



HAL
open science

Summarizing Contrasts by Recursive Pattern Mining

Arnaud Soulet, Bruno Crémilleux, Marc Plantevit

► **To cite this version:**

Arnaud Soulet, Bruno Crémilleux, Marc Plantevit. Summarizing Contrasts by Recursive Pattern Mining. IEEE 11th International Conference on Data Mining Workshops (ICDMW), Dec 2011, Vancouver, Canada. pp.1155-1162, <10.1109/ICDMW.2011.161>. <hal-01021980>

HAL Id: hal-01021980

<https://hal.science/hal-01021980v1>

Submitted on 11 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Summarizing Contrasts by Recursive Pattern Mining

Arnaud Soulet
University of Tours, LI
3 place Jean Jaurès
41026 Blois, France
arnaud.soulet@univ-tours.fr

Bruno Crémilleux
University of Caen, GREYC CNRS
Campus Côte de Nacre
14032 Caen Cedex, France
bruno.cremilleux@unicaen.fr

Marc Plantevit
University of Lyon, LIRIS CNRS
43 Bd du 11 Novembre 1918
69622 Villeurbanne Cedex, France
marc.plantevit@liris.cnrs.fr

Abstract—A lot of constrained patterns (e.g., emerging patterns, subgroup discovery, classification rules) emphasize the contrasts between data classes and are at the core of many classification techniques. Nevertheless, the extremely large collection of generated patterns hampers the end-user interpretation and the deep understanding of the knowledge revealed by the whole collection of patterns. The key idea of this paper is to *summarize* the contrasts of a dataset in order to provide understandable characterizations of data classes. We first introduce a novel framework, called *recursive pattern mining*, for only discovering few as well as relevant patterns. We demonstrate that this approach encompasses usual pattern mining framework and we study its key properties. Then, we use recursive pattern mining for extracting k recursive emerging patterns. Taken together, these patterns form a *REP k -summary* which summarizes the contrasts of the dataset. Finally, we validate our approach on benchmarks and real-world applications on the biological domain, showing the efficiency and the usefulness of the approach.

Keywords—pattern mining; contrasts; summary

I. INTRODUCTION

Contrast mining and more generally supervised descriptive rule discovery [1] is a significant field of research in Knowledge Discovery in Databases. Initially introduced in [2], emerging patterns (EPs) are patterns whose frequency strongly varies between two parts of a dataset (i.e., two classes). EPs characterize the classes in a quantitative and qualitative way. Thanks to their ability to emphasize the distinctions between classes, EPs are well suited to design classifiers [3] or to propose a help for diagnosis [4]. Unfortunately, as many tasks based on local patterns, the discovery of EPs suffers from limitations that may hinder its use.

Firstly, the large number of output patterns hamper the individual and global analysis performed by end-users of data. Many works address methods to reduce the collection of discovered patterns and a promising direction is the general issue of pattern set mining [5]. Nevertheless, most of them are dedicated to frequent patterns [6], [7], [8], [9] and are ineffective to handle classes. In [10], the authors focus on the most expressive emerging patterns which are only present in one class. But, there are still numerous, they may be over-specified and thus irrelevant. On the other hand, condensed representations of EPs [11], [12] are not sufficiently compact

for enabling user-friendly interpretation.

Second, setting thresholds (e.g., minimal support) is often too subtle. Both the quantity and the quality of desired patterns are unpredictable. A high threshold may generate no answer, a small one may generate millions of patterns. Adjusting thresholds to diminish the number of patterns is actually counterproductive. Typically, increasing the minimal support focuses the mining on the most frequent patterns which are often too trivial. On the other hand, thresholds of constraints based on interestingness measures appear as too rigid barriers. This crisp effect already studied in [13] motivates approaches where parameters have a softer impact or are more understandable for users.

In this paper, we aim at summarizing contrasts stemmed from the different classes of a dataset by extracting at most k emerging patterns having a good trade-off between significance and representativeness. Our summarization process copes with the drawbacks enumerated above. In particular, the tunable concision of mined patterns favours users' interpretation. One originality of our approach is to design a novel framework melting recursive data mining and pattern mining. A key idea is to iterate the pattern mining process on the output until the latter is reduced enough. Resulting recursive patterns bring forward information coming from each mining step and they are meaningful. Moreover, we show that our method enables us to control the number of produced patterns. Our approach is sound and complete contrary to the summarization issue addressed in [14] which is based on heuristics mixing the interestingness and the distribution of discovered patterns.

Our contributions are as follows:

- We introduce a new mining framework, called recursive pattern mining. The key principle is to successively refine the mined patterns until few and relevant patterns are obtained. We prove that recursive pattern mining encompasses the usual pattern mining paradigm.
- We define the notion of recursive emerging patterns (REPs) which are discovered by the recursive pattern mining framework. These patterns not only characterize the original dataset, but they depict all the contrasts of all the mining steps. Then, we gather them for building summaries of contrasts and we show how to restrain

the maximal size of a summary at most k patterns.

- A set of experiments carried out on benchmarks and applications on biological data show that the recursive pattern mining enables us to efficiently build concise and representative summaries. By discovering useful information in the application context, these experiments shows the great interest of recursive mining on real-world applications.

The remainder of the paper is organized as follows. Section II introduces essential notions, motivates and formalizes the problem of summarizing the contrasts of a dataset. In Section III, we present our recursive pattern mining framework. Then in Section IV we instantiate this approach to summarize the contrasts of a dataset by presenting recursive emerging patterns. Experiments are described in Section V. We review related work in Section VI.

II. PROBLEM STATEMENT

A. Pattern Mining and Contrasts

Let \mathcal{I} be a set of distinct literals called *items*, an itemset (or pattern) is a non-null subset of \mathcal{I} . These patterns are gathered together in the language $\mathcal{L}_{\mathcal{I}} = 2^{\mathcal{I}} \setminus \emptyset$. A transactional dataset is a multi-set of patterns of $\mathcal{L}_{\mathcal{I}}$. Each pattern, named *transaction*, is a database entry. Table I gives an example of a transactional dataset \mathcal{D} with 8 transactions t_1, \dots, t_8 described by 6 items A, \dots, F . In contrast mining, \mathcal{D} is subdivided into classes (corresponding to the datasets \mathcal{D}_1 and \mathcal{D}_2 in Table I). Every transaction has a class label.

| Trans. | \mathcal{D} | | | | | |
|--------|---------------|-----|-----|-----|-----|-----------------|
| | Items | | | | | |
| t_1 | | | C | | | \mathcal{D}_1 |
| t_2 | A | B | D | | F | |
| t_3 | A | | D | E | | |
| t_4 | A | B | D | | | |
| t_5 | A | | D | | | \mathcal{D}_2 |
| t_6 | A | B | C | | F | |
| t_7 | | B | C | D | | |
| t_8 | | | C | E | | |

Table I
EXAMPLE OF A TRANSACTIONAL DATASET

Constraint-based pattern mining aims at discovering all the patterns of $\mathcal{L}_{\mathcal{I}}$ satisfying a given predicate q , named *constraint*, and occurring in \mathcal{D} . The problem is traditionally formulated as the problem of computing the *theory* $Th(\mathcal{D}, \mathcal{L}_{\mathcal{I}}, q)$ [15]. As mining patterns under constraints requires the exploration of the huge search space depicted by $\mathcal{L}_{\mathcal{I}}$, pruning conditions are necessary. A lot of pruning methods are based on the property of monotonicity (a constraint q is anti-monotone (resp. monotone) iff for all $X \in \mathcal{L}_{\mathcal{I}}$ satisfying q , any subset (resp. superset) of X also satisfies q). Constraints are a way to only select interesting patterns according to the task (e.g., finding regularities, exceptions or contrasts).

Many measures such as the well-used *frequency* are involved in constraints [16]. The *frequency* of a pattern X , denoted by $freq(X, \mathcal{D})$, is the number of transactions in \mathcal{D} containing X . The minimal frequency constraint focuses on itemsets having a frequency exceeding a given minimal threshold $minfreq > 0$ (i.e., $freq(X, \mathcal{D}) \geq minfreq$). Following on, we also use the relative frequency, named *support*: $supp(X, \mathcal{D}) = freq(X, \mathcal{D})/|\mathcal{D}|$ (where $|\mathcal{D}|$ is the cardinality of set \mathcal{D}). For instance, $freq(AD, \mathcal{D}) = 4$ and $supp(AD, \mathcal{D}) = 4/8 = 0.5$ with dataset given by Table I. The property of anti-monotonicity satisfied by the frequency constraint is likely a key point to explain the success of this measure.

Emerging patterns reveal contrasts between two parts of a dataset. Intuitively, an emerging pattern is a pattern whose frequency increases significantly from one class to another [2]. The capture of contrast brought by such a pattern is evaluated by the growth rate measure. The *growth rate* $gr_i(X, \mathcal{D})$ of a pattern X from $\mathcal{D} \setminus \mathcal{D}_i$ to \mathcal{D}_i is defined as:

$$gr_i(X, \mathcal{D}) = \begin{cases} \frac{supp(X, \mathcal{D}_i)}{supp(X, \mathcal{D} \setminus \mathcal{D}_i)} & \text{if } supp(X, \mathcal{D} \setminus \mathcal{D}_i) > 0 \\ +\infty & \text{if } supp(X, \mathcal{D}_i) > 0 \text{ and } \\ & supp(X, \mathcal{D} \setminus \mathcal{D}_i) = 0 \\ 0 & \text{otherwise} \end{cases}$$

where $supp(X, \mathcal{D} \setminus \mathcal{D}_i) = \frac{\sum_{k \neq i} freq(X, \mathcal{D}_k)}{\sum_{k \neq i} |\mathcal{D}_k|}$. Thus, the definition of an *emerging pattern* (EP in short) is given by:

Definition 1 (Emerging pattern): Given a threshold $\rho > 1$, a pattern X is said to be an emerging pattern from $\mathcal{D} \setminus \mathcal{D}_i$ to \mathcal{D}_i if $gr_i(X, \mathcal{D}) \geq \rho$.

All the emerging patterns of a given class i correspond to the theory $Th(\mathcal{D}, \mathcal{L}_{\mathcal{I}}, gr_i(X, \mathcal{D}) \geq \rho)$. Following our example in Table I, with $\rho = 3$, AD and DF are EPs of \mathcal{D}_1 . Indeed, $gr_1(AD, \mathcal{D}) = 0.75/0.25 = 3$ and $gr_1(DF, \mathcal{D}) = 0.25/0 = \infty$. Conversely, AB is not an EP: $gr_1(AB, \mathcal{D}) = 0.50/0.25 = 2 (< \rho)$. When X is not present in $\mathcal{D} \setminus \mathcal{D}_i$ (i.e., $supp(X, \mathcal{D} \setminus \mathcal{D}_i) = 0$), we get $gr_i(X, \mathcal{D}) = \infty$ and such a pattern is called a *jumping emerging pattern* (JEP). For instance, DF is a jumping emerging pattern in \mathcal{D}_1 . Unless otherwise indicated, we consider that the growth rate of a pattern X must be higher than 1 in order that X is an EP. The minimal growth rate constraint has no good properties of monotonicity, but effective methods to mine EPs are proposed in literature [2], [10], [12].

B. Summarizing Contrasts of a Database

In practice, the theory is often a huge collection of patterns and experts cannot directly analyze this massive output. For instance, even with the small dataset *wine* and $\gamma = 10\%$ (see experiments in Section V), there are 852 emerging patterns having a growth rate greater than 3 and 472 *closed* emerging patterns [17] which remains a too large collection. Increasing support or growth rate thresholds is not satisfactory: the mined patterns become too general and are often trivial.

Moreover, the rise of the minimal growth rate threshold does not reduce enough the number of patterns. For instance, there are still 215 JEPs in *wine* with $\gamma = 10\%$.

Another well-known problem about pattern mining is the choice of measure thresholds (e.g., minimal support, minimal growth rate). Firstly, the number of mined patterns with given thresholds deeply depends on the studied dataset. It is then difficult to set the proper values of the thresholds and several attempts are often necessary to tune them. Second, thresholds are usually managed with a crisp effect [13]. The patterns having a measure value slightly over the threshold are selected whereas those with a measure slightly under are rejected. The thresholds induce a boundary which is too strongly marked. Typically, selecting emerging patterns with $\rho = 3$ probably lead to miss interesting patterns with a growth rate around 2.9 and below ρ .

Summarizing the contrasts of datasets is a way to cope with these difficulties. For that purpose, we define a method based on the key idea of recursive data mining. Our method produces a concise and relevant summarization of patterns such as EPs called *summary of contrasts*:

Definition 2 (k-Summary of contrasts): Given an integer $k > 0$ and a transactional dataset \mathcal{D} divided into n classes $\mathcal{D}_1, \dots, \mathcal{D}_n$, a summary of contrasts is a representative set of at most k patterns characterizing data classes.

A k -summary of contrasts is a concise description of the dataset relevant with respect to the different classes. About conciseness, our method guarantees that the size of a k -summary does not exceed k patterns. Moreover, k can be chosen as small as wanted so that an analyzable and understandable result is obtained. Another advantage is that this parameter has a clear meaning and it is easier to fix than usual measure thresholds. In order to be representative, the summary has to cover a large part of the dataset. The selected EPs have to be sufficiently general in order that most transactions contain at least one EP characterizing the transaction class. In this paper, we chose EPs for capturing contrasts, but our method deals with any other interestingness measure. The innovative recursive mining process is presented in the next section and Section IV depicts the summary of contrasts made of EPs named *recursive emerging patterns*.

III. RECURSIVE PATTERN MINING

This section presents the framework of recursive pattern mining and gives essential properties for the following.

A. Framework of Recursive Pattern Mining

The key idea of recursive pattern mining is to reduce the output by successively repeating the mining process in order to preserve the most significant patterns. More precisely, for each step, the previous result becomes the new dataset (i.e., the mined patterns constitute the new dataset). This recursive process is ended as soon as the result becomes stable. More

formally, the aim of recursive pattern mining is to find the *recursive theory* defined as below:

Definition 3 (Recursive theory): Let $\mathcal{L}_{\mathcal{I}}$ be a language, \mathcal{D} be a dataset and $(q_j)_{j \geq 1}$ be a constraint sequence, the recursive theory, denoted by $\mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, (q_j)_{j \geq 1})$, is the limit (if it exists) of the sequence $(\mathcal{D}^j)_{j \geq 1}$ given by:

$$\begin{cases} \mathcal{D}^1 = \mathcal{D} \\ \mathcal{D}^{j+1} = \mathcal{Th}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}^j, q_j), j \geq 1 \end{cases}$$

This definition of recursive theory relies on the definition of the sequence of datasets $(\mathcal{D}^j)_{j \geq 1}$. For any $j \geq 1$, the dataset \mathcal{D}^{j+1} contains exactly all the patterns of $\mathcal{L}_{\mathcal{I}}$ satisfying q_j and occurring in the previous dataset \mathcal{D}^j . There is no assumption on the constraints q_j , these constraints can be the same or different. Patterns belonging to $\mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, (q_j)_{j \geq 1})$ are named *recursive patterns*. Intuitively, recursive patterns retrace the phenomena occurring in each theory $\mathcal{Th}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}^j, q_j)$. It may happen that the recursive theory is not defined because the sequence $(\mathcal{D}^j)_{j \geq 1}$ does not converge to a limit. Property 1 (see below) gives an important result about this issue showing that in practice this sequence converges in most of the cases.

Table II illustrates the recursive pattern mining with the constraint sequence $(\text{supp}(X, \mathcal{D}^j) \geq 0.25)_{j \geq 1}$. As the support depends on the cardinality of the dataset \mathcal{D}^j , the constraint $\text{supp}(X, \mathcal{D}^j) \geq 0.25$ evolves during the process. The first step \mathcal{D}^2 corresponds to the theory $\mathcal{Th}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}^1, \text{supp}(X, \mathcal{D}^1) \geq 0.25)$ (where $\mathcal{D}^1 = \mathcal{D}$). On the left, this theory \mathcal{D}^2 is represented as a transactional dataset and contains the patterns A (first line), AF (second line), etc. The second step computes \mathcal{D}^3 which gives the patterns satisfying $\text{supp}(X, \mathcal{D}^2) \geq 0.25$ in \mathcal{D}^2 . Property 1 ensures us that \mathcal{D}^3 is the recursive theory.

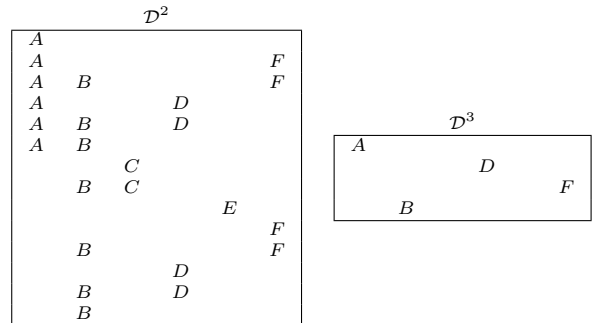


Table II
MINING OF THE RECURSIVE THEORY $\mathcal{D}^3 = \mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, (\text{supp}(X, \mathcal{D}^j) \geq 0.25)_{j \geq 1})$

Obviously, the support alone is not a measure highlighting the contrasts. With the example in Table II, the recursive frequent patterns B and F do not characterize data classes \mathcal{D}^1 or \mathcal{D}^2 (e.g., $gr_1(B, \mathcal{D}) = gr_2(B, \mathcal{D}) = 1$). In order to summarize contrasts of dataset \mathcal{D} , Section IV instantiates

the recursive pattern mining framework with both support and growth rate measures.

B. Scope and Properties

The framework of recursive pattern mining is very general. The choice of the constraint sequence enables us to extract many suitable kinds of patterns according to the application. The next theorem proves that the recursive pattern mining encompasses the pattern mining:

Theorem 1: Let $\mathcal{L}_{\mathcal{I}}$ be a language, \mathcal{D} be a dataset and q be a constraint, there exists a constraint sequence $(q_j)_{j \geq 1}$ such that the recursive theory $\mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, (q_j)_{j \geq 1})$ equals to the theory $\mathcal{Th}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, q)$.

The proof is straightforwardly obtained by choosing the constraint sequence with $q_1 \equiv q$ and $q_{j>1} \equiv \text{true}$. Theorem 1 shows that pattern mining is a particular case of recursive pattern mining. Note that the paradigm of recursive pattern mining is different from that of iterative pattern mining. Iterative mining applies different mining processes to the same original database in order to modify or adjust the approach whereas, in recursive pattern mining, a new database is considered for each step.

A crucial issue of recursive pattern mining is to know whether the recursive theory exists and (if it exists) to compute it. The next property provides a practical way to perform this twofold task:

Property 1 (Practical stability): Let $\mathcal{L}_{\mathcal{I}}$ be a language, \mathcal{D} be a transactional dataset and $(q_j)_{j \geq 1}$ be a constraint sequence, if there exists l such that $\mathcal{D}^l = \mathcal{D}^{l+1}$ and $q_{l'} \equiv q_l$ for all $l' \geq l$, then the recursive theory $\mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, (q_j)_{j \geq 1})$ exists and is \mathcal{D}^l .

This property (the proof is omitted due to lack of space) is very important in practice. Indeed, whenever the constraint sequence is stable and two successive datasets \mathcal{D}^l and \mathcal{D}^{l+1} are equal, Property 1 ensures that \mathcal{D}^l is the desired recursive theory. In other words, as soon as the mining context is not altered, the recursive pattern mining is over. For instance, $\mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, (\text{supp}(X, \mathcal{D}^j) \geq 0.25)_{j \geq 1})$ is given by \mathcal{D}^3 in Table II because frequent patterns in \mathcal{D}^3 are exactly the transactions of \mathcal{D}^3 (i.e., $\mathcal{D}^4 = \mathcal{D}^3$) and $\text{supp}(X, \mathcal{D}^{l'}) = \text{supp}(X, \mathcal{D}^3)$ with $l' \geq 3$.

Other significant properties of recursive pattern mining (a sequence of anti-monotone constraints is stable, inclusion of recursive theory according to $(q_j \wedge q'_j)_{j \geq 1}$ in recursive theory of $(q_j)_{j \geq 1}$, number of steps, etc) are not developed here because they are not essential for the following.

IV. SUMMARIZING CONTRASTS WITH REPS

This section presents our method to build a summary of contrasts thanks to the recursive pattern mining framework. We define the notion of recursive emerging patterns (REPs) and we show that all REPs taken together are a tunable summary.

A. Recursive Pattern Mining of Emerging Patterns

Intuitively, recursive emerging patterns are patterns which frequently occur within the EPs of a class. They rely on the assumption that the most significant EPs are those often occurring within the output, summarizing the main contrasts through the output. The definition of REPs is an instance of the recursive pattern mining framework. Nevertheless, from a technical point of view, the REP mining is slightly different because it builds in parallel n sub-datasets \mathcal{D}_i^j linked together. Each sub-dataset \mathcal{D}_i^j describes the i^{th} class of the j^{th} mining step and is required for computing the growth rate of patterns in the i^{th} class.

Following on, the j^{th} dataset \mathcal{D}^j is always divided into several sub-datasets \mathcal{D}_i^j such that $\mathcal{D}^j = \bigcup_{i \in \{1, \dots, n\}} \mathcal{D}_i^j$. For clarity, we also introduce the new constraint $\text{is-an-EP}(X, \mathcal{D}, \rho) = \bigvee_{i \in \{1, \dots, n\}} \text{gr}_i(X, \mathcal{D}) \geq \rho$ which selects emerging patterns occurring in \mathcal{D} and having a growth rate exceeding ρ in at least one class. Now we give a more formal definition of REPs (which is a particular case of Definition 3):

Definition 4 (Recursive emerging patterns): Let $\mathcal{L}_{\mathcal{I}}$ be a language, \mathcal{D} be a transactional dataset, recursive emerging patterns are the recursive theory $\mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}^j, (\text{supp}(X, \mathcal{D}^j) \geq \gamma \wedge \text{is-an-EP}(X, \mathcal{D}^j, \rho))_{j \geq 1})$ which corresponds to the limit of the sequence $(\mathcal{D}^j)_{j \geq 1}$:

$$\begin{cases} \mathcal{D}^1 = \mathcal{D} & \text{(a)} \\ \forall i \in \{1, \dots, n\}, \mathcal{D}_i^1 = \mathcal{D}_i & \text{(b)} \\ \mathcal{D}^{j+1} = \mathcal{Th}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}^j, \text{supp}(X, \mathcal{D}^j) \geq \gamma \wedge \text{is-an-EP}(X, \mathcal{D}^j, \rho)), j \geq 1 & \text{(c)} \\ \forall i \in \{1, \dots, n\}, \mathcal{D}_i^{j+1} = \{X \in \mathcal{D}^{j+1} \mid \text{gr}_i(X, \mathcal{D}^j) \geq \rho\}, j \geq 1 & \text{(d)} \end{cases}$$

Definition 4 states that REPs are frequent emerging patterns recursively mined starting from the different classes. The final recursive theory, called *REP theory*, depends on thresholds γ and ρ . Lines a and b initialize the process. In particular, Line b distinguishes each class. Line c builds the $(j+1)^{\text{th}}$ dataset by mining frequent patterns (i.e., $\text{supp}(X, \mathcal{D}^j) \geq \gamma$) which are also an EP from one class to others (i.e., $\text{is-an-EP}(X, \mathcal{D}^j, \rho)$). Finally, Line d discriminates the EPs of \mathcal{D}^{j+1} according to their class (i.e., $\mathcal{D}_i^{j+1} = \mathcal{Th}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}^j, \text{supp}(X, \mathcal{D}^j) \geq \gamma \wedge \text{gr}_i(X, \mathcal{D}^j) \geq \rho)$). A pattern may appear in several datasets \mathcal{D}_i^j , but obviously it occurs once in dataset \mathcal{D}^j which is a set.

Table III depicts the mining of REPs from data given in Table I with $\gamma = 0.1$ and $\rho = 2$. The process requires only three steps including the checking of stability. Indeed, as EPs from \mathcal{D}^3 are exactly the patterns of \mathcal{D}^4 , the constraint sequence $(\text{supp}(X, \mathcal{D}^j) \geq \gamma \wedge \text{is-an-EP}(X, \mathcal{D}^j, \rho))_{j \geq 1}$ is stable for any $j \geq 3$. Thereby, Property 1 guarantees that the REP theory is extracted (this result is generalized to any REP theory as soon as there exists j with $\mathcal{D}^{j+1} = \mathcal{D}^j$). The datasets \mathcal{D}_1^2 and \mathcal{D}_2^2 exactly correspond to EPs from \mathcal{D} with

$\gamma = 0.1$ and $\rho = 2$. We can observe that the REP theory (i.e., union of \mathcal{D}_1^3 and \mathcal{D}_2^3 : 9 patterns) is smaller than the theory of EPs (i.e., union of \mathcal{D}_1^2 and \mathcal{D}_2^2 : 22 patterns). In the next section, Theorem 2 shows that we can fix the maximal number of patterns belonging to the REP theory.

| \mathcal{D}_1^2 | | | | \mathcal{D}_2^2 | | | |
|-------------------|---|---|---|-------------------|---|---|---|
| A | | | E | A | C | | |
| A | | D | E | A | B | C | |
| A | | D | | A | B | C | F |
| A | B | D | | A | | C | F |
| A | | D | | | | C | |
| A | B | D | | | | C | E |
| A | | D | | | | C | F |
| | | D | E | B | C | | F |
| | | D | | B | C | D | |
| | B | D | | B | C | D | |
| | B | D | | B | C | | |

| \mathcal{D}_1^3 | | | | \mathcal{D}_2^3 | | | |
|-------------------|---|---|---|-------------------|---|--|---|
| A | | D | E | A | C | | |
| | | D | | | C | | F |
| | B | D | | B | C | | |
| | | D | | | C | | |
| | | D | | | C | | |

Table III
REPs MINED FROM \mathcal{D} WITH $\gamma = 0.1$ AND $\rho = 2$

B. REP k -Summary

This section defines the REP k -summary and describes its main features: tunable concision and representativeness. We start by introducing the notion of *recursive emerging pattern (REP) k -summary* which is closely linked to REP theory:

Definition 5 (REP k -summary): Let $k > 0$, the REP k -summary with ρ is the REP theory with $\gamma = 1/k$ and ρ (if it exists): $\mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}^j, (supp(X, \mathcal{D}^j) \geq \gamma \wedge is-an-EP(X, \mathcal{D}^j, \rho)))_{j \geq 1}$.

A REP k -summary corresponds to a REP theory with particular thresholds. Theorem 2 explains the key role of the parameter k to bound the size of the summary.

Let us illustrate this notion with our running example. Table IV provides the REP 10-summary with $\rho = 2$ (patterns are indicated in this table with their support and growth rate in the initial transactional dataset \mathcal{D}). As $\gamma = 1/10$, this summary is exactly the REP theory given by Table III. Interestingly, the growth rate values of the patterns of the REP k -summary may be lower than ρ (e.g., D is under the threshold since $gr_1(D, \mathcal{D}) = 1.5$). Nevertheless, except E , all the patterns are emerging patterns (i.e., their growth rate exceeds 1) and most of them have a significant growth rate. This avoids the crisp effect of a threshold with usual pattern mining where a promising pattern is deleted only because its value for the measure is just under the threshold (cf. Section II-B). Experiments in Section V confirm this phenomenon.

We give now the most important theorem showing that the cardinality of a REP k -summary does not exceed k patterns:

Theorem 2 (Maximal size k): The REP k -summary with ρ contains at most k patterns.

| REPs of \mathcal{D}_1 | | | REPs of \mathcal{D}_2 | | |
|-------------------------|--------|----------|-------------------------|--------|----------|
| REP | $supp$ | gr_1 | REP | $supp$ | gr_2 |
| AD | 0.5 | 3 | AC | 0.125 | ∞ |
| E | 0.25 | 1 | CF | 0.125 | ∞ |
| DF | 0.125 | ∞ | BC | 0.25 | ∞ |
| BD | 0.375 | 2 | C | 0.5 | 3 |
| D | 0.625 | 1.5 | | | |

Table IV
REP 10-SUMMARY OF \mathcal{D} WITH $\rho = 2$

Proof: Let $S = \mathcal{RTh}(\mathcal{L}_{\mathcal{I}}, \mathcal{D}, (supp(X, \mathcal{D}^j) \geq \gamma \wedge is-an-EP(X, \mathcal{D}^j, \rho)))_{j \geq 1}$ be a REP theory. If $|S| = 0$, obviously $|S| \leq 1/\gamma$. Otherwise, there exists a maximal pattern $X \in S$. As X is maximal and S is a theory, X occurs in only one transaction of S because S is a set: $freq(X, S) = 1$. Besides, $S = \mathcal{Th}(\mathcal{L}_{\mathcal{I}}, S, (supp(X, S) \geq \gamma \wedge is-an-EP(X, S, \rho)))$ and $supp(X, S) \geq \gamma$. As $supp(X, S) = freq(X, S)/|S|$ and $freq(X, S) = 1$, we obtain that $1/|S| \geq \gamma$. We conclude that $|S| \leq 1/\gamma$. If γ is set at $1/k$, the recursive theory S contains at most k itemsets. ■

In other words, to ensure that a summary does not exceed k contrasts, it is enough to fix the minimal support threshold γ at $1/k$. Thus it is very easy to set the parameter of the mining process according to the number of patterns desired by the end-user. For instance, the REP summary of Table IV contains 9 patterns (fewer than 10 since $\gamma = 1/0.1$). Theorem 2 is intuitively coherent because decreasing the support threshold leads to mine more recursive patterns, as usual in pattern mining.

More generally, the cardinality of a recursive theory is independent from the initial dataset. For instance, the recursive theory shown in Table II contains only four itemsets with \mathcal{D} . With the same parameters (i.e., $\gamma = 0.25$), the recursive theory of larger datasets will not return more than 4 ($= 1/0.25$) itemsets. This point clearly differentiates recursive pattern mining from usual pattern mining. As said in Section II-B, the number of output patterns is really unknown in advance with usual constraint-based pattern mining. We infer that the constraint sequence appears as a global constraint handling the recursive theory. The constraint sequence puts together patterns and gives a global sense to them. For this reason, we think that recursive patterns are promising patterns having a nature which differs from the usual patterns.

The next property justifies why REPs have an important support and explains their representativeness:

Property 2 (Frequent patterns): A recursive emerging pattern is frequent in the transactional context \mathcal{D} .

Proof: Let X be a REP. We can note that X appears at least in one transaction Y of dataset \mathcal{D}^2 (otherwise it would not be mined in the following steps). As any transaction of \mathcal{D}^2 is a frequent pattern of \mathcal{D} , we have $supp(Y, \mathcal{D}) \geq \gamma$. Then, we conclude that $supp(X, \mathcal{D}) \geq \gamma$ because $X \subseteq Y$ and $supp$ decreases. ■

This property guarantees a good individual representative-

ness of each REP. For instance, in the REP 10-summary given by Table IV, the support of any pattern exceeds 0.1 which is considerable. In fact, the process ensures that REPs are frequent patterns in any dataset \mathcal{D}^j . Thus, they are representative not only of \mathcal{D} , but also of \mathcal{D}^1 which gathers all the emerging patterns from \mathcal{D} exceeding ρ . From an abstract point of view, REPs can be seen as generalizations of emerging patterns.

V. EXPERIMENTS

The aim of the experiments is to evaluate both the quantitative and qualitative benefits brought by REP k -summaries on benchmarks and real-world applications. In all our experiments, the collections of recursive patterns converge and thus the REP theories are defined, it illustrates the fact that the recursive paradigm properly runs in practice. Moreover, the most important time-consuming step is the first mining. Therefore, mining recursive patterns is not significantly more expensive than usual pattern mining.

A. Benchmarks

The used datasets are benchmarks available on UCI repository¹. In this section, we evaluate the size of REP 10-summaries, the growth rate of REPs and the number of steps per summary. Figure 1 reports information on the processes for computing REP summaries according to various thresholds. Each histogram plots the number of patterns per step. The last step is the REP summary, its number of patterns gives the REP summary size. Quality of summaries is estimated by computing the growth rate of all the patterns with respect to the initial dataset. Patterns are categorized in 4 groups. The black (resp. white) color designates the weakest (resp. the strongest) contrasts. Note that the y-axis is a logarithmic scale. The class coverage is indicated under each histogram (the class coverage is the number of transactions containing at least one contrast (of its own class) among the summary divided by the cardinality of the dataset).

Obviously, the size of REP 10-summaries does not exceed 10 patterns (see Theorem 2). In practice, most of them have a number of patterns close to 10. This number also depends on the minimal growth rate threshold. In all cases, the REP summary appears as a very compact representation in comparison to the collection of emerging patterns (obtained by the first mining step). In particular, the REP 10-summary of *ionosphere* with $\rho = 3$ is very concise in comparison with the 100,000 EPs.

An important result is that the growth rates of REPs may be lower than ρ (see for example the REP 10-summaries of *wine* with EPs having a growth rate lower than the threshold $\rho = 6$). This phenomenon is due to the several steps of the pattern mining which aim at finding a trade-off between

high growth rates and good representativeness. Thus, this method overcomes the crisp effect of minimal growth rate threshold mentioned in Section II-B. From an abstract point of view, each step can be seen as a generalization of the previous one. Then, the recursive theory ensures a global coverage of the original dataset. Indeed, the class coverage of any REP 10-summary always exceeds 25% of transactions (even with *abalone* dataset which contains more than 4000 transactions).

B. Application: SAGE Data Analysis

In the context of genomic data, the study of simultaneous expression of thousands of genes is requested by biologists to characterize classes of biological situations. In this section, we outline how REP summaries contribute to discover few potential relevant genes which may be associated to cancer from SAGE (Serial Analysis of Gene Expression) data. The dataset provides the level-expression of 27679 genes in 90 biological situations. Gene expressions are quantitative values and the property of overexpression has to be encoded for each gene. Biological situations (resp. overexpressed gene) correspond to transactions (resp. items). Besides, biological situations are divided into two classes: *cancer* and *no cancer*. 59 situations are labeled *cancer* and 31, *no cancer* (i.e., normal situations).

| Sequence | Description of cancer | <i>supp</i> | <i>gr</i> |
|------------|---|-------------|-----------|
| CATCCAAAAC | HNRPH1 Heterogeneous nuclear ribonucleoprotein H1 (H) | 0.28 | 2.10 |
| CTCTTCGAGA | GPX1 Glutathione peroxidase 1 | 0.32 | 3.28 |
| CTTGCTGCC | NIFIE14 Seven transmembrane domain protein | 0.26 | 2.50 |

Class coverage : 40%

Table V
REP 4-SUMMARY OF SAGE DATA WITH $\rho = 2$

Table V provides the REP 4-summary with $\rho = 2$. At first, we observe that all the patterns describe the class *cancer*. Other values of k and ρ also lead to characterize only the class *cancer*. The three mined patterns characterize 40% of biological situations and 61% of cancerous situations. Interestingly, this REP summary confirms the results obtained in [18] with characterization rules. Indeed, genes HNRPH1 and GPX1 may have an influence on the development of cancer. In particular, the expression of GPX1 has been found in several studies to be correlated with cancerous situations [19]. The interest of our approach is to directly isolate the same genes associated to cancer without requiring a manual inspection of rules.

C. Application: Biomedical Text Mining

Recursive pattern mining is a general paradigm and also can be used with other kinds of data (e.g., sequences) and with various constraints. In this section, we sketch the

¹www.ics.uci.edu/~mllearn/MLRepository.html

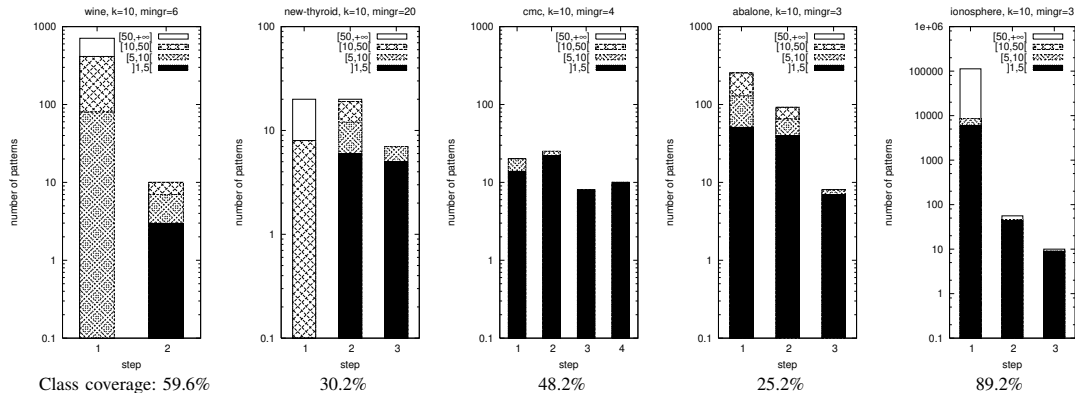


Figure 1. REP 10-summaries of UCI benchmarks

use of recursive pattern mining to support the detection of interactions between genes in biological and medical papers.

Literature on biology and medicine represents a huge amount of knowledge (e.g. more than 19 million publications are available in PubMed repository). A critical challenge is then to extract, from these text collections, relevant and useful knowledge such as linguistic patterns characterizing gene interactions. To this end, recursive *sequential* pattern mining with specific constraints was successfully applied [20]. In this work, the constraint $is-an-EP(X, \mathcal{D}^j, \rho)$ was replaced by constraints derived from expert’s prior (i.e. “an interaction must contain at least two genes” and “an interaction must contain a verb or a noun”). In addition, a maximality constraint was added to ensure the termination of the recursion and to give prominence to longer sequences.

In this application, recursive pattern mining enables us to return a manageable output to the expert who can then easily validate or discard each of recursive patterns. Indeed, even if around 65,000 patterns satisfied the constraints, the recursive pattern mining returns only 667 recursive patterns that were examined in 90 minutes by an expert. At the end, 232 sequential recursive patterns representing several forms of interactions between genes were discovered. Among those patterns, some explicitly convey interactions. For instance, `AGENE@np interact@vvz with@in AGENE@np` or `activation@nn of@in AGENE@np by@in AGENE@np`. Other patterns represent more general interactions between genes, meaning that a gene plays a role in the activity of another like `AGENE@np involve@vvn in@in AGENE@np` and `AGENE@np play@vvz role@nn in@in of@in AGENE@np`. Note that these verbs do not belong to available word lists devoted to interaction discovery.

Finally, the validated patterns were then considered as information extraction rules. Their application in classification gave good results to detect gene interactions (i.e. precision=0.83, recall = 0.75 and f-score = 0.79) and categorize them (i.e. precision= 0.88, recall = 0.69 and f-score = 0.77) [20].

VI. RELATED WORK

In Section II, we have mentioned that our problem differs from mining local contrasts [10], [11], [1], [12] which returns large collections of patterns in comparison with a k -summary. That is why Hu et al. [21] propose to find the k patterns having the best support. In our context, discovering the k patterns optimizing the growth rate is harmful because we would obtain again all the JEPs. Other mining techniques exist to restrain the number of mined patterns. Most of them are dedicated to frequent patterns and then, they are useless to summarize contrasts. Siebes et al. [7] compress the dataset by exploiting Minimum Description Length Principle, but their method does not take into account several classes and it still returns numerous patterns. Mielikäinen and Mannila [6], Yan et al. [9] and Wang and Parthasarathy [8] discover k representative patterns with probabilistic models for summarizing/approximating only *frequent* patterns.

Classifiers based on local patterns (e.g., classification association rules [22] or EPs [3]) have also been proposed in literature. In the same way that predictive and descriptive rules are different [1], *predictive* models like classifiers [23] do not target the same goals as *descriptive* models like summaries. In particular, the size of such models is not tunable and remains too large.

Our summarization problem is very similar to the redundancy-aware top- k approach proposed by Xin et al. [14], which find k patterns satisfying a trade-off between high-significance (according to an interestingness measure) and low-redundancy. The originality of our work is to propose a new data mining technique based on a recursive process providing a correct and complete mining whereas all the previous summarization approaches use heuristics and greedy algorithms. Therefore, our proposal can be extended to other languages (e.g., sequences) by changing $\mathcal{L}_{\mathcal{I}}$ in the recursive theory. To the best of our knowledge, recursive data mining has received little attention. The closest work to our framework, done by Szymanski and Zhang [24], mines

frequent patterns and encodes them for rewriting the original dataset. This process is repeated several times. The recursive pattern mining formulates a different view where the mined patterns directly constitute the new dataset. Furthermore, our framework provides a more general method since we can adapt the constraint sequence according to the application.

VII. CONCLUSION

We presented a novel approach for summarizing contrasts using recursive pattern mining. We introduced the paradigm of recursive pattern mining and studied its convergence. This framework keeps patterns satisfying a given sequence of constraints. Our summarization method ensures to discover at most k recursive emerging patterns. Such patterns capture frequent contrasts stemming from the different mining steps. Experiments show that recursive pattern mining is a viable option for reducing the number of mined patterns and focusing on the most relevant ones. In particular, the success of our approach on biological data analysis indicates that REP k -summaries are useful in real-world applications.

In the future, we want to perform further experiments on other real-world applications with more complex data (e.g., trees or graphs). Furthermore, recursive pattern mining opens a new direction on finding both various and significant patterns which may lead to promising uses. We particularly intend to exploit recursive pattern mining for highlighting exceptions embedded in data as for instance [25].

REFERENCES

- [1] P. K. Novak, N. Lavrac, and G. I. Webb, "Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining," *Journal of Machine Learning Research*, vol. 10, pp. 377–403, 2009.
- [2] G. Dong and J. Li, "Efficient mining of emerging patterns: discovering trends and differences," in *Proc. of KDD*. ACM Press, 1999, pp. 43–52.
- [3] G. Dong, X. Zhang, L. Wong, and J. Li, "CAEP: Classification by aggregating emerging patterns," in *Discovery Science*, 1999, pp. 30–42.
- [4] J. Li and L. Wong, "Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns," in *Genome Informatics 18*, 2002, pp. 725–734.
- [5] L. De Raedt and A. Zimmermann, "Constraint-based pattern set mining," in *Proc. of SDM*. SIAM, 2007.
- [6] T. Mielikäinen and H. Mannila, "The pattern ordering problem," in *Proc. of PKDD*, 2003, pp. 327–338.
- [7] J. Vreeken, M. van Leeuwen, and A. Siebes, "Krimp: mining itemsets that compress," *Data Min. Knowl. Discov.*, vol. 23, no. 1, pp. 169–214, 2011.
- [8] C. Wang and S. Parthasarathy, "Summarizing itemset patterns using probabilistic models," in *Proc. of KDD*. ACM, 2006, pp. 730–735.
- [9] X. Yan, H. Cheng, J. Han, and D. Xin, "Summarizing itemset patterns: a profile-based approach," in *Proc. of KDD*. ACM, 2005, pp. 314–323.
- [10] J. Bailey, T. Manoukian, and K. Ramamohanarao, "Fast algorithms for mining emerging patterns," in *Proc. of PKDD*, 2002, pp. 39–50.
- [11] J. Li, G. Liu, and L. Wong, "Mining statistically important equivalence classes and delta-discriminative emerging patterns," in *Proc. of KDD*. ACM, 2007, pp. 430–439.
- [12] A. Soulet and B. Crémilleux, "Adequate condensed representations of patterns," *DMKD*, vol. 17, no. 1, pp. 94–110, 2008.
- [13] S. Bistarelli and F. Bonchi, "Soft constraint based pattern mining," *Data Knowl. Eng.*, vol. 62, no. 1, pp. 118–137, 2007.
- [14] D. Xin, H. Cheng, X. Yan, and J. Han, "Extracting redundancy-aware top-k patterns," in *Proc. of KDD*. ACM, 2006, pp. 444–453.
- [15] H. Mannila and H. Toivonen, "Levelwise search and borders of theories in knowledge discovery," *Data Min. Knowl. Discov.*, vol. 1, no. 3, pp. 241–258, 1997.
- [16] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB*. Morgan Kaufmann, 1994, pp. 487–499.
- [17] G. C. Garriga, P. Kralj, and N. Lavrac, "Closed sets for labeled data," *Journal of Machine Learning Research*, vol. 9, pp. 559–580, 2008.
- [18] B. Crémilleux, A. Soulet, J. Klema, C. Hébert, and O. Gandrillon, "Discovering knowledge from local patterns in sage data," in *Data Mining and Medical Knowledge Management: Cases and Applications*. IGI Global publications, 2009, pp. 251–267.
- [19] M. Nasr, M. Fedele, K. Esser, and A. Diamond, "GPx-1 modulates akt and p70s6k phosphorylation and gadd45 levels in mcf-7 cells," *Free Radical Biology and Medicine*, vol. 37, no. 2, pp. 187–195, 2004.
- [20] P. Cellier, T. Charnois, and M. Plantevit, "Sequential patterns to discover and characterise biological relations," in *Proc. of CICLing*, 2010, pp. 537–548.
- [21] A. W.-C. Fu, R. W. w. Kwong, and J. Tang, "Mining n -most interesting itemsets," in *ISMIS*, 2000, pp. 59–67.
- [22] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proc. of KDD*, 1998, pp. 80–86.
- [23] A. J. Knobbe and E. K. Y. Ho, "Pattern teams," in *Proc. of PKDD*, 2006, pp. 577–584.
- [24] B. K. Szymanski and Y. Zhang, "Recursive data mining for masquerade detection and author identification," in *3rd Annual IEEE Information Assurance Workshop*, 2004.
- [25] W. Duivesteijn, A. J. Knobbe, A. Feelders, and M. van Leeuwen, "Subgroup discovery meets bayesian networks – an exceptional model mining approach," in *ICDM*, 2010, pp. 158–167.