiding some intensional knowledge (i.e. rules/patterns) considered sensitive, which could be inferred from the data which is going to be disclosed. This hiding is usually obtained by *sanitizing* the database in input in such a way that the sensitive knowledge can no longer be inferred, while the original database is changed as less as possible [6,11,15, 35,36,43]. Notice that this approach aims to a kind of privacy which is more related to keep secret some corporate information, rather than individuals identity: thus, we should better refer to it as *secrecy*.

Another approach, more aimed at the privacy of the individual, is the *Extensional Knowledge Hiding* approach, sometimes referred to as *distribution reconstruction*. This approach addresses the issue of privacy preservation by perturbing the data in order to avoid the identification of the original database rows, while at the same time allowing the reconstruction of the data distribution at an aggregate level, in order to perform the mining [2, 4,18–22,27,31,34,41,25]. In other words, the extensional knowledge in the dataset is hidden, but is still possible to extract valid intensional knowledge.

Both approaches described above are applicable in contexts where what is disclosed is the data. During the last year a novel approach, shifting the privacy problem from the data to the mining models, has emerged in privacy preserving data mining [23,30,37]. All the previous approaches were focussed on producing a valid mining model without disclosing private data, but they still leave a privacy question open [30]: *do the data mining results themselves violate privacy?*

So far, just few works have investigated this issue. The work in [30] compliments the line of research in *secure distributed data mining* [12,14,16,17,26,29,40,46], but focussing on the possible privacy threat caused by the data mining results. In particular the authors study the case of a classifier trained over a mixture of different kind of data: *public* (known to every one including the adversary), *private/sensitive* (should remain unknown to the adversary), and *unknown* (neither sensitive nor known by the adversary). The authors propose a model for privacy implication of the learned classifier, and within this model, they study possible ways in which the classifier can be used by an adversary to compromise privacy. The work in [37] has some common aspects with the line of research in intensional knowledge hiding. But this time, instead of the problem of sanitizing the data, the problem of *association rule sanitization* is addressed. The data owner, rather than sharing the data prefer to mine it and share the discovered association rules. The basic assumption is that the data owner knows a set of restricted association rules that he does not want to disclose. The authors propose a framework to sanitize a set of association rules protecting the restricted ones: they show that in this context sanitizing directly the patterns, is some more information-preserving than sanitizing the data and mine the patterns from the sanitized data. This is somehow similar, even if in a different context, to our finding discussed in Section 8.3. In [23] a framework for evaluating classification rules in terms of their perceived privacy and ethical sensitivity is described. The proposed framework empowers the data miner with alerts for sensitive rules which can be accepted or dismissed by the user as appropriate. Such alerts are based on an aggregate *Sensitivity Combination Function*, which assigns to each rule a value of sensitivity by aggregating the sensitivity value (an integer in the range $0 \dots 10$) of each attribute involved in the rule. The process of labelling each attribute with its sensitivity value must be accomplished by the domain expert.

Our proposal clearly collocates within this new emerging area, but with distinctive original features. A common aspect of the three works above, is that they all require some *a priori* knowledge of what is sensitive. Instead we study when data mining results represent *per se* a threat to privacy, without any background knowledge of what is sensitive. The fundamental difference lies in generality: we propose a novel, objective definition of privacy compliance of patterns without any reference to a preconceived knowledge of sensitive data or patterns, on the basis of the rather intuitive and realistic constraint that the anonymity of individuals should be guaranteed. It should also be noted the different setting w.r.t. the other works in privacy preserving data mining: in our context no data perturbation or sanitization is performed, as we allow real mining on the real data, while focussing on the anonymity preservation properties of the extracted patterns.

3 k-Anonymity: from Data to Patterns

When the objective of a data owner is to disclose the data, *k-anonymity* is an important method for protecting the privacy of the individuals recorded in the data. The concept of k-anonymity was introduced by Samarati and Sweeney in [42,44]. In these works, it is shown that protection of individual sources does not guarantee protection when sources are cross-examined: a sensitive medical record, for instance, can be uniquely linked to a *named* voter record in a publicly available voter list through some shared attributes. The objective of k-anonymity is to eliminate such opportunities of inferring private information through cross linkage. According to this approach, the data holder identifies all possible attributes in the private information that can be found in other public databases, and thus could be exploited by a malicious adversary by means of cross linkage. Sets of attributes (like gender, date of birth, and zip code in the example above) that can be linked with external data to uniquely identify individuals in the population are called *quasi-identifiers*. Once the quasi-identifiers are known, a "sanitization" of the source data takes place: the data is transformed in such a way that, for every combination of values of the quasi-identifiers in the sanitized data, there are at least k records that share those values. Such a sanitization is obtained by generalization of attributes (the quasi-identifiers) and, when needed, suppression of tuples [45].

As stated in the Introduction, in this article we do not study how to safely disclose data, but instead we focus on the disclosure of patterns extracted by means of data mining. In our context the data owner is not willing to share the data – on the contrary, it is often legally responsible for protecting the data – and, instead, it is interested in publishing the knowledge discovered by mining the data. Therefore, a malicious adversary could only attack privacy exploiting the information which is present in the patterns, plus his own background knowledge about the technique used to extract the patterns.

A pattern produced by data mining techniques can be seen as a `select` query, which returns the set of tuples in the database which are captured by the given pattern. Thus we can shift the concept of k-anonymity from the data to the patterns in a straightforward way: we say that the result of a data mining extraction is k-anonymous if from any pattern inferred from such results is not possible to identify a group of tuples of cardinality less than k. More precisely:

- a single pattern p with support count $s > 0$ (i.e., occurring s times in the source database) is k-anonymous iff $s \geq k$, i.e., there are at least k tuples in the database satisfying p;
- a collection of patterns, each with support count, is k-anonymous iff each pattern in it is k-anonymous as well as any further pattern whose support can be inferred from the collection.

As in the classical k-anonymity framework for databases, also in our framework we can assume that attributes and their corresponding values are divided into two groups by the data owner: *quasi-identifiers* (i.e., public available attributes) and *sensitive attributes* (i.e., private attributes to be kept secret). In general, we can have patterns containing only quasi-identifiers, only sensitive attributes, or both. Notice that the information that can possibly violate the anonymity of individuals, as in the classical k-anonymity framework for databases, is not the release of sensitive values, but the possibility of linking individuals identities to sensitive data through quasi-identifiers.

In the following we show an example of a possible attack done by exploiting non k-anonymous patterns holding in the quasi-identifiers.

Example 3 Consider the two following association rules:

$Age = 27 \wedge ZIP = 45254 \wedge Religion = Christian \Rightarrow Native\_Country = USA$ $[sup = 758, conf = 99.8\%]$

$Age = 27 \wedge ZIP = 45254 \Rightarrow Native\_Country = USA$ $[sup = 1053, conf = 99.9\%]$

Both rules have high support and high confidence, so they are likely to be disclosed as interesting rules. Notice that they are also apparently safe (not anonymity violating). In fact, in the first rule there's one sensitive attribute/value ($Religion = Christian$) in the premise of the rule, so a malicious attacker should know this sensitive attribute in order to apply the rule. But being a sensitive attribute, *Religion* can not be known by any attacker. In the second rule instead, we have only quasi-identifiers, therefore we are not releasing any linking opportunity to sensitive information. Therefore these two rules are apparently safe, but from them we can learn that:

$Age = 27 \wedge ZIP = 45254 \wedge \neg(Native\_Country = USA) \Rightarrow Religion = Christian$ $[sup = 1, conf = 100\%]$

In fact from the first rule we can infer that there's only one individual such that: $Age = 27 \wedge ZIP = 45254 \wedge Religion = Christian \wedge \neg(Native\_Country = USA)$. Moreover, from the second rule, we know that there is only one individual such that: $Age = 27 \wedge ZIP = 45254 \wedge \neg(Native\_Country = USA)$. Therefore we can easily conclude that this individual is the same as before and she is *Christian*. Age, postcode, and native country are public available information (quasi-identifiers), which in this 100%-confidence rule, covering just one individual, are linked to a sensitive information as individual's religion.

This is exactly the linking attack for which k-anonymity defence has been developed. In the database framework, once we have k-anonymized our data, we can not identify a singular individual by means of a combination of quasi-identifiers. Similarly, in our framework, by sanitizing non k-anonymous patterns where all the items are quasi-identifiers, we block this kind of attack. More concretely, in our framework, we identify and sanitize (by increasing its support to k, or by decreasing it to 0) the pattern $Age = 27 \wedge ZIP = 45254 \wedge \neg(Native\_Country = USA)$, which is made only of quasi-identifiers.

In conclusion, in our framework by sanitizing the patterns that non k-anonymous and made only of quasi-identifiers we avoid linking attacks, unless they regards at least k individuals.

In the following we study the problem of how to produce a set of patterns that are k-anonymous, and of course, as close as possible to the real patterns holding in the data. We set our investigations in the very general context where the source data is a binary database, and the kind of patterns extracted (and disclosed) are *frequent itemsets*, i.e., sets of attributes which appear all set to 1 in a number of tuples larger than a given frequency threshold. Therefore, with the aim of facilitating the theoretical investigation, and for sake of clarity of the presentation, for the moment we do not consider the semantics of the attributes, and the possibility of generalizing them. Moreover we do not distinguish between quasi-identifiers and other attributes: in our context all attributes are quasi-identifier. Note that these two assumptions do not weaken our contribution; on the contrary we develop a very general theory that could be easily instantiated to the more concrete case of categorical data originating from relational tables. In the experiments reported in the article, on binary databases, we considered all the attributes as quasi-identifiers in order to test our approach in the worst case scenario. Introducing the distinction between quasi-identifiers and non, as well as the possibility of generalizing some attributes, will just make our patterns sanitization task easier, and would reduce the amount of distortion needed to k-anonymize the patterns.

Finally, notice that a trivial solution to our problem would be to first k-anonymize the data using some well known technique, and then mine the patterns from the k-anonymized data. In fact, by mining a k-anonymized database no patterns threatening anonymity can be obtained. But such approach would produce patterns impoverished by the information loss which is intrinsic in the generalization and suppression techniques. Since our objective is to extract valid and interesting patterns, we propose to postpone k-anonymization after the actual mining step. In other words, we do not enforce k-anonymity onto the source data, but instead we move such a concept to the extracted patterns. Since there is a clear correspondence between patterns and data, k-anonymizing the patterns can be seen as k-anonymizing just the portion of interest of data, the portion corresponding to the patterns. Following this way we introduce much less distortion. This issue will be further analyzed in Section 8.3.

4 *k*-Anonymous Patterns

A preliminary version of this section, containing the basic definitions is in [8]. We start by defining binary databases and patterns following the notation in [24].

Definition 1 (Binary Database) *A binary database $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ consists of a finite set of binary variables $\mathcal{I} = \{i_1, \ldots, i_p\}$, also known as* items*, and a finite multiset $\mathcal{T} = \{t_1, \ldots, t_n\}$ of p-dimensional binary vectors recording the values of the items. Such vectors are also known as* transactions*.*

Definition 2 (Pattern) *A pattern for the variables in $\mathcal{I}$ is a logical (propositional) sentence built by AND ($\wedge$), OR ($\vee$) and NOT ($\neg$) logical connectives, on variables in $\mathcal{I}$. The domain of all possible patterns is denoted $\mathcal{P}at(\mathcal{I})$.*

A binary database $\mathcal{D}$ is given in Figure 1(a). In this context, a row or transaction of $\mathcal{D}$ is a tuple recording the values of some attributes (or items) of an individual. Therefore in this context, the objective of our analysis is the anonymity of transactions.

According to Definition 2, $e \wedge (\neg b \vee \neg d)$, where $b, d, e \in \mathcal{I}$, is a pattern. One of the most important properties of a pattern is its frequency in the database, i.e. the number of individuals (transactions) in the database which make the given pattern true[1].

Definition 3 (Support) *Given a database $\mathcal{D}$, a transaction $t \in \mathcal{D}$ and a pattern p, we write $p(t)$ if t makes p true. The support of p in $\mathcal{D}$ is given by the number of transactions which makes p true:*

$$sup_{\mathcal{D}}(p) = |\{t \in \mathcal{D} \mid p(t)\}|.$$

If for a given pattern this number is very low (i.e. smaller than an anonymity threshold k) but not null, then the pattern represents a threat for the anonymity of the individuals about which the given pattern is true.

Definition 4 (k-Anonymous Pattern) *Given a binary database $\mathcal{D}$ and an anonymity threshold k, a pattern p is said to be* k-anonymous *if $sup_{\mathcal{D}}(p) \geq k$ or $sup_{\mathcal{D}}(p) = 0$.*

The objective of this article is to study when the output of a mining extraction could be exploited by a malicious adversary to identify *non* k-anonymous patterns, i.e., small groups of individuals. In particular, we focus on *frequent itemset mining*: one of the most basilar and fundamental mining tasks.

Itemsets are a particular class of patterns: conjunctions of positive valued variables, or in other words, sets of items. The retrieval of itemsets which satisfy a minimum frequency property is the basic step of many data mining tasks, including (but not limited to) association rules [3,5].

[1] The notion of truth of a pattern w.r.t. a transaction t is defined in the usual way: t makes p true iff t is a model of the propositional sentence p.

Notation: patterns	Notation: itemsets
$sup_{\mathcal{D}}(a \vee f) = 11$	$sup_{\mathcal{D}}(abc) = 6$
$sup_{\mathcal{D}}(e \wedge (\neg b \vee \neg d)) = 4$	$sup_{\mathcal{D}}(abde) = 7$
$sup_{\mathcal{D}}(h \wedge \neg b) = 1$	$sup_{\mathcal{D}}(cd) = 9$

Table 1 Notation: example of general patterns and itemsets with their support in the database $\mathcal{D}$ of Figure 1(a).

Definition 5 (σ-Frequent Itemset) *The set of all itemsets $2^{\mathcal{I}}$, is a pattern class consisting of all possible conjunctions of the form $i_1 \wedge i_2 \wedge \ldots \wedge i_m$. Given a database $\mathcal{D}$ and a minimum support threshold σ, the set of σ-frequent itemsets in $\mathcal{D}$ is denoted:*

$$\mathcal{F}(\mathcal{D}, \sigma) = \{\langle X, sup_{\mathcal{D}}(X)\rangle \mid X \in 2^{\mathcal{I}} \wedge sup_{\mathcal{D}}(X) \geq \sigma\}.$$

Frequent Itemset Mining (FIM), i.e., computing $\mathcal{F}(\mathcal{D}, \sigma)$, is one of the most studied algorithmic problems in data mining: hundreds of algorithms have been developed and compared (see [1] for a good repository) since the first proposal of the well-known APRIORI algorithm [5]. As shown in Figure 1(b), the search space of the FIM problem is a lattice, which is typically visited breadth-first or depth-first, exploiting the following interesting property: $\forall X \subset Y \in 2^{\mathcal{I}}. sup_{\mathcal{D}}(X) \geq sup_{\mathcal{D}}(Y)$.

This property, also known as *"anti-monotonicity of frequency"* or *"Apriori trick"*, is exploited by the APRIORI algorithm (and by almost all FIM algorithms) with the following heuristic: if an itemset X does not satisfy the minimum support constraint, then no superset of X can be frequent, and hence they can be pruned from the search space. This pruning can affect a large part of the search space, since itemsets form a lattice.

Itemsets are usually denoted in the form of set of the items in the conjunction, e.g. $\{i_1, \ldots, i_m\}$; or sometimes, simply $i_1 \ldots i_m$. Table 1 shows the different notation used for general patterns and for itemsets.

Example 4 Given the binary database $\mathcal{D}$ in Figure 1(a), and a minimum support threshold $\sigma = 8$, we have that: $\mathcal{F}(\mathcal{D}, 8) = \{\langle\emptyset, 12\rangle, \langle a, 9\rangle, \langle b, 8\rangle, \langle c, 9\rangle, \langle d, 10\rangle, \langle e, 11\rangle, \langle ab, 8\rangle, \langle ae, 8\rangle, \langle cd, 9\rangle, \langle ce, 9\rangle, \langle de, 10\rangle, \langle cde, 9\rangle\}$.

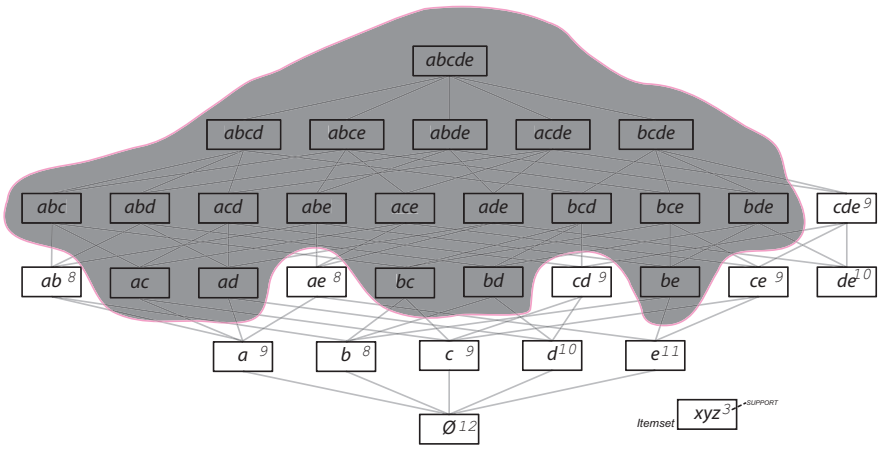
As already stated, the problem addressed in this article is given by the possibility of inferring from the output of frequent itemset mining, i.e, $\mathcal{F}(\mathcal{D}, \sigma)$, the existence of patterns with very low support (i.e., smaller than an anonymity threshold k, but not null): such patterns represent a threat for the anonymity of the individuals about which they are true.

Recall our motivating example: from the two disclosed frequent itemsets $\{a_1, a_2, a_3\}$ and $\{a_1, a_2, a_3, a_4\}$

$\mathcal{D}$

	a	b	c	d	e	f	g	h
t_1	1	1	1	1	1	1	1	1
t_2	1	1	1	1	1	0	1	0
t_3	1	1	1	1	1	0	0	0
t_4	1	1	1	1	1	1	1	0
t_5	1	1	1	1	1	0	0	0
t_6	1	1	1	1	1	0	0	0
t_7	1	1	0	1	1	0	0	0
t_8	1	0	0	0	1	1	1	0
t_9	0	0	1	1	1	1	1	0
t_{10}	0	0	1	1	1	0	0	0
t_{11}	0	0	1	1	1	1	1	1
t_{12}	1	1	0	0	0	1	1	0

(a)



(b)

Fig. 1 Running example: (a) the binary database $\mathcal{D}$; and (b) a graphical representation of the lattice of itemsets $2^{\mathcal{I}}$ for $\mathcal{I} = \{a, b, c, d, e\}$: the set of σ-frequent ($\sigma = 8$) itemsets over $\mathcal{D}$ is displayed together with their supports (i.e., $\mathcal{F}(\mathcal{D}, 8)$). This is what is disclosed, i.e., the whole information that a malicious adversary can use, while the grey area represents what is not known. The singleton items f, g and h, and all their supersets, are not displayed: since they are infrequent, the adversary can not even know that these items (or attributes) are present in the source data.

(and their supports) it was possible to infer the existence of the *non* k-anonymous pattern $a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4$.

Informally, we call *inference channel* any collection of itemsets (with their respective supports), from which it is possible to infer non k-anonymous patterns. In the following we formally define and characterize inference channels.

4.1 Inference Channels

Before introducing our anonymity preservation problem, we need to define the inference of supports, which is the basic tool for the attacks to anonymity.

Definition 6 (Database Compatibility) *A set S of pairs $\langle X, n\rangle$, where $X \in 2^{\mathcal{I}}$ and $n \in \mathbb{N}$, and a database $\mathcal{D}$ are said to be compatible if $\forall \langle X, n\rangle \in S. sup_{\mathcal{D}}(X) = n$.*

Definition 7 (Support Inference) *Given a set S of pairs $\langle X, n\rangle$, where $X \in 2^{\mathcal{I}}$ and $n \in \mathbb{N}$, and a pattern $p \in \mathcal{P}at(\mathcal{I})$ we say that $S \models sup(p) > x$ (respectively $S \models sup(p) < x$) if, for all databases $\mathcal{D}$ compatible with S, we have that $sup_{\mathcal{D}}(p) > x$ (respectively $sup_{\mathcal{D}}(p) < x$).*

Definition 8 (Inference Channel) *An inference channel $\mathcal{C}$ is a set of pairs $\langle X, n\rangle$, where $X \in 2^{\mathcal{I}}$ and $n \in \mathbb{N}$, such that:*

$$\exists p \in \mathcal{P}at(\mathcal{I}) : \mathcal{C} \models 0 < sup(p) < k.$$

As suggested by Example 2, a simple inference channel is given by any itemset X which has a superset $X \cup \{a\}$ such that $0 < sup_{\mathcal{D}}(X) - sup_{\mathcal{D}}(X \cup \{a\}) < k$. In this case the pair $\langle X, sup_{\mathcal{D}}(X)\rangle, \langle X \cup \{a\}, sup_{\mathcal{D}}(X \cup \{a\})\rangle$ is an inference channel for the non k-anonymous pattern $X \wedge \neg a$, whose support is directly given by $sup_{\mathcal{D}}(X) - sup_{\mathcal{D}}(X \cup \{a\})$. This is a trivial kind of inference channel.

Do more complex structures of itemsets exist that can be used as inference channels?

In general, the support of a conjunctive pattern $p = i_1 \wedge \cdots \wedge i_m \wedge \neg a_1 \wedge \cdots \wedge \neg a_n$ can be inferred if we know the support of itemsets $I = \{i_1, \ldots, i_m\}$, $J = I \cup \{a_1, \ldots, a_n\}$, and every itemset L such that $I \subset L \subset J$.

Lemma 1 *Given a pattern $p = i_1 \wedge \cdots \wedge i_m \wedge \neg a_1 \wedge \cdots \wedge \neg a_n$ we have that:*

$$sup_{\mathcal{D}}(p) = \sum_{I \subseteq X \subseteq J} (-1)^{|X \setminus I|} sup_{\mathcal{D}}(X)$$

where $I = \{i_1, \ldots, i_m\}$ and $J = I \cup \{a_1, \ldots, a_n\}$.

Proof (Sketch) The proof follows directly from the definition of support and the well-known *inclusion-exclusion principle* [32].

Following the notation in [10], we denote the right-hand side of the equation above as $f_I^J(\mathcal{D})$.

Example 5 In the database $\mathcal{D}$ in Figure 1(a) we have that $sup_{\mathcal{D}}(b \wedge \neg d \wedge \neg e) = f_b^{bde}(\mathcal{D}) = sup_{\mathcal{D}}(b) - sup_{\mathcal{D}}(bd) - sup_{\mathcal{D}}(be) + sup_{\mathcal{D}}(bde) = 8 - 7 - 7 + 7 = 1$.

Definition 9 *Given a database $\mathcal{D}$, and two itemsets $I, J \in 2^{\mathcal{I}}$, $I = \{i_1, \ldots, i_m\}$ and $J = I \cup \{a_1, \ldots, a_n\}$, if $0 < f_I^J(\mathcal{D}) < k$, then the set*

$$\{\langle X, sup_{\mathcal{D}}(X)\rangle \mid I \subseteq X \subseteq J\}$$

constitutes an inference channel for the non k-anonymous pattern $p = i_1 \wedge \cdots \wedge i_m \wedge \neg a_1 \wedge \cdots \wedge \neg a_n$. We denote such inference channel $\mathcal{C}_I^J$ and we write $sup_{\mathcal{D}}(\mathcal{C}_I^J) = f_I^J(\mathcal{D})$.

Example 6 Consider the database $\mathcal{D}$ of Figure 1(a), and suppose $k = 3$. We have that $\mathcal{C}_{\emptyset}^{cde}$ is an inference channel of support 1. In fact we got that:
$sup_{\mathcal{D}}(\mathcal{C}_{\emptyset}^{cde}) = f_{\emptyset}^{cde}(\mathcal{D}) = sup_{\mathcal{D}}(\emptyset) - sup_{\mathcal{D}}(c) - sup_{\mathcal{D}}(d) - sup_{\mathcal{D}}(e) + sup_{\mathcal{D}}(cd) + sup_{\mathcal{D}}(ce) + sup_{\mathcal{D}}(de) - sup_{\mathcal{D}}(cde) = 12 - 9 - 10 - 11 + 9 + 9 + 10 - 9 = 1$.

This means that there is only one transaction $t \in \mathcal{D}$ is such that $\neg c \wedge \neg d \wedge \neg e$ (transaction t_{12}). A graphical representation of this channel is given in Figure 2.

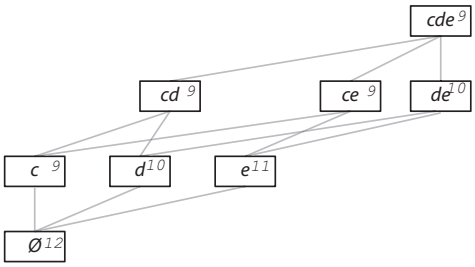


Fig. 2 A detail of Figure 1(b): this is an inference channel, (i.e, $\mathcal{C}_{\emptyset}^{cde}$ for $k = 3$).

The next Theorem states that if there exists a non k-anonymous pattern, then there exists a pair of itemsets $I \subseteq J \in 2^{\mathcal{I}}$ such that $\mathcal{C}_I^J$ is an inference channel.

Theorem 1

$$\forall p \in \mathcal{P}at(\mathcal{I}) : 0 < sup_{\mathcal{D}}(p) < k \;.\; \exists\, I, J \in 2^{\mathcal{I}} : \mathcal{C}_I^J$$

Proof Let us consider a generic pattern $p \in \mathcal{P}at(\mathcal{I})$. Without loss of generality p is in *normal disjunctive form*: $p = p_1 \vee \ldots \vee p_q$, where each $p_1, \ldots, p_q$ is a conjunctive pattern, i.e., a pattern made only by conjunction and negation as the one in Lemma 1. We have that:

$$sup_{\mathcal{D}}(p) \geq \max_{1 \leq i \leq q} sup_{\mathcal{D}}(p_i).$$

Since $sup_{\mathcal{D}}(p) < k$ we have for all patterns p_i that $sup_{\mathcal{D}}(p_i) < k$. Moreover, since $sup_{\mathcal{D}}(p) > 0$ there is at least a pattern p_i such that $sup_{\mathcal{D}}(p_i) > 0$. Therefore, there is at least a conjunctive pattern p_i such that $0 < sup_{\mathcal{D}}(p_i) < k$.

From Lemma 1, we have that $\exists\, I \subseteq J \in 2^{\mathcal{I}} : sup_{\mathcal{D}}(p_i) = f_I^J(\mathcal{D})$. Since $0 < sup_{\mathcal{D}}(p_i) = f_I^J(\mathcal{D}) < k$ we have that $\mathcal{C}_I^J$ is an inference channel.

Corollary 1 *Given a database $\mathcal{D}$, and an anonymity threshold k:*

$$\nexists\, I, J \in 2^{\mathcal{I}} : \mathcal{C}_I^J \Rightarrow \nexists p \in \mathcal{P}at(\mathcal{I}) : 0 < sup_{\mathcal{D}}(p) < k$$

5 Detecting Inference Channels

The problem addressed in the following is the detection of anonymity threats in the output of a frequent itemset extraction. From Corollary 1 we can conclude that by detecting and sanitizing all inference channels of the form $\mathcal{C}_I^J$, we can produce a k-anonymous output which can be safely disclosed. However, when we instantiate the general theory above to the concrete case in which we want to disclose a set of frequent itemsets (.i.e., $\mathcal{F}(\mathcal{D}, \sigma)$ and not the whole $2^{\mathcal{I}}$), the situation is made more complex by the frequency threshold. In fact, frequency divides the lattice of itemsets $2^{\mathcal{I}}$ in two parts: the frequent part which is disclosed, and the infrequent part (i.e., the grey area in Figure 1(b)) which is not disclosed.

This division induces a distinction also on the kind of patterns:

- patterns made only by composing pieces of $\mathcal{F}(\mathcal{D}, \sigma)$;
- patterns made also using infrequent itemsets.

The following definition precisely characterize the first kind.

Definition 10 (σ-vulnerable Pattern) *Given a general pattern $p \in \mathcal{P}at(\mathcal{I})$, we can assume without loss of generality that p is in normal disjunctive form: $p = p_1 \vee \ldots \vee p_q$, where each p_i is a conjunctive pattern. Given a database $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ and a minimum support threshold σ, we define $\mathcal{P}at(\mathcal{D}, \sigma) \subseteq \mathcal{P}at(\mathcal{I})$ as the set of patterns $p = p_1 \vee \ldots \vee p_q$ such that, for each conjunctive pattern p_i, if we consider the set of all items in it, i.e., the itemset obtained removing the negation symbols from it, such itemset is frequent. We call σ-vulnerable each pattern in $\mathcal{P}at(\mathcal{D}, \sigma)$.*

Example 7 Given a database $\mathcal{D}$ and a minimum support threshold σ the pattern $(a \wedge \neg b) \vee (d \wedge \neg c)$ is in $\mathcal{P}at(\mathcal{D}, \sigma)$ if $\langle ab, sup_{\mathcal{D}}(ab) \rangle \in \mathcal{F}(\mathcal{D}, \sigma)$ and $\langle cd, sup_{\mathcal{D}}(cd) \rangle \in \mathcal{F}(\mathcal{D}, \sigma)$. In the same case the pattern $a \wedge d \wedge \neg c$ is not in $\mathcal{P}at(\mathcal{D}, \sigma)$ if $\langle acd, sup_{\mathcal{D}}(acd) \rangle \notin \mathcal{F}(\mathcal{D}, \sigma)$.

The interest in the σ-vulnerable patterns lies in the consideration that a malicious attack starts from the delivered knowledge, i.e, the frequent itemsets.

We next project Theorem 1 on this class of patterns.

Theorem 2 *Given a database $\mathcal{D}$, a minimum support threshold σ and an anonymity threshold k, we have that $\forall p \in \mathcal{P}at(\mathcal{D}, \sigma) : 0 < sup_{\mathcal{D}}(p) < k$ there exist two itemsets $I, J \in 2^{\mathcal{I}}$ such that $\langle I, sup_{\mathcal{D}}(I) \rangle, \langle J, sup_{\mathcal{D}}(J) \rangle \in \mathcal{F}(\mathcal{D}, \sigma)$ and $\mathcal{C}_I^J$.*

Proof (Sketch) The proof follows directly from Theorem 1 and Definition 10.

Corollary 2 *Given a database $\mathcal{D}$, and an anonymity threshold k: $\nexists\, \langle I, sup_{\mathcal{D}}(I) \rangle, \langle J, sup_{\mathcal{D}}(J) \rangle \in \mathcal{F}(\mathcal{D}, \sigma) : \mathcal{C}_I^J \Rightarrow \nexists p \in \mathcal{P}at(\mathcal{D}, \sigma) : 0 < sup_{\mathcal{D}}(p) < k$.*

Therefore, if we know that the set of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$ that we have extracted does not contain any inference channel of the form $\mathcal{C}_I^J$, we can be sure that a malicious adversary will never infer from it a *non* k-anonymous, σ-vulnerable pattern. In the rest of the article we focus on this essential kind of patterns: as stated above, a malicious adversary can easily find inference channels made up only of elements which are present in the disclosed output. However, these inference channels are not the unique possible source of inference: further inference channels involving also infrequent itemsets could be possibly discovered, albeit in a much more complex way. In fact, in [10] deduction rules to derive tight bounds on the support of itemsets are introduced. Given an itemset J, if for each subset $I \subset J$ the support $sup_{\mathcal{D}}(I)$ is known, such rules allow to compute lower and upper bounds on the support of J. Let l be the greatest lower bound we can derive, and u the smallest upper bound we can derive: if we find that $l = u$ then we can infer that $sup_{\mathcal{D}}(J) = l = u$ without actual counting. In this case J is said to be a *derivable itemset*. We transpose such deduction techniques in our context and observe that they can be exploited to discover information about infrequent itemsets (i.e., infer supports in the grey area of Figure 1(b)), and from these to discover inference channels crossing the border with the grey area, or even inference channels holding completely within the grey area.

This higher-order problem is discussed later in Section 9. However, here we can say that the techniques to detect this kind of inference channels and to block them are very similar to the techniques for the first kind of channels. This is due to the fact that both kinds of channels rely on the same concept: inferring supports of larger itemsets from smaller ones. Indeed, the key equation of our work (Lemma 1) is also the basis of the deduction rules proposed in [10]. For the moment, let us restrict our attention to the essential form of inference channel, namely those involving frequent itemsets only.

Problem 1 (Inference Channels Detection) Given a collection of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$ and an anonymity threshold k, our problem consists in detecting all possible inference channels $\mathcal{C} \subseteq \mathcal{F}(\mathcal{D}, \sigma) \ : \ \exists p \in \mathcal{P}at(\mathcal{D}, \sigma) : \mathcal{C} \models 0 < sup_{\mathcal{D}}(p) < k$.

Our mining problem can be seen as a second-order frequent pattern extraction with two frequency thresholds: the usual minimum support threshold σ for itemsets (as defined in Definition 5), and an anonymity threshold k for general patterns (as defined in Definition 2).

Note that an itemset with support less than k is itself a non k-anonymous, and thus dangerous, pattern. However, since we are dealing with σ-frequent itemsets, and since we can reasonably assume that $\sigma \gg k$, such pattern would be discarded by the usual mining algorithms.

We just stated that we can reasonably assume σ to be much larger than k. In fact σ, in real-world applications is usually in the order of hundreds, or thousands, or (more frequently) much larger. Consider that having a small σ on a real-world database, would produce an extremely large number of associations in output, or it would lead to an unfeasible computation. On the other hand, the required level of anonymity k is usually in the order of tens or even smaller. Therefore, it is reasonable to assume $\sigma \gg k$. However, for sake of completeness, if we have $\sigma < k$ then our mining problem will be trivially solved by adopting k as minimum support threshold in the mining of frequent itemsets.

In the rest of this paper we will avoid discussing this case again, and we will always assume $\sigma > k$.

Definition 11 *The set of all $\mathcal{C}_I^J$ holding in $\mathcal{F}(\mathcal{D}, \sigma)$, together with their supports, is denoted* $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma)) = \{\langle \mathcal{C}_I^J, f_I^J(\mathcal{D})\rangle \mid 0 < f_I^J(\mathcal{D}) < k \wedge \langle J, sup_{\mathcal{D}}(J)\rangle \in \mathcal{F}(\mathcal{D}, \sigma)\}$.

Example 8 Consider the database $\mathcal{D}$ in Figure 1(a) and suppose $\sigma = 8$ and $k = 3$. The following is the set of inference channels holding in $\mathcal{F}(\mathcal{D}, 8)$:
$\mathcal{C}h(3, \mathcal{F}(\mathcal{D}, 8)) = \{\langle \mathcal{C}_\emptyset^d, 2\rangle, \langle \mathcal{C}_\emptyset^{de}, 1\rangle, \langle \mathcal{C}_e^{de}, 1\rangle, \langle \mathcal{C}_d^{dc}, 1\rangle, \langle \mathcal{C}_\emptyset^{dc}, 2\rangle, \langle \mathcal{C}_{de}^{dce}, 1\rangle, \langle \mathcal{C}_\emptyset^{dce}, 1\rangle, \langle \mathcal{C}_e^{dce}, 1\rangle, \langle \mathcal{C}_\emptyset^{ce}, 1\rangle, \langle \mathcal{C}_e^{ce}, 2\rangle, \langle \mathcal{C}_a^{ae}, 1\rangle, \langle \mathcal{C}_a^{ab}, 1\rangle, \langle \mathcal{C}_\emptyset^{e}, 1\rangle\}$.

Algorithm 1 Naïve Inference Channel Detector

Input: $\mathcal{F}(\mathcal{D}, \sigma), k$
Output: $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$
1: $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma)) = \emptyset$
2: **for all** $\langle J, sup(J)\rangle \in \mathcal{F}(\mathcal{D}, \sigma)$ **do**
3: **for all** $I \subseteq J$ **do**
4: **compute** f_I^J;
5: **if** $0 < f_I^J < k$ **then**
6: **insert** $\langle \mathcal{C}_I^J, f_I^J\rangle$ **in** $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$;

Algorithm 1 detects all possible inference channels $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$ that hold in a collection of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$ by checking all possible pairs of itemsets $I, J \in \mathcal{F}(\mathcal{D}, \sigma)$ such that $I \subseteq J$. This could result in a very large number of checks. Suppose that $\mathcal{F}(\mathcal{D}, \sigma)$ is formed only by a maximal itemset Y and all its subsets (an itemset is maximal if none of its proper supersets is in $\mathcal{F}(\mathcal{D}, \sigma)$). If $|Y| = n$ we get $|\mathcal{F}(\mathcal{D}, \sigma)| = 2^n$ (we also count the empty set), while the number of possible $\mathcal{C}_I^J$ is $\sum_{1 \leq i \leq n} \binom{n}{i} (2^i - 1)$. In the following Section we study some interesting properties that allow to dramatically reduce the number of checks needed to retrieve $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$.

6 A Condensed Representation

In this section we introduce a condensed representation of $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$. A condensed representation of a collection of patterns (in our case of $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$) is a

subset of the collection which is more efficient to compute, and from which we can reconstruct the original collection without accessing the database again. In other words it removes redundancy while preserving all the information.

The benefits of having such condensed representation go far beyond mere efficiency in the detection phase. In fact, by removing the redundancy existing in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, we also implicitly avoid redundant sanitization, when blocking the channels holding in $\mathcal{F}(\mathcal{D}, \sigma)$, to produce a safe output (the issue of how to sanitize the inference channels found in $\mathcal{F}(\mathcal{D}, \sigma)$ is addressed in Section 7) .

Consider, for instance, the two inference channels $\langle \mathcal{C}^e_\emptyset, 1\rangle$ and $\langle \mathcal{C}^{de}_\emptyset, 1\rangle$ holding in $\mathcal{C}h(3, \mathcal{F}(\mathcal{D}, 8))$ of our running example: one is more specific than the other, but they both uniquely identify transaction t_{12}. It is easy to see that many other families of equivalent, and thus redundant, inference channels can be found. *How can we directly identify one and only one representative inference channel in each family of equivalent ones?* The theory of *closed itemsets* can help us with this problem.

Closed itemsets were first introduced in [38] and received a great deal of attention especially by an algorithmic point of view [49,39]. They are a concise and lossless representation of all frequent itemsets, i.e., they contain the same information without redundancy. Intuitively, a closed itemset groups together all its subsets that have its same support; or in other words, it groups together itemsets which identify the same group of transactions.

Definition 12 (Closure Operator) *Given the function $f(T) = \{i \in \mathcal{I} \mid \forall t \in T, i(t)\}$, which returns all the items included in the set of transactions T, and the function $g(X) = \{t \in \mathcal{T} \mid X(t)\}$ which returns the set of transactions supporting a given itemset X, the composite function $c = f \circ g$ is the closure operator.*

Definition 13 (Closed Itemset) *An itemset I is closed if and only if $c(I) = I$. Alternatively, a closed itemset can be defined as an itemset whose supersets have a strictly smaller support. Given a database $\mathcal{D}$ and a minimum support threshold σ, the set of frequent closed itemsets is denoted: $\mathcal{C}l(\mathcal{D}, \sigma) = \{\langle X, sup_{\mathcal{D}}(X)\rangle \in \mathcal{F}(\mathcal{D}, \sigma) | \nexists Y \supset X s.t. \langle Y, sup_{\mathcal{D}}(X)\rangle \in \mathcal{F}(\mathcal{D}, \sigma)\}$.*

Example 9 Given the binary database $\mathcal{D}$ in Figure 1(a), and a minimum support threshold $\sigma = 8$, we have that: $\mathcal{C}l(\mathcal{D}, 8) = \{\langle \emptyset, 12\rangle, \langle a, 9\rangle, \langle e, 11\rangle, \langle ab, 8\rangle, \langle ae, 8\rangle, \langle de, 10\rangle, \langle cde, 9\rangle\}$.

Definition 14 (Maximal Frequent Itemset) *An itemset $I \in \mathcal{C}l(\mathcal{D}, \sigma)$ is said to be maximal if and only if $\nexists J \supset I$ s.t. $J \in \mathcal{C}l(\mathcal{D}, \sigma)$.*

Analogously to what happens for the pattern class of itemsets, if we consider the pattern class of conjunctive patterns we can rely on the *anti-monotonicity property of frequency*. For instance, the number of transactions for which the pattern $a \wedge \neg c$ holds is always larger than the number of transactions for which the pattern $a \wedge b \wedge \neg c \wedge \neg d$ holds. This can be straightforwardly transposed to inference channels.

Definition 15 *Given two inference channels $\mathcal{C}^J_I$ and $\mathcal{C}^L_H$ we say that $\mathcal{C}^J_I \preceq \mathcal{C}^L_H$ when $I \subseteq H$ and $(J \setminus I) \subseteq (L \setminus H)$.*

Proposition 1 $\mathcal{C}^J_I \preceq \mathcal{C}^L_H \Rightarrow \forall \mathcal{D}\ .\ f^J_I(\mathcal{D}) \geq f^L_H(\mathcal{D})$.

It follows that, when detecting inference channels, whenever we find a two itemsets $H \subseteq L$ such that $f^L_H(\mathcal{D}) \geq k$, we can avoid checking the support of all $\mathcal{C}^J_I \preceq \mathcal{C}^L_H$, since they will not be inference channels.

Definition 16 (Maximal Inference Channel) *An inference channel $\mathcal{C}^J_I$ is said to be* maximal *w.r.t. $\mathcal{D}$, k and σ, if $\forall H, L$ such that $I \subseteq H$ and $(J \setminus I) \subseteq (L \setminus H)$, $f^L_H = 0$. The set of maximal inference channels is denoted $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$.*

Proposition 2

$\mathcal{C}^J_I \in \mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma)) \Rightarrow I \in \mathcal{C}l(\mathcal{D}, \sigma) \wedge J$ *is maximal.*

Proof *i)* $I \in \mathcal{C}l(\mathcal{D}, \sigma)$: if I is not closed then consider its closure $c(I)$ and consider $J' = J \cup (c(I) \setminus I)$. For the definition of closure, the set of transactions containing I is the same of the set of transactions containing $c(I)$, and the set of transactions containing J' is the same of the set of transactions containing J. It follows that $\mathcal{C}^{J'}_{c(I)} \succeq \mathcal{C}^J_I$ and $f^J_{c(I)} = f^J_I > 0$. Then, if I is not closed, $\mathcal{C}^J_I$ is not maximal.

ii) J is maximal: if J is not maximal then consider its frequent superset $J' = J \cup \{a\}$ and consider $I' = I \cup a$. It is straightforward to see that $f^J_I = f^{J'}_I + f^{J'}_{I'}$ and that $\mathcal{C}^{J'}_I \succeq \mathcal{C}^J_I$ and $\mathcal{C}^{J'}_{I'} \succeq \mathcal{C}^J_I$. Therefore, since $f^J_I > 0$, at least one among $f^{J'}_I$ and $f^{J'}_{I'}$ must be not null. Then, if J is not maximal, $\mathcal{C}^J_I$ is not maximal as well.

Example 10 Consider the database $\mathcal{D}$ in Figure 1(a) and suppose $\sigma = 8$ and $k = 3$. The following is the set of maximal inference channels: $\mathcal{MC}h(3, \mathcal{C}l(\mathcal{D}, 8)) = \{\langle \mathcal{C}^{cde}_\emptyset, 1\rangle, \langle \mathcal{C}^{ab}_a, 1\rangle, \langle \mathcal{C}^{ae}_a, 1\rangle, \langle \mathcal{C}^{cde}_e, 1\rangle, \langle \mathcal{C}^{cde}_{de}, 1\rangle\}$.

The next Theorem shows how the support of any channel in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$ can be reconstructed from $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$.

Theorem 3 *Given $I \subseteq J \in 2^{\mathcal{I}}$, let M be any maximal itemset such that $M \supseteq J$. The following equation holds:*

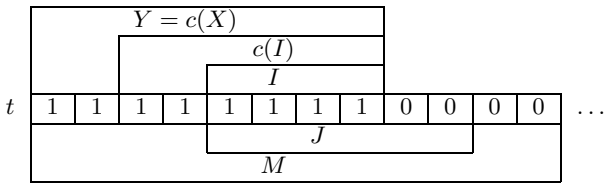
$$f^J_I(\mathcal{D}) = \sum_{c(X)} f^M_{c(X)}(\mathcal{D})$$

where $c(I) \subseteq c(X) \subseteq M$ and $c(X) \cap (J \setminus I) = \emptyset$.

Proof By definition, f_I^J is equal to the number of transactions t s.t. $\mathcal{C}_I^J(t)$, i.e., all items in I are set to 1 and all items in $J \setminus I$ are set to 0. We prove that, in the summation in the right-hand side of the equation: (i) each of such transactions is counted once, (ii) only once, and (iii) no other transaction is counted.

i) we must show that every such transaction t is considered at least once. This means that exists an itemset X such that $c(I) \subseteq c(X) \subseteq M$ and $c(X) \cap (J \setminus I) = \emptyset$.

Let Y denote the set of items in the M-projection of t that are set to 1. Y is necessarily a frequent closed itemset [2]. Let X be the itemset such that $c(X) = Y$. We have that $c(I) \subseteq c(X) \subseteq M$ and $c(X) \cap (J \setminus I) = \emptyset$.



ii) we must show that in the summation we count each such transaction t exactly once: this means that $\mathcal{C}^M_{c(X)}$ (with M fixed and varying $c(X)$) forms a partition of the set of transactions t such that $\mathcal{C}_I^J(t)$. In fact, given $c(X_1) \subset c(X_2)$, we have that each item in $c(X_2) \setminus c(X_1)$ is set to 0 in the transactions t such that $\mathcal{C}^M_{c(X_1)}(t)$, and set to 1 in the transactions t such that $\mathcal{C}^M_{c(X_2)}(t)$. As a consequence, the same transaction can not be considered by both $\mathcal{C}^M_{c(X_1)}$ and $\mathcal{C}^M_{c(X_2)}$.

iii) For a transaction t in order to be counted, it must exist an itemset X such that $c(X)$ holds in t. Since we require $c(I) \subseteq c(X) \subseteq M$ and $c(X) \cap (J \setminus I) = \emptyset$, no transaction having an item in I set to 0, or an item in $J \setminus I$ set to 1, can be counted.

Corollary 3 *For all* $\langle \mathcal{C}_I^J, f_I^J(\mathcal{D}) \rangle \in \mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$ *we have that, for any* $c(X)$ *s.t.* $c(I) \subseteq c(X) \subseteq M$ *and* $c(X) \cap (I \setminus J) = \emptyset$, $0 \leq f^M_{c(X)}(\mathcal{D}) < k$.

Proof Since $\mathcal{C}_I^J \preceq \mathcal{C}^M_{c(X)}$, and $f_I^J(\mathcal{D}) < k$, we conclude that $f^M_{c(X)}(\mathcal{D}) \leq f_I^J(\mathcal{D}) < k$. Moreover, for at least one $c(X)$ we have that $f^M_{c(X)}(\mathcal{D}) > 0$, otherwise we get a contradiction to Theorem 3.

From Corollary 3 we conclude that all the addends needed to compute $f_I^J(\mathcal{D})$ for an inference channel are either in $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ or are null. Therefore, as the set of all closed frequent itemsets $\mathcal{C}l(\mathcal{D}, \sigma)$ contains all the information of $\mathcal{F}(\mathcal{D}, \sigma)$ in a more compact representation, analogously the set $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ represents, without redundancy, all the information in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$.

In the database $\mathcal{D}$ of Figure 1(a), given $\sigma = 6$ and $k = 3$, we have that $|\mathcal{C}h(3, \mathcal{F}(\mathcal{D}, 6))| = 58$ while $|\mathcal{MC}h(3, \mathcal{C}l(\mathcal{D}, 6))| = 5$ (Figure 1(e)), a reduction of one order of magnitude. On the same database for $\sigma = 6$ and $k = 3$ (our running example, we got $|\mathcal{C}h(3, \mathcal{F}(\mathcal{D}, 6))| = 13$ while $|\mathcal{MC}h(3, \mathcal{C}l(\mathcal{D}, 6))| = 5$.

Such compression of the condensed representation is also confirmed by our experiments on various datasets from the FIMI repository [1], reported in Figure 3(a).

Another important benefit of our condensed representation, is that, in order to detect all inference channels holding in $\mathcal{F}(\mathcal{D}, \sigma)$, we can limit ourselves to retrieve only the inference channels in $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$, thus taking in input $\mathcal{C}l(\mathcal{D}, \sigma)$ instead of $\mathcal{F}(\mathcal{D}, \sigma)$ and thus performing a much smaller number of checks. Algorithm 2 exploits the anti-monotonicity of frequency (Proposition 1) and the property of maximal inference channels (Proposition 2) to compute $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ from $\mathcal{C}l(\mathcal{D}, \sigma)$. Thanks to these two properties, Algorithm 2 dramatically outperform the naive inference channel detector (Algorithm 1), and scales well even for very low support thresholds, as reported in Figure 3(b). Note that the run-time of both algorithms is independent from k, while it depends from the minimum support threshold.

Algorithm 2 Optimized Inference Channel Detector

Input: $\mathcal{C}l(\mathcal{D}, \sigma), k$
Output: $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$

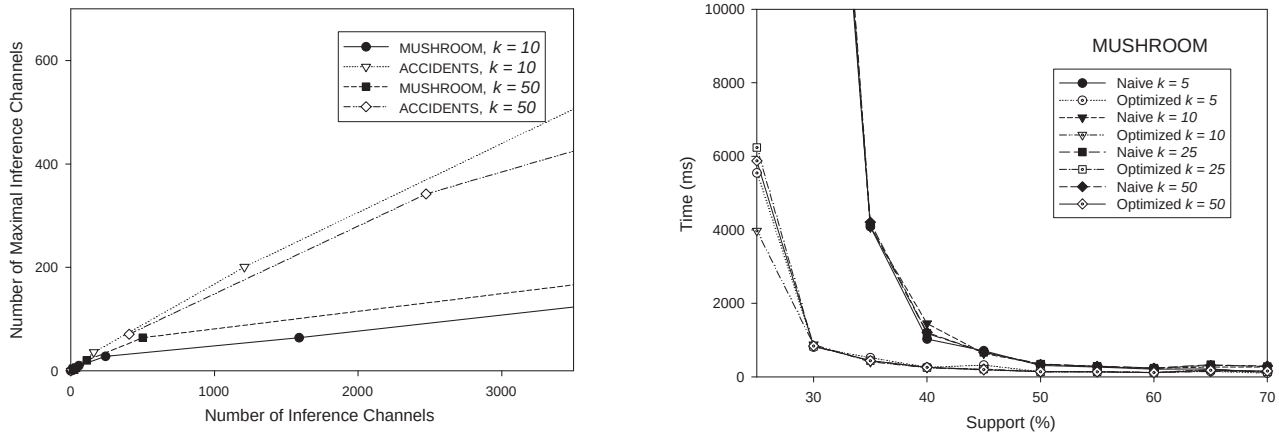
```
1: M = {I ∈ Cl(D,σ) | I is maximal};
2: MCh(k, Cl(D,σ)) = ∅;
3: for all J ∈ M do
4:    for all I ∈ Cl(D,σ) such that I ⊆ J do
5:       compute f_I^J;
6:       if 0 < f_I^J < k then
7:          insert ⟨C_I^J, f_I^J⟩ in MCh(k, Cl(D,σ));
```

6.1 Anonymity vs. Accuracy: Empirical Observations

Algorithm 2 represents an optimized way to identify all threats to anonymity. Its performance revealed adequate in all our empirical evaluations using various datasets from the FIMI repository [1]; in all such cases the time improvement from the Naïve (Algorithm 1) to the optimized algorithm is about one order of magnitude, as reported in Figure 3(b).

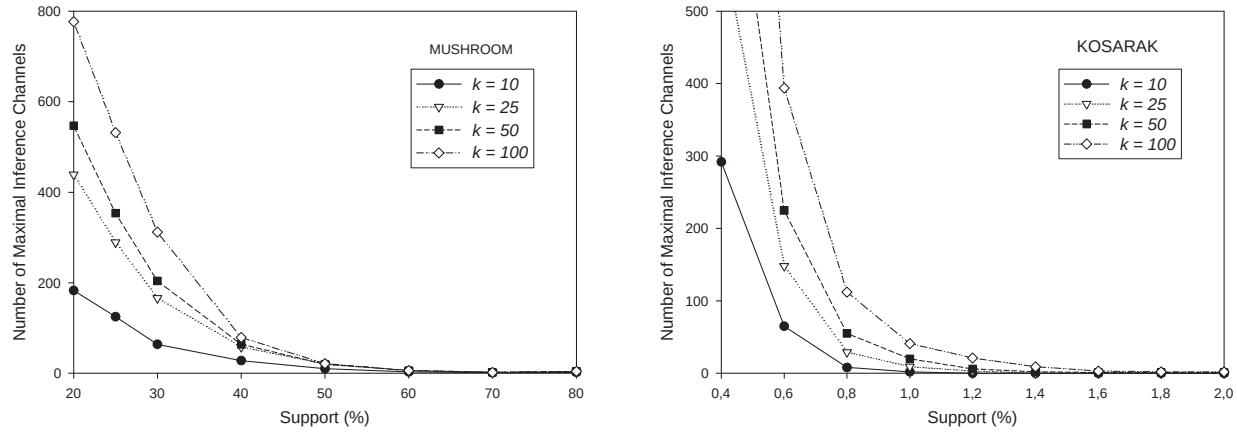
This level of efficiency allows an interactive-iterative use of the algorithm by the analyst, aimed at finding the best trade-off among privacy and accuracy of the collection of patterns. To be more precise, there is a conflict among keeping the support threshold as low as possible, in order to mine all interesting patterns, and avoiding the generation of anonymity threats.

[2] By contradiction, if Y is not closed, then there is at least one other item a which is always set to 1 in all transactions t such that $Y(t)$. Since Y is the positive part the M-projection of t, it follows that a is not in M, hence $M \cup \{a\}$ is frequent, hence M is not maximal.



(a) (b)

Fig. 3 Benefits of the condensed representation: size of the representations (a), and run time (b).



(a) (b)

Fig. 4 Experimental results on cardinality of $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ on two datasets.

The best solution to this problem is precisely to find out the minimum support threshold that generates a collection of patterns with no threats, thus avoiding to introduce the distortion needed to block the threats, and thus preserving accuracy.

The plots in Figure 4(a) and (b) illustrate this point: on the x-axis we report the minimum support threshold, on the y-axis we report the total number of threats (the cardinality of $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$), and the various curves indicate such number according to different values of the anonymity threshold k. In Figure 4(a) we report the plot for the MUSHROOM dataset (a dense one), while in Figure 4(b) we report the plot for the KOSARAK dataset which is sparse. In both cases, it is evident the value of the minimum support threshold that represents the best trade-off, for any given value of k. However, in certain cases, the best support threshold can still be too high to mine a sufficient quantity of interesting patterns. In such cases, the only option is to allow lower support thresholds and then to block the inference channels in the mining outcome. This problem will be the central topic of the following sections.

7 Blocking Inference Channels

In the previous sections we have studied Problem 1: how to detect inference channels in the output of a frequent itemset extraction. Obviously, a solution to this problem directly yields a method to formally prove that the disclosure of a given collection of frequent itemsets does not violate the anonymity constraint: it is sufficient to check that no inference channel exists for the given collection. In this case, the collection can be safely distributed even to malicious adversaries. On the contrary, if this is not the case, we can proceed in two ways:

– mine a new collection of frequent itemsets under different circumstances, e.g., higher minimum support threshold, to look for an admissible collection;
– transform (sanitize) the collection to remove the inference channels.

When it is needed to pursue the second alternative, we are faced with a second problem, which is addressed in this section.

Problem 2 (Inference Channels Sanitization) Given a collection of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$, and the set of all its inference channels $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, transform $\mathcal{F}(\mathcal{D}, \sigma)$ in a collection of frequent itemsets $\mathcal{O}^k$, which can be safely disclosed. $\mathcal{O}^k$ is the output of our problem, and it must satisfy the following conditions:

1. $\mathcal{C}h(k, \mathcal{O}^k) = \emptyset$;
2. $\exists \mathcal{D}' : \mathcal{O}^k = \mathcal{F}(\mathcal{D}', \sigma)$;
3. the effects of the transformation can be controlled by means of appropriate measures.

The first condition imposes that the sanitized output $\mathcal{O}^k$ does not contain any inference channel. Note that in the sanitization process, while introducing distortion to block the "real" inference channels holding in $\mathcal{F}(\mathcal{D}, \sigma)$, transforming it in $\mathcal{O}^k$, we could possibly create some new "fake" inference channels (not existing in the original database and thus not violating the anonymity of real individuals). We do not allow this possibility: although fake, such inference channels could be the starting point for a backward reasoning of a malicious adversary, in other terms, could open the door to inverse mining attacks.

The second condition constraints the output collection of itemsets to be *"realistic"*; i.e., to be compatible with at least a database. Note that, according to Definition 7, we can infer everything from $\mathcal{F}(\mathcal{D}, \sigma)$, if $\mathcal{F}(\mathcal{D}, \sigma)$ itself is not compatible with any database, or, in other words, if it contains contradictions (for instance, $sup\_tdb(X) < sup_{\mathcal{D}}(Y)$ with $X \subset Y$). Disclosing an output which is not compatible with any database could represent a threat. In fact, a malicious adversary could recognize that the set of pattern disclosed is not "real", and he could exploit this leak by reconstructing the missing patterns, starting from those ones present in the output. We call this kind of threat *inverse mining attacks*.

The inverse mining problem, i.e. given a set of σ-frequent itemsets reconstruct a database compatible with it, has been shown NP-complete [9]. However such a problem can be tackled by using some heuristics [48]. In this paper, in order to avoid this kind of attacks, we study how to sanitize a set of patterns in such a way that the output produced is always compatible with at least one database. Doing so, we avoid the adversary to distinguish an output which as been k-anonymized from a non k-anonymized one.

Finally, the third condition of Problem 2 requires to control the distortion effects of the transform of the original output by means of appropriate distortion measures (see Section 7).

Note that our output $\mathcal{O}^k$ always contains also the number of individuals in the database, or at least a sanitized version of such number. In fact, since $\mathcal{O}^k$ must be realistic, for the anti-monotonicity of frequency it must always contain the empty itemset with its support, which corresponds to the number of transactions in the database. More formally, we can say that $\langle \emptyset, sup_{\mathcal{D}'}(\emptyset) \rangle \in \mathcal{O}^k$ and $sup_{\mathcal{D}'}(\emptyset) = |\mathcal{D}'|$, where $\mathcal{D}'$ is a database compatible with $\mathcal{O}^k$. The relevance of this fact is twofold. On one hand the size of the database in analysis is a important information to disclose: for instance, in a medical domain, the number of patients on which a novel treatment has been experimented, and to which the set of extracted association rules refers, can not be kept secret. On the other hand, disclosing such number can help a malicious adversary to guess the support of non k-anonymous patterns.

One naïve attempt to solve Problem 2 is simply to eliminate from the output any pair of itemsets I, J such that $\mathcal{C}_I^J$ is an inference channel. Unfortunately, this kind of sanitization would produce an output which is, in general, not compatible with any database. As stated before, we do not admit this kind of solution, because disclosing an inconsistent output could open the door to inverse mining attacks.

In this Section, under the strong constraints imposed above, we develop two dual strategies to solve Problem 2. The first one blocks inference channels by increasing the support of some itemsets in $\mathcal{F}(\mathcal{D}, \sigma)$. Such support increasing is equivalent to adding transaction to the original database $\mathcal{D}$, and thus such strategy is named *additive sanitization*. The second strategy, named *suppressive sanitization*[3], blocks inference channels by decreasing the support of some itemsets (equivalent to suppress transactions from $\mathcal{D}$).

7.1 Avoiding Redundant Distortion

Using our condensed representation of maximal inference channels introduced in Section 6, we remove the redundancy existing in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, but this means to implicitly avoid redundant sanitization, and thus we dramatically reduce the distortion needed to block all the inference channels. In fact, to block an inference channel $\mathcal{C}_I^J \in \mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$ we have two main options:

– making the inference channel anonymous enough, i.e., forcing $f_I^J(\mathcal{D}) \geq k$;
– making the inference channel disappear, i.e., forcing $f_I^J(\mathcal{D}) = 0$.

The following two propositions show that, whichever option we choose, we can just block the channels in

[3] A preliminary version of this second strategy is in [7].

$\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$, obtaining to block all the inference channels in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$.

Proposition 3 *Given a database $\mathcal{D}$, consider a database $\mathcal{D}'$ s.t. $\forall \langle \mathcal{C}_H^L, f_H^L(\mathcal{D})\rangle \in \mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ it holds that $f_H^L(\mathcal{D}') \geq k$. Then from Proposition 1 it follows that $\forall \langle \mathcal{C}_I^J, f_I^J(\mathcal{D})\rangle \in \mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, $f_I^J(\mathcal{D}') \geq k$.*

Proposition 4 *Given a database $\mathcal{D}$, consider a database $\mathcal{D}'$ s.t. $\forall \langle \mathcal{C}_H^L, f_H^L(\mathcal{D})\rangle \in \mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ it holds that $f_H^L(\mathcal{D}') = 0$. Then from Proposition 3 it follows that $\forall \langle \mathcal{C}_I^J, f_I^J(\mathcal{D})\rangle \in \mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, $f_I^J(\mathcal{D}') = 0$.*

In the following we exploit these properties to reduce the distortion needed to sanitize our output.

7.2 Additive Sanitization

Given the set of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$, and the set of channels holding in it, the simplest solution to our problem is the following: for each $\mathcal{C}_I^J \in \mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, increase the support of the itemset I by k to force $f_I^J > k$. In order to maintain database-compatibility, the support of all subsets of I is increased accordingly. This is equivalent to add k transactions $t = I$ to the original database $\mathcal{D}$. Clearly, we are not really adding transactions: this is just to highlight that the transformed set of frequent itemsets maintains database-compatibility. Moreover, it will contain exactly the same itemsets, but with some supports increased.

One could observe that it is not strictly necessary to increase the support of I by k, but it is sufficient to increase it by $k - f_I^J$ to block the inference channel (making it reach the anonymity threshold k). This solution has two drawbacks. First, it creates new fake inference channels (which are not allowed by our problem definition). Second, it would produce an output on which a malicious adversary, could compute and find out a lot of $f_I^J = k$. This could suggest to the adversary: (*i*) that a k-anonymization has taken place, (*ii*) the exact value of k, and (*iii*) the set of possible patterns which have been distorted. Increasing the support of I by k we avoid all these problems.

The third requirement of our problem is to minimize the distortion introduced during the anonymization process. Since in our sanitization approach the idea is to increment supports of itemsets and their subsets (by virtually adding transactions in the original dataset), minimizing distortion means reducing as much as possible the increments of supports (i.e., number of transactions virtually added). To do this, we exploit the anti-monotonicity property of patterns, and the condensed representation introduced in Section 6. Therefore we will actually feed the sanitization algorithm with $\mathcal{C}l(\mathcal{D}, \sigma)$ and $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$, instead of $\mathcal{F}(\mathcal{D}, \sigma)$ and $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$. But we can do something more. In fact, when adopting an additive strategy, some redundancy can be found and avoided, even in $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$, as described in the following.

Example 11 Consider the two maximal inference channels $\langle \mathcal{C}_a^{ab}, 1\rangle, \langle \mathcal{C}_a^{ae}, 1\rangle \in \mathcal{MC}h(3, \mathcal{C}l(\mathcal{D}, 8))$ (Example 10). According to the additive strategy we should virtually add 3 transactions $'a'$ for the first channel, and other 3 transactions $'a'$ for the second channel. Obviously we can just add 3 transactions to block both channels.

Definition 17 (Maximal Channels Merging) *Two inference channels $\mathcal{C}_I^J$ and $\mathcal{C}_H^L$ can be merged if $I \subseteq H$ and $H \cap (J \setminus I) = \emptyset$; or viceversa, $H \subseteq I$ and $I \cap (L \setminus H) = \emptyset$. Their merge is $\mathcal{C}_I^J \bowtie \mathcal{C}_H^L = \mathcal{C}_{I \cup H}^{J \cup L}$.*

Example 12 Mining the MUSHROOM dataset (8124 transactions) at 60% minimum support level (absolute support $\sigma = 4874$) we obtain 52 frequent itemsets (counting also the empty set). With $k = 10$ the detection algorithm can find 20 inference channels. Among them, only 3 are maximal:

$$\mathcal{C}_{\{85,34\}}^{\{85,86,39,34\}}, \mathcal{C}_{\{85,34\}}^{\{85,59,86,34\}}, \mathcal{C}_{\{85,90,34\}}^{\{85,86,90,36,34\}}$$

In this case all of them can be merged into a unique inference channel: $\mathcal{C}_{\{85,34,90\}}^{\{85,59,86,39,34,90,36\}}$. Therefore, increasing the support of the itemset $\{85, 34, 90\}$ and of all its subsets by 10, we remove all the 20 inference channels holding in the output of frequent itemset mining on the MUSHROOM dataset at 60% of support.

Algorithm 3 implements the additive sanitization just described: it takes in input $\mathcal{C}l(\mathcal{D}, \sigma)$ and $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$, and returns $\mathcal{O}^k$, which in this case is the sanitized version of $\mathcal{C}l(\mathcal{D}, \sigma)$. Obviously, if we want to output a sanitized version of $\mathcal{F}(\mathcal{D}, \sigma)$ instead of $\mathcal{C}l(\mathcal{D}, \sigma)$, we can simply reconstruct it from the sanitized version of $\mathcal{C}l(\mathcal{D}, \sigma)$ (recall that $\mathcal{F}(\mathcal{D}, \sigma)$ and $\mathcal{C}l(\mathcal{D}, \sigma)$ contain exactly the same information). The algorithm is composed

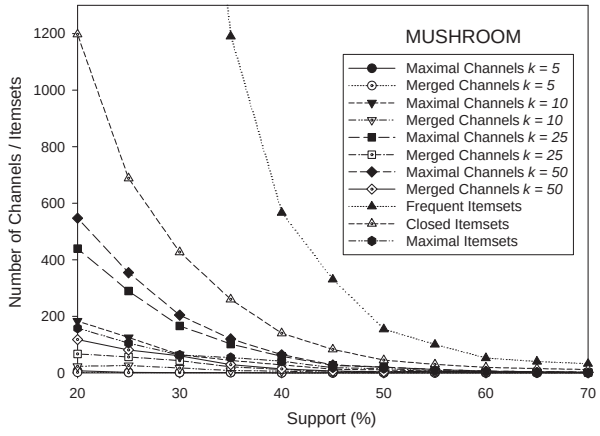


Fig. 5 The benefits of the channels merging operation.

Algorithm 3 Additive Sanitization

Input: $\mathcal{C}l(\mathcal{D},\sigma), \mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$
Output: $\mathcal{O}^k$

1: *//Merging phase*
2: $S \leftarrow \emptyset$;
3: **for all** $\langle \mathcal{C}_I^J, f_I^J\rangle \in \mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$ **do**
4: **if** $\exists \mathcal{C}_{I'}^{J'} \in S$ s.t. $\mathcal{C}_I^J$ and $\mathcal{C}_{I'}^{J'}$ can be merged **then**
5: $S \leftarrow S \setminus \{\mathcal{C}_{I'}^{J'}\}$;
6: $S \leftarrow S \cup (\mathcal{C}_I^J \bowtie \mathcal{C}_{I'}^{J'})$;
7: **else**
8: $S \leftarrow S \cup \{\mathcal{C}_I^J\}$;
9: *//Distortion phase*
10: **for all** $\langle I, sup_{\mathcal{D}}(I)\rangle \in \mathcal{C}l(\mathcal{D},\sigma)$ **do**
11: **for all** $\mathcal{C}_{I'}^{J'} \in S$ s.t. $I \subseteq I'$ **do**
12: $sup_{\mathcal{D}}(I) \leftarrow sup_{\mathcal{D}}(I) + k$;
13: $\mathcal{O}^k \leftarrow \mathcal{C}l(\mathcal{D},\sigma)$

Algorithm 4 Suppressive Sanitization

Input: $\mathcal{C}l(\mathcal{D},\sigma), \mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma)), \mathcal{D}$
Output: $\mathcal{O}^k$

1: **while** $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma)) \neq \emptyset$ **do**
2: *//Scan the database*
3: **for all** $t \in \mathcal{D}$ **do**
4: **if** $\exists \langle \mathcal{C}_I^J, f_I^J\rangle \in \mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$ s.t. $I \subseteq t$ and $(J \setminus I) \cap t = \emptyset$ **then**
5: *//Transaction suppression*
6: **for all** $\langle X, sup_{\mathcal{D}}(X)\rangle \in \mathcal{C}l(\mathcal{D},\sigma)$ s.t. $X \subseteq t$ **do**
7: $sup_{\mathcal{D}}(X) \leftarrow sup_{\mathcal{D}}(X) - 1$;
8: *//Compact* $\mathcal{C}l(\mathcal{D},\sigma)$
9: **for all** $\langle X, s\rangle \in \mathcal{C}l(\mathcal{D},\sigma)$ **do**
10: **if** $\exists \langle Y, s\rangle \in \mathcal{C}l(\mathcal{D},\sigma)$ s.t. $Y \supset X$ **or** $s < \sigma$ **then**
11: $\mathcal{C}l(\mathcal{D},\sigma) \leftarrow \mathcal{C}l(\mathcal{D},\sigma) \setminus \langle X, s\rangle$;
12: **detect** $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$ **in** $\mathcal{C}l(\mathcal{D},\sigma)$;
13: $\mathcal{O}^k \leftarrow \mathcal{C}l(\mathcal{D},\sigma)$;

by two phases: during the first phase all maximal channels are merged as much as possible, according to Definition 17; then the resulting set of merged channels is used in the second phase to select the itemsets whose support must be increased.

Example 13 Consider $\mathcal{MC}h(3,\mathcal{C}l(\mathcal{D},8))$ in Example 10. The 5 maximal inference channels can be merged giving 3 channels: $\mathcal{C}_a^{abcde}$,$\mathcal{C}_e^{cde}$ and $\mathcal{C}_{de}^{cde}$. Therefore, according to the additive strategy, we sanitize $\mathcal{C}l(\mathcal{D},8)$ (Example 9) virtually adding 3 transactions $'a'$, 3 transactions $'e'$, and 3 transactions $'de'$. The resulting set is $\mathcal{O}^3 = \{\langle\emptyset,21\rangle, \langle a,12\rangle, \langle e,17\rangle, \langle ab,8\rangle, \langle ae,8\rangle, \langle de,13\rangle, \langle cde,9\rangle\}$.

7.3 Suppressive Sanitization

The basic idea of Suppressive Sanitization is to hide inference channels, sending their support to 0: this can be done by removing transactions t s.t. $I \subseteq t \wedge (J \setminus I) \cap t = \emptyset$. Unfortunately, we can not simulate such suppression of transactions simply by decreasing the support of the itemset I by f_I^J for each $\mathcal{C}_I^J \in \mathcal{C}h(k,\mathcal{F}(\mathcal{D},\sigma))$, since we would lose database-compatibility due to the other items appearing in the dangerous transactions. Consider for instance a transaction $I \cup \{x,y,z\}$: removing it we reduce the support of I, but as uncontrolled side effect, we also reduce the support of the itemset $I \cup \{x\}$. Therefore, in order to maintain database-compatibility, we must take into account these other items carefully. On way of achieving this is to really access the database, suppress the dangerous transactions, and reduce the support of all itemsets contained in the suppressed transactions accordingly. But this is not enough. In fact, while in the additive strategy described before, is sufficient to raise the supports by k to be sure that no novel (fake) inference channel is created, the case is more subtle with the suppressive strategy. The unique solution here is to perform again the detection algorithm on the transformed database, and if necessary, to block the novel inference channels found. Obviously, this process can make some frequent itemsets become infrequent. This is a major drawback of the suppressive strategy w.r.t. the additive strategy which has the nice feature of maintaining the same set of frequent itemsets (even if with larger supports).

Algorithm 4 implements the suppressive sanitization which access the database $\mathcal{D}$ on the basis of the information in $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$, and adjust $\mathcal{C}l(\mathcal{D},\sigma)$ with the information found in $\mathcal{D}$. As for the additive strategy, the following pseudo-code outputs a sanitized version of $\mathcal{C}l(\mathcal{D},\sigma)$ but nothing prevents us from disclosing a sanitized version of $\mathcal{F}(\mathcal{D},\sigma)$.

Example 14 Consider $\mathcal{C}l(\mathcal{D},8)$ and $\mathcal{MC}h(3,\mathcal{C}l(\mathcal{D},8))$ in Fig.1(d) and (e). Suppressive Sanitization removes transactions which are directly involved in maximal inference channels. In our example we got 5 maximal inference channels $\langle\mathcal{C}_\emptyset^{cde},1\rangle, \langle\mathcal{C}_a^{ab},1\rangle, \langle\mathcal{C}_a^{ae},1\rangle, \langle\mathcal{C}_e^{cde},1\rangle$ and $\langle\mathcal{C}_{de}^{cde},1\rangle$ corresponding to transactions t_{12}, t_8, t_{12}, t_8 and t_7 respectively. The suppression of these 3 transactions reduces the support of some closed itemsets. In particular, at the end of the suppression phase (line 8 of Algorithm 4) we got that $\mathcal{C}l(\mathcal{D},8) = \{\langle\emptyset,9\rangle, \langle a,6\rangle, \langle e,9\rangle, \langle ab,6\rangle, \langle ae,6\rangle, \langle de,9\rangle, \langle cde,9\rangle\}$. Compacting $\mathcal{C}l(\mathcal{D},8)$ means to remove from it itemsets which, due to the transactions suppression, are no longer frequent or no longer closed (lines 9 – 12), i.e., $\mathcal{C}l(\mathcal{D},8) = \{\langle cde,9\rangle\}$. At this point Algorithm 4 invokes the optimized detection algorithm (Algorithm 2) to find out the maximal channels in the new $\mathcal{C}l(\mathcal{D},8)$, and if necessary, starts a new suppression phase. In our example this is not the case, since we have no more inference channels. Therefore the resulting output, which can be safely disclosed, is given by the itemset $'cde'$ and all its subsets, all having the same support 9.

8 Experimental Analysis

In this section we report the results of the experimental analysis that we have conducted in order to:

- assess the distortion introduced by our sanitization strategies;
- measure time needed by our sanitization framework (inference channels detection plus blocking);
- compare empirically the differences between k-anonymizing the data and k-anonymizing the patterns.

8.1 Distortion Empirical Evaluation

In the following we report the distortion empirical evaluation we have conducted on three different datasets from the FIMI repository [1], with various σ and k thresholds, recording the following four measures of distortion:

1. Absolute number of transaction virtually added (by the additive strategy) or suppressed (by the suppressive strategy).
2. The fraction of itemsets in $\mathcal{F}(\mathcal{D},\sigma)$ which have their support changed in $\mathcal{O}^k$:
$$\frac{|\{\langle I, sup_{\mathcal{D}}(I)\rangle \in \mathcal{F}(\mathcal{D},\sigma) \,:\, sup_{\mathcal{O}^k}(I) \neq sup_{\mathcal{D}}(I)\}|}{|\mathcal{F}(\mathcal{D},\sigma)|}$$
where $sup_{\mathcal{O}^k}(I) = s$ if $\langle I, s\rangle \in \mathcal{O}^k$; 0 otherwise.
3. The average distortion w.r.t. the original support of itemsets:
$$\frac{1}{|\mathcal{F}(\mathcal{D},\sigma)|} \sum_{I\in\mathcal{F}(\mathcal{D},\sigma)} \frac{|sup_{\mathcal{O}^k}(I) - sup_{\mathcal{D}}(I)|}{sup_{\mathcal{D}}(I)}$$
4. The worst-case distortion w.r.t. the original support of itemsets:
$$\max_{I\in\mathcal{F}(\mathcal{D},\sigma)} \left\{ \frac{|sup_{\mathcal{O}^k}(I) - sup_{\mathcal{D}}(I)|}{sup_{\mathcal{D}}(I)} \right\}$$

The first row of plots in Figure 6 reports the first measure of distortion. Since the size of $\mathcal{MCh}(k, \mathcal{C}l(\mathcal{D},\sigma))$ grows for larger k and for smaller σ, the number of transactions added (by the additive strategy) or suppressed (by the suppressive strategy) behaves accordingly. However, in few cases, the additive sanitization exhibits a non-monotonic behavior (e.g., the local maximum for $\sigma = 55\%$ and $k = 50$ in Figure 6(a)): this is due to the non-monotonicity of the merge operation (Definition 17 and lines 4–6 of Algorithm 1). In general, for reasonable values of σ, the number of transactions involved in the sanitization is always acceptable (the total number of transactions in MUSHROOM, KOSARAK and RETAIL are respectively 8124, 990002 and 88162).

The percentage of itemsets whose support has been distorted is reported in the second row of plots of Figure 6. Additive sanitization always behaves better than suppressive: in fact, for each inference channel $\mathcal{C}_I^J$, additive only changes the support of I and its subsets, while suppressive changes the support of other itemsets contained in the dangerous transaction. Non-monotonicity (evident in Figure 6(d), but also present in Figure 6(e) and (f)) is due to the fact that lowering the value of σ some infrequent itemsets became frequent but they do not contribute to new inference channels.

The third row of plots of Figure 6 reports the average distortion introduced. Notice that this measure drastically penalizes the suppression approach that possibly makes some itemsets become infrequent: for this itemsets we count a maximum distortion of 1 ($sup_{\mathcal{O}^k}(I) = 0$).

It is worth noting that, while the number of itemsets distorted is usually very large, the average distortion on itemsets is very low: this means that quite all itemsets are touched by the sanitization, but their supports are changed just a little. Consider, for instance, the experiment on KOSARAK (Figure 6(e) and (h)) with $k = 50$ and $\sigma = 0.6\%$: the additive strategy changes the support of the 40% of itemsets in the output, but the average distortion is 0.4%; the suppressive sanitization changes the 100% of itemsets but the average distortion is 2.1%.

Finally, the fourth row of plots of Figure 6 reports the worst-case distortion, i.e., the maximum distortion of the support of an itemset in $\mathcal{F}(\mathcal{D},\sigma)$. Note that this measure is 100% whenever the suppressive strategy makes at least one frequent itemset infrequent. While for the additive strategy the itemset maximally distorted is always the empty itemset, i.e. the bottom of the itemset lattice, which, as discussed in Section 7, is always present in our output $\mathcal{O}^k$, and which represents the number of transactions in the database. Since the empty itemset is subset of any set, every single transaction virtually added by the additive strategy increases its support.

Also w.r.t. this measure the additive strategy outperforms the suppressive strategy, providing a reasonable distortion. However, note that when the suppressive strategy does not make any frequent itemset become infrequent (for instance in Figure 6(j) for $k = 5$ and a support of 25%) the worst-case distortion is less than the distortion introduced by the additive strategy.

A major drawback of the additive approach is that the transactions virtually added are fake. The suppressive approach, on the other hand, induces a stronger distortion, but such distortion is only due to the hiding of some nuggets of information containing threats to anonymity. Therefore, in many practical situations, the suppressive strategy could be a more reasonable choice, guaranteeing a more meaningful output.

8.2 Run-time Analysis

Although time performance is not a central issue in our proposal, it is worth noting that the execution of both strategies was always very fast. Figure 7 reports the time

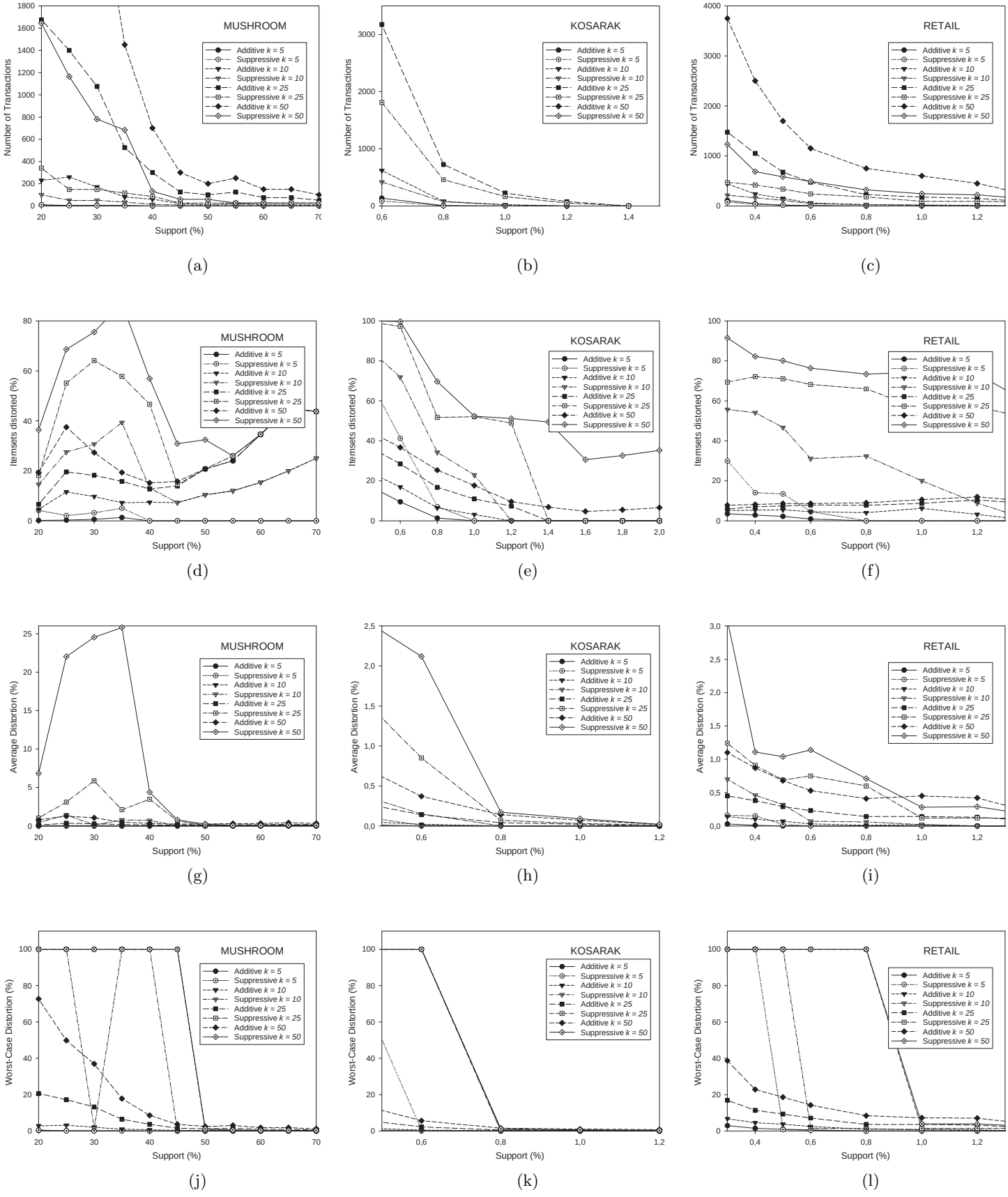
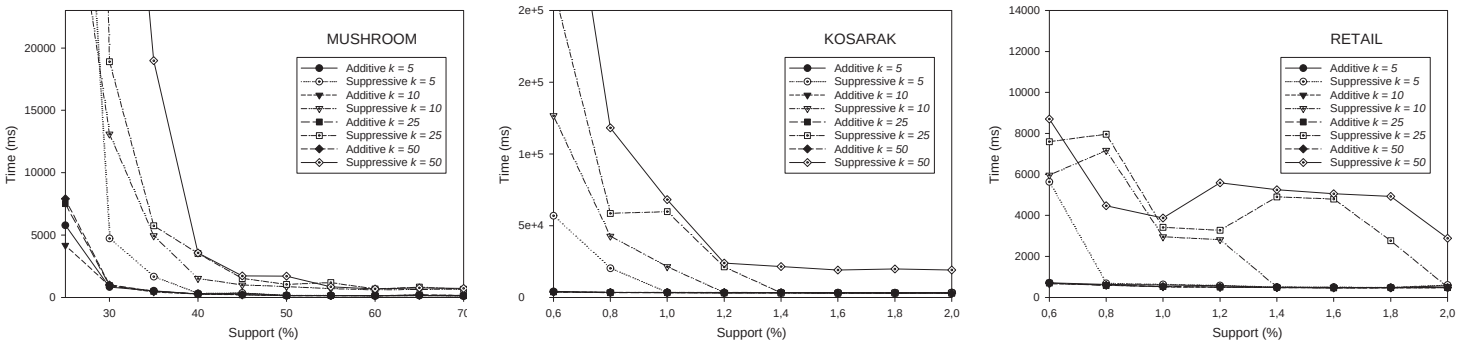


Fig. 6 Distortion empirical evaluation.



(a) (b) (c)

Fig. 7 Experiments on run-time (both detection and sanitization).

needed to detect and sanitize inference channels, i.e., Algorithm 2 followed by Algorithm 3 (Additive) or 4 (Suppressive). Note that (quite obviously) run-time generally grows for growing k and shrinking σ. However, in Figure 7(c) the suppressive strategy has an evident non monotonic behaviour w.r.t. σ: this is due to the possibly different number of sanitization cycles required by the suppressive strategy for different σ.

8.3 Anonymizing Data Vs. Anonymizing Patterns

In the following we resume the thread of the discourse we began in Section 3. As stated before, a trivial solution to our pattern sanitization problem could be to first k-anonymize the source database $\mathcal{D}$ using some well known techniques (for instance the DATAFLY algorithm [45], or the INCOGNITO algorithm [33]), obtaining a k-anonymized database $\mathcal{D}'$, and then to mine the frequent itemsets from $\mathcal{D}'$, using for instance APRIORI. In fact, by mining a k-anonymized database, the set of frequent itemsets that we obtain is clearly safe, in the sense that it does not contain inference channels. The situation is described by Figure 8. We claim that, if the goal is to disclose the result of the mining and not the source data, such solution is overkilling.

Example 15 Consider again our running example: the database $\mathcal{D}$ in Figure 1(a) with $\sigma = 8$ and $k = 3$. Suppose that all items in $\mathcal{I}$ are quasi-identifier. If we try to k-anonymize the data, the sanitization process would involve also the tuples not covering any patterns. In particular, also the three singleton items which are infrequent (i.e., f, g and h) would be involved in the sanitization. Consider for instance the tuples t_9, t_{10} and t_{11}: if we consider their projection on the frequent items (i.e., a, b, c, d and e) these tuples are identical. In other words, the pattern $p = \neg a \wedge \neg b \wedge c \wedge d \wedge e$, describing the three transaction above, is k-anonymous having $sup_{\mathcal{D}}(p) = 3 = k$. Therefore, if we sanitize the patterns such tuples would not be involved. On the contrary, if we first try to sanitize the data, we must somehow sanitize these three tuples due to differences in f, g and h.

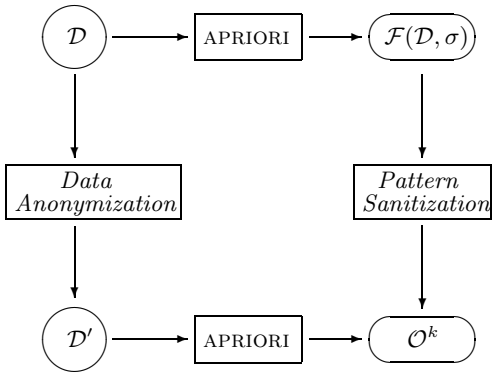


Fig. 8 Two alternatives paths from a source database $\mathcal{D}$ to an output set of itemsets, $\mathcal{O}^k$, which can be safely disclosed. One path first k-anonymizes the data, obtaining a k-anonymized database $\mathcal{D}'$ from which $\mathcal{O}^k$ is mined by means of APRIORI. The second path first mines the set of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$ from the original database $\mathcal{D}$, and then k-anonymizes the result of the mining using, for instance our Algorithm 3 or 4.

It is natural to argue that the traditional k-anonymization on data is developed for multivalued attributes (not for binary), and it is particularly effective in the cases in which there are only few quasi-identifiers among the whole set of attributes. Following this consideration, we further analyze the issue (graphically described in Figure 8) by means of an experimentation on the ADULT dataset from the UCI repository.

The following comparison is qualitative rather than quantitative. This is due to the fact that data anonymization algorithms perform attributes generalization, and thus change the vocabulary of items, while our method does not. This makes impossible to quan-

Itemset	Support	ADD	SUP
{Native-Country = United-States, Capital-Loss = 0, WorkClass = Private}	19237	19237	19237
{Capital-Loss = 0, Sex = Male}	19290	19290	19290
{ Race = White, Capital-Gain = 0, Income = Low}	18275	18275	18249
{Race = White, Capital-Loss = 0, Income = Low}	18489	18489	18489
{Sex = Male, Capital-Gain = 0}	18403	18403	18263
{Race = White, Capital-Loss = 0, WorkClass = Private}	18273	18273	18273
{Native-Country = United-States, Sex = Male}	18572	18572	18512
{Race = White, Native-Country = United-States, Capital-Loss = 0, Capital-Gain = 0}	20836	20836	20836
{Native-Country = United-States, WorkClass = Private, Capital-Gain = 0}	18558	18558	18493
{Hours-per-Week = (20,40]}	18577	18577	18446
{Capital-Loss = 0, WorkClass = Private, Capital-Gain = 0}	19623	19623	19573
{Native-Country = United-States, Capital-Loss = 0, Capital-Gain = 0, Income = Low}	19009	19009	19009

Table 2 The set of 12 maximal frequent itemset extracted from the database $\mathcal{D}$ with $\sigma = 18100$, and their supports. The third and the fourth columns report their supports in the output $\mathcal{O}^k$ produced by our additive strategy (ADD), and suppressive strategy (SUP).

I	J	f_I^J
{ }	{ Native-Country = United-States, Capital-Loss = 0, Workclass = Private}	36
{ }	{ Race = White, Capital-Loss = 0, WorkClass = Private}	49
{ Capital-Gain = 0 }	{ Race = White, Native-Country = United-States, Capital-Loss = 0, Capital-Gain = 0}	48
{ Capital-Gain = 0 }	{ Native-Country = United-States, Capital-Loss = 0, Capital-Gain = 0, Income = Low}	48

Table 3 The set of maximal inference channels $\mathcal{MC}h(50, \mathcal{C}l(\mathcal{D}, 18100))$.

titatively compare the output of the two strategies, for instance, by means of the measures of distortion introduced in Section 8.1.

As usually done when preprocessing data for association rule or frequent itemset mining, we have discretized the continuous-valued attributes. It is worth noting that a multivalued attribute can be transformed into a set of binary ones: one binary attribute for each possible value of the multivalued attribute. This simple transformation allows to perform frequent itemsets (and thus association rule) mining on real-world categorical data. Applying it to the ADULT dataset we obtained 341 items (or attribute-value pairs). Moreover, in the preprocessing phase, we have removed tuples containing missing values, obtaining a dataset with 30162 tuples. Let us denote this preprocessed dataset $\mathcal{D}$. In the following we describe the results of our experiments on $\mathcal{D}$ with a frequency threshold $\sigma = 18100$ (i.e., $\approx 60\%$ of the total number of tuples in $\mathcal{D}$), and an anonymity threshold $k = 50$.

As a first step we have mined frequent itemsets from $\mathcal{D}$: the result is useful for comparing the distortions introduced by the two different philosophies. We obtained 41 frequent itemsets, which are also closed itemsets ($\mathcal{F}(\mathcal{D}, 18100) = \mathcal{C}l(\mathcal{D}, 18100)$). Among these 12 are maximal, and they are reported in Table 2.

We have then applied the Optimized Inference Channel Detector (Algorithm 2) to $\mathcal{C}l(\mathcal{D}, 18100)$ obtaining four maximal inference channels, which are reported in Table 3. For instance consider the first inference channel: it indicates that in our dataset there are only 36 individuals such that: Native-Country $\neq$ United-States, Capital-Loss $\neq 0$, Workclass $\neq$ Private. The fact that our detector has found some inference channels, means that the output $\mathcal{F}(\mathcal{D}, 18100)$ can not be safely disclosed if we want an anonymity level larger than $k = 50$.

We then proceed with our pattern sanitization. If we apply the Additive strategy (Algorithm 3) we must try to merge (see Definition 17) the four maximal inference channels found by the channels detector. The merging operation returns a unique inference channel $\mathcal{C}_I^J$, where $I =$ { Capital-Gain = 0 } and $J =$ { Race = White, Native-Country = United-States, WorkClass = Private, Capital-Gain = 0, Capital-Loss = 0, Incomes = Low}.

Therefore, according to the additive sanitization, we increase the support of the itemset {Capital-Gain = 0} by k, which in this case is 50.

The Suppressive strategy, instead, virtually removed 195 transactions, but however no frequent itemset became infrequent. In conclusion, both the additive and the suppressive strategy maintain the original set of frequent itemsets and they only slightly change the support of a few itemsets. This is not the case with the other philosophy (first k-anonymize the data and then mine the frequent itemsets), as described in the following.

We implemented the well known DATAFLY algorithm [45] which k-anonymizes a database by generalizing attribute which are quasi-identifier, and by removing some tuples when needed. DATAFLY is a heuristic-driven greedy algorithm which does not guarantee minimality of the distortion introduced. We have also experimented using the INCOGNITO algorithm [33], kindly provided by

Itemset	Support
{Age = ∗, Native-Country = ∗, Race = ∗, Hours-per-Week = (20,40]}	18577
{Age = ∗, Native-Country = ∗, Race = ∗, Sex = Male, Capital-Gain = 0}	18403
{Age = ∗, Native-Country = ∗, Race = ∗, Sex = Male, Capital-Loss = 0}	19290
{Age = ∗, Native-Country = ∗, Race = ∗, WorkClass = Private, Capital-Gain = 0, Capital-Loss = 0}	19623
{Age = ∗, Native-Country = ∗, Race = ∗, Income = Low, Capital-Gain = 0, Capital-Loss = 0}	21021

Table 4 The set of maximal frequent itemsets obtained by mining $\mathcal{D}'$ (the database k-anonymized by DATAFLY and by INCOGNITO). Note that the ∗ symbol stands for *any*, i.e., maximal generalization.

Itemset	Support
{Age = ∗, Race = ∗, Income = ∗, Hours-per-Week = (20,40]}	18577
{Age = ∗, Race = ∗, Income = ∗, Sex = Male, Capital-Gain = 0}	18403
{Age = ∗, Race = ∗, Income = ∗, Sex = Male, Native-Country = North-America}	18652
{Age = ∗, Race = ∗, Income = ∗, Sex = Male, Capital-Loss = 0}	19290
{Age = ∗, Race = ∗, Income = ∗, WorkClass = Private, Capital-Gain = 0, Native-Country = North-America}	18642
{Age = ∗, Race = ∗, Income = ∗, WorkClass = Private, Capital-Gain = 0, Capital-Loss = 0}	19623
{Age = ∗, Race = ∗, Income = ∗, WorkClass = Private, Native-Country = North-America, Capital-Loss = 0}	19322
{Age = ∗, Race = ∗, Income = ∗, Capital-Gain = 0, Native-Country = North-America, Capital-Loss = 0}	23921

Table 5 The set of maximal frequent itemsets obtained by mining $\mathcal{D}'$ (another solution provided by INCOGNITO).

the authors, which can be considered the-state-of-the-art of data anonymization. In fact INCOGNITO produces *minimal* full-domain generalizations (i.e., it performs the minimal number of generalization steps, therefore maintaining as much information as possible while guaranteeing k-anonymity). Note that INCOGNITO outputs all possible minimal generalizations, and that, although it allows tuples suppression, in our experimentation the parameter defining the maximum number of allowed suppressions has been kept at the default value 0. Both DATAFLY and INCOGNITO take in input the database $\mathcal{D}$, the set of quasi identifiers $QI \subseteq \mathcal{I}$, and for each attribute in QI a *domain generalization hierarchy*.

In the pattern sanitization framework we consider every attribute as a possible source of threat. The same approach is unapplicable when using DATAFLY or INCOGNITO(i.e., considering every attribute as a quasi-identifier): in our case, we lost all the information contained in the database, since all attributes were maximally generalized. We therefore drastically reduced the number of quasi-identifier attributes to 5 (Age, Race, Sex, Native-Country and Income). We had that only 3170 tuples out of 30162 were non k-anonymous, leading to attributes generalization (no tuple was suppressed by DATAFLY[4]). After attribute generalization, both by means of DATAFLY and INCOGNITO, we have mined the resulting anonymized database, $\mathcal{D}'$ by means of the APRIORI algorithm.

In Table 4 are reported the 5 maximal frequent itemsets obtained by mining the $\mathcal{D}'$ produced by DATAFLY. Note that the same generalization, and thus the same anonymized database and the consequent maximal frequent itemsets, is produced by INCOGNITO as one of the minimal generalizations. Recall that INCOGNITO outputs all possible minimal generalizations. In Table 5 we report the maximal frequent itemsets obtained by mining the anonymized database given by the first minimal generalization produced by INCOGNITO.

By comparing Table 2 with Table 4 and 5, we can readily observe the huge loss of information obtained by mining a k-anonymized database. The itemsets in tables 4 and 5 are extremely generic, if compared with the 12 patterns of Table 2, due to unnecessary generalization in the source data. We have also conducted the same kind of analysis on other datasets (e.g., the CENSUS-INCOME dataset from the UCI repository), obtaining the same kind of unnecessary generalizations w.r.t. the sanitization focussed on patterns.

Another important difference is given by the computation time requirements. In fact, while sanitizing the patterns always requires few seconds, computing a minimal k-anonymization of the database requires instead a time ranging from tens of seconds (ADULT dataset using 5 quasi-identifiers) to several hours (CENSUS-INCOME dataset using 8 quasi-identifiers). All known minimal database anonymization algorithms require exponential time (in the number of quasi-identifiers), since the decisional version of the problem has been proved to be NP-Hard, while existing bound for approximated solutions are currently far from optimality.

In conclusion, if the goal is not to disclose data, but the result of mining, sanitizing the mined patterns yields better quality results than mining anonymized source data: the pattern sanitization process focuses only on the portions of data pertinent to the harmful patterns, instead of the whole source dataset.

[4] Note that this choice is taken by the algorithm.

9 Conclusion and Future Work

We introduced in this paper the notion of k-anonymous patterns. Such notion serves as a basis for a formal account of the intuition that a collection of patterns, obtained by data mining techniques and made available to the public, should not offer any possibilities to violate the privacy of the individuals whose data are stored in the source database. To the above aim, we formalized the threats to anonymity by means of inference channel through frequent itemsets, and provided practical algorithms to (i) check whether or not a collection of mined patterns exhibits threats, and (ii) eliminate such threats, if existing, by means of a controlled distortion of the pattern collection. The overall framework provides comprehensive means to reason about the desired tradeoff between anonymity and quality of the collection of patterns, as well as the distortion level needed to block the threatening inference channels. Concerning the blocking strategies, it is natural to confront our method with the traditional sanitization approach where the source dataset is transformed in such a way that the forbidden patterns are not extracted any longer. We, on the contrary, prefer to transform the patterns themselves, rather than the source data. In our opinion, this is preferable for two orders of reasons. First, in certain cases the input data cannot be accessed more than once: a situation that occurs increasingly often as data streams become a typical source for data mining. In this case there is no room for repeated data pre-processing, but only for pattern post-processing. Second, as a general fact the distortion of the mined patterns yields better quality results than repeating the mining task after the distortion of the source data, as thoroughly discussed in Section 8.3.

The first objective of our on-going investigation is the characterization of *border crossing inference channels*; i.e., inference channels also made of itemsets which are not frequent. Recall that in Section 5 we defined a subset of all possible patterns, namely σ-vulnerable patterns (Definition 10), and proved that our detection and sanitization methodology produces an output which is k-anonymous w.r.t. σ-vulnerable patterns.

Example 16 Consider again our running example of Figure 1 with $\sigma = 8$ and $k = 3$. As shown in Example 13, the set of closed frequent itemsets resulting from the additive sanitization is $\mathcal{O}^3 = \{\langle\emptyset, 21\rangle, \langle a, 12\rangle, \langle e, 17\rangle, \langle ab, 8\rangle, \langle ae, 8\rangle, \langle de, 13\rangle, \langle cde, 9\rangle\}$. From Theorem 2 we know that all σ-vulnerable patterns such as $(a \wedge \neg b) \vee (c \wedge \neg e)$ are k-anonymous, i.e. there exists at least a database, compatible with $\mathcal{O}^3$, in which such pattern does not appear or it appears at least k times. Therefore, a malicious attacker can not infer, in any possible way, that the cardinality of the group of individuals described by such pattern is for sure lower than k.

Unfortunately, Theorem 2 applies to σ-vulnerable pattern such as $(a \wedge \neg b) \vee (c \wedge \neg e)$, i.e., both itemsets $\{ab\}$ and $\{ce\}$ are frequent; but it does not apply, for instance, to the pattern $a \wedge b \wedge \neg c$, since the itemset $\{abc\}$ is not frequent.

In the particular case of our running example, all non σ-vulnerable patterns are also k-anonymous, in other terms no *border crossing inference channels* can be found. This means that $\mathcal{O}^3$ is completely safe. To prove this, we show that exists a 3-anonymous database compatible with $\mathcal{O}^3$ (w.r.t. $\sigma = 8$).

a	b	c	d	e	
1	1	1	1	1	x4
0	0	1	1	1	x5
0	0	0	1	1	x4
1	0	0	0	1	x4
1	1	0	0	0	x4

Similarly, we can show that also the $\mathcal{O}^3$ produced by the suppressive strategy is completely safe.

Unfortunately this ad-hoc proofs can not be applied in general. We are provable protected under attacks to σ-vulnerable patterns, but, although hardly, it is possible to find set of sanitized itemsets that allow the attacker to discover non σ-vulnerable patterns that are non k-anonymous.

Example 17 Suppose that for a given database, and with thresholds $\sigma = 5$ and $k = 3$, after our pattern sanitization we got the following set of itemsets: $\mathcal{O}^3 = \{\langle ab, 9\rangle, \langle ac, 8\rangle, \langle bc, 5\rangle, \langle a, 14\rangle, \langle b, 14\rangle, \langle c, 13\rangle, \langle\emptyset, 22\rangle\}$.

By applying deduction rules [10], we can infer some information on the support of the infrequent itemsets $\{abc\}$. In fact we got that:
$sup(abc) \geq sup(ab) + sup(ac) - sup(a) = 3$, and
$sup(abc) \leq sup(\emptyset) - sup(a) - sup(b) - sup(c) + sup(ab) + sup(ac) + sup(bc) = 22 - 14 - 14 - 13 + 9 + 8 + 5 = 3$.
Therefore, although it has not been disclosed, we can conclude that the support of the infrequent itemset $\{abc\}$ is exactly 3. From this we can discover a *border crossing inference channels*: $\mathcal{C}_{bc}^{abc}$ which as support $2 < k$. We have inferred that the non σ-vulnerable patterns $b \wedge c \wedge \neg a$ is also non k-anonymous.

This kind of attacks are very difficult but, in some cases, possible. We are actually characterizing them, and studying ad hoc detection and sanitization techniques.

Other issues, emerging from our approach, are worth a deeper investigation and are left to future research.

One path of research regards the mapping of our theoretical framework to the more concrete case of categorical data originating from relational tables. In this context, we could exploit the semantics of the attributes in order to apply generalization techniques similar to what done by classical k-anonymization [45]. Moreover, we could introduce in our framework the distinction between quasi-identifiers and sensitive attributes and focus

our pattern sanitization techniques only to the projection of patterns on the quasi-identifiers. After this concretization step, we should be able to provide a more comprehensive empirical evaluation of our approach: to this purpose we intend to conduct a large-scale experiment with real life bio-medical data about patients to assess both applicability and scalability of the approach in a realistic, challenging domain.

Another open problem, deserving further investigation, regards the release multiple collections of k-anonymous itemsets extracted from the same source database, but with different support thresholds. A malicious adversary receiving more than one of this collections can violate the k-anonymity defense. However, in the case we want to release multiple collections of patterns we can adopt the following straightforward solution. If we want to share N set of patterns with support thresholds respectively $\sigma_1, \sigma_2, \ldots, \sigma_N$, then we have to mine and k-anonymize the set of itemsets extracted with $\min_{1 \leq i \leq N} \sigma_i$. From this set of itemsets (k-anonymized) we can select the various collections corresponding to the various σ_i. The most informative set of patterns released is the one with $\sigma = \min_{1 \leq i \leq N} \sigma_i$; but this is k-anonymous and thus there is no problem, while the other sets are just subsets of this, thus containing less information, and conjoining them we cannot infer any non k-anonymous pattern. The solution described could not be always applicable. In fact, the data owner must know in advance the value $\min_{1 \leq i \leq N} \sigma_i$. If he does not know this value, then he can still use a value small enough for σ and then share subsets w.r.t. the σ required situation by situation.

Finally, we plan to investigate whether the proposed notion of anonymity preserving pattern discovery may be applied to other forms of patterns and models, for instance, classification or clustering models. In those scenarios, we should study how to produce a model that, while maintaining accuracy, it provably does not allow an attacker to infer information regarding less than k individuals in the original training set.

In any case, the importance of the advocated form of privacy-preserving pattern discovery is evident: demonstrably trustworthy data mining techniques may open up tremendous opportunities for new knowledge-based applications of public utility and large societal and economic impact.

References

1. `http://fimi.cs.helsinki.fi/data/`.
2. D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM PODS*, 2001.
3. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD*.
4. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD on Management of Data*.
5. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the Twentieth VLDB*, 1994.
6. M. Atallah, A. Elmagarmid, M. Ibrahim, E. Bertino, and V. Verykios. Disclosure limitation of sensitive rules. In *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange*, page 45. IEEE Computer Society, 1999.
7. M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Blocking anonymity threats raised by frequent itemset mining. In *Proceedings of Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 561–564.
8. M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. k-anonymous patterns. In *Proceedings of 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05), Porto, Portugal, 2005.*
9. T. Calders. Computational complexity of itemset frequency satisfiability. In *Proc. PODS Int. Conf. Princ. of Database Systems*, 2004.
10. T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proceedings of the 6th PKDD*, 2002.
11. L. Chang and I. S. Moskowitz. An integrated framework for database inference and privacy protection. In *Data and Applications Security*, 2000.
12. D. Cheung, J. Han, V. Ng, A. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In *4th International Conference on Parallel and Distributed Information Systems (PDIS '96)*, 1996.
13. C. Clifton, M. Kantarcioglu, and J. Vaidya. Defining privacy for data mining. In *Natural Science Foundation Workshop on Next Generation Data Mining*, 2002.
14. C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explor. Newsl.*, 4(2), 2002.
15. E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. Hiding association rules by using confidence and support. In *Proceedings of the 4th International Workshop on Information Hiding*, 2001.
16. W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *Proceedings of the 2001 workshop on New security paradigms*, 2001.
17. W. Du and Z. Zhan. Building decision tree classifier on private data. In *Proceedings of the IEEE international conference on Privacy, security and data mining*, 2002.
18. W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.
19. V. Estivill-Castro and L. Brankovic. Data swapping: Balancing privacy against precision in mining for logic rules. In *Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery*, 1999.
20. A. Evfimievski. Randomization in privacy preserving data mining. *SIGKDD Explor. Newsl.*, 4(2), 2002.
21. A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2003.

22. A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.
23. P. Fule and J. F. Roddick. Detecting privacy and ethical sensitivity in data mining results. In *Proc. of the 27th conference on Australasian computer science*, 2004.
24. D. Hand, H. Mannila, and P. Smyh. *Principles of Data Mining*. The MIT Press, 2001.
25. Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *SIGMOD'05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 37–48, 2005.
26. I. Ioannidis, A. Grama, and M. Atallah. A secure protocol for computing dot-products in clustered and distributed environments. In *Proceedings of the International Conference on Parallel Processing (ICPP'02)*, 2002.
27. M. Z. Islam and L. Brankovic. A framework for privacy preserving classification in data mining. In *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 163–168, 2004.
28. J. Kacprzyk and K. Cios, editors. *Medical Data Mining and Knowledge Discovery*. Physica-Verlag, 2001.
29. M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In *In The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02)*, 2002.
30. M. Kantarcioglu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *Proceedings of the tenth ACM SIGKDD*, 2004.
31. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Proceedings of the Third IEEE International Conference on Data Mining*, 2003.
32. D. Knuth. *Fundamental Algorithms*. Addison-Wesley, Reading, Massachusetts, 1997.
33. K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60, 2005.
34. K. Muralidhar and R. Sarathy. Security of random data perturbation methods. *ACM Trans. Database Syst.*, 24(4), 1999.
35. S. R. M. Oliveira and O. R. Zaiane. Privacy preserving frequent itemset mining. In *Proceedings of the IEEE international conference on Privacy, security and data mining*, 2002.
36. S. R. M. Oliveira and O. R. Zaiane. Protecting sensitive knowledge by data sanitization. In *Third IEEE International Conference on Data Mining (ICDM'03)*, 2003.
37. S. R. M. Oliveira, O. R. Zaiane, and Y. Saygin. Secure association rule sharing. In *Proc.of the 8th PAKDD*, 2004.
38. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. ICDT '99*, 1999.
39. J. Pei, J. Han, and J. Wang. Closet+: Searching for the best strategies for mining frequent closed itemsets. In *SIGKDD '03*, 2003.
40. B. Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.*, 4(2), 2002.
41. S. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *Proceedings of the 28th VLDB Conference*, 2002.
42. P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1-3, 1998, Seattle, Washington*, 1998.
43. Y. Saygin, V. S. Verykios, and C. Clifton. Using unknowns to prevent discovery of association rules. *SIGMOD Rec.*, 30(4), 2001.
44. L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 2002.
45. L. Sweeney. k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 2002.
46. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.
47. V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 33(1):50–57, 2004.
48. X. Wu, Y. Wu, Y. Wang, and Y. Li. Privacy aware market basket data set generation: A feasible approach for inverse frequent set mining. In *Proc. 2005 SIAM Int. Conf. on Data Mining*, 2005.
49. M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemsets mining. In *2nd SIAM International Conference on Data Mining*, 2002.