k-Anonymous Patterns

Maurizio Atzori[†‡] Francesco Bonchi[‡] Fosca Giannotti[‡] Dino Pedreschi[†]

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

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

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 association rule mining. In this paper we show that this belief is ill-founded. By shifting the concept of k-anonymity 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 might arise from the disclosure of a set of extracted patterns. On this basis we obtain a formal and effective notion of privacy protection that allows the disclosure of the extracted knowledge together with the proof that it does not violate the anonymity of the individuals in the source database. Finally, 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.

1 Introduction

Privacy Preserving Data Mining, i.e., the analysis of data mining side-effects on privacy, has recently become a key research issue and is receiving a growing attention from the research community [1, 3, 9, 11, 26, 37]. However, despite such efforts, a common understanding of what is meant by "privacy" is still missing. This fact has led to the proliferation of many completely different approaches to privacy preserving data mining, all sharing the same generic goal: producing a valid mining model without disclosing "private" data.

As highlighted in [22], 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? During the last year, few works [19, 22, 28] have tried to address this problem by some different points of view, but they all require some *a priori* knowledge of what is sensitive and what is not.

In this paper we study when data mining results represent *per se* a threat to privacy, without any background knowledge of what is sensitive. In particular, we focus on *individual privacy*, which is mainly concerned with the *anonymity* of individuals.

A prototypical application instance is in the medical domain, where the collected data is typically very sensitive, and the kind of privacy usually required is the anonymity of the patients in a survey. 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.

At a first sight, it seems 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 association rule mining. The next example shows that the above belief is ill-founded.

Example 1 *Consider the following association rule:*

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

ISTI-2005-TR, ISTI-CNR, May 2005

where sup and conf are the usual interestingness measures of support and confidence as defined in [2]. 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:

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

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.

The knowledge inferred is a clear threat to the anonymity of that individual: on one hand the pattern identifying the individual could itself contain sensitive information; on the other hand it could be used to re-identify the same individual in other databases.

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.

k-Anonymity: From Data To Patterns

An important method for protecting individual privacy is *k-anonymity*, introduced in [34], a notion that establishes that the cardinality of the answer to any possible query will be at least k. In this work, 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. In particular, this is obtained by a "sanitization" of the source data that must be transformed in such a way that, for all possible queries, at least k tuples will be returned. Such a sanitization is obtained by generalization and suppression of attributes and tuples [35].

Trivially, by mining a k-anonymized database nothing threatening the anonymity can be obtained. But such mining would produce models 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.

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. By this point of view, we can shift the concept of k-anonymity from the data to the extracted patterns in a straightforward way: we say that the result of a data mining extraction is k-anonymous if from any patterns inferred from such results is not possible to identify a group of tuples of cardinality less than k.

Paper Contributions and Organization

In this paper we study the privacy problem described above in the very general setting of patterns which are boolean formulas over a binary database. Our contributions are the following.

- A general characterization of k-anonymous patterns and of the inference channels among patterns which may threat anonymity of source data.
- An effective and efficient 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.
- A strategy to eliminate the threats to anonymity by introducing distortion on the dangerous patterns in a controlled way, by measuring the effects of the distortion.

It should be noted that the capability of detecting the potential threats is extremely useful for the analyst to determine a trade-off among the quality of mining result and the privacy guarantee, by means of an iterative interaction with the proposed detection algorithm. Our empirical experiments, reported in this paper, bring evidence to this observation.

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, we allow real mining on the real data, while focussing on the anonymity preservation properties of the extracted patterns.

The plan of the paper follows. In the next Section we formalize the idea of k-anonymity for patterns, then we describe the kinds of inference that a malicious adversary can exploit to retrieve non k-anonymous patterns. At the end of this Section we formalize the privacy preserving data mining problem addressed by this paper. Then in Section 3 we will study the properties that allow to identify dangerous patterns hidden in a set of frequent itemsets, and then in Section 4, we propose a simple yet very effective way to eliminate these threats.

2 k-Anonymous Patterns

We start by defining binary databases and patterns according to [20].

$\mathcal{D}$

	a	b	c	d	e
t_1	1	1	1	1	1
t_2	0	1	1	0	0
t_3	0	1	1	1	1
t_4	1	1	1	1	0
t_5	1	1	1	0	1
t_6	0	1	1	1	1
t_7	0	1	1	0	1
t_8	1	1	0	1	0

$$sup_{\mathcal{D}}(a \vee e) = 7$$
$$sup_{\mathcal{D}}(e \wedge (\neg a \vee \neg d)) = 4$$

$$sup_{\mathcal{D}}(abc) = 3$$
$$sup_{\mathcal{D}}(bcde) = 3$$

Figure 1: An example binary database, two example patterns and their supports, two example itemsets and theirs supports.

Definition 1 *A binary database $\mathcal{D}$ 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. *A* pattern *for the variables in $\mathcal{I}$ is a logical (propositional) sentence obtained by connecting conditions on the value of some of the variables, using the AND ($\wedge$) and OR ($\vee$) logical connectives. The domain of all possible patterns is denoted $\mathcal{P}at(\mathcal{I})$.*

According to Definition 1 $(i_j = 1) \wedge (i_k = 1 \vee i_l = 0)$ is a pattern. In the rest of the paper we will denote this pattern simply by $i_j \wedge (i_k \vee \neg i_l)$.

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.

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 about which the given pattern is true.

Definition 2 *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 make 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 3 *Given a 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 most studied *pattern class* is the itemset, i.e., a conjunction of positive valued variables, or in other words, a set 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 [2, 4].

Definition 4 *The set of all itemsets $2^{\mathcal{I}}$ is a pattern class consisting of all possible conjunctions of the form $i_j = 1 \wedge i_k = 1 \ldots \wedge i_l = 1$. Given a database $\mathcal{D}$ and a minimum support threshold σ, the frequent itemsets mining problem requires to compute:*

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

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

Example 2 *In the database $\mathcal{D}$ in Figure 1, we have that: $\mathcal{F}(\mathcal{D}, 7) = \{\langle \emptyset, 8\rangle, \langle b, 8\rangle, \langle c, 7\rangle, \langle bc, 7\rangle\}$.*

Inference of Supports and Anonymity Threats

We address the problem of patterns *anonymity* in the result of a frequent itemsets extraction. In the following we formally define the kinds of attacks we consider. Since we disclose only a set of frequent itemsets, the attacks can only consist in inferring, from this collection, knowledge about the existence of *non k-anonymous patterns*; i.e., any pattern p such that $0 < sup_{\mathcal{D}}(p) < k$.

Definition 5 *A set of σ-frequent itemsets is a set of pairs $\langle X, n\rangle$, where $X \in 2^{\mathcal{I}}$ and $n \in \mathbb{N}, n \geq \sigma$. A set S of σ-frequent itemsets and a database $\mathcal{D}$ are said to be compatible if $S = \mathcal{F}(\mathcal{D}, \sigma)$.*

Definition 6 *Given a set S of σ-frequent itemsets and a pattern p 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 $sup_{\mathcal{D}}(p) < x$).*

Note that, according to this definition, we can infer everything from a set S of σ-frequent itemsets if S itself is not compatible with any database, or, in other words, if it contains contradictions (e.g., $sup(X) < sup(Y)$ 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 [5]. However such a problem can be tackled by using some heuristics [38]. 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 has been k-anonymized from a non k-anonymized one.

These considerations lead to the following problems statement.

Problems Definition

The first problem that we address is the detection of anonymity threats in the output of a frequent itemset extraction. Informally, we call *inference channel* any substructure of the collection of itemsets (with their respective supports), from which it is possible to infer non k-anonymous patterns.

Problem 1 *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 $\mathcal{C}$ which exist in $\mathcal{F}(\mathcal{D}, \sigma)$:*

$$\mathcal{C} \subseteq \mathcal{F}(\mathcal{D}, \sigma) \ : \ \exists p \in \mathcal{P}at(\mathcal{I}) : \mathcal{C} \models 0 < sup_{\mathcal{D}}(p) < k.$$

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, specified as follows.

Problem 2 *Given a collection of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$, and the set of all its inference channels (output of Problem 1), our problem consists in transforming $\mathcal{F}(\mathcal{D}, \sigma)$ 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. $\nexists p \in \mathcal{P}at(\mathcal{I}) : \mathcal{O}^k \models 0 < sup_{\mathcal{D}}(p) < k;$
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 a malicious adversary should not be able to infer from $\mathcal{O}^k$, in any possible way, the existence of any non k-anonymous pattern. The second condition constraints the output collection of itemsets to be *"realistic"*; while the third condition requires to control the distortion effects of the transform of the original output by means of appropriate distortion measures (see Section 4).

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 a 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 a number can help a malicious adversary to guess the support of non k-anonymous patterns.

Frequency and Anonymity Thresholds

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 4), and an anonymity threshold k for general patterns (as defined in Definition 1).

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 $\sigma > k$.

3 Detecting Inference Channels

In this Section we study Problem 1: how information about non k-anonymous patterns can be possibly inferred from a collection of σ-frequent itemsets. Such a Problem can be further decomposed in two subproblems:

1. detecting inference channels involving only frequent itemsets;
2. detecting inference channels involving also infrequent itemsets.

The first problem, addressed in the rest of this Section, is the most essential. In fact, 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 [6] 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.* Such deduction techniques can be exploited to discover information about infrequent itemsets, and from these to infer non k-anonymous patterns.

For lack of space, this second-order problem is not discussed here, and left to the extended version of this paper. 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, which will be presented in the following. This is due to the fact that both kinds of channels rely on the same concept: inferring supports of larger itemsets from smaller ones. Actually, the key equation of our work (Lemma 1) is also the basis of the deduction rules proposed in [6].

From now on we restrict our attention to the essential form of inference channel, namely those involving frequent itemsets only.

Characterization of Inference Channels

As suggested by Example 1, 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 pattern $p = i_1 \wedge \dots \wedge i_m \wedge \neg a_1 \wedge \dots \wedge \neg a_n$ can be inferred if we know the support of itemsets $I = \{i_1, \dots, i_m\}$, $J = I \cup \{a_1, \dots, a_n\}$, and every itemset L such that $I \subset L \subset J$.

Lemma 1 *Given a pattern $p = i_1 \wedge \dots \wedge i_m \wedge \neg a_1 \wedge \dots \wedge \neg a_n$ it holds that:*

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

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

Proof(Sketch) The proof follows directly from the definition of support and the well-known *inclusion-exclusion principle* [24]. □

Following the notation in [6], we denote the right-hand side of the equation above as $f_I^J(\mathcal{D})$.

Example 3 *In the database $\mathcal{D}$ in Figure 1 we have that $sup_{\mathcal{D}}(b \wedge \neg a \wedge \neg e) = f_b^{abe}(\mathcal{D}) = sup_{\mathcal{D}}(b) - sup_{\mathcal{D}}(ab) - sup_{\mathcal{D}}(be) + sup_{\mathcal{D}}(abe) = 8 - 4 - 5 + 2 = 1$.*

Definition 7 *Given two itemsets $I, J \in 2^{\mathcal{I}}$ such that $I = \{i_1, \dots, i_m\}$ and $J = I \cup \{a_1, \dots, a_n\}$, we denote the conjunctive pattern $p = i_1 \wedge \dots \wedge i_m \wedge \neg a_1 \wedge \dots \wedge \neg a_n$ with the symbol $\mathcal{C}_I^J$. Given a database $\mathcal{D}$ we have that $sup_{\mathcal{D}}(\mathcal{C}_I^J) = f_I^J(\mathcal{D})$. If $0 < f_I^J(\mathcal{D}) < k$, then $\mathcal{C}_I^J$ is called an* inference channel.

Example 4 *Consider the database $\mathcal{D}$ of Figure 1, and suppose $k = 3$. We have that $\mathcal{C}_b^{abe}$ is an inference channel of support 1. This means that there is only one transaction $t \in \mathcal{D}$ such that $b \wedge \neg a \wedge \neg e$.*

The next Theorem states that, for every possible non k-anonymous pattern, there is always a conjunctive pattern which is non k-anonymous as well. This means that if there exists a non k-anonymous pattern, we can always find a pair of itemsets $I \subseteq J \in 2^{\mathcal{I}}$ such that $\mathcal{C}_I^J$ is an inference channel.

Theorem 1
$\forall p \in \mathcal{P}at(\mathcal{I}) : 0 < sup_{\mathcal{D}}(p) < k \; . \; \exists \mathcal{C}_I^J : 0 < f_I^J(\mathcal{D}) < k$

Proof The case of a conjunctive pattern p is a direct consequence of Lemma 1. Let us now consider a generic pattern $p \in \mathcal{P}at(\mathcal{I})$. A generic pattern can always be transformed in normal disjunctive form: $p = p_1 \vee \dots \vee p_q$, where $p_1 \dots p_q$ are conjunctive patterns. In this case 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 Theorem 1 we conclude that all possible threats to anonymity can be linked to inference channels of the form $\mathcal{C}_I^J$.

Algorithm 1 Naïve Inference Channel Detector

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

Definition 8 *We say that an inference channel $\mathcal{C}_I^J$ holds in $\mathcal{F}(\mathcal{D}, \sigma)$ if $\{\langle I, sup_{\mathcal{D}}(I)\rangle, \langle J, sup_{\mathcal{D}}(J)\rangle\} \subseteq \mathcal{F}(\mathcal{D}, \sigma)$. The set of all inference channels holding in $\mathcal{F}(\mathcal{D}, \sigma)$, together with their supports, is denoted $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$.*

Example 5 *Consider the database $\mathcal{D}$ in Figure 1 and suppose $k = 2$. The following is the set of inference channels holding in $\mathcal{F}(\mathcal{D}, 3))$:*
$\mathcal{C}h(2, \mathcal{F}(\mathcal{D}, 3)) = \{\langle \mathcal{C}_a^{ac}, 1\rangle, \langle \mathcal{C}_b^{dbe}, 1\rangle, \langle \mathcal{C}_a^{da}, 1\rangle, \langle \mathcal{C}_\emptyset^{de}, 1\rangle, \langle \mathcal{C}_{db}^{dcb}, 1\rangle, \langle \mathcal{C}_\emptyset^{c}, 1\rangle, \langle \mathcal{C}_b^{cbe}, 1\rangle, \langle \mathcal{C}_{ab}^{dab}, 1\rangle, \langle \mathcal{C}_{dcb}^{dcbe}, 1\rangle, \langle \mathcal{C}_{cb}^{dcbe}, 1\rangle, \langle \mathcal{C}_{db}^{dcbe}, 1\rangle, \langle \mathcal{C}_b^{cb}, 1\rangle, \langle \mathcal{C}_d^{dc}, 1\rangle, \langle \mathcal{C}_d^{dce}, 1\rangle, \langle \mathcal{C}_{dc}^{dce}, 1\rangle, \langle \mathcal{C}_c^{dce}, 1\rangle, \langle \mathcal{C}_\emptyset^{ce}, 1\rangle, \langle \mathcal{C}_{ab}^{acb}, 1\rangle\}$.

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 $\sum_{1 \le i \le n} \binom{n}{i} (2^i - 1)$.

In the following we study some properties that allow to dramatically reduce the number of checks needed to retrieve $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$.

Anti-monotonicity

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 larger than the number of transactions for which the pattern $a \wedge b \wedge \neg c \wedge \neg d$ holds.

Definition 9 *Given two conjunctive patterns $\mathcal{C}_I^J$ and $\mathcal{C}_H^L$ we say that $\mathcal{C}_I^J \preceq \mathcal{C}_H^L$ when $I \subseteq H$ and $(J \setminus I) \subseteq L$.*

Proposition 1 $\mathcal{C}_I^J \preceq \mathcal{C}_H^L \Rightarrow \forall \mathcal{D} \,.\, f_I^J(\mathcal{D}) \geq f_H^L(\mathcal{D})$.

Therefore, when detecting inference channels, if we move from the larger ones to the smaller, whenever we find a conjunctive pattern $\mathcal{C}_I^J$ such that $f_I^J(\mathcal{D}) \geq k$, we can avoid checking the support of all conjunctive patterns below $\mathcal{C}_I^J$ in the $\preceq$ ordering.

Redundant Inference Channels

In Example 5 we have many inference channels which are clearly redundant. Consider, for instance, the two inference channels $\langle \mathcal{C}_{dc}^{dce}, 1\rangle \preceq \langle \mathcal{C}_{dcb}^{dcbe}, 1\rangle$: among the two associated non k-anonymous patterns, one is more specific than the other, but they both uniquely identify transaction t_4. It is easy to see that many other families of equivalent, and thus redundant, inference channels can be found in $\mathcal{C}h(2, \mathcal{F}(\mathcal{D}, 3))$.

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 [29] and since then they have received a great deal of attention especially by an algorithmic point of view [39, 30]. By using closed itemsets, we implicitly benefit from data correlations which allow to strongly reduce problem complexity and output size, by discarding all redundancies. Closed itemsets are a concise and lossless representation of all frequent itemsets: 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 10 *Given the function $f(T) = \{i \in \mathcal{I} \mid \forall t \in T, i \in t\}$, which returns all the items included in the set of transactions T, and $g(X) = \{t \in \mathcal{T} \mid \forall i \in X, i \in t\}$ which returns the set of transactions supporting a given itemset X, the composite function $f \circ g$ is called Galois operator or closure operator. An itemset I is said to be closed (w.r.t $\mathcal{I}$ and $\mathcal{T}$) if and only if $c(I) = (f \circ g)(I) = f(g(I)) = I$.*

Definition 11 *Given a database $\mathcal{D}$ and a minimum support threshold σ, the frequent closed itemsets mining problem requires to compute:*

$$\mathcal{C}l(\mathcal{D}, \sigma) = \{\langle X, sup_{\mathcal{D}}(X)\rangle \in \mathcal{F}(\mathcal{D}, \sigma) \mid X = c(X)\}$$

An itemset I is said to be frequent maximal if and only if it is frequent closed and $\nexists J \supset I$ s.t. $J \in \mathcal{C}l(\mathcal{D}, \sigma)$.

Now we can define a set of equivalence classes over the lattice of frequent itemsets, where two itemsets X, Y belong to the same class if and only if $c(X) = c(Y)$. Closed itemsets are exactly the maximal elements of these equivalence classes. Figure 2 shows the lattice of frequent itemsets derived from the same simple dataset of Figure 1. Each equivalence class contains elements sharing the same supporting transactions, and closed itemsets are the largest element of each class.

Example 6 *In the database $\mathcal{D}$ in Figure 1, we have that: $\mathcal{C}l(\mathcal{D}, 5) = \{\langle b, 8\rangle, \langle bc, 7\rangle, \langle bd, 5\rangle, \langle bce, 5\rangle\}$. In this situation bd and bce are maximal frequent itemsets.*

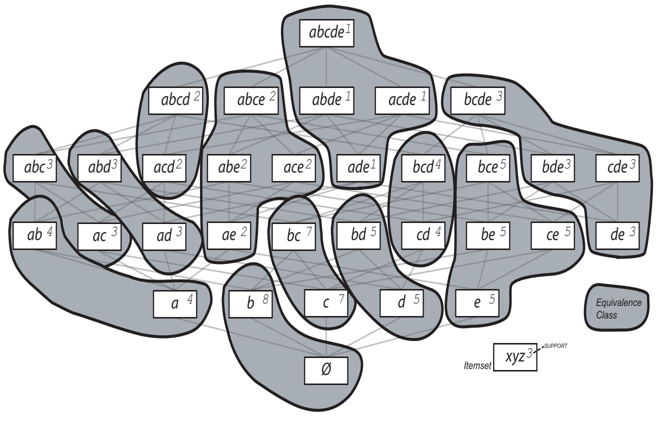


Figure 2: Equivalence classes of itemsets for the dataset $\mathcal{D}$ defined in Figure 1.

The very same concept can be used to discard redundant inference channels and to identify only the relevant ones. Indeed, since itemsets belonging to the same equivalence class are supported by the same transactions, we can think of inference channels as holding among equivalence classes of frequency, instead of between pairs of itemsets. In the following, we exploit this consideration to reduce the number of checks needed to detect all possible threats to anonymity in the output of a frequent itemset extraction.

Definition 12 *Given the set of all frequent closed itemsets $\mathcal{C}l(\mathcal{D},\sigma)$, we define:*

$$\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma)) = \{\langle \mathcal{C}_I^J, f_I^J(\mathcal{D})\rangle | I \in \mathcal{C}l(\mathcal{D},\sigma), J \textit{ maximal}\}$$

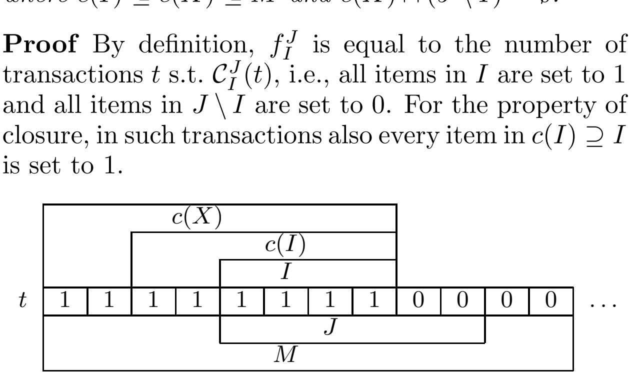
The next Theorem shows how for any pair of itemsets I, J we can compute $f_I^J(\mathcal{D})$ from $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$.

Theorem 2 *Given a database $\mathcal{D}$, a frequency threshold σ and two frequent itemsets $I, J \in \mathcal{F}(\mathcal{D},\sigma)$ such that $I \subseteq J$, let M be any maximal (w.r.t. set inclusion in $\mathcal{F}(\mathcal{D},\sigma)$) itemset such that $M \supseteq J$. The following equation holds:*

$$f_I^J(\mathcal{D}) = \sum_{c(X)} f_{c(X)}^M(\mathcal{D})$$

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. For the property of closure, in such transactions also every item in $c(I) \supseteq I$ is set to 1.



Consider now the M-projection of every such t. Trivially, we also have that $M \supseteq J \supseteq I$ and $M \supseteq (J \setminus I)$. In the summation of the right-hand side of the equation, we do not consider transactions that have zeros in $c(I)$ or have ones in $J \setminus I$. We must show that in the summation we count each such transaction t exactly once: this means that $\mathcal{C}_{c(X)}^M$ (with M fixed and varying $c(X)$) forms a partition of the set of transactions t we are considering (the transactions in which $\mathcal{C}_I^J$ holds). Every pattern $\mathcal{C}_{c(X)}^M$ is mutually exclusive w.r.t. the other patterns. In fact, given $\mathcal{C}_{c(X_1)}^M$ and $\mathcal{C}_{c(X_2)}^M$ s.t. $c(X_1) \subset c(X_2)$, we have that the items in $c(X_2) \setminus c(X_1)$ are set to 0 in the transactions in which $\mathcal{C}_{c(X_1)}^M$ holds, and set to 1 in the transactions in which $\mathcal{C}_{c(X_2)}^M$ holds. As a consequence, the same transaction can not be considered by both $\mathcal{C}_{c(X_1)}^M$ and $\mathcal{C}_{c(X_2)}^M$. Finally, we must show that every such transaction t is considered at least once: let Y denote the set of items in the M-projection of t that are set to 1. Since Y is necessarily a closed itemset[1], $c(X) = Y$ is considered in the summation, and $\mathcal{C}_{c(X)}^M$ trivially holds in t. □

Corollary 1 *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 $c(X) \cap (I \setminus J) = \emptyset$, $0 \leq f_{c(X)}^M(\mathcal{D}) < k$.*

Proof Since $\mathcal{C}_I^J \preceq \mathcal{C}_{c(X)}^M$, and $f_I^J(\mathcal{D}) < k$, we conclude that $f_{c(X)}^M(\mathcal{D}) \leq f_I^J(\mathcal{D}) < k$. Moreover, for at least one $c(X)$ we have that $f_{c(X)}^M(\mathcal{D}) > 0$, otherwise we get a contradiction to Theorem 2. □

From Corollary 1 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, we have that the set $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$ represents, without redundancy, all inference channels holding in $\mathcal{F}(\mathcal{D},\sigma)$. This means 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 performing a much smaller number of checks.

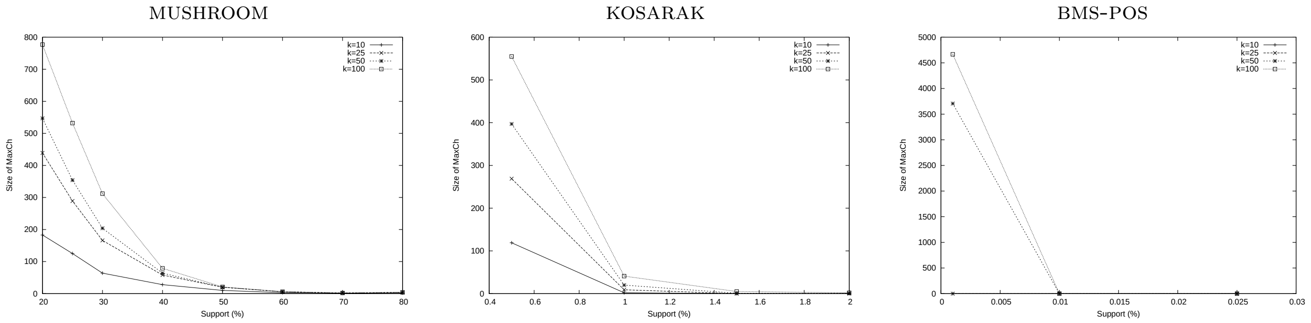
Example 7 *In the database $\mathcal{D}$ in Figure 1 we have that:*

$$\mathcal{MC}h(2,\mathcal{C}l(\mathcal{D},3)) = \{\langle\mathcal{C}_{ab}^{dab},1\rangle, \langle\mathcal{C}_{cb}^{dcbe},1\rangle, \langle\mathcal{C}_{db}^{dcbe},1\rangle, \langle\mathcal{C}_{ab}^{acb},1\rangle\}$$

Anonymity vs. Accuracy: Empirical Observations

Algorithm 2 represents an optimized way to identify all threats to anonymity. Its performance revealed ad-

[1] In fact M is maximal and $Y \subseteq M$ (therefore Y is frequent). By contradiction, if Y is not closed, then Y appears always together with at least one other items a which is not in M, therefore also $M \cup \{a\}$ is frequent. It follows that M is not maximal.



(a) (b) (c)

Figure 3: Cardinality of $\mathcal{MCh}(k, \mathcal{C}l(\mathcal{D}, \sigma))$ for different σ and k on three datasets.

Algorithm 2 Optimized Inference Channel Detector

```
Input: $\mathcal{C}l(\mathcal{D}, \sigma), k$
Output: $\mathcal{MCh}(k, \mathcal{C}l(\mathcal{D}, \sigma))$
$M = \{I \in \mathcal{C}l(\mathcal{D}, \sigma) | I\ is\ maximal\}$;
$\mathcal{MCh}(k, \mathcal{C}l(\mathcal{D}, \sigma)) = \emptyset$;
for all $J \in M$ do
    for all $I \in \mathcal{C}l(\mathcal{D}, \sigma)$ such that $I \subseteq J$ do
        compute $f_I^J$;
        if $0 < f_I^J < k$ then
            insert $\langle \mathcal{C}_I^J, f_I^J \rangle$ in $\mathcal{MCh}(k, \mathcal{C}l(\mathcal{D}, \sigma))$;
```

equate in all our empirical evaluations using various datasets from the FIMI repository[2]; in all such cases the time improvement from the Naïve (Algorithm 1) to the optimized algorithm is about one order of magnitude. 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. The best solution to this problem is precisely to find out the minimum support threshold that generates a collection of patterns with no threats. The plots in Figure 3 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 $\mathcal{MCh}(k, \mathcal{C}l(\mathcal{D}, \sigma))$), and the various curves indicate such a number according to different values of the anonymity threshold k. In Figure 3(a) we report the plot for the MUSHROOM dataset (a dense one), while in Figure 3(b) and Figure 3(c) we report the plot for, respectively, the KOSARAK dataset and the BMS-POS dataset, both sparse. In all 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 is the problem we tackle next.

4 Blocking Inference Channels

In this section we study the problem of how to block the threats to anonymity described in the previous section. One naïve attempt to solve the problem 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 not (in general) compatible with any database, and, as observed before, could open the door to inverse mining attacks.

A possible way to sanitize the output while maintaining compatibility with some source database could be to access the database, retrieve the set of tuples which contain the inference channels detected by Algorithm 2, and directly sanitize such tuples. Then we should mine frequent itemsets again from the sanitized database. This solution could be not always feasible: for instance in a context where we have a stream of data that can be read only once.

Therefore, we focus on a simpler solution that acts simply as a post-processor, taking the frequent itemsets and directly sanitizing them accordingly to the inference channels detected by Algorithm 2. Such simple yet effective pattern sanitization is performed in such a way to maintain database compatibility.

The basic idea we propose is roughly as follows: for all inference channels $\mathcal{C}_I^J$ increase the support of the itemset I by k to enforce $f_I^J > k$. In order to maintain database-compatibility, the support of all subsets of I is increased accordingly.

Proposition 2 *Let S be a set of σ-frequent itemsets compatible with at least a database $\mathcal{D}$; then, incrementing by k the support of an itemset $I \in S$ and of every subset of I, we obtain another set of σ-frequent itemsets which is compatible with a database $\mathcal{D}'$, obtained by adding to $\mathcal{D}$ k transactions containing only I.*

Clearly, we are not really adding transactions, we are only asserting that, increasing the supports this

[2] http://fimi.cs.helsinki.fi/data/

way is equivalent to adding transactions, and thus database-compatibility is maintained. Moreover, the set S will contain the same σ-frequent itemsets, but some of them will have their support increased. Note that, by modifying the set of itemsets this way, we also avoid creating new inference channels.

Minimizing the Number of Tuple Insertions

Although our main goal is to hide every inference channel, this is not enough: we also want 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 transaction in the original dataset), minimize noise actually means reducing as much as possible the increments of supports (i.e. number of transactions virtually added). To do this, we exploit the antimonotony property of patterns, defining the natural partial order of inclusion among patterns and the related concept of maximal patterns (as the one of the maximal itemsets).

Observation: given two different patterns $\mathcal{C}_{ab}^{abcd}$ and $\mathcal{C}_{bt}^{bgt} =<,>$ we can join them into $\mathcal{C}_{abt}^{abcdgt}$ In general, given two pattern $\mathcal{C}_{I}^{J}$ and $\mathcal{C}_{I'}^{J'}$, it is possible to join them into $\mathcal{C}_{I''}^{J''}$ if and only if it exists a $\mathcal{C}_{I''}^{J''}$ such that $\mathcal{C}_{I''}^{J''} \supseteq \mathcal{C}_{I}^{J}$ and $\mathcal{C}_{I''}^{J''} \supseteq \mathcal{C}_{I'}^{J'}$, that is $I \cap (J' \setminus I') = \emptyset$ and $I' \cap (J \setminus I) = \emptyset$. In this case we can have $J'' = J \cup J'$ and $I'' = I \cup I'$. In the following algorithm *smax* denotes the set of such joined maximal supersets.

```
Algorithm 3 Inference Channel Sanitization
Input: $k, \mathcal{F}(\mathcal{D},\sigma)$
Output: $\mathcal{O}^k$
 1: compute $\mathcal{C}l(\mathcal{D},\sigma)$ from $\mathcal{F}(\mathcal{D},\sigma)$;
 2: compute $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$ from $\mathcal{C}l(\mathcal{D},\sigma)$;
 3: $smax := \emptyset$;
 4: for all $< \mathcal{C}_I^M, f_I^M > \in \mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$ do
 5:   if $\exists < \mathcal{C}_A^B, f_A^B > \in smax$ such that
        $A \cap (M \setminus I) = \emptyset$ and $I \cap (B \setminus A) = \emptyset$; then
 6:     $smax := smax \setminus \{< \mathcal{C}_A^B, f_A^B >\}$;
 7:     $smax := smax \cup \{< \mathcal{C}_{I\cup A}^{M\cup B}, f_{I\cup A}^{M\cup B} >\}$;
 8:   else
 9:     $smax := smax \cup \{< \mathcal{C}_I^M, f_I^M >\}$;
10: for all $< X, sup(X) > \in \mathcal{F}(\mathcal{D},\sigma)$ s.t. $X \subseteq I$ do
11:   $sup^k(I) := sup(I)$;
12:   for all $< \mathcal{C}_I^J, f_I^J > \in smax$ do
13:     $sup^k(I) := sup^k(I) + k$;
14:   $\mathcal{O}^k = \mathcal{O}^k \cup \{< I, sup^k(I) >\}$;
```

From line 3 to 9 Algorithm 3 computes the set of inference channels *smax* from $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$ by exploiting the observation above. Since it always holds that $|smax| \leq |\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))|$, this can help to reduce the total number of insertions (i.e. we reduce the difference between $\mathcal{F}(\mathcal{D},\sigma)$ and $\mathcal{F}(\mathcal{D},\sigma)^k$ (see experiments below). From line 10 to 14 the algorithm increments the support of the itemsets that fall into inference channels in *smax*. If the same itemset falls in more then one channel, we add k for each channel.

Example 8 *Mining the* MUSHROOM *dataset (8124 transactions) with minimum support equals to 60% (absolute support equals to 4874) we get 51 frequent itemsets. With* $k = 10$ *Algorithm in Fig. 3 can find 20 inference channels with* $sup = 8$. *Among the 20 inference channels found, there are only 3 which are in* $\mathcal{MC}h(k,\mathcal{C}l(\mathcal{D},\sigma))$*:*

$1 : \{85, 34\}\{85, 86, 39, 34\}$
$2 : \{85, 34\}\{85, 59, 86, 34\}$
$3 : \{85, 90, 34\}\{85, 86, 90, 36, 34\}$

In this case all of them can be joined together into a unique inference channel (smax as denoted in Algorithm in Fig. 3):

$$\{85, 34, 90\}\{85, 59, 86, 39, 34, 90, 36\}$$

i.e. this pattern is the minimal superset of all the other 3 patterns. Therefore, incrementing the support of the itemset $\{85, 34, 90\}$ *and of all its subsets by 10, we remove all these 20 inference channels holding in the output of* MUSHROOM *at 60% of support.*

Distortion Measures: Empirical Evaluation

We tested our algorithm against both synthetic and real datasets. Since the sanitization approach described in this paper always guarantees that the resulting frequent itemsets are k-anonymized, we only have to measure how much the sanitized version of the frequent itemsets differ from the original non anonymized ones. In other words, we measure the noise added by the algorithm. The only operation done by the sanitization algorithm is the increment of already frequent itemsets, therefore the set of frequent itemsets is preserved. Here are some metrics we used to evaluate the distortion to support values introduced in the sanitization:

- the maximal increment to the original support of an itemset (worst case distortion ratio):

$$\max_{I \in \mathcal{F}(\mathcal{D},\sigma)} \left\{ \frac{sup_{\mathcal{O}^k}(I) - sup_{\mathcal{F}(\mathcal{D},\sigma)}(I)}{sup_{\mathcal{F}(\mathcal{D},\sigma)}(I)} \right\}$$

- the average increment to the original support of itemsets (average distortion ratio):

$$\sum_{I \in \mathcal{F}(\mathcal{D},\sigma)} \left(\frac{sup_{\mathcal{O}^k}(I) - sup_{\mathcal{F}(\mathcal{D},\sigma)}(I)}{sup_{\mathcal{F}(\mathcal{D},\sigma)}(I)} \right)$$

We tested these measures with a wide set of public available datasets. We report in Figure 4 the results for the datasets MUSHROOM, KOSARAK and BMS-POS. As it is possible to see, the sanitization is very conservative for low values of k and high values of the support threshold. For BMS-POS the distortion is sensibly higher, due to the large number of anonymity threats found by the inference channel detector algorithm and the very low support threshold needed to mine a sufficient number of frequent itemsets. In general, in dense datasets the distortion is very low since the increments are relatively small w.r.t. the original support values. Anyway, in all cases we tested, the average and worst case distortion was acceptable for real uses.

5 Related Work

Three main approaches in privacy preserving data mining can be identified [37]. We briefly review here the main aspects of the three classic approaches; then we describe a fourth emerging research theme, where our contribution is collocated, which focuses on the potential privacy breaches within the extracted patterns.

Intensional Knowledge Hiding

This approach, also known as *sanitization*, is aimed at hiding some intensional knowledge (i.e., extracted rules/patterns) considered sensitive. 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 [7, 12, 27, 33].

Extensional Knowledge Hiding

This approach, sometimes referred to as *distribution reconstruction*, addresses the issue of privacy preservation by perturbing the data in order to avoid the identification of the original database tuples, while at the same time allowing the reconstruction of the data distribution at an aggregate level, in order to perform the mining [1, 3, 15, 16, 17, 18, 23, 25, 32]. In other words, the extensional knowledge in the dataset is hidden, but is still possible to extract valid intensional knowledge.

Distributed Extensional Knowledge Hiding

This approach, also known as *Secure Multiparty Computation* and based on *cryptographic* techniques is aimed at computing a common data mining model from several distributed datasets, where each party owning a dataset does not communicate its extensional knowledge (its dataset) to the other parties involved in the computation [8, 10, 13, 14, 21, 31, 36]. The datasets are either vertically or horizontally distributed, and the multi-party computation occurs on the basis of *secure* elementary operations, such as sum or scalar product.

Secure Intensional Knowledge Sharing

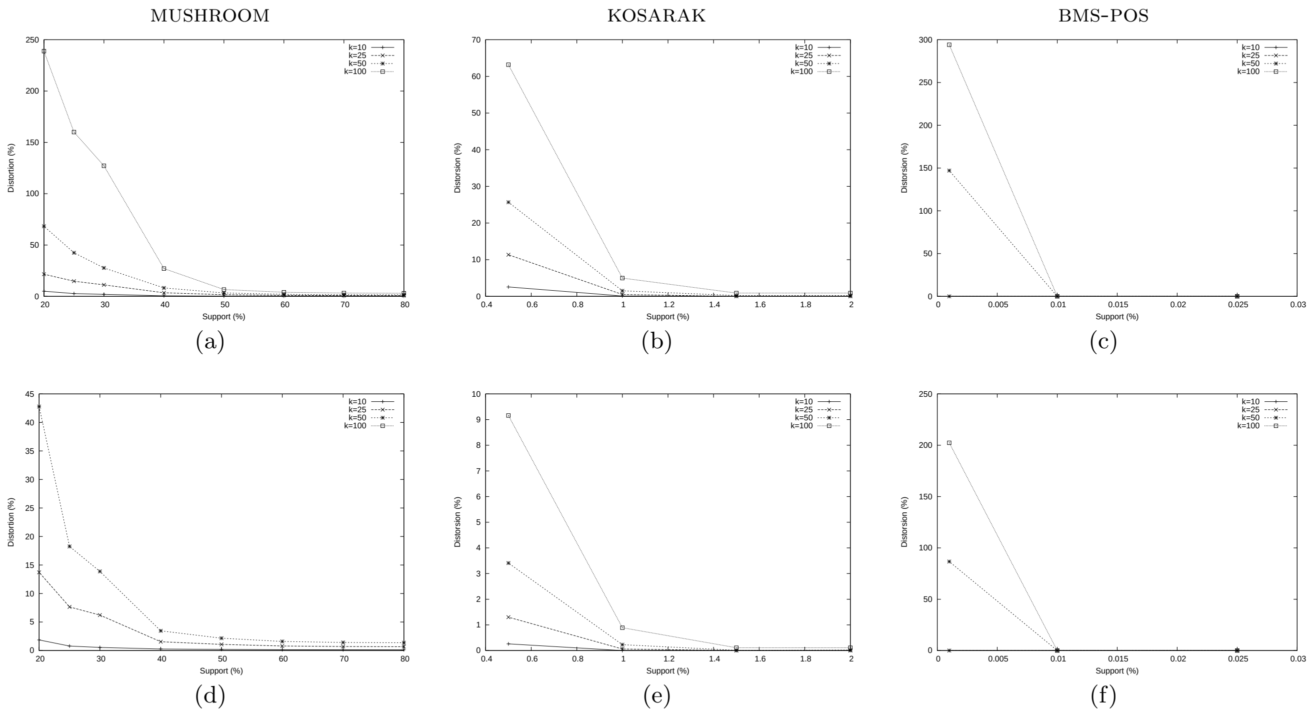
During the last year a novel problem has emerged in privacy preserving data mining [19, 22, 28]. All the previous approaches were focussed on producing a valid mining model without disclosing private data, but they still leave a crucial privacy question open [22]: do the data mining results themselves violate privacy? This issue has been preliminarily investigated in a few papers, with approaches that are deeply different among themselves and with respect to the one proposed in this paper.

The work in [22] follows the line of research in distributed extensional knowledge hiding, 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 everyone 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 [28] has some common aspects with 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 prefers to mine it and share the discovered association rules. As usual for all works in intensional knowledge hiding, the data owner knows a set of restricted association rules that (s)he does not want to disclose. The authors propose a framework to sanitize a set of association rules protecting the restricted ones by blocking some inference channels.

In [19] a framework for evaluating classification rules in terms of their perceived privacy and ethical sensitivity is described. The proposed framework empowers the analyst 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, which knows what is sensitive and what is not.

The fundamental difference of these approaches with ours 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 in-



(a) (b) (c)

(d) (e) (f)

Figure 4: Worst case distortion (a,b,c) and average distortion (d,e,f).

tuitive and realistic constraint that the anonymity of individuals should be guaranteed.

6 Conclusions

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 trade-off between anonymity and quality of the collection of patterns, as well as the distortion level needed to block the threatening inference channels that cannot be removed.

Concerning the blocking algorithm, 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. A thorough discussion of this point is left for a forthcoming extended version of this paper.

Other issues, emerging from our approach, are worth a deeper investigation and are left to future research. These include: (i) a thorough comparison of the various different approaches to block inference channels, considering all the alternatives to the approach taken in Section 4; (ii) a more comprehensive empirical evaluation of our approach: to this purpose we are conducting 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; (iii) an investigation whether the proposed notion of privacy-preserving pattern discovery may be generalized to other forms of patterns and models.

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] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2001.

[2] 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.*

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD on Management of Data.*

[4] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the Twentieth International Conference on Very Large Databases*, 1994.

[5] T. Calders. Computational complexity of itemset frequency satisfiability. In *Proc. PODS Int. Conf. Princ. of Database Systems*, 2004.

[6] T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proceedings of the 6th PKDD*, 2002.

[7] L. Chang and I. S. Moskowitz. An integrated framework for database inference and privacy protection. In *Data and Applications Security*, 2000.

[8] 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.

[9] C. Clifton, M. Kantarcioglu, and J. Vaidya. Defining privacy for data mining. In *Natural Science Foundation Workshop on Next Generation Data Mining*, 2002.

[10] 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.

[11] C. Clifton and D. Marks. Security and privacy implications of data mining. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 15–19, 2 1996.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] A. Evfimievski. Randomization in privacy preserving data mining. *SIGKDD Explor. Newsl.*, 4(2), 2002.

[17] 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.

[18] 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.

[19] P. Fule and J. F. Roddick. Detecting privacy and ethical sensitivity in data mining results. In *Proceedings of the 27th conference on Australasian computer science*, 2004.

[20] D. Hand, H. Mannila, and P. Smyh. *Principles of Data Mining.* The MIT Press, 2001.

[21] 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.

[22] M. Kantarcioglu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004.

[23] 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.

[24] D. Knuth. *Fundamental Algorithms.* Addison-Wesley, Reading, Massachusetts, 1997.

[25] K. Muralidhar and R. Sarathy. Security of random data perturbation methods. *ACM Trans. Database Syst.*, 24(4), 1999.

[26] D. E. O'Leary. Knowledge discovery as a threat to database security. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, pages 507–516, Menlo Park, CA, 1991. AAAI Press/MIT Press.

[27] 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.

[28] S. R. M. Oliveira, O. R. Zaiane, and Y. Saygin. Secure association rule sharing. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2004.

[29] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. ICDT '99*, 1999.

[30] J. Pei, J. Han, and J. Wang. Closet+: Searching for the best strategies for mining frequent closed itemsets. In *SIGKDD '03*, 2003.

[31] B. Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.*, 4(2), 2002.

[32] S. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *Proceedings of the 28th VLDB Conference*, 2002.

[33] Y. Saygin, V. S. Verykios, and C. Clifton. Using unknowns to prevent discovery of association rules. *SIGMOD Rec.*, 30(4), 2001.

[34] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 2002.

[35] L. Sweeney. k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 2002.

[36] 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.

[37] 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.

[38] 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.

[39] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemsets mining. In *2nd SIAM International Conference on Data Mining*, 2002.