Exploring the Disjunctive Search Space towards Discovering New Exact Concise Representations for Frequent Patterns

T. Hamrouni\textsuperscript{1,2}, I. Denden\textsuperscript{1}, S. Ben Yahia\textsuperscript{1} and E. Mephu Nguifo\textsuperscript{2,3}

\textsuperscript{1} Department of Computer Sciences, Faculty of Sciences of Tunis University Campus, 1060 Tunis, Tunisia
\{tarek.hamrouni, sadok.benyahia\}@fst.rnu.tn
\textsuperscript{2} CRIL-CNRS, Artois University
Rue de l’université, 62307 Lens cedex, France
\{hamrouni, mephu@cril.univ-artois.fr\}
\textsuperscript{3} LIMOS-CNRS, Blaise Pascal Clermont 2 University
Campus des cèzeaux, 63173 Aubière cedex, France
engelbert.mephuNguifo@univ-bpclermont.fr

October 2007

Abstract. Extracting concise representations seems to be a milestone towards the emerging knowledge extraction field. In fact, it is a quite survival reflex towards providing a manageably-sized and reliable knowledge. Thus, we bashfully witness the emergence of a trend towards extracting concise representations, e.g., closed patterns, non-derivable patterns and essential patterns. The essential pattern-based concise representation presents very interesting properties since it also allows the direct derivation of the disjunctive and negative supports of a pattern, in contrast with almost all known concise representations. In addition, it offers a respectable compactness rates. However, these properties are shadowed by the burden of a positive border maintained in the sake of preserving the exactness property. In this technical report, we introduce a new exact concise representation standing at the crossroads of closure operators and essential patterns. The introduced concise representation required the definition of a new closure operator. Since the latter operator makes possible mapping many elements to a unique one, the new representation permits to drastically reduce the number of handled patterns while avoiding the use of the positive border. It also maintains the interesting properties of essential patterns. Furthermore, this representation makes it possible to bridge the gap with various association rule forms. Carried out experiments show an important lossless reduction of the number of extracted patterns vs. those performed by the concise representations based on frequent closed, (closed) non-derivable and essential patterns, respectively.

Keywords: Frequent pattern, Concise representation, Disjunctive closure operator, Disjunctive closed pattern, Generalized association rule, Itemset.

1 Introduction and motivations

Within the traditional framework for extracting association rules, the conjunctive operator – linking items – got the monopoly. To palliate a hardly manageable number
of frequent patterns extracted from real-life datasets \(^{(1)}\), a growing number of works explored then the conjunctive search space to get out an information lossless nucleus of patterns from which the remaining ones can be derived. Such a reduced set is better known as exact concise representation. Beyond high compactness rates, an exact concise representation makes it possible to guess the frequency status of a pattern and to exactly retrieve its exact support in case of that pattern is (potentially) interesting.

The main concise representations proposed are those based on frequent closed patterns \([1]\), frequent non-derivable patterns \([2]\), frequent closed non-derivable patterns \([3]\) and frequent essential patterns \([4]\). The main originality of the latter concise representation stands in the fact that it mainly explores the disjunctive search space where elements are characterized by their respective disjunctive supports, instead of conjunctive ones. It also heavily relies on the inclusion-exclusion identities \([5]\) to bridge both conjunctive and disjunctive search spaces. Thus, it permits to straightforwardly derive richer forms of associations rules, in which various logical operators may cohabit in the same rule.

In spite of such originality, the concise representation based on frequent essential patterns presents two major disadvantages. First, it is not self-contained in the sense that essential patterns do not make it possible by themselves to decide whether a pattern is frequent or not. Hence, such a set has to be burdened by additional elements belonging to the maximal elements of the umbrella marked by the minimum conjunctive support value, namely \(\text{minsup}\). Second, several essential patterns can characterize the same set of data and, therefore, they present a certain form of redundancy. In this situation, finding a closure operator related to essential patterns would be of paramount importance to get a more reduced concise representation. Indeed, thanks to this operator, many essential patterns will be mapped to the same element within the disjunctive search space. Thus, the obtained representation will be much more compact.

Furthermore, the simultaneous use of essential patterns and disjunctive closed ones can also ease the detection of their respective disjunctive equivalence classes and, hence, the traversal of the disjunctive search space. This can intensively be explored in many applications as done within the conjunctive search space thanks to their correspondences; minimal generators and closed patterns respectively (see \([6]\) for a study). Indeed, these particular patterns are structurally localized within the associated lattice what gives them more semantics, contrary to other patterns numerically retained (like non-derivable patterns) independently from their localization.

Our main contributions are two-fold. First, we show that the set of disjunctive closed patterns, whose associated essential patterns are pruned \(\text{w.r.t. minsup}\), cannot constitute an exact concise representation for frequent patterns. Thus, we lead a thorough study of the finest set of elements that should be added to get the exactness label. Carried out experiments, focusing on the conciseness aspect, show that the proposed concise representation is more compact than the main ones in the literature. Second, we propose an introductory discussion about the derivation of richer forms of association rules thanks to our representation.

The remainder of the paper is organized as follows. The next section recalls the key notions to be used throughout the remainder. Section 3 presents the disjunctive closure operator and details its main properties. Then, new disjunctive closure-based concise representation...
representations are introduced in Section 4, followed by an algorithm allowing to extract the best one among them in Section 5. The empirical evidences about the utility of our approach are provided in Section 6. Section 7 is dedicated to the description of generalized association rules that can be extracted thanks to our representation. We also discuss the main related work in Section 8. The paper ends with a conclusion of our contributions in Section 9.

2 Key notions

In this section, we briefly sketch the key notions used in the paper.

Definition 1. \textbf{(EXTRACTION CONTEXT)} An extraction context is a triplet $K = (O, I, R)$, where $O$ represents a finite set of objects, $I$ is a finite set of items and $R$ is a binary (incidence) relation (i.e., $R \subseteq O \times I$). Each couple $(o, i) \in R$ expresses that the object $o \in O$ contains the item $i \in I$.

Example 1. In the remainder, we will consider the extraction context depicted by Table 1 with $O = \{1, 2, 3, 4, 5, 6, 7\}$ and $I = \{a, b, c, d\}$.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. An extraction context.

A pattern can be characterized by three kinds of supports as sketched by the following definition.

Definition 2. \textbf{[5] (SUPPORTS OF A PATTERN)} Let $K = (O, I, R)$ be an extraction context. We distinguish three kinds of supports associated to a pattern $I$:

- \textbf{Conjunctive support:} $\text{Supp}(I) = \{ o \in O \mid (\forall i \in I, (o, i) \in R) \}$
- \textbf{Disjunctive support:} $\text{Supp}(\lor I) = \{ o \in O \mid (\exists i \in I, (o, i) \in R) \}$
- \textbf{Negative support:} $\text{Supp}(\neg I) = \{ o \in O \mid (\forall i \in I, (o, i) \notin R) \}$

A pattern $I$ is said to be frequent if $\text{Supp}(I)$ is greater than or equal to a user-specified minimum support threshold, denoted $\text{minsup}$ [7]. Since frequent patterns fulfill the order ideal property [8], the supersets of infrequent items will also be infrequent. The set of items $I$ (and consequently the extraction context $K$) will hence be considered as only containing frequent ones. Infrequent items will thus be pruned. The following definition introduces the notion of a frequent essential pattern [4].
**Definition 3. (Frequent Essential Pattern)** Let $K = (O, I, R)$ be an extraction context and $I \subseteq I$. $I$ is an essential pattern iff $\text{Supp}(\lor I) \neq \max\{\text{Supp}(\lor I) \mid i \in I\}$. An essential pattern $I$ is also frequent if $\text{Supp}(I) \geq \text{minsup}$.

It is important to mention that the set of frequent essential patterns, denoted $\mathcal{FEP}_K$, was shown in [4] to be an order ideal in $(2^I, \subseteq)$. The following theorem presents the frequent essential pattern-based concise representation.

**Theorem 1.** [4] The set $\mathcal{FEP}_K$ of frequent essential patterns augmented by $BD^+(FP_K)$ constitutes an exact concise representation for $FP_K$.

Given the respective disjunctive supports of a pattern’s subsets, we are able to derive its conjunctive support using the inclusion-exclusion identities [5]. Furthermore, thanks to the De Morgan’s law, we are even able to derive its negative support. Lemma 1 shows these important properties.

**Lemma 1. (Derivation of Conjunctive and Negative Supports)** Let $I \subseteq I$ be an arbitrary pattern. Its conjunctive and negative supports are respectively derived as follows [5]:

\[
\text{Supp}(I) = \sum_{\emptyset \subseteq I_1 \subseteq I} (-1)^{|I_1| - 1} \text{Supp}(\lor I_1) \quad (1)
\]

\[
\text{Supp}(\neg I) = |O| - \text{Supp}(\lor I) \quad (2)
\]

**Example 2.** Let us consider the extraction context of Table 1. Given the disjunctive supports of $bc$’ subsets, its conjunctive and negative supports are inferred as follows:

- $\text{Supp}(bc) = (-1)^{|bc|} - 1 \text{Supp}(\lor bc) + (-1)^{|b|} - 1 \text{Supp}(\lor b) + (-1)^{|c|} - 1 \text{Supp}(\lor c) = -5 + 3 + 3 = 1$.
- $\text{Supp}(\neg bc) = |O| - \text{Supp}(\lor bc) = 7 - \text{Supp}(\lor bc) = 7 - 5 = 2$.

3. **The Disjunctive Closure Operator and Associated Properties**

Here we detail the main constructs related to the disjunctive closure operator [10, 11], which will make possible to map several essential patterns into a unique element within the disjunctive search space. This operator will be at the roots of the new concise representations introduced in the next section.

Let us start by defining the disjunctive closure operator.

**Definition 4. (Disjunctive Closure Operator)** Let $K = (O, I, R)$ be an extraction context. The disjunctive closure operator $h_d : \mathcal{P}(I) \rightarrow \mathcal{P}(I)$ is defined as follows:

\[
h_d(I) = \{i \in I \mid [(\exists o \in O) ((o, i) \in R)] \land [\forall o_1 \in O) ((o_1, i) \in R) \Rightarrow ((\exists i_1 \in I) ((i_1, i) \in R))]\}
\]

Let us semantically explain the disjunctive closure. Let $I$ be a pattern, $h(I)$ is equal to the maximal set of items which only appear in the objects that contain at least an item of $I$. Actually, Definition 4 gives an explicit expression of the disjunctive closure.
operator, free from the connection operators linking $\mathcal{P}(\mathcal{I})$ and $\mathcal{P}(\mathcal{O})$. Such definition structurally characterizes the disjunctive closure of any pattern $X$ and, hence, allows to straightforwardly compute it from any extraction context. To the best of our knowledge, our work is the first allowing the extraction of a concise representation for frequent patterns based on a disjunctive closure operator.

We will use $\text{DCP}_K$ to denote the set of disjunctive closed patterns extracted from a context $K$. Thanks to the closure operator $h$, the disjunctive search space is partitioned into distinct equivalence classes. In the latter classes, disjunctive closed patterns (resp. essential patterns) are the largest (resp. minimal) elements, w.r.t. set inclusion.

The following proposition allows to establish the relation between the smallest disjunctive closed pattern containing a pattern $I$ and $h(I)$.

**Proposition 1.** Let $I \subseteq \mathcal{I}$. $h_d(I)$ is the smallest disjunctive closed pattern containing $I$: $h_d(I) = \min \subseteq \{ f \in \text{DCP}_K \mid I \subseteq f \}$. Proposition 2 shows the existing link between the disjunctive support of a pattern and that of its closure.

**Proposition 2.** Let $I \subseteq \mathcal{I}$. $\text{Supp}(\bigvee I) = \text{Supp}(\bigvee h_d(I))$.

The following proposition makes possible to deduce the disjunctive closure of a pattern using the disjunctive closure of one of its subsets.

**Proposition 3.** Let $X \subseteq \mathcal{I}$ and $Y \subseteq \mathcal{I}$ be two patterns. We then have:

$$(X \subseteq Y \subseteq h(X)) \Rightarrow (h(Y) = h(X)).$$

Proposition 4 establishes the link between disjunctive closed patterns and frequent essential patterns.

**Proposition 4.** Let $I \subseteq \mathcal{I}$ and $\mathcal{FP}_K$ be the set of frequent patterns. We then have:

$$(I \in \mathcal{FP}_K) \Rightarrow (\exists f \in \text{DCP}_K \text{ and } I_1 \in \mathcal{FEP}_K \text{ s.t. } h(I_1) = h(I) = f \text{ and } I_1 \subseteq I).$$

In the remainder of the paper, we denote by $\mathcal{EDCP}_K$ (stands for Essential Disjunctive Closed Patterns) the subset of $\text{DCP}_K$ whose elements have for generators at least a frequent essential pattern. Please note that the disjunctive closures of the patterns belonging to $BD^+ (\mathcal{FP}_K)$ are obviously contained in $\mathcal{EDCP}_K$ (cf. Proposition 4).

Once we have defined the disjunctive closure operator and presented its main properties, we can then introduce the exact concise representations based on it.

4 New exact disjunctive closure-based concise representations

4.1 Problem settings

In [10], the authors applied the disjunctive closure operator on frequent essential patterns (i.e., towards extracting $\mathcal{EDCP}_K$) to generate a new exact concise representation that is always smaller, in size terms, than that based on essential patterns. Unfortunately, the proposed concise representation has been shown to miss some cases and is hence not an exact one. For example, let us consider the context shown in Table 1. For
The disjunctive support of any pattern, whose size is less than 2, can be correctly retrieved from \( EDCP_K \) (using Proposition 2). However, when we retrieve the disjunctive support of \( bcd \) from \( EDCP_K \), we will assign to \( bcd \) the disjunctive support of the smallest element (w.r.t. set inclusion) in \( EDCP_K \) subsuming it, i.e., \( abcd \). However, its actual disjunctive support is equal to 6 and not 7 as for \( abcd \). Consequently, when computing the conjunctive support of \( bcd \) using Lemma 1, we obtain \( \text{Supp}(bcd) = 1 \), while the actual conjunctive support of \( bcd \) is equal to 0. This flaw is due to the pruning of the disjunctive closed pattern \( bcd \) whose unique generator is the infrequent essential pattern \( bcd \) (i.e., itself).

A straightforward way to palliate the aforementioned problem is to add other elements to \( EDCP_K \). These elements depend, on the one hand, on the way we will retrieve the conjunctive support of frequent patterns candidates and, on the other hand, on our characterization of the patterns whose supports will be computed in a wrong way if such elements will not be added.

The following sections introduce a thorough discussion on how to palliate such insufficiency. Since the conjunctive support of a pattern \( X \) can be computed only if the disjunctive supports of all its subsets are available, \( X \) will be treated only if all its proper subsets were already treated and proved to be frequent patterns \(^2\).

4.2 First concise representation

The following theorem introduces the first disjunctive closure-based concise representation, namely \( EDCP_K \cup BD^+(FP_K) \).

**Theorem 2.** The set \( EDCP_K \cup BD^+(FP_K) \) is an exact concise representation for \( FP_K \).

**Proof.** When we add \( BD^+(FP_K) \) to \( EDCP_K \), we can easily check the frequency status of a pattern. If it is frequent, we can correctly retrieve its conjunctive support from \( EDCP_K \) thanks to both Proposition 4 and Lemma 1. □

Despite the fact that this representation allows to exactly retrieve frequent patterns and their conjunctive supports, it suffers from the following drawbacks:

1. **Heterogeneity:** \( EDCP_K \) provides disjunctive closed patterns while \( BD^+(FP_K) \) contains maximal conjunctive frequent patterns. The heterogeneity is induced, on the one hand, by the two different supports (conjunctive and disjunctive ones) characterizing each set and, on the other hand, by the distinct kinds of their elements. It is worth mentioning that the heterogeneity requires the membership storage of each pattern to each set composing the concise representation.

2. **Oversize:** Especially for dense contexts, the size of \( BD^+(FP_K) \) constitutes a highly sized burden.

\(^2\) This can be ensured thanks to a level-wise (or a depth-first right to left) traversal of the search space.
4.3 Second concise representation

Through the following theorem, we introduce another concise representation based on the disjunctive closure.

**Theorem 3.** The set $\mathcal{EDCP}_K \cup \mathcal{FEP}_K$ is an exact concise representation for $\mathcal{FP}_K$.

**Proof.** For a pattern $X$, if it exists $Y$ s.t. $Y \in \mathcal{FEP}_K$ verifying $Y \subseteq X \subseteq h(Y)$, we can deduce from Proposition 3 that $h(X) = h(Y)$ and we can ensure a correct computation of the conjunctive support of $X$. If it is not the case, we can deduce that $X$ contains an infrequent essential pattern. Hence, $X$ is infrequent (since the frequent patterns fulfill the order ideal property [7]). □

This second representation reduces the heterogeneity problem since it is only induced by the elements having different structural properties. Indeed, both disjunctive closed patterns and essential patterns are characterized by their disjunctive support. Disjunctive closed patterns (resp. essential patterns) are the largest (resp. minimal) elements w.r.t. set inclusion within the equivalence classes induced by the closure operator $h$. However, the size of $\mathcal{FEP}_K$ can be larger than that of $\mathcal{BD}^+ (\mathcal{FP}_K)$ for dense and, especially, sparse contexts.

4.4 Third concise representation

To palliate the drawbacks of both aforementioned concise representations, we present a third one. Let us begin by defining the set $\mathcal{ADCP}_K$ (stands for Added Disjunctive Closed Patterns) ensuring the new representation to be an exact one. The exactness of this representation is proved through Theorem 4.

**Definition 5.** Let $\mathcal{EP}_K$ be the set of all essential patterns that can be extracted from a context $K$ and $\mathcal{BD}^-(\mathcal{FEP}_K) = \min \subseteq \{Y \in \mathcal{P}(I) \setminus \mathcal{FEP}_K\}$. The set $\mathcal{ADCP}_K$ is defined as follows: $\mathcal{ADCP}_K = \{h(X) \mid (X \in \mathcal{BD}^-(\mathcal{FEP}_K) \cap \mathcal{EP}_K) \land ((-1)^{|X|} = -1) \land (\forall X' \subseteq I, h(X') = h(X) \Rightarrow \text{Supp}(X') < \text{minsup})\}$.

Roughly speaking, $\mathcal{ADCP}_K$ is the set of the disjunctive closed patterns, generated by infrequent essential patterns of odd sizes and belonging to the negative border of frequent essential patterns, which have all their corresponding essential patterns infrequent. For example, for $\text{minsup} = 1$, $bcd \in \mathcal{ADCP}_K$ since its unique essential pattern is $bcd$ which is infrequent and of odd size (since equal to 3). In the remainder, the concise representation $\mathcal{EDCP}_K \cup \mathcal{ADCP}_K$ will be denoted $\mathcal{DCP}_K$.

**Theorem 4.** The set $\mathcal{DCP}_K$ is an exact concise representation for $\mathcal{FP}_K$.

**Proof.** Only infrequent patterns can have their conjunctive supports wrongly computed (cf. Proposition 4). Since we process the patterns by an ascending order of their size, we can avoid the computation errors of conjunctive supports if we detect the minimal infrequent patterns for which the computed disjunctive supports may be erroneous \(^3\). Let $X$ be a minimal pattern whose the disjunctive support may be wrongly computed. A careful scrutiny of $X$ points out that it is an essential pattern fulfilling two properties:

\(^3\) The computation of their conjunctive supports is inevitable since we cannot beforehand predict whether they are frequent or not.
1. $X$ belongs to $\mathcal{BD}^-(\mathcal{FE}_P_K)$.
2. $h(X)$ does not contain any frequent essential pattern $X'$ s.t. $h(X') = h(X)$ (otherwise, we have the disjunctive closure of $X$ in $\mathcal{EDCP}_K$ and, hence, its disjunctive support).

Since $h(X)$ does not contain any frequent essential pattern, its disjunctive closure is not present in $\mathcal{EDCP}_K$. Let $Sh(X)$ be the smallest disjunctive closed pattern in $\mathcal{EDCP}_K$ containing $X$. If $Sh(X)$ does not exist then $X$ is immediately guessed as infrequent. Otherwise, from equation (1) of Lemma 1, we have:

$$\text{Supp}(X) = \sum_{\emptyset \subset X' \subseteq X} (-1)^{|X'| - 1}\text{Supp}(\vee X') = (-1)^{|X| - 1}\text{Supp}(\vee X) + \sum_{\emptyset \subset X' \subseteq X} (-1)^{|X'| - 1}\text{Supp}(\vee X').$$

Two cases have to be distinguished:

**a.** $|X|$ is even: $\text{Supp}(X) = -\text{Supp}(\vee X) + \sum_{\emptyset \subset X' \subseteq X} (-1)^{|X'| - 1}\text{Supp}(\vee X') < \text{minsup}$.

Since $X \subset Sh(X)$, $\text{Supp}(\vee X) \leq \text{Supp}(\vee Sh(X))$. Hence, $-\text{Supp}(\vee Sh(X)) + \sum_{\emptyset \subset X' \subseteq X} (-1)^{|X'| - 1}\text{Supp}(\vee X') \leq \text{Supp}(\vee X) + \sum_{\emptyset \subset X' \subseteq X} (-1)^{|X'| - 1}\text{Supp}(\vee X') < \text{minsup}$. This inequality points out that even with an incorrect disjunctive support of $X$, we can detect that $X$ is infrequent if $|X|$ is even.

**b.** $|X|$ is odd: By applying the same process as for the previous case, we are not able to detect in all the cases the frequency status of $X$. Indeed, in this case, $(-1)^{|X| - 1}\text{Supp}(\vee X) = ±\text{Supp}(\vee X)$ and not $-\text{Supp}(\vee X)$ as in the case where $|X|$ is even. Hence, it is necessary to add $h(X)$ to $\mathcal{EDCP}_K$ in order to ensure the correct detection of the status of $X$.

Thus, the set $\mathcal{DCP}_{K, rep}$ is an exact concise representation for $\mathcal{FP}_K$. □

The proof of Theorem 4 can be treated as a naive algorithm for deriving frequent patterns and their associated supports.

It is important to mention that in $\mathcal{ADCP}_K$, we did not consider the disjunctive closures of infrequent non-essential patterns belonging to $\mathcal{BD}^-(\mathcal{FE}_P_K)$ since they are already included in $\mathcal{EDCP}_K$ (cf. Proposition 3). On the other hand, we can further reduce the cardinality of $\mathcal{ADCP}_K$ by pruning its elements which are not covered by any disjunctive closed pattern of $\mathcal{EDCP}_K$. Hereafter, $\mathcal{ADCP}_K$ will denote the set presented in Definition 5, from which we delete the elements not covered by those of $\mathcal{EDCP}_K$.

The next theorem states the correctness of the regeneration process of frequent patterns after this pruning [12].

**Theorem 5.** After the reduction, the set $\mathcal{DCP}_{K, rep}$ is still an exact concise representation for $\mathcal{FP}_K$.

**Proof.** The proof is based on that of Theorem 4 and on the following fact: a pattern $X$ is eligible to be frequent only if it is covered by a pattern of $\mathcal{EDCP}_K$. Indeed, as stated in Proposition 4, if $X$ is not covered by any element of $\mathcal{EDCP}_K$, then we can directly affirm that $X$ is infrequent. Therefore, the set $\mathcal{DCP}_{K, rep}$ is still an exact concise representation for $\mathcal{FP}_K$. □
In addition to the exact retrieval of frequent patterns as well as their various supports, $DCP_K$ offers three other main properties [12]:

1. **Homogeneity**: $DCP_K$ overcomes the heterogeneity problem since it only involves disjunctive closed patterns (vs. $\mathcal{FEP}_K \cup \mathcal{BD}^+ (\mathcal{FP}_K)$). Its elements hence have the same structural properties. Indeed, they are the top elements of their associated equivalence classes within the disjunctive search space. This ensures the homogeneity of the representation since all its elements are also provided with the same kind of support; the disjunctive one.

2. **Small size**: In [10], the size of $EDCP_K$ is shown to be significantly smaller than those of the best known concise representations. In addition, the size of $ADCP_K$ is very small since its elements must fulfill many easy-to-check constraints. Hence, the size of $DCP_K$ will be, in most cases, smaller than those of the other representations.

3. **Low regeneration cost**: It is worth mentioning that our concise representation allows retrieving the conjunctive support faster than from (closed) non-derivable patterns [2, 3]. Indeed, for a pattern $X$ s.t. $|X| = n$, the retrieval process of $\text{Supp}(X)$ from these representations requires the costly evaluation of $2^n$ deduction rules based on Bonferroni-inequalities [13]. The computation cost for inferring supports is then awfully high. While the retrieval of $\text{Supp}(X)$ from our concise representation only needs to evaluate a unique inclusion-exclusion identity. Furthermore, it allows the straightforward retrieval of the disjunctive and negative supports of frequent patterns.

## 5 The $DCPRMINER$ algorithm

We describe a generate-and-test algorithm, called $DCPRMINER$ (4), allowing the extraction of the disjunctive closure-based concise representation $DCP_K$. The notations used in this algorithm are summarized in Table 2. The pseudo-code of $DCPRMINER$ is given by Algorithm 1.

<table>
<thead>
<tr>
<th>$C_i$ (resp. $L_i$)</th>
<th>the set of candidate (resp. frequent) essential patterns of size $i$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_i$</td>
<td>a pattern of size $i$.</td>
</tr>
<tr>
<td>$X_i,\text{Disj}Clos$</td>
<td>the disjunctive closure of $X_i$.</td>
</tr>
<tr>
<td>$X_i,\text{Disj}los$</td>
<td>the set of items that necessarily do not belong to the disjunctive closure of $X_i$.</td>
</tr>
<tr>
<td>$X_i,\text{Conj}Supp$ (resp. $X_i,\text{Disj}Supp$)</td>
<td>the conjunctive (resp. disjunctive) support of $X_i$.</td>
</tr>
</tbody>
</table>

Table 2. The notations used in the $DCPRMINER$ algorithm.

The $\text{COMPUTE\_DISJUNCTIVE\_CLOSED\_PATTERNS}$ procedure makes possible the computation of the disjunctive and conjunctive supports as well as the disjunctive closures of candidates of size $i$. Then, only the required disjunctive closed patterns will be added to $DCP_K$ (cf. the previous section). During the process, a candidate having a subset which is not a frequent essential pattern is withdrawn, since the set $\mathcal{FEP}_K$ is minimized.

---

4 $DCPRMINER$ stands for Disjunctive Closed Pattern Representation Miner.
Algorithm 1: DCPR_Miner

\[ \text{Data: An extraction context } k = (O, I, R), \text{ and the minimum support threshold } \minsup. \]
\[ \text{Results: The concise representation } DCP_Kep. \]

1 Begin
2 \[ DCP_Kep = \emptyset; i = 1; C_i = I; \]
3 While \((C_i \neq \emptyset)\) do
4 \[ \text{COMPUTE_DISJUNCTIVE_CLOSED_PATTERNS}(K, \minsup, C_i, L_i, DCP_Kep); \]
5 \[ i = i + 1; \]
6 Return \(DCP_Kep; \]
7 End

9 Procedure \text{COMPUTE_DISJUNCTIVE_CLOSED_PATTERNS}(K, \minsup, C_i, L_i, DCP_Kep)\]
10 Begin
11 Foreach \((O \in O)\) do
12 Foreach \((X_i \in C_i)\) do
13 \[ \Omega = X_i \cap I; \]
14 If \((\Omega = \emptyset)\) then
15 \[ X_i.\text{Disj} = X_i.\text{Disj} - \bigcup I; \]
16 Else
17 \[ X_i.\text{Conj} = X_i.\text{Conj} + 1; \]
18 If \((\Omega = X_i)\) then
19 \[ X_i.\text{Supp} = X_i.\text{Conj} + 1; \]
20 Foreach \((X_i \in C_i)\) do
21 If \((X_i.\text{Conj} \geq \minsup)\) then
22 \[ L_i = L_i \cup \{X_i\}; \]
23 \[ X_i.\text{Disj} = X_i.\text{Disj} - \bigcup X_i.\text{Disj}; \]
24 \[ DCP_Kep = DCP_Kep \cup \{(X_i.\text{Disj}, X_i.\text{Supp})\}; \]
25 Else If \((i \text{ is odd})\) then
26 \[ P \in BD^i(FEP_K) \text{ and } X_i.\text{Conj} < \minsup; \]
27 \[ X_i.\text{Disj} = X_i.\text{Disj} - \bigcup X_i.\text{Disj}; \]
28 \[ DCP_Kep = DCP_Kep \cup \{(X_i.\text{Disj}, X_i.\text{Supp})\}; \]
30 End

is an order ideal. Proposition 3 also allows pruning a candidate which is included in the disjunctive closure of one of its subsets, since it is necessarily not an essential pattern. These pruning strategies are ensured by a slight modification of the \text{APRIORI-GEN} procedure (cf. line 5 in Algorithm 1). The latter procedure is the standard one of the \text{APRIORI} algorithm \cite{7} to generate new candidates of size \((i + 1)\) starting from the retained ones in \(L_i\). In fact, the set \text{APRIORI-GEN}(\(L_i\)) equals \(\{X \subseteq I \mid \|X\| = i + 1, \forall Y \subseteq X \text{ s.t. } \|Y\| = i; (Y \in L_i) \lor (X \not\subseteq Y.\text{Disj} \cap \text{Disj})\}\).

As aforementioned, the regeneration of all frequent patterns starting from \(DCP_Kep\) can straightforwardly be done in a level-wise manner that starts from 1-patterns, 2-patterns and so forth. The corresponding pseudo-code is omitted here.

6 Experimental results

In this section, our objective is to show, through various experiments, that in most cases the size of our concise representation is smaller than those of the exact concise representations based on frequent closed patterns, frequent (closed) non-derivable patterns and frequent essential patterns. This is done in the most critical cases, i.e., for strongly
correlated datasets. Indeed, within such datasets, the ratio between the cardinality of the frequent pattern set and those of concise representations is high. Thus, we are in the most interesting cases. Moreover, equivalence classes extracted from sparse datasets are often reduced to the associated generators and cannot be compacted anymore. The number of extracted frequent patterns is hence small even for low minsup values. This makes the size reduction rates brought by concise representations meaningless in such datasets while paying for the overhead of useless tests and computations required by a concise representation’s extraction. For example, the closed non-derivable patterns, supposed to be the smallest exact concise representation, cannot considerably reduce the number of frequent patterns (e.g. in the T10I4D100K dataset and for minsup = 0.02%, the number of frequent closed non-derivable patterns is equal to 102,869 while the number of frequent patterns is equal to 129,876) [3].

The characteristics of tested benchmark datasets are summarized in Table 3 (5). All experiments were carried out on a PC equipped with a 1.73GHz Centrino Duo Core and 2GB of main memory, and running the Linux version Fedora Core 6 (with 2GB of swap memory). Obtained results are shown in Table 4 (6). The abbreviation “FPK” (resp. “FCPKrep”, “NDPKrep”, “CNDPKrep”, and “FEPKrep”) is used to stand for the set of frequent patterns (resp. frequent closed, frequent non-derivable, frequent closed non-derivable and frequent essential pattern-based concise representation). It is important to note that in [3], the authors have chosen a specific interval of minsup for each dataset to extract closed non-derivable patterns. But beyond these intervals, we noticed that the program allowing the extraction of CNDPKrep comes to an end with an execution error. Therefore, we use the symbol “-” to designate a case where an execution error occurred. At a glance, we can also deduce the following assertions:

1. Even for high values of minsup, the size of the concise representation DCPKrep is considerably reduced compared to that of frequent patterns.
2. Especially for CHESS, CONNECT and PUMSB datasets, the size of DCPKrep is significantly reduced compared to those of the concise representations based on frequent closed patterns, frequent (closed) non-derivable patterns and frequent essential patterns.

Table 3. Dataset characteristics.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Number of items</th>
<th>Number of objects</th>
<th>Average size of objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT</td>
<td>129</td>
<td>67,557</td>
<td>43</td>
</tr>
<tr>
<td>MUSHROOM</td>
<td>119</td>
<td>8,124</td>
<td>23</td>
</tr>
<tr>
<td>CHESS</td>
<td>75</td>
<td>3,196</td>
<td>37</td>
</tr>
<tr>
<td>PUMSB</td>
<td>2,113</td>
<td>49,046</td>
<td>74</td>
</tr>
<tr>
<td>PUMSB*</td>
<td>2,088</td>
<td>49,046</td>
<td>50.50</td>
</tr>
</tbody>
</table>

5 These datasets are available at: http://fimi.cs.helsinki.fi/data.
6 For more details, interested readers are referred to [14].
7 To extract frequent (closed) patterns, we have used the source codes available at: http://fimi.cs.helsinki.fi/src.
3. It is easily observable that $DCP_{K, rep}$ is less sensible to the variation of $\text{minsup}$ than the other concise representations.

4. In some cases, the size of $DCP_{K, rep}$ is quite greater than the size of the other concise representations (e.g., MUSHROOM for $\text{minsup} = 1\%$).

Unlike frequent closed patterns and frequent (closed) non-derivable patterns, the introduced exact concise representation $DCP_{K, rep}$ allows to directly retrieve the disjunctive and negative supports of frequent patterns in addition to their respective conjunctive supports. This interesting property allows to derive many forms of generalized association rules investigated in the following section.

7 Generalized association rules

Generalized association rules grasped the interest of many researchers because of their usefulness in many applications, especially disjunctive association rules. Indeed, the latter rules were considered for two main purposes. On the one hand, they were used to define exact concise representations (e.g., (generalized) disjunction-free patterns/generators, please see [15] for a survey). On the other hand, they were exploited in other works (e.g., [16, 17]) to provide users with new forms of association rules.

Let $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathcal{I}$ and $Y = \{y_1, y_2, \ldots, y_m\} \subseteq \mathcal{I}$ be two patterns s.t. $X \cap Y = \emptyset$.

Since we can retrieve the disjunctive support of any pattern whose the disjunctive closure is present in $DCP_{K, rep}$, we can derive the following dual rules with their exact support and confidence values:

1. $\mathcal{R}_1: x_1 \vee x_2 \vee \ldots \vee x_n \rightarrow y_1 \vee y_2 \ldots \vee y_m$
2. $\mathcal{R}_2: \neg x_1 \land \neg x_2 \land \ldots \land \neg x_n \rightarrow \neg y_1 \land \neg y_2 \land \ldots \land \neg y_m$

The support and the confidence of $\mathcal{R}_1$ are given by applying the inclusion-exclusion identities:

$$\text{Supp}(\mathcal{R}_1) = \text{Supp}(x_1 \vee x_2 \vee \ldots \vee x_n) \land (y_1 \vee y_2 \vee \ldots \vee y_m)) = \text{Supp}(x_1 \vee x_2 \vee \ldots \vee x_n) + \text{Supp}(y_1 \vee y_2 \vee \ldots \vee y_m) - \text{Supp}(x_1 \vee x_2 \vee \ldots \vee x_n \vee y_1 \vee y_2 \vee \ldots \vee y_m).$$

$$\text{Conf}(\mathcal{R}_1) = \frac{\text{Supp}(x_1 \vee x_2 \vee \ldots \vee x_n \land (y_1 \vee y_2 \vee \ldots \vee y_m))}{\text{Supp}(x_1 \vee x_2 \vee \ldots \vee x_n) + \text{Supp}(y_1 \vee y_2 \vee \ldots \vee y_m) - \text{Supp}(x_1 \vee x_2 \vee \ldots \vee x_n \vee y_1 \vee y_2 \vee \ldots \vee y_m)}.$$

Using the De Morgan law (cf. Formula (2) in Lemma 1), we can straightforwardly deduce the support and the confidence of $\mathcal{R}_2$:

$$\text{Supp}(\mathcal{R}_2) = \text{Supp}(-x_1 \land -x_2 \land \ldots \land -x_n \land -y_1 \land -y_2 \land \ldots \land -y_m) = \lvert \mathcal{O} \rvert$$

$$\text{Conf}(\mathcal{R}_2) = \frac{\text{Supp}(-x_1 \land -x_2 \land \ldots \land -x_n \land -y_1 \land -y_2 \land \ldots \land -y_m)}{\lvert \mathcal{O} \rvert - \text{Supp}(x_1 \vee x_2 \vee \ldots \vee x_n \vee y_1 \vee y_2 \vee \ldots \vee y_m)}.$$
| minsup (%) | \( |FP_K| \) | \( |FCP_K, rep| \) | \( |NPDP_K, rep| \) | \( |CNPDP_K, rep| \) | \( |FEP_K, rep| \) | \( |DCP_K, rep| \) |
|-----------|--------|--------|--------|--------|--------|--------|
| 10        | 27,128 | 3,487  | 199    | 177    | 398    | 22     |
| 80        | 533,976| 15,108 | 348    | 305    | 977    | 83     |
| 70        | 4,129,840 | 35,876 | 545    | 491    | 1,710  | 161    |
| 60        | 21,250,672 | 68,344 | 894    | -      | 2,925  | 293    |
| 50        | 88,173,344 | 130,112 | 1,397  | -      | 5,063  | 589    |
| 40        | 339,915,256 | 239,373 | 2,066  | -      | 8,161  | 1,062  |
| 30        | 1,331,673,368 | 460,357 | 3,221  | -      | 14,083 | 1,986  |
| 20        | 6,157,510,380 | 1,483,199 | 7,574  | -      | 39,203 | 5,514  |
| 10        | -     | 8,035,412 | 29,167 | -      | 153,928 | 22,400 |
| 5         | -     | 28,384,574 | 91,050 | -      | 488,398 | 82,738 |
| 40        | 566   | 140    | 146    | 117    | 151    | 91     |
| 30        | 2,736 | 427    | 329    | 275    | 310    | 213    |
| 20        | 53,584 | 1,197  | 1,143  | 731    | 1,258  | 941    |
| 10        | 574,432 | 4,885  | 4,347  | 2,655  | 6,530  | 5,457  |
| 5         | 3,755,512 | 12,843 | 11,569 | 6,546  | 24,407 | 20,554 |
| 4         | 5,131,853 | 16,733 | 14,382 | 8,249  | 28,316 | 25,160 |
| 3         | 9,987,059 | 22,231 | 19,426 | 10,824 | 50,552 | 43,791 |
| 2         | 23,596,651 | 31,768 | 28,253 | -      | 92,267 | 76,435 |
| 1         | 90,751,402 | 51,640 | 48,719 | -      | 237,242 | 197,055 |
| 90        | 623   | 499    | 95     | 93     | 118    | 43     |
| 80        | 8,228 | 5,084  | 281    | 276    | 467    | 150    |
| 70        | 48,732 | 23,893 | 684    | 669    | 1,482  | 420    |
| 60        | 254,945 | 98,395 | 1,596  | 1,567  | 4,637  | 917    |
| 50        | 1,272,933 | 369,451 | 3,425  | 3,341  | 14,272 | 1,971  |
| 40        | 6,439,703 | 1,361,158 | 7,185  | 7,015  | 44,027 | 4,118  |
| 30        | 37,282,963 | 5,316,468 | 15,147 | -      | 147,777 | 8,824  |
| 20        | 289,154,814 | 22,808,625 | 34,761 | -      | 542,540 | 22,517 |
| 10        | 4,553,779,005 | 123,243,073 | 98,664 | -      | 2,453,744 | 76,198 |
| 10        | 2,608 | 1,467  | 586    | 460    | 788    | 318    |
| 85        | 20,535 | 8,514  | 1,792  | 1,147  | 2,628  | 647    |
| 80        | 142,157 | 33,308 | 3,642  | 2,136  | 6,251  | 1,079  |
| 75        | 672,630 | 101,083 | 5,549  | 3,171  | 11,707 | 1,469  |
| 70        | 2,698,265 | 241,259 | 7,875  | 4,564  | 18,318 | 2,143  |
| 65        | 8,099,128 | 496,199 | 12,609 | 7,575  | 28,591 | 3,551  |
| 60        | 19,529,992 | 1,074,628 | 21,323 | 12,081 | 54,644 | 5,550  |
| 55        | 48,790,118 | 2,729,796 | 32,121 | -      | 118,121 | 8,144  |
| 50        | 165,903,541 | 7,121,265 | 47,764 | -      | 232,581 | 11,551 |
| 40        | 3,474,538,312 | 44,434,213 | 149,211 | -      | 865,727 | 35,575 |
| 30        | -     | 698,928,543 | 470,828 | -      | 5,641,149 | 112,429 |
| 20        | -     | 7,453,502,677 | -     | -      | 32,404,484 | 500,505 |

| PUMSB * | 70 | 30 | 18 | 21 | 18 | 23 | 17 |
| 60 | 168 | 69 | 76 | 69 | 77 | 71 |
| 50 | 680 | 249 | 277 | 238 | 275 | 255 |
| 40 | 27,355 | 2,611 | 1,884 | 1,595 | 2,028 | 1,593 |
| 30 | 432,699 | 16,155 | 7,926 | 6,596 | 10,400 | 7,555 |
| 20 | 7,122,280,454 | 122,202 | 49,642 | - | 102,274 | 56,587 |
| 10 | - | 1,512,866 | 450,855 | - | 1,414,102 | 527,968 |

Table 4. Size of the different concise representations for benchmark datasets.
Since our concise representation allows the correct retrieval of the conjunctive support of all frequent patterns, we can also quantify the support and the confidence of the rule \( x_1 \land x_2 \land \ldots \land x_n \rightarrow y_1 \land y_2 \land \ldots \land y_m \) using their usual formulae.

In general, the following forms of generalized association rules can be extracted from our concise representation (8):

1. \( R_1: x_1 \land x_2 \land \ldots \land x_n \rightarrow y_1 \land y_2 \land \ldots \land y_m. \)
2. \( R_2: x_1 \land x_2 \land \ldots \land x_n \rightarrow y_1 \lor y_2 \lor \ldots \lor y_m. \)
3. \( R_3: x_1 \land x_2 \land \ldots \land x_n \rightarrow \neg y_1 \land \neg y_2 \land \ldots \land \neg y_m. \)
4. \( R_4: x_1 \lor x_2 \lor \ldots \lor x_n \rightarrow y_1 \land y_2 \land \ldots \land y_m. \)
5. \( R_5: x_1 \lor x_2 \lor \ldots \lor x_n \rightarrow y_1 \lor y_2 \lor \ldots \lor y_m. \)
6. \( R_6: x_1 \lor x_2 \lor \ldots \lor x_n \rightarrow \neg y_1 \land \neg y_2 \land \ldots \land \neg y_m. \)
7. \( R_7: \neg \neg x_1 \land \neg \neg x_2 \land \ldots \land \neg \neg x_n \rightarrow y_1 \land y_2 \land \ldots \land y_m. \)
8. \( R_8: \neg \neg x_1 \land \neg \neg x_2 \land \ldots \land \neg \neg x_n \rightarrow y_1 \lor y_2 \lor \ldots \lor y_m. \)
9. \( R_9: \neg \neg x_1 \land \neg \neg x_2 \land \ldots \land \neg \neg x_n \rightarrow \neg y_1 \land \neg y_2 \land \ldots \land \neg y_m. \)

These generalized association rules bring richer information to the user than those presented in the literature since they involve various forms of Boolean connectors in both premise and conclusion parts, respectively. In this respect, it is worth noting that an interesting case of the rule \( R_5 \) occurs when \( \{y_1, y_2, \ldots, y_m\} \) is an essential pattern and \( \{x_1, x_2, \ldots, x_n\} \bigcup \{y_1, y_2, \ldots, y_m\} = h(\{y_1, y_2, \ldots, y_m\}) \). Indeed, in such case, this rule is an exact one since its confidence is equal to 1:

\[
\text{Conf}(R_5) = \frac{\text{Supp}(x_1, x_2, \ldots, x_n) \cup \text{Supp}(y_1, y_2, \ldots, y_m) - \text{Supp}(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)}{\text{Supp}(x_1, x_2, \ldots, x_n) \cup \text{Supp}(y_1, y_2, \ldots, y_m) - \text{Supp}(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)} = 1 \quad (\text{cf. Proposition 2}).
\]

Moreover, \( R_9 \) will have a maximal premise and a minimal conclusion, which is the most interesting case for disjunctive association rules (16).

The respective support and confidence values of each rule are computed using their classical formulae:

\[
\text{Supp}(X \rightarrow Y) = \text{Supp}(X \land Y) \quad \text{and} \quad \text{Conf}(X \rightarrow Y) = \frac{\text{Supp}(X \rightarrow Y)}{\text{Supp}(X)} = \frac{\text{Supp}(X \rightarrow Y)}{\text{Supp}(X)}.
\]

From the confidence formula, it is obvious that if we can compute \( \text{Supp}(X \rightarrow Y) \), the evaluation of the confidence will be straightforward since \( \text{Supp}(X) \) can be directly computed from our exact concise representation. Here we will present an overview of the process by which we are able to retrieve the supports of association rules presented above. To compute these latter supports, Lemma 1 will be useful in addition to the two following formulæ:

1. \( \text{Supp}(x_1 \land x_2 \land \ldots \land x_n \land (y_1 \lor y_2 \lor \ldots \lor y_m)) = \text{Supp}(x_1 \land x_2 \land \ldots \land x_n) - \text{Supp}(x_1 \land x_2 \land \ldots \land x_n \land \neg y_1 \land \neg y_2 \land \ldots \land \neg y_m) \) (inclusion-exclusion identity) [5].
2. \( \text{Supp}(x_1 \land x_2 \land \ldots \land x_n \land \neg y_1 \land \neg y_2 \land \ldots \land \neg y_m) = \sum_{Y' \subseteq \{y_1, \ldots, y_m\}} (-1)^{|Y'|}\text{Supp}(x_1 \land \ldots \land x_n). \)

Although they can also be extracted, association rules with premises and/or conclusions under the conjunctive or disjunctive normal form are not treated here.
The respective supports of generalized association rules are then computed as follows:

1. \( \text{Supp}(R_1) = \text{Supp}(x_1 \land x_2 \land \ldots \land x_n \land y_1 \land y_2 \land \ldots \land y_m) \).
2. \( \text{Supp}(R_2) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} \text{Supp}(x_1 \land x_2 \land \ldots \land x_n) \).
3. \( \text{Supp}(R_3) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} (-1)^{|Y'|} \text{Supp}(x_1 \land x_2 \land \ldots \land x_n \land Y') \).
4. \( \text{Supp}(R_4) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} (-1)^{|X'|} \text{Supp}(X' \land y_1 \land \ldots \land y_m) \).
5. \( \text{Supp}(R_5) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} (-1)^{|X'|} \text{Supp}(X' \land y_1 \land \ldots \land y_m) \).
6. \( \text{Supp}(R_6) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} (-1)^{|Z'|} \text{Supp}(Z') \).
7. \( \text{Supp}(R_7) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} (-1)^{|X'|} \).
8. \( \text{Supp}(R_8) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} (-1)^{|X'|} \).
9. \( \text{Supp}(R_9) = \sum_{(y_1 \lor y_2 \lor \ldots \lor y_m)} (-1)^{|X'|} \).

8. Related work

In this section, we review works closely related to the disjunctive closed patterns and to generalized association rules.

First of all, let us make an alignment between the disjunctive search space and the conjunctive one. We will hence find that an essential pattern is the mapping of the concept of minimal generator [19] (aka key pattern [20] and free-set [21] in the literature) when the conjunctive search space is considered. While the disjunctive closed patterns are the mapping of conjunctive ones [1] (9).

\[^9\] A critical survey on algorithms extracting closed patterns within the conjunctive search space can be found in [22].
The concepts of essential and disjunctive closed patterns can be considered as particular cases of composite items [23] where the disjunction of items is used to compose new items, the composite ones. This is an attempt towards making useful infrequent items in some applications. For example, consider the context of Table 1 and let minsup = 4, b and c are hence infrequent items since their support is equal to 3. Nevertheless, the support of b ∨ c is equal to 5 and, hence, Supp(b ∨ c) ≥ minsup. b ∨ c will be considered as a new item (a composite one) even if, actually it is composed of two items. It will be used during the mining process since it is frequent what makes b and c useful.

It is important to make the link between our work and that of Zhao et al. Indeed, in [24], the authors proposed connection operators to link $P(I)$ and $P(O)$ for the case of disjunctive Boolean expressions, called OR-clauses. Nevertheless, their definition of the operator linking $P(O)$ to $P(I)$ depends on that ensuring the opposite direction and was not independently given. The following definition presents the proposed operators [24]:

**Definition 6.** Let $K = (O, I, R)$ be an extraction context, $P(I)$ the set of all possible OR-clauses over $I$, and $P(O)$ the power-set of objects. Let $X \in P(I)$ and $Y \in P(O)$.

Given two partially ordered sets ($P(I)$, $\subseteq$) and ($P(O)$, $\subseteq$), the following operators form a Galois connection over $P(I)$ and $P(O)$:

\[
\phi : P(I) \rightarrow P(O) \\
X \mapsto \phi(X) = \{ O \in O \mid \exists i \in X \text{ s.t. } i \in O \}
\]

\[
\psi : P(O) \rightarrow P(I) \\
Y \mapsto \psi(Y) = \{ i \in I \mid \phi(i) \subseteq Y \}
\]

However, the authors neither proposed the expression of the resulting closure operator nor carried out a thorough analysis of inherent structural properties. Indeed, we can directly notice that these operators do not allow the computation of the disjunctive closure of a pattern $X$ directly from the context, without knowing beforehand the items tidsets. Hence, from an algorithmic point of view, Zhao et al. proposed an algorithm based on a combination of a depth-first traversal of the search space and the use of tidsets. Also, the authors did not make the connection between minimal OR-clauses and essential patterns. Finally, it is important to mention that these connection operators were used in [24] in order to extract closed OR-clauses (the equivalent of disjunctive closed patterns in our case) whose disjunctive supports are encompassed between two user-defined thresholds. However, they were not applied within the framework of frequent pattern concise representations.

The disjunction operator (i.e., the operator $\lor$) has also been used to define some concise representations for frequent patterns, like those based on disjunction-free sets [25] and (generalized) disjunction-free generators [26]. These representations only explore the conjunctive search space and required the introduction of what is called disjunctive rule [25]. Such a rule has a premise part composed by a conjunction of items and a conclusion part, distinct from the premise one, containing a specified number of items linked using the disjunction operator, as follows: $X \rightarrow y_1 \lor y_2 \lor \ldots \lor y_n$ where

---

10. The tidset of an item is the list of identifiers of objects to which the item belongs.
11. We did not use these representations in our experiments since $N\!D\!P_K$rep and consequently, $C\!N\!D\!P_K$rep is shown in [2] to provide better results.
Exploring the Disjunctive Search Space towards Discovering...

$X \subseteq I$, $\{y_1, y_2, \ldots, y_n\} \subseteq I$, $X \cap \{y_1, y_2, \ldots, y_n\} = \emptyset$ and $X$ is a conjunction of items. For more details on these representations, interested readers are referred to [15, 27], where a survey on some concise representations can be found, and to Chapter 3 in [26] where other ones are described.

Some works [17, 16] were interested in using the disjunction connector within association rules to define what is called generalized association rules. These rules grasped the interest of many researchers since they offer wealthier types of knowledge in many applications. In addition to the inclusive disjunction operator, i.e., the operator $\lor$, the authors in [16] were also interested in the exclusive disjunction operator, denoted $\oplus$. Nanavati et al. also proposed two kinds of rules which are the simple disjunctive rules and the generalized disjunctive rules. Simple disjunctive rules are those having either the premise or the conclusion (i.e., not simultaneously both) composed by a disjunction of items. This disjunction can be inclusive (the simultaneous occurrence of items is possible) or exclusive (two items cannot occur together). On the other hand, generalized disjunctive rules are disjunctive rules whose premises or conclusions contain a conjunction of disjunctions. These disjunctions can either be inclusive or exclusive.

In [17], the author mainly focuses on association rules having conclusions containing mutually exclusive items, i.e., the presence of one of them leads to the absence of the others, what is expressed in [16] using the operator $\oplus$.

Other forms of generalized association rules were also described in [28]. They are as follows: for all $x_i, y_j \in I$,

1. A bidirectional rule is expressed as $(x_1 \Leftrightarrow x_2) \land (x_2 \Leftrightarrow x_3) \land \ldots \land (x_{n-1} \Leftrightarrow x_n)$.
2. A mutually exclusive rule is defined as a set $X$ of items or predicates that cannot occur together. If an item of $X$ occurs in an object $O$, then all other items of $X$ are not present in $O$.
3. Rules having the premise or conclusion part composed of negated items, i.e., those of the form $x_1 \land x_2 \land \ldots \land x_n \rightarrow \neg y_1 \land \neg y_2 \land \ldots \land \neg y_m$ or $\neg x_1 \land \neg x_2 \land \ldots \land \neg x_n \rightarrow y_1 \land y_2 \land \ldots \land y_m$.
4. Rules having implication with disjunctive premises or conclusions, i.e., those of the form $x_1 \land x_2 \land \ldots \land x_n \rightarrow y_1 \lor y_2 \lor \ldots \lor y_m$ or $x_1 \lor x_2 \lor \ldots \lor x_n \rightarrow y_1 \land y_2 \land \ldots \land y_m$.

All these association rules forms are included and enriched thanks to the forms offered by our concise representation.

9 Conclusion and perspectives

In this technical report, we presented a new disjunctive closure operator as well as its main properties. Based on this operator, we introduced new concise representations which correct the claim of [10, 11] where the associated representation can miss some cases. This required the addition of few further elements what ensures the correctness of the whole regeneration process of frequent patterns. In addition to interesting compactness rates, our concise representation allows a straightforward computation of the disjunctive and negative supports. The experimental results showed that, in most cases, its size is significantly smaller than those of the best known concise representations. After that, we have shown various forms of generalized association rules that can be
extracted from our concise representation. We also discussed the main related work. It is worth noting that our approach can easily be extended when negative items are handled.

Acknowledgements: The authors would like to thank T. Calders and B. Goethals for providing the source code allowing the extraction of frequent non-derivable patterns, A. Casali and S. Lopes for providing the source code allowing the extraction of the concise representation based on frequent essential patterns, and, J. Muhonen and H. Toivonen for providing the source code allowing the extraction of frequent closed non-derivable patterns. This work is partly supported by the French-Tunisian CMCU project 05G1412.

References


