Constraint-Based Search of Straddling Biclusters and Discriminative Patterns

Israel Guerra, Loïc Cerf, João Foscarini, Michel Boaventura and Wagner Meira Jr.

Universidade Federal de Minas Gerais, Brazil
{guerra, lcerf, jfoscarini, michel, meira}@dcc.ufmg.br

Abstract. The state-of-the-art Data-Peeler algorithm extracts closed patterns in n-ary relations. Because it refines a lower bound and an upper bound of the pattern space, Data-Peeler can, in some circumstances, guarantee that a region of the pattern space does not contain any closed n-set satisfying some relevance constraint, allowing the algorithm to not perform any further pattern search in that region. If it is so, this region is left unexplored and some time is saved. Not all constraints enable such a pruning of the pattern space but both the monotone and the anti-monotone constraints do. This article shows that a minimal (resp. maximal) cover of some arbitrary groups of elements is anti-monotone (resp. monotone). As a consequence, Data-Peeler may prune the search space with those constraints and efficiently discover many different patterns. For instance, it can list the so-called straddling biclusters, which cover at least some given portions of every group. It can also discover closed n-sets that discriminate a group from the others, what has potential applications to supervised classification.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords: closed patterns, data mining, discriminative patterns, straddling biclusters

1. INTRODUCTION

Given a binary relation, i. e., a set of *objects* described with Boolean *attributes*, the complete extraction of the closed itemsets is a popular task. It generically aims to discover maximal sets of objects sharing the same maximal set of attributes. However, not every closed itemset is worth an interpretation. In fact, in practical contexts, the complete collection of all closed itemsets is so large that it is humanly impossible to read them all, if not computationally impossible to list them all. That is why, since the pioneer works, only a relevant subset of all closed itemsets is searched. This relevance is defined by means of additional constraints the closed itemset must satisfy. The discovery of the relevant closed itemsets can be faster if the algorithm is able to guarantee, without exploring a region of the pattern space, that this region does not contain any pattern satisfying the relevance constraints. This ability depends on both the enumeration principles of the algorithm and the properties of the constraints.

The Data-Peeler algorithm [Cerf et al. 2009] efficiently handles a broad class of constraints that includes (but is not restricted to) the so-called *monotone* and *anti-monotone* constraints. Moreover, it generalizes the search for closed patterns to n-ary relations, i. e., datasets with n dimensions rather than only one *object* dimension and one *attribute* dimension. In many contexts, the elements of the relation belong to some predefined groups. For instance, in the classical two-dimensional case, the objects can be partitioned into several groups and learning an associative classifier starts with the discovery of patterns that discriminate a group from the other ones. On the contrary, an analyst may be interested in the so-called *straddling bliclusters* [Owens III 2009], which are patterns that span across several groups.

This article tackles, for the first time (to the best of our knowledge), the problem of discovering relevant patterns in an n-ary relation with arbitrary groups of elements, i. e., the groups can overlap

This work was partially suppported by CNPq, Capes, Fapemig, and Inweb.

Table I: Example relation and groups

	A	B	C	A	B	C	A	B	C
1	**1**	1	1	1	1	1	1	1	0
2	1	1	**0**	1	0	0	1	1	0
3	0	1	0	0	0	1	1	0	1
4	0	0	1	1	0	1	1	1	1
	α			β			γ		

(a) $\mathcal{R}_E \subseteq \{\alpha, \beta, \gamma\} \times \{1, 2, 3, 4\} \times \{A, B, C\}$.

	$\subseteq \{\alpha, \beta, \gamma\}$		$\subseteq \{1, 2, 3, 4\}$		$\subseteq \{A, B, C\}$
G_1	$\emptyset$	$\cup$	$\{1, 2\}$	$\cup$	$\emptyset$
G_2	$\emptyset$	$\cup$	$\{3, 4\}$	$\cup$	$\emptyset$
G_3	$\{\alpha, \beta\}$	$\cup$	$\{2, 3, 4\}$	$\cup$	$\{A, B\}$

(b) Three groups defined from the dimensions of $\mathcal{R}_E$.

and contain elements from any of the n dimensions. Section 2 defines two constraints. One forces the patterns to contain a minimal number of elements in a group, whereas the other imposes a maximal cover of a group. Combining several instantiations of the two constraints allows to specify various types of relevant patterns. For example, in the context of associative classification in $c \in \mathbb{N}$ classes, a conjunction of c constraints can define a relevant pattern: it must cover, for instance, at least 20% of the class to discriminate and at most 10% of every other class. In Section 3, the minimal (resp. maximal) cover is proven monotone (resp. anti-monotone). As a consequence, DataPeeler can handle them in an efficient way (pruning of the pattern space). Section 4 proposes an incremental computation of the minimal (resp. maximal) possible covers of the groups and derives the computational cost of verifying the constraints. Section 5 experimentally compares the approach with the extraction of all frequent patterns followed by a filtering of those satisfying the constraints. Handling the constraints during the extraction allows the extraction of both the discriminative patterns and the straddling biclusters to be up to several orders of magnitude faster. That same section reports as well the discovery of interesting patterns in a real-life ternary relation. In this way, it shows it is valuable to define the constraints in the general context of a dataset with possibly more than two dimensions. The related work is given in Section 6 and Section 7 concludes the article.

2. PROBLEM STATEMENT

Given $n \in \mathbb{N}$ dimensions of analysis (i. e., n finite sets) $(D_i)_{i=1..n}$, the dataset is a relation $\mathcal{R} \subseteq \times_{i=1}^{n} D_i$, i. e., a set of n-tuples. Table Ia represents such a relation $\mathcal{R}_E \subseteq \{\alpha, \beta, \gamma\} \times \{1, 2, 3, 4\} \times \{A, B, C\}$, hence a ternary relation. In this table, every '1' at the intersection of three elements stands for the presence of the related triple in $\mathcal{R}_E$. For example, the bold '**1**', in Table Ia, is at the intersection of the elements α, 1 and A. It represents the presence of $(\alpha, 1, A)$ in $\mathcal{R}_E$. On the contrary a '0' in Table Ia is at the intersection of three elements which form a triple absent from $\mathcal{R}_E$. For example the bold '**0**' in Table Ia means $(\alpha, 2, C) \notin \mathcal{R}_E$.

A subset of the elements in any of the n dimensions (without loss of generality, they are assumed disjoint) is called *group*. More precisely, a group is a subset of $\cup_{i=1}^{n} D_i$. Table Ib lists three groups, which are defined from the dimensions of $\mathcal{R}_E$. Notice that a group (such as G_3 in Table Ib) may contain elements in different dimensions and that two groups may overlap (such as G_1 and G_3 in Table Ib) or even be included into each other (such as G_2 and G_3 in Table Ib). This article deals with the discovery of the closed n-sets involving *at least*, or *at most*, certain user-defined quantities of elements in the groups, as well as defined by the user. After recalling the definition of a closed n-set, we formally define these two constraints.

The *closed n-set* [Cerf et al. 2009] straightforwardly generalizes the famous *closed itemset* [Pasquier et al. 1999; Zaki and Hsiao 1999] to relations defined on (possibly) more than two dimensions. It is defined as a conjunction of two constraints, namely $\mathcal{C}_{\text{connected}}$ and $\mathcal{C}_{\text{closed}}$. $\mathcal{C}_{\text{connected}}$ constrains the pattern to only cover tuples present in the relation:

Definition 1 Connectedness. *The pattern* $(X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}$ *is* connected, *denoted* $\mathcal{C}_{connected}(X_1, \ldots, X_n)$, *if and only if* $\times_{i=1}^{n} X_i \subseteq \mathcal{R}$.

$\mathcal{C}_{\text{closed}}$ constrains the pattern to be maximal, i. e., adding any additional element (in any dimension) leads to violating $\mathcal{C}_{\text{connected}}$:

Definition 2 Closedness. *The pattern* $(X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}$ *is* closed, *denoted* $\mathcal{C}_{closed}(X_1, \ldots, X_n)$, *if and only if* $\forall (X'_1, \ldots, X'_n) \in \times_{i=1}^{n} D_i$, *then we have:*
$$\begin{cases} \forall i = 1..n, X_i \subseteq X'_i \\ \mathcal{C}_{connected}(X'_1, \ldots, X'_n) \end{cases} \Rightarrow \forall i = 1..n, X_i = X'_i.$$

Definition 3 Closed n-set. *The pattern* $(X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}$ *is a* closed n-set *if and only if* $\mathcal{C}_{connected}(X_1, \ldots, X_n) \wedge \mathcal{C}_{closed}(X_1, \ldots, X_n)$.

Example 1. *In* $\mathcal{R}_E$, *represented in Table Ia,* $(\{\alpha, \gamma\}, \{1, 2\}, \{A, B\})$ *is a closed* 3*-set.* $(\{\alpha, \beta\}, \{1, 2\}, \{A, B\})$ *is not a closed* 3*-set because it is not connected (*$(\beta, 2, B) \notin \mathcal{R}_E$*).* $(\{\alpha, \gamma\}, \{1, 2\}, \{A\})$ *is not a closed* 3*-set because it is not closed (*$(\{\alpha, \gamma\}, \{1, 2\}, \{A, B\})$ *is a strict super-pattern of it and does not violate* $\mathcal{C}_{connected}$*).*

Given a group $G \subseteq \cup_{i=1}^{n} D_i$, the constraint "covering at least $\mu \in \mathbb{N}$ elements in this group" is defined as follows:

Definition 4 Minimal group cover. *The pattern* $(X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}$ *covers at least* $\mu \in \mathbb{N}$ *elements in* G, *denoted* $\mathcal{C}_{G,\geq,\mu}(X_1, \ldots, X_n)$, *if and only if* $|(\cup_{i=1}^{n} X_i) \cap G| \geq \mu$.

Example 2. *Given the groups in Table Ib, the pattern* $(\{\alpha, \gamma\}, \{1, 2\}, \{A, B\})$ *does not cover any element in* G_2. *It covers both elements in* G_1 *and four elements — α,* 2, *A and B — in* G_3. *As a consequence, it minimally covers these three groups if and only if the related minimal thresholds are less than or equal to, respectively,* 0, 2 *and* 4.

In a similar way, "covering at most $\mu \in \mathbb{N}$ elements in this group" is defined as follows:

Definition 5 Maximal group cover. *The pattern* $(X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}$ *covers at most* $\mu \in \mathbb{N}$ *elements in* G, *denoted* $\mathcal{C}_{G,\leq,\mu}(X_1, \ldots, X_n)$, *if and only if* $|(\cup_{i=1}^{n} X_i) \cap G| \leq \mu$.

Example 3. *Continuing Example 2, the pattern* $(\{\alpha, \gamma\}, \{1, 2\}, \{A, B\})$ *maximally covers* G_1, G_2 *and* G_3 *if and only if the related maximal thresholds are greater than or equal to, respectively,* 2, 0 *and* 4.

Given an n-ary relation $\mathcal{R} \subseteq 2^{\times_{i=1}^{n} D_i}$ and a finite set $\mathcal{T}$ of triples in $2^{\cup_{i=1}^{n} D_i} \times \{\geq, \leq\} \times \mathbb{N}$ (i. e., a finite set of *groups* associated with *minimal* or *maximal* cover *thresholds* to respect), the problem solved in this article is the computation of $\{(X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i} \mid \begin{cases} (X_1, \ldots, X_n) \text{ is a closed } n\text{-set} \\ \wedge_{t \in \mathcal{T}} \mathcal{C}_t(X_1, \ldots, X_n) \end{cases}\}$.

For the case $n = 2$, specific instantiations of this problem can be found in the literature. In particular, let us mention the discovery of the following types of patterns:

Straddling biclusters. [Owens III 2009] are patterns significantly straddling across user-defined subsets of one dimension D_i, i. e., every group is a subset of D_i and it is always associated with a *minimal* cover threshold (restriction of the domain of $\mathcal{T}$ to $2^{D_i} \times \{\geq\} \times \mathbb{N}$).

Discriminative patterns. [Cerf et al. 2008] are patterns covering at least a given number of elements in a group and at most another number of elements in any other group, i. e., $\mathcal{T}$ contains one and only one triple with a $\leq$ component. Moreover, the groups are subsets of one dimension D_i.

3. TRAVERSAL OF THE PATTERN SPACE

The problem stated in the previous section is solvable by first extracting all closed n-set and then filtering those satisfying the group cover constraints. Nevertheless, that solution is intractable unless the relation $\mathcal{R}$ is very small. Indeed, there are, in the worst case, $2^{\sum_{i \neq j} |D_i|}$ closed n-sets to be extracted from $\mathcal{R}$ (where j is the index of the largest dimension). To lower the time requirements to solve the problem, the two constraints must be used during the closed n-set extraction. More precisely, they must guide the search of the valid patterns, i. e., allow to prune regions of the pattern space that need not be traversed because they do not contain any valid closed n-set. Fortunately, those two constraints have properties that help in identifying the fruitless regions of the pattern space. The two following theorems state those properties. In both of them, G is a group and μ is a (minimal or maximal) size threshold.

Any sub-pattern of a pattern violating $\mathcal{C}_{G,\geq,\mu}$ violates it as well:

THEOREM 1 MONOTONICITY OF $\mathcal{C}_{G,\geq,\mu}$. *Given a pattern $(U_1, \ldots, U_n)$ violating $\mathcal{C}_{G,\geq,\mu}$ (i. e., $\neg\mathcal{C}_{G,\geq,\mu}(U_1, \ldots, U_n)$), we have:*

$$\forall (X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}, (\forall i = 1..n, X_i \subseteq U_i) \Rightarrow \neg\mathcal{C}_{G,\geq,\mu}(X_1, \ldots, X_n) \ .$$

PROOF.

$$\begin{aligned}
(\forall i = 1..n, X_i \subseteq U_i) &\Rightarrow \cup_{i=1}^{n} X_i \subseteq \cup_{i=1}^{n} U_i \\
&\Rightarrow \cup_{i=1}^{n} X_i \cap G \subseteq \cup_{i=1}^{n} U_i \cap G \\
&\Rightarrow |\cup_{i=1}^{n} X_i \cap G| \leq |\cup_{i=1}^{n} U_i \cap G| \\
&\Rightarrow |\cup_{i=1}^{n} X_i \cap G| < \mu \text{ because } \neg\mathcal{C}_{G,\geq,\mu}(U_1, \ldots, U_n) \\
&\Leftrightarrow \neg\mathcal{C}_{G,\geq,\mu}(X_1, \ldots, X_n)
\end{aligned}$$

□

Any super-pattern of a pattern violating $\mathcal{C}_{G,\leq,\mu}$ violates it as well:

THEOREM 2 ANTI-MONOTONICITY OF $\mathcal{C}_{G,\leq,\mu}$. *Given a pattern $(L_1, \ldots, L_n)$ violating $\mathcal{C}_{G,\leq,\mu}$ (i. e., $\neg\mathcal{C}_{G,\leq,\mu}(L_1, \ldots, L_n)$), we have:*

$$\forall (X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}, (\forall i = 1..n, L_i \subseteq X_i) \Rightarrow \neg\mathcal{C}_{G,\leq,\mu}(X_1, \ldots, X_n) \ .$$

PROOF.

$$\begin{aligned}
(\forall i = 1..n, L_i \subseteq X_i) &\Rightarrow \cup_{i=1}^{n} L_i \subseteq \cup_{i=1}^{n} X_i \\
&\Rightarrow \cup_{i=1}^{n} L_i \cap G \subseteq \cup_{i=1}^{n} X_i \cap G \\
&\Rightarrow |\cup_{i=1}^{n} L_i \cap G| \leq |\cup_{i=1}^{n} X_i \cap G| \\
&\Rightarrow \mu < |\cup_{i=1}^{n} X_i \cap G| \text{ because } \neg\mathcal{C}_{G,\leq,\mu}(L_1, \ldots, L_n) \\
&\Leftrightarrow \neg\mathcal{C}_{G,\leq,\mu}(X_1, \ldots, X_n)
\end{aligned}$$

□

The state-of-the-art closed n-set extractor, namely DATA-PEELER, explores the pattern space by traversing a binary tree whose nodes are associated with two patterns, namely $(L_1, \ldots, L_n)$ and $(U_1, \ldots, U_n)$, which are computed from the parent node according to the mathematical expressions in Figure 1. Please refer to [Cerf et al. 2009] for a detailed presentation of this traversal and why it supports a correct and complete discovery of the closed n-sets. For this article, the only relevant property is that, at any node, $(L_1, \ldots, L_n)$ and $(U_1, \ldots, U_n)$, respectively, are the smallest and the largest patterns that can be considered in the sub-tree the node roots. More precisely, any closed

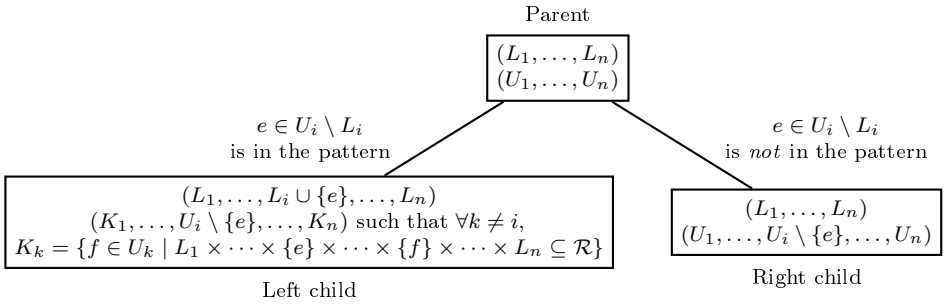

Fig. 1: Data-Peeler's traversal of the pattern space. At the root of the tree, $(L_1, \ldots, L_n) = (\emptyset, \ldots, \emptyset)$ and $(U_1, \ldots, U_n) = (D_1, \ldots, D_n)$.

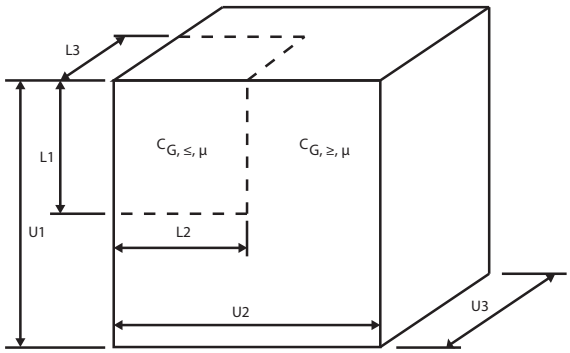

Fig. 2: At every enumeration node and for every group G, Data-Peeler checks whether the smallest and the largest pattern that can be considered in the sub-tree respectively satisfy $\mathcal{C}_{G,\leq,\mu}$ and $\mathcal{C}_{G,\geq,\mu}$. Any test that fails triggers a safe pruning of the enumeration tree at the current node.

n-set $(X_1, \ldots, X_n)$ that can be discovered along the traversal of the sub-tree is such that $\forall i = 1..n$, $L_i \subseteq X_i \subseteq U_i$, i. e., $(L_1, \ldots, L_n)$ and $(U_1, \ldots, U_n)$ respectively are a lower and an upper bound of the pattern space that would be explored "below" the current node. The conditional tense applies because, thanks to those bounds, Data-Peeler efficiently verifies whether that pattern space can possibly contain a closed n-set satisfying the group cover constraints and, if it does not, this pattern space is not explored, i. e., the binary tree is pruned at the current node.

Here is the predicate that Data-Peeler evaluates to decide whether it is safe to prune the binary tree at the current node:

$$(\exists G, \geq, \mu \in \mathcal{T} \mid \neg\mathcal{C}_{G,\geq,\mu}(U_1, \ldots, U_n)) \vee (\exists G, \leq, \mu \in \mathcal{T} \mid \neg\mathcal{C}_{G,\leq,\mu}(L_1, \ldots, L_n)) \quad .$$

Figure 2 graphically represents this predicate. If it is satisfied then Theorem 1 or/and 2 guarantee that, for all pattern $(X_1, \ldots, X_n) \in \times_{i=1}^{n} 2^{D_i}$ such that $\forall i = 1..n$, $L_i \subseteq X_i \subseteq U_i$, we have:

$$(\exists G, \geq, \mu \in \mathcal{T} \mid \neg\mathcal{C}_{G,\geq,\mu}(X_1, \ldots, X_n)) \vee (\exists G, \leq, \mu \in \mathcal{T} \mid \neg\mathcal{C}_{G,\leq,\mu}(X_1, \ldots, X_n)) \quad .$$

In other terms, any pattern, that would be considered in the sub-tree rooted by the current node, is violating at least one group cover constraint. As a consequence, there is no need to traverse this sub-tree.

4. IMPLEMENTATION

The pattern space being pruned, the time requirements to solve the problem, stated in Section 2, may be significantly lowered in comparison to extracting all closed n-sets and then filtering those satisfying the group cover constraints. This is particularly true when those constraints are strong, i. e., for a

given constraint, when many elements in the related group are to be present (for a minimal group cover) or absent (for a maximal group cover) in the discovered closed n-sets. In the opposite case, the predicate, stated at the end of the previous section, will almost always be evaluated to "false" and those evaluations increase the time requirements without being compensated by a significant pruning of the pattern space. Anyway, an efficient evaluation of the predicate obviously is preferable to a naive one. A naive one would simply stick to the Definitions 4 and 5, i. e., it would intersect the elements in each group with those in the (lower or upper) bound. Assuming the elements are maintained ordered, the time complexity of this naive evaluation would therefore be linear in the total number of elements (in all groups and in the bounds).

As depicted in Figure 1, Data-Peeler refines the lower and upper bounds of the pattern space while going down the binary tree, i. e., from a node to one of its two children, some elements are added to the lower bound and some elements are removed from the upper bound. This incremental (resp. decremental) computation of the lower (resp. upper) bound allows a faster evaluation of the predicate. It is only about maintaining updated, along the pattern space traversal, the quantity $l_G = |\cup_{i=1}^{n} L_i \cap G|$ (resp. $u_G = |\cup_{i=1}^{n} U_i \cap G|$) for every group that must be maximally (resp. minimally) covered. In this way, these cardinalities can be compared, in constant time, to the maximal (resp. minimal) cover thresholds. Given the set $\cup_{i=1..n} L_i^{\text{child}} \setminus L_i^{\text{parent}}$ (resp. $\cup_{i=1..n} U_i^{\text{parent}} \setminus U_i^{\text{child}}$) of the elements that are added to (resp. removed from) the lower (resp. upper) bound, when going from the parent node to the child node, we respectively have:

—$l_G^{\text{child}} = l_G^{\text{parent}} + |\{e \in \cup_{i=1}^{n} L_i^{\text{child}} \setminus L_i^{\text{parent}} \mid e \in G\}|$;
—$u_G^{\text{child}} = u_G^{\text{parent}} - |\{e \in \cup_{i=1}^{n} U_i^{\text{parent}} \setminus U_i^{\text{child}} \mid e \in G\}|$.

Testing whether an element e is in a group G requires a constant time if a bitset represents the group[1]. Overall, the time complexity of this evaluation of the predicate is linear in the number of groups $|\mathcal{T}|$ multiplied by the number $|E|$ of elements that were added to the lower bound or removed from upper bound. In practice, this complexity is far below that of the naive evaluation.

5. EXPERIMENTAL STUDY

Data-Peeler was written in C++ and compiled with GCC 4.5.3 and the O3 optimizations. This section reports experiments performed on a GNU/Linux system running on top of a 3.4 GHz core.

5.1 Extracting Straddling Biclusters and Discriminative Patterns

To the best of our knowledge, Data-Peeler is the fastest closed n-sets extractor when $n \geq 3$. In those contexts, and considering the additional constraints presented in this article, it obviously remains faster than its competitors. Indeed, the patterns they output need to be post-processed, whereas our approach enables more pruning along Data-Peeler's enumeration, hence improved running times. That is why we decided to first consider the case $n = 2$ and compare our approach to LCM [Uno et al. 2005], which is often considered the fastest algorithm to completely list the closed itemsets in a binary relation. LCM's post-processing step, which filters the closed itemsets satisfying the group cover constraints, was implemented in C++. Two problems, already proposed in literature, are tackled: the extraction of the *straddling biclusters* [Owens III 2009] and that of the *discriminative patterns* [Cerf et al. 2008] (see Section 2). In both problems, the groups partition one single dimension, the objects. We have chosen normalized and discretized datasets [Coenen 2003] that are often used to evaluate classification algorithms. The groups simply are the classes defined in those datasets.

[1]The bitsets representing the groups are constructed after the relation $\mathcal{R}$ is parsed (so that all elements are known and attributed an id, its index in the bitsets). They are never modified afterward.

Table II: Characteristics of the datasets: number of objects, number of objects in the smallest class, number of attributes, number of classes, number of tuples and density (in this order).

Datasets	$\lvert D_{\text{objects}}\rvert$	$\lvert G_{\min}\rvert$	$\lvert D_{\text{attributes}}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{R}\rvert/\lvert D_{\text{objects}} \times D_{\text{attributes}}\rvert$
`cylBands`	540	228	34	2	1080	0.0588
`letRecog`	20000	734	15	26	40000	0.1333
`soybean-large`	683	8	35	19	1366	0.0571
`waveform`	5000	1647	21	3	10000	0.0950

Table II lists some characteristics of the most demanding datasets, i. e., those requiring the largest running times. (except `connect4` because mining it requires prohibitive time and space requirements in some of the considered settings). Figure 0**??** reports the times, in seconds, to extract the straddling biclusters or the discriminative patterns from those datasets. In the case of the straddling biclusters, the same minimal proportion is chosen for all groups to cover. It is in the x-axis of the curves and varies from 0 to 1 with a 0.1 step. When it comes to list discriminative patterns, the class to characterize is to be minimally covered, whereas every other class is to be maximally covered. We chose to always minimally cover the smallest class because it is usually considered the hardest to characterize. The minimal cover is the proportion P in the x-axis of the curves. The maximal cover of every other class is fixed to $P/2$. Although LCM cannot enforce the group cover constraints, it can force the extracted patterns to have a minimal number of objects, a consequence of the constraints. More precisely, those numbers are the sums of the μ thresholds of the $\mathcal{C}_{G,\geq,\mu}$ constraints.

Thanks to the minimal size constraint, LCM's running time decreases when the group cover constraints are stronger (right sides of the curves). So does Data-Peeler's, thanks to the group cover constraints. Data-Peeler usually outperforms LCM followed by its post-processing step. The gain is larger when searching discriminative patterns rather than straddling biclusters. This is expected since only minimal cover constraints lead to a minimal size constraint that LCM uses to prune the pattern space. In some cases, Data-Peeler runs orders of magnitude faster than LCM. For instance, in `letRecog`, it takes Data-Peeler four seconds to list the 21 discriminative patterns with $P = 0.2$. For the same tasks, LCM first lists 546,158 closed itemsets and the 21 discriminative patterns are filtered during the post-process for a total running time of 238 seconds. The straddling biclusters extracted at the abscissas 0 of the curves are all the closed itemsets. Indeed, the constraint becomes "covering at least 0 element in every group". In this context, LCM runs faster than Data-Peeler. It is therefore expected that comparing Data-Peeler's performances with and without the group cover constraints would give an advantage to the former that is larger than the advantage it has over LCM.

5.2 Extracting Patterns of Influence in Twitter

Twitter is a particularly popular social network. As of 2012, more than 500 million users are registered. They publish, on Twitter, over 340 million messages a day. Brazil is the second country in terms of the number of Twitter registered users. In February 2012, this number reached the 30 million mark. Data-Peeler can help in identifying, in Twitter's data, patterns of influence w.r.t. a given topic. Two scenarios are considered in this section: the Brazilian soccer championship and the Brazilian municipal elections. In both cases, our analysis focuses on *retweets*, i. e., Twitter messages that are endorsed by some users [Kwak et al. 2010]. Every *retweet* is classified w.r.t. the *entities* discussed in it (supervised classification method, which is out of the scope of this article). An entity is a soccer team (among 20) in the case of the Brazilian championship. It is a candidate in the case of the municipal elections. By considering as well the week during which the original message was written, we end up with a ternary relation. Every triple (`user`,`week`,`entity`) in it stands for a `user` who published, during `week`, a Twitter message that refers to `entity` and was *retweeted* at least once.

Between the 45^{th} week of 2010 and the 6^{th} week of 2011 (14 weeks of collect), 126,566 users were retweeted when writing about the Brazilian soccer championship. Let us assume the analyst wants to discover, in the $126,566 \times 14 \times 20$ relation, sets of users who were only influential when they wrote

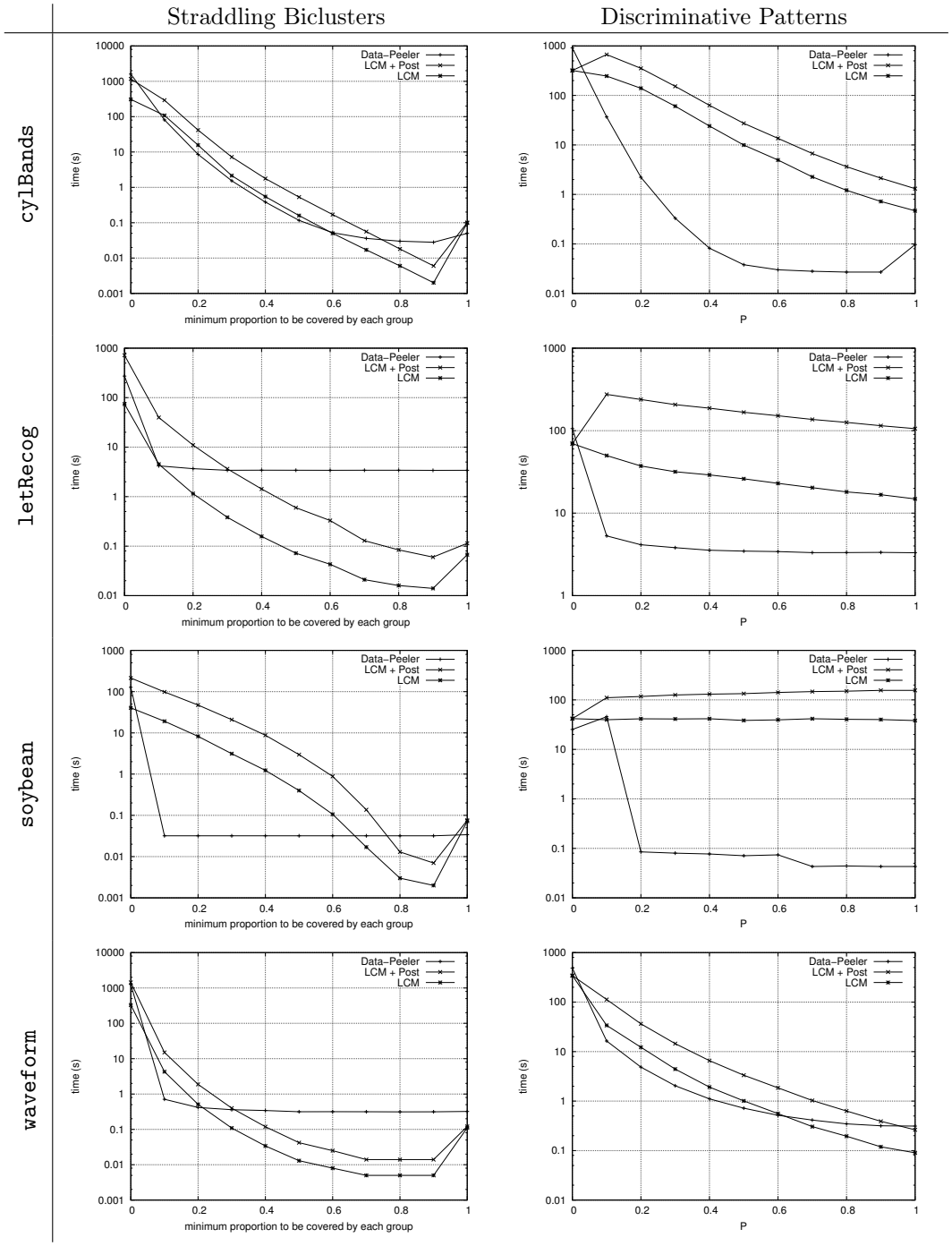

Fig. 3: Extraction times of straddling biclusters and discriminative patterns.

about some teams from Rio de Janeiro and who remained influential (i. e., were still *retweeted*) after the championship was over. To answer this query, three groups are defined: G_1 contains the four teams from Rio de Janeiro, G_2 contains the remaining sixteen teams, and G_3 contains the nine weeks after the end of the championship. It takes Data-Peeler 9 minutes and 20 seconds to extract the 4,913 closed 3-sets involving at least two weeks and 100 users. By additionally requiring the satisfaction of $\mathcal{C}_{G_1,\geq,2} \wedge \mathcal{C}_{G_2,\leq,1} \wedge \mathcal{C}_{G_3,\geq,1}$, only four patterns are returned within 8 minutes. They specifically answer the considered query. They all involve Flamengo and Vasco, which are, among Rio's soccer teams, those with the more supporters. The 382 users (in at least one of the four patterns) are journalists, pep rallies and ordinary users. 46 of them have "fla" or "vasco" in their login. That confirms their use of Twitter as a mean to comment on their favorite team. To discover users who are *constantly* influential, the minimal number of weeks for a closed 3-set to be valid is increased to ten, while the minimal number of users is lowered to five. The conjunction of group cover constraints is left unchanged. It takes Data-Peeler 3 minutes and 6 seconds to output the one single pattern satisfying the new set of constraints. Again, it involves Vasco and Flamengo. The users in the closed 3-set are: "gilmarferreira", "globoesportecom", "SporTV", "TorcedorVasco" and "reFlamengo".

Between the 29^{th} and the 45^{th} week of 2012 (17 weeks of collect), 98,194 users were retweeted when writing about at least one of the 100 candidates who ran for mayor of one of the 14 largest Brazilian cities. Let us assume the analyst wants to discover sets of users who were influential when they discussed the top-three candidates in the cities of São Paulo, Rio de Janeiro or Belo Horizonte[2]. To do so, the groups $(G_c)_{c\in\{\text{São Paulo,Rio,BH}\}}$ are defined. Each of them contains the top three candidates of the related city. Those cities are then taken one by one. It takes Data-Peeler about 30 minutes to list the patterns with at least two candidates in the considered city (use of the constraint $\mathcal{C}_{G_c,\geq,2}$) and having a number of users at least equal to 1% the number triples involving a candidate from the city c (929 for São Paulo, 478 for Rio de Janeiro and 45 for Belo Horizonte). Among the 21 patterns for all three cities, the most common week is the 40^{th} (present in 17 of those patterns). During this week, the main debates were broadcast on TV and Twitter users were reacting to them.

6. RELATED WORK

Our approach exploits the so-called *anti-monotonicity* and *monotonicity* of $\mathcal{C}_{G,\geq,\mu}$ (Definition 4) and $\mathcal{C}_{G,\leq,\mu}$ (Definition 5) respectively. Those two classes of constraints were first defined in [Ng et al. 1998] and [Grahne et al. 2000] respectively. DualMiner [Bucila et al. 2002] was the first algorithm to efficiently discover the (not necessarily closed) itemsets satisfying both monotone and anti-monotone constraints. Data-Peeler [Cerf et al. 2009] generalized this task to n-ary relations. To the best of our knowledge it remains, today, the most efficient proposal (even in the restricted case $n = 3$) and the only one able to prune the n-dimensional pattern space with both monotone and anti-monotone constraints when $n \geq 3$.

Data-Peeler explores the pattern space by traversing a binary tree. Its nodes are associated with a lower bound and an upper bound of the pattern space to be (or not) explored in the sub-tree the node roots. As instantiated in this article, Data-Peeler prunes the sub-tree if the lower bound does not satisfy a monotone constraint or if the upper bound does not satisfy an anti-monotone constraint. Notice, however, that knowing those bounds allows to efficiently enforce a larger class of constraints, namely the piecewise (anti)-monotone constraints. [Cerf et al. 2009] defines it in a "divisive" way: they are constraints which are either monotone or anti-monotone w.r.t. each *occurrence* of a variable in their expression. However, Soulet and Crémilleux [2005] have independently proposed a "constructive" definition of what happens to be the same class of constraints: those constraints are recursively defined from arbitrary primitives that either increase or decrease w.r.t. each of their arguments. The proposed algorithm relies as well on refining a lower bound and an upper bound of the pattern space during its exploration. It is, however, restricted to binary relations. To the best of our knowledge, Soulet and

[2]The top-three is defined *a posteriori*, from the results of the election.

Crémilleux [2009] present the most comprehensive study of the piecewise (anti)-monotone constraints and their ability to prune the pattern space.

7. CONCLUSION

Given an n-ary relation, an analyst may be interested in discovering closed n-sets that cover minimal or, in the contrary, maximal quantities of elements in some groups of interest. For instance, in a supervised classification context, a class is a group that can be characterized by the closed n-sets covering a large portion of its elements and small portions of those in every other class. This article has shown that minimally (resp. maximally) covering a group is an anti-monotone (resp. a monotone) constraint. Knowing an upper (resp. a lower) bound of the patterns in a region of the search space allows to efficiently verify whether this region contains patters satisfying the minimal (resp. maximal) group covers. If it does not, the region can be safely left unexplored and the constrained closed n-sets are discovered faster. The state-of-the-art closed n-set extractor, namely Data-Peeler, refines such bounds along the pattern space traversal. That is why it is easy to make it to additionally enforce the group cover constraints in an efficient way. Data-Peeler may be up to orders of magnitude faster than algorithms that first list the closed n-sets and then filter those satisfying the additional group cover constraints. We have observed such gains even in the case $n = 2$ when Data-Peeler was compared to LCM listing the frequent (threshold corresponding to the sum of the minimal possible covers of the transactions in the groups) closed itemsets and then post-processing them.

REFERENCES

Bucila, C., Gehrke, J., Kifer, D., and White, W. M. DualMiner: a dual-pruning algorithm for itemsets with constraints. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Canada, pp. 42–51, 2002.

Cerf, L., Besson, J., Robardet, C., and Boulicaut, J.-F. Closed Patterns Meet n-ary Relations. *ACM Transactions on Knowledge Discovery from Data* 3 (1): 1–36, 2009.

Cerf, L., Gay, D., Selmaoui, N., and Boulicaut, J.-F. A Parameter-Free Associative Classification Method. In *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery*. Turin, Italy, pp. 293–304, 2008.

Coenen, F. The LUCS-KDD Discretised/Normalised ARM and CARM Data Library, 2003. `http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN/`.

Grahne, G., Lakshmanan, L. V. S., and Wang, X. Efficient Mining of Constrained Correlated Sets. In *Proceedings of the International Conference on Data Engineering*. San Diego, USA, pp. 512–521, 2000.

Kwak, H., Lee, C., Park, H., and Moon, S. What is Twitter, a Social Network or a News Media? In *Proceedings of the International Conference on World Wide Web*. Raleigh, USA, pp. 591–600, 2010.

Ng, R. T., Lakshmanan, L. V. S., Han, J., and Pang, A. Exploratory Mining and Pruning Optimizations of Constrained Association Rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Seattle, USA, pp. 13–24, 1998.

Owens III, C. C. *Mining Truth Tables and Straddling Biclusters in Binary Datasets*. M.S. thesis, Faculty of the Virginia Polytechnic Institute and State University, 2009.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. Efficient Mining of Association Rules Using Closed Itemset Lattices. *Information Systems* 24 (1): 25–46, 1999.

Soulet, A. and Crémilleux, B. Exploiting Virtual Patterns for Automatically Pruning the Search Space. In *Proceedings of the International Workshop on Knowledge Discovery in Inductive Databases, Revised Selected and Invited Papers*. Porto, Portugal, pp. 202–221, 2005.

Soulet, A. and Crémilleux, B. Mining Constraint-Based Patterns Using Automatic Relaxation. *Intelligent Data Analysis* (1): 109–133, 2009.

Uno, T., Kiyomi, M., and Arimura, H. LCM ver.3: collaboration of array, bitmap and prefix tree for frequent itemset mining. In *Proceedings of the Open Source Data Mining Workshop on Frequent Pattern Mining Implementations*. Chicago, USA, pp. 77–86, 2005.

Zaki, M. J. and Hsiao, C.-J. ChARM: an efficient algorithm for closed association rule mining. Tech. Rep. 99-10, Computer Science Department, Rensselaer Polytechnic Institute, Troy, USA. oct, 1999.