Steady Patterns

Willy Ugarte, Alexandre Termier, Miguel Santana

Steady Patterns

Willy Ugarte
LIG Laboratory - Slide Team
University of Grenoble Alpes
Grenoble, France
Email: willy.ugarte-rojas@imag.fr

Alexandre Termier
IRISA lab - Lacodam team
University of Rennes 1
Rennes, France
Email: alexandre.termier@irisa.fr

Miguel Santana
STMicroelectronics
Crolles, France
Email: miguel.santana@st.com

***Abstract*—Skypatterns are an elegant answer to the pattern explosion issue, when a set of measures can be provided. Skypatterns for all possible measure combinations can be explored thanks to recent work on the skypattern cube. However, this leads to too many skypatterns, where it is difficult to quickly identify which ones are more important. First, we introduce a new notion of pattern *steadiness* which measures the conservation of the skypattern property across the skypattern cube, allowing to see which are the "most universal" skypatterns. Then, we extended this notion to partitions of the dataset, and show in our experiments that this both allows to discover especially stable skypatterns, and identify interesting differences between the partitions.**

I. Introduction

Pattern mining is an important field of Data Mining, whose goal is to detect *regularities* in the data. It has been successfully for applications as diverse as market basket analysis in supermarkets, debugging with execution trace for embedded systems [1] and understanding characteristics of carcinogenic molecules [2].

However, such success stories rely on an important investment of human expert time. The reason is that pattern mining suffers from the so-called *pattern explosion*: there is a huge number of output patterns of mixed interest, requiring tedious manual inspection. One of the promising solution to this problem is the notion of *skypatterns* [3]. Inspired by *skyline queries* from the database community, the idea of skypatterns is to consider a set of measures of interest for the user, and to output only patterns that are not dominated on any of these measures: these are the skypatterns. Skypatterns are thus a solution to a multi-criteria optimization problem, and are the Pareto front over the proposed measures. There are usually very few skypatterns compared to the total number of patterns, and the measures help in the understanding of their interest.

A difficulty is to consider the many possible combination of measures over which a pattern can be a skypattern. This difficulty is addressed with the notion of *skypattern cube* recently proposed [4]. The skypattern cube is the lattice of all possible measure combinations, with for each measure combinations its skypaterns. The authors of [4] have shown that this allowed to determine, for each combination of measure, its *proper skypatterns*: patterns which are skypatterns only for the given combination of measures, and not for the others.

A natural question arises: what about patterns that are skypatterns on most combinations of measures? This includes top-level skypatterns (i.e. patterns that are skypatterns for all combinations of measures), but also patterns that are skypatterns for many combinations of measures, and fall short to be skypatterns on other combinations of measures, for sake of robustness. Such patterns are interesting, because they cannot be dominated on most the combinations of measures considered, and can thus be seen as "universal dominants". This characteristic allow them to give a good summary of the most important elements of the dataset over the proposed measures. We call these patterns *steady patterns*.

Steady patterns express a notion of robust global dominance over the measures, and are thus expected to be especially important in the data according to the selected measures. This leads to another question: if there exists natural ways to partition the dataset (by a class, or along a temporal attribute), are the "global" steady patterns also steady over the partitions? If yes, the "steadiness" of these patterns is not affected by the partitioning, they thus have some property of stability over the data which is worth reporting. If patterns are only steady on some partitions, this can help to understand the different characteristics of these partitions. For example, if the partitioning of the data is temporal, it can be interesting to see if a global steady pattern is steady on most periods of time except one: it can quickly pinpoint this period of time as different, prompting a look at the change in measures and possibly at more details of the data.

This paper proposes answers to the questions above, through the following contributions:

- We define a robust *measure of steadiness*, which allows to evaluate how much a pattern verifies the skypattern property over all combinations of measures;
- We extend this definition to a dataset and its partitions, with a measure of *local-aware steadiness* which indicates the stability of a pattern over the partitions of the dataset;
- We propose efficient methods to compute both measures;
- We perform a thorough experimental study over real datasets, which helps to understand the nature of steady patterns and show their interest for real data analysis.

This paper is organized as follows. After introducing the background in Section II, we present in Section III our definitions of steady patterns. We then present in Section IV our method to compute steady patterns from the skypattern cube. We discuss related work in Section V. Section VI presents the experiments and finally we conclude.

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F

Item	A	B	C	D	E	F
Price ($)	30	40	10	40	70	55
Weight (Kg)	4	6	2	0.5	1	2

TABLE I
TRANSACTIONAL DATASET $\mathcal{T}$.

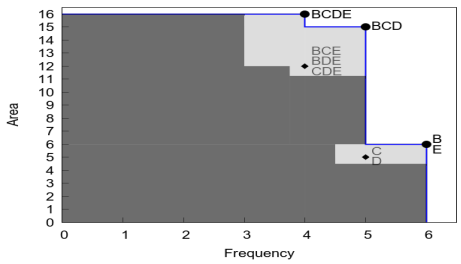


Fig. 1. (Thick)-Skypatterns for M=$\{\texttt{freq}, \texttt{area}\}$ with $\epsilon = 0.25$.

II. BACKGROUND

We introduce in this section the definitions of skypattern and skypattern cube that are necessary for the rest of this paper.

A. Context and Definitions

Let $\mathcal{I}$ be a set of distinct literals called *items*. An itemset (or pattern) is a non-null subset of $\mathcal{I}$. The language of itemsets corresponds to $\mathcal{L}_\mathcal{I} = 2^\mathcal{I} \backslash \{\emptyset\}$. A transactional dataset $\mathcal{T}$ is a multiset of patterns of $\mathcal{L}_\mathcal{I}$ (i.e. multiset of transactions). Table I depicts a transactional dataset $\mathcal{T}$ where each transaction t_i is described by items denoted $A, \ldots, F$. The traditional example is a supermarket database in which each transaction corresponds to a customer and every item in the transaction is a product bought by the customer. An attribute (*price*) is associated to each product (see Table I).

Constraint-based pattern mining aims to mine all patterns $\mathtt{x}$ of $\mathcal{L}_\mathcal{I}$ satisfying a query $q(\mathtt{x})$ (i.e. conjunction of constraints) which is called *theory* [5]: $Th(q) = \{\mathtt{x} \in \mathcal{L}_\mathcal{I} \mid \ q(\mathtt{x}) \text{ is true}\}$. A common example is the minimal frequency constraint ($q(\mathtt{x}) \equiv \texttt{freq}(\mathtt{x}) \geq \min_{fr}$). Formally, $\texttt{freq}(\mathtt{x}) = |T(\mathtt{x})|$ where $T(\mathtt{x})$ is the support of $\mathtt{x}$ (i.e $T(\mathtt{x}) = \{t \in \mathcal{T} \mid \ \mathtt{x} \subseteq t\}$). The latter provides patterns $\mathtt{x}$ having a number of occurrences in the dataset exceeding a given minimal threshold $\min_{fr}$. There are other usual measures for a pattern $\mathtt{x}$:

- $\texttt{area}(\mathtt{x}) = \texttt{freq}(\mathtt{x}) \times \texttt{size}(\mathtt{x})$.
- $\texttt{min}(\mathtt{x}.att)$ (resp. $\texttt{max}(\mathtt{x}.att)$) is the lowest (resp. highest) value of the set of item values of $\mathtt{x}$ for attribute att.
- $\texttt{mean}(\mathtt{x}.att) = (\texttt{min}(\mathtt{x}.att) + \texttt{max}(\mathtt{x}.att))/2$.

Example 1. For the dataset in Figure I, $\texttt{freq}(BC)$=5, $\texttt{area}(BC)$=10 and $\texttt{mean}(BCD.price)$=25.

B. Skypatterns and Skypattern Cube

1) Skypatterns: Skypatterns enable to express a user-preference point of view according to a dominance relation [3].

Definition 1 (Pareto-Dominance). Given a set of measures M, a pattern $\mathtt{x}_i$ dominates another pattern $\mathtt{x}_j$ w.r.t. M (noted by $\mathtt{x}_i \succ_M \mathtt{x}_j$), iff $\forall m \in M, m(\mathtt{x}_i) \geq m(\mathtt{x}_j)$ and $\exists m \in M, m(\mathtt{x}_i) > m(\mathtt{x}_j)$.

Example 2. From $\mathcal{T}$ and with M=$\{\texttt{freq}, \texttt{area}\}$, BCD dominates BC as $\texttt{freq}$ (BCD)=$\texttt{freq}$ (BC)=5 and $\texttt{area}(BCD)$>$\texttt{area}(BC)$ (see Figure I).

Definition 2 (Skypattern and skypattern operator). Given a set of measures M and a dataset $\mathcal{T}$, a skypattern w.r.t. M is a pattern not dominated w.r.t. M. The skypattern operator $Sky(\mathcal{T}, M)$ returns all the skypatterns w.r.t. M for $\mathcal{T}$: $Sky(\mathcal{T}, M) = \{\mathtt{x}_i \in \mathcal{L}_\mathcal{I} \mid \nexists \mathtt{x}_j \in \mathcal{L}_\mathcal{I}, \mathtt{x}_j \succ_M \mathtt{x}_i\}$

Example 3. Let M=$\{\texttt{freq}, \texttt{area}\}$, we have that $Sky(\mathcal{T}, M)$=$\{BCDE, BCD, B, E\}$ and Figure 1 provides its graphical representation. The dark gray area is the *forbidden area* since it cannot contain any (thick-)skypattern. The light gray area is the neighborhood of the skypatterns. The other part is called the *dominance area*. The edge of this area (blue line) marks the boundary between them.

2) Soft-skypatterns: Skypatterns can be stringent, thus soft-skypatterns have been recently introduced by [6] capturing patterns that do not strictly satisfy the Pareto-dominance, but are nevertheless valuable.

Simply, by changing the dominance relation we can mine different types of soft-skypatterns. For instance, we can easily extend the notion of thick-skylines [7] to obtain thick-skypatterns. Thick-skylines [7] are non-skyline points that are close in a ϵ distance to a skyline point.

Definition 3 (Strongly Dominating Relation [7]). Let M a set of measures, a pattern $\mathtt{x}_i$ strongly dominates another pattern $\mathtt{x}_j$ w.r.t. M (noted by $\mathtt{x}_i \succ^\epsilon_M \mathtt{x}_j$), iff $\forall m \in M, (1-\epsilon) \times m(\mathtt{x}_i) \geq m(\mathtt{x}_j) \wedge \exists m' \in M, (1-\epsilon) \times m'(\mathtt{x}_i) > m'(\mathtt{x}_j)$.

Definition 4 (Thick-skypattern and Thick-skypattern operator). Given a set of measures M and a dataset $\mathcal{T}$, a thick-skypattern w.r.t. M is a pattern not strongly dominated w.r.t. M and is close for a distance ϵ to a skypattern. The thick-skypattern operator $Thick\text{-}Sky(\mathcal{T}, M)$ returns all the thick-skypatterns:

$$Thick\text{-}Sky(\mathcal{T}, M) = \{\mathtt{x}_i \in \mathcal{L}_\mathcal{I} \mid \nexists\, \mathtt{x}_j \in \mathcal{L}_\mathcal{I}, \mathtt{x}_j \succ^\epsilon_M \mathtt{x}_i) \wedge (\exists \mathtt{x}_j \in Sky(\mathcal{T}, M), \epsilon\text{-}neighbors(\mathtt{x}_i, \mathtt{x}_j))\}$$

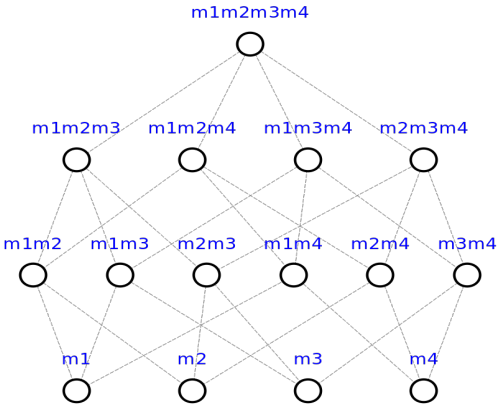
where $\epsilon\text{-}neighbors(\mathtt{x}_i, \mathtt{x}_j) = \forall m \in M, \frac{|m(\mathtt{x}_i) - m(\mathtt{x}_j)|}{\max\{m(\mathtt{x}_i), m(\mathtt{x}_j)\}} \leq \epsilon$

Example 4. Let M=$\{\texttt{freq}, \texttt{area}\}$ and ϵ=0.25, thus $Thick\text{-}Sky(\mathcal{T}, M)$=$Sky(\mathcal{T}, M) \cup \{BCE, BDE, CDE, C, D\}$. Figure 1 provides its graphical representation (e.g. BDE is a thick-skypattern because there exists the skypattern BCD and $\epsilon\text{-}neighbors(BCD, BDE)$).

3) Skypattern Cube: Skypattern cube [4] for a set of measures M consists of the $2^{|M|}-1$ skypattern sets on all possible non empty subsets $M_u \subseteq M$.

Definition 5 (Skypattern Cube). Let M be a set of measures. $SkyCube(M) = \{(M_u, Sky(\mathcal{T}, M_u)) \mid M_u \subseteq M, M_u \neq \emptyset\}$.

Example 5. Figure 2a depicts the lattice associated to M. Second column in Figure 2b associates to each non-empty subset of M its skypattern set.



(a) Lattice associated to M.

Subset of M	Global Skypatterns	Local Skypatterns	
		$\mathcal{T}_1$	$\mathcal{T}_2$
$\{m_1, m_2, m_3, m_4\}$	{BCDEF, BCDE, BCD, BDE, BEF, BE, BF, B, EF, E}	{BEF, BF, B, EF, E, F}	{BCDEF, BCDE, BCEF, BCF, BDEF, BDE, BEF, BF, EF, E}
$\{m_1, m_2, m_3\}$	{BCDE, BCD, BDE, BE, EF, E}	{EF, E}	{BCDE, BDE, E}
$\{m_1, m_2, m_4\}$	{BCDEF, BCDE, BCD, BEF, BF, B}	{BEF, BF, B, EF}	{BCDEF, BCDE, BCEF, BCF}
$\{m_1, m_3, m_4\}$	{BEF, BE, BF, B, EF, E}	{BEF, BF, B, EF, E, F}	{BEF, BF, EF, E, BDE, BE}
$\{m_2, m_3, m_4\}$	{BCDEF, BCDE, BDE, BEF, BF, EF, E}	{BEF, BF, EF, E}	{BCDEF, BCDE, BCEF, BCF, BDEF, BDE, BEF, BF, EF, E}
$\{m_1, m_2\}$	{BCDE, BCD, B, E}	{EF}	{BCDE}
$\{m_1, m_3\}$	{E}	{E}	{E}
$\{m_1, m_4\}$	{BF, B}	{BF, B}	{BCF, BF, BCDE, BCD, BCE, BC, BDE, BD, BE, B}
$\{m_2, m_3\}$	{BCDE, BDE, EF, E}	{EF, E}	{BCDE, BDE, E}
$\{m_2, m_4\}$	{BCDEF, BCDE, BEF, BCF, BF}	{BEF, BF, EF}	{BCDEF, BCDE, BCEF, BCF}
$\{m_3, m_4\}$	{BEF, BF, EF, E}	{BEF, BF, EF, E}	{BEF, BF, EF, E}
$\{m_1\}$	{B, E}	{B, EF, E, F}	{BCDE, BCD, BCE, BC, BDE, BD, BE, B, CDE, CD, CE, C, DE, D, E}
$\{m_2\}$	{BCDE}	{EF}	{BCDE}
$\{m_3\}$	{E}	{E}	{E}
$\{m_4\}$	{ABF, BCF, BF}	{ABCF, ABF, BCF, BF}	{ABCF, ABF, BCF, BF}

(b) Skypattern cube for M.

Fig. 2. $M = \{m_1 : \texttt{freq(x)}, m_2 : \texttt{area(x)}, m_3 : \texttt{mean(x.}price\texttt{)}, m_4 : \texttt{mean(x.}weight\texttt{)}\}$.

III. Steady patterns

We can now propose our main contributions, which are the measures of steadiness.

A. Steadiness

In [4], the main issue of skypattern cube is that the number of measure combinations is exponential w.r.t the number of measures, even if they proposed a concise representation of the cube with equivalence classes to regroup nodes in the measure lattice (similarly to closed and free pattern in a pattern lattice). Thanks to skypattern cube we can extract hundreds or thousands of patterns ensuring that they are Pareto-optimal w.r.t at least one combination of measures, but this amount of patterns could be untreatable by a human-user. In this case, steadiness for a skypattern refers to its nature as skypatterns through all the cube, in other words a skypattern is steady (resp. unsteady) if it is a skypattern or close to be (e.g. thick-skypatterns) for many (resp. few) measure combinations.

Since in the skypattern cube there are $2^{|M|} - 1$ subsets of measures, we define Steadiness of a skypattern for the cube.

Definition 6 (Steadiness of a pattern). Given a set of measures M and a dataset $\mathcal{T}$. The steadiness of a pattern $\texttt{x}$ is the average steadiness of $\texttt{x}$ for each subset of measures $M_u \subseteq M$.

$$\texttt{steadiness}(\texttt{x}, \mathcal{T}) = \left(\sum_{M_u \subseteq M} \texttt{std}(\texttt{x}, M_u, \mathcal{T}) \right) / \left(2^{|M|} - 1\right)$$

$$\text{where } \texttt{std} : \mathcal{L}_{\mathcal{I}} \times \{2^M \setminus \emptyset\} \times \{dataset\} \rightarrow [0, 1]$$

$$\texttt{std}(\texttt{x}, M_u, \mathcal{T}) = \begin{cases} 1 & \text{if } \texttt{x} \in Sky(\mathcal{T}, M_u) \\ (1-\epsilon) & \text{if } \texttt{x} \in Thick\text{-}Sky(\mathcal{T}, M_u) \\ 0 & \text{otherwise} \end{cases}$$

B. Local-awareness

1) Locality: As we already defined a transactional dataset $\mathcal{T}$ is a multiset of transactions. A local sub-dataset of $\mathcal{T}$ is sub-multiset of transactions of $\mathcal{T}$. A local-aware dataset $\mathcal{T}^p = \{\mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_n\}$, corresponding to $\mathcal{T}$, is a set of n disjoint local sub-datasets $\mathcal{T}_i$ of $\mathcal{T}$ with $1 \leq i \leq n$.

Example 6. From dataset $\mathcal{T}$ from Figure I and with $n = 2$, we obtain its local-aware dataset $\mathcal{T}^p = \{\mathcal{T}, \mathcal{T}_1, \mathcal{T}_2\}$ where $\mathcal{T}_1 = \{t_1, t_2, t_3\}$ and $\mathcal{T}_2 = \{t_4, t_5, t_6, t_7\}$.

Thus, instead of studying a dataset $\mathcal{T}$ only as a whole, we are going to study its corresponding local-aware dataset $\mathcal{T}^p$. We extend the notion of skypattern cube to this kind of dataset.

Definition 7 (Local and Global Skypattern and Local-aware Skypattern Operator). Given a set of measures M and a local-aware dataset $\mathcal{T}^p = \{\mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_n\}$. A skypattern w.r.t. M is local (resp. global) iff is a skypattern w.r.t. M in any of the local sub-datasets of $\mathcal{T}_i$ (resp. in the global dataset $\mathcal{T}$). The local-aware skypattern operator $Sky^p(M)$ returns all the local and global skypatterns w.r.t. M:

$$Sky^p(M) = Sky(\mathcal{T}, M) \cup \bigcup_{1 \leq i \leq n} Sky(\mathcal{T}_i, M)$$

Example 7. For the local-aware dataset $\mathcal{T}^p$ in Example 6 and a set of measures M=$\{freq, area\}$, we have that $Sky^p(M)$=$\{BCDE, BCD, B, E, EF_1, BCDE_2\}$ as $Sky(\mathcal{T}, M)$=$\{BCDE, BCD, B, E\}$, $Sky(\mathcal{T}_1, M) = \{EF\}$ and $Sky(\mathcal{T}_2, M)$=$\{BCDE\}$.

Definition 8 (Local-aware Skypattern Cube). Let M be a set of measures.

$SkyCube^p(M) = \{(M_u, Sky^p(M_u)) \mid M_u \subseteq M, M_u \neq \emptyset\}$.

Example 8. Consider the dataset in Figure I. As in Example 5, we have the same lattice from Figure 2a. In Figure 2b, for each non-empty subset of M its local-aware skypattern set is the union from second to fourth columns.

2) Local-aware Steadiness: The notion of steadiness (see Definition 6) can be extended for a local-aware dataset, in order to estimate a measure of steadiness based on steadiness values for different partitions of the dataset.

Definition 9 (Local-aware Steadiness of a pattern). Given a set of measures M and a local-aware dataset $\mathcal{T}^p = \{\mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_n\}$. The local-aware steadiness of a pattern $\texttt{x}$ is the average local-aware steadiness of $\texttt{x}$ for every subset of measures $M_u \subseteq M$.

$$\texttt{steadiness}^p(\texttt{x}, \mathcal{T}^p) = \frac{\sum_{M_u \subseteq M} \texttt{std}^p(\texttt{x}, M_u, \mathcal{T}^p)}{2^{|M|} - 1}$$

where $\texttt{std}^p : \mathcal{L}_\mathcal{I} \times \{2^M \setminus \emptyset\} \times \mathcal{T}^p \rightarrow [0,1]$

$$\texttt{std}^p(\texttt{x}, M_u, \mathcal{T}^p) = 0.5 \times \left(\texttt{std}(\texttt{x}, M_u, \mathcal{T}) + \sum_{1 \leq i \leq n} \frac{\texttt{std}(\texttt{x}, M_u, \mathcal{T}_i)}{n} \right)$$

In the rest of this paper, we denote as steady (resp. unsteady) patterns the k skypatterns with highest (resp.lowest) steadiness, for a given value of k. As we are going to show in experiments steady skypatterns are interesting because they show skypatterns with a stability through a local-aware cube, and unsteady skypatterns are interesting because they have few occurrences as skypatterns through a local-aware cube (i.e. they occur exceptionally or under special conditions).

IV. Computation

This section presents our approach for computing steady skypatterns by using the local-aware skypattern cube. The key idea is to compute the local-aware skypattern cube by adapting existing methods for a regular skypattern cube. Then we compute the steady skypatterns from this local-aware cube.

A. Computing Skypattern Cube

In the best of our knowledge there are two methods to extract directly the skypattern cube from a dataset.

1) Regular Methods: [4] relies in the relations between nodes of the measures lattice, defining derivation rules in order to avoid computing the whole skypattern set for a node by deriving most of the skypatterns for a node from its child nodes. And [8] applies a more direct approach by computing a superset of the cube (i.e. edge-skypatterns), to finally filter the skypattern cube.

2) Generalizing the Regular Methods for Periodic Patterns: The regular methods extract skypatterns w.r.t to a set measures directly but there are other types of patterns that add an extra layer, besides mining patterns they have to satisfy a condition that cannot be modeled as a measure.

For instance, periodic patterns can be directly mined [1]. Even if it could seem easy to modeling period as a measure for patterns, it's not the case. A pattern can have different period values regarding its occurrences, meaning that we can have different instances of periodic patterns corresponding to a single mined pattern. Unlike frequent, emerging or closed patterns to which we have one value by pattern.
In this case, we can define a general method:

a) Mining periodic patterns with an *ad hoc* tool.

b) Post-processing the output with an SKYCUBE extracting tool [9]. SKYCUBE is a multidimensional generalization of skylines extracting all skylines sets for a set of dimensions.

3) From SKYCUBE *to skypattern cube:* This section shows how the problem of computing a skypattern cube w.r.t. a set of measures M can be converted into an equivalent problem of computing a skyline cube in $|M|$ dimensions.

Let f be a mapping from $\mathcal{L}_\mathcal{I}$ to $\mathbb{R}^k$ that associates, to each pattern $\texttt{x} \in \mathcal{L}_\mathcal{I}$, a data point $f(\texttt{x}) \in \mathbb{R}^k$ with coordinates $(m_1(\texttt{x}), m_2(\texttt{x}), \ldots, m_k(\texttt{x}))$. Let $P = \{f(\texttt{x}) \mid \texttt{x} \in \mathcal{L}_\mathcal{I}\}$. P is a multiset: let $\texttt{x}_i$ and $\texttt{x}_j$ s.t. $\texttt{x}_i \neq \texttt{x}_j$. If $\texttt{x}_i$ and $\texttt{x}_j$ are indistinct w.r.t. M then $f(\texttt{x}_i) = f(\texttt{x}_j)$.

Pattern	m_1	m_2	m_3	m_4
ABC	2	6	25.00	3.00
ABDEF	1	5	50.00	3.25
⋮	⋮	⋮	⋮	⋮
BCD	5	15	25.00	3.00
⋮	⋮	⋮	⋮	⋮
BF	3	6	47.50	4.00
B	6	6	40.00	3.00
E	6	6	70.00	0.50

TABLE II
MULTIDIMENSIONAL VIEW OF PATTERNS FOR M IN FIGURE 1.

Example 9. Table II reports the mapping between all the patterns in $\mathcal{T}$ and data points of $\mathbb{R}^4$ ($|M| = 4$). $f(B)$ (resp. $f(BCD)$) is the data point with coordinates $(6, 40, 6, 40)$ (resp. $(5, 15, 25, 3)$).

B. Computing Local-aware Skypattern Cube

In the case of a local-aware dataset $\mathcal{T}^p = \{\mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_n\}$, we need to compute the local-aware skypattern cube.

This could be done easily, by computing n+1 skypattern cubes (each one for each dataset in $\mathcal{T}^p$) and unifying these cubes into a single one. By applying the method described in precedent sections (see Sections IV-A1, IV-A3) to each dataset $\mathcal{T}_i \in \mathcal{T}^p$, we can obtain n+1 skypattern cubes.

Unifying these n+1 skypattern cubes is an easy task. Foreach set of measures M_u (i.e. node in the cube) we regroup all skypattern sets $Sky(\mathcal{T}_i, M_u)$, in other words $\forall \mathcal{T}_i \in \mathcal{T}^p$, $\bigcup_{M_u \subseteq M} Sky(\mathcal{T}_i, M_u)$) that corresponds to the local-aware skypattern set $Sky^p(M_u)$ (see Definition 8).

Finally we regroup these local-aware skypattern sets into form the local-aware skypattern cube (see Definition 8).

C. Computing Steady Skypatterns

For all extracted skypatterns we compute their steadiness w.r.t. the obtained local-aware cube (see Definition 6) and take the k skypatterns with the highest (resp. lowest) steadiness i.e. steady (resp. unsteady) skypatterns. This could be seen as a top-k of steady skypatterns, but using as score (or interestingness measure) the steadiness of the mined skypatterns.

V. Related Work

There are two methods to compute the skypattern cube: [4] propose a bottom-up approach based on derivation rules exploiting the relation between the nodes in the measure lattice and [8] that proposes a more direct aproximation-based method, that computes a superset of the whole set of skypatterns in the skypattern cube, then filter edge-skypatterns that are not present in the skypattern cube.

Although these methods are not directly adaptable for periodic pattern mining because the period is not a straightforward measure over the patterns but it is taken into account the complexity of the problem (i.e $P \times \mathcal{L}_\mathcal{I}$ where $P \in \{1, n\}$) furthermore for more complex notions as Periodic Concepts [1].

VI. Experimental Evaluation

In this section we show the results for our experiments over two study cases (execution trace analysis and weather analysis). In each case we show: i) a quantitative analysis for CPU-time and space, and ii) a qualitative analysis, first only with global steadiness (see Definition 6) and after showing the benefit of adding local-awareness (see Definition 9)

A. Experimental protocol

All experiments were conducted on a computer running Linux with a core i5 processor at 2.13 GHz. Our method consists of the following steps:

- Preprocessing the logfile into a transactional dataset.
- Mining periodic patterns: we use PerMiner [1].
- Computing the skypattern cube: we use Orion [9].
- Post-processing the output by computing its steadiness.

1) Preprocessing of logfiles into transactional datasets: In these experiments, our data always have the form of a logfile. A logfile is a sequence of events, where each event is associated with a timestamp. This temporal component will allow use to exploit periodic patterns, with allow to have a rich set of measures for skypattern computation.

From a practical point of view, pattern mining algorithms aim to find common patterns in a transactional dataset Therefore, in order to make logfiles exploitable by pattern mining algorithms (e.g. PerMiner), the logfile has to be split into a sequence of sets of events. In literature there are many possible methods to split a logfile, for instance for execution traces the most common method is a **Time Interval Method**.

Example 10. In Figure 3 we have the result of splitting a logfile using a time interval of 0.1 ms.

2) Measures: Now we describe some of the measures that we use for computing the steady patterns for our study cases. An important advantage of using the skypattern cube is that we can combine heterogeneous measures, in other words in one query we can have measures based on (Periodic) Pattern Mining, on Statistics or on Expert Knowledge.

(i) Classical Periodic Pattern Mining Measures

- *Frequency:* This measure is to maximize because frequent event sets help to detect unusual behaviors (e.g. a hard-drive writing function frequently is followed by an interruption).
- *Size:* This measure is to maximize because user would like large patterns having the most possible combinations of events together in order to detect anomalies in their performance.
- *Period:* The constant distance between some occurrences of a pattern x, formally noted as per(x). This measure is to minimize to obtain event sets that appear often periodically in the trace.

(ii) Statistical Measures

- *% of Rupture of periodicity:* The relative gap between the average distance among the occurrences of a pattern and the period computed by PerMiner. Formally, %Rupture(x) This measure is to maximize because we would like to obtain patterns that seem periodic (in a subset of their occurrences) but not for all of their occurrences.

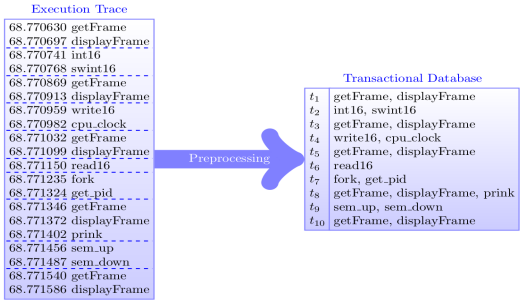


Fig. 3. Splitting an execution trace into time intervals of 0.0001 sec (0.1 ms).

(iii) Knowledge-based Measures: We will detail these measures in each subsection depending on the study case.

B. Steady skypatterns for Trace Analysis

In this section, we report an experimental evaluation on datasets obtain from execution traces. The aim of Execution trace analysis is to trace any event that happens during the execution of a software application and carry out a post-mortem analysis of the execution trace. This allows to detect subtle timing bugs between several hardware or software components. These are not functional bugs and thus cannot be detected with traditional debugging software, whose intrusiveness changes the timings of the different components.
In our case we use a trace that has been obtained from a STMicroelectronics board equipped with a dual core SMP ARM [1] used in a set-top box. The trace has been collected while the box presented failures when displaying a video stream from the HDMI port.

1) Events and Processes: This trace contains various types of low-level events generated by the operating system:
- **I-**$\{i\}(p)$: Interruption number $\{i\}$ generated by process p.
- **C**$\{id_1\}$**-**$\{id_2\}$: Switch from process id_1 to process id_2.
- **E-**$\{f\}$ / **X-**$\{f\}$: Enter/Exit syscalls of a function f.

These events are generated by various running processes:
- 0: Main process.
- 1402 **(irq/140-vsync0)**: Interrupt handler process for interruption-140 (vertical synchronization in main screen).
- 1403 **(irq/141-vsync1)**: Interrupt handler process for interruption-141 (vertical sync in secondary screen).
- 1577 **(irq/138-display)**: Interrupt handler process for interruption-138 (display in main screen).

2) Measures: Now we describe the **knowledge-based measures** commonly used in execution trace analysis, in addition to the other measures described in Section VI-A2.
- *% of CPU Usage:* The percentage of CPU-time in the processors used by event in a pattern x. %CPU(x) is the ratio between the sum of all durations for all x occurrences and the whole duration of the trace. This measure is to maximize because a pattern x with a high value of %CPU(x) its a pattern consuming a lot of CPU-time of the processors.

[1]http://www.st.com/content/st_com/en/products/digital-set-top-box-ics/uhd-set-top-box-processors/stih312.html

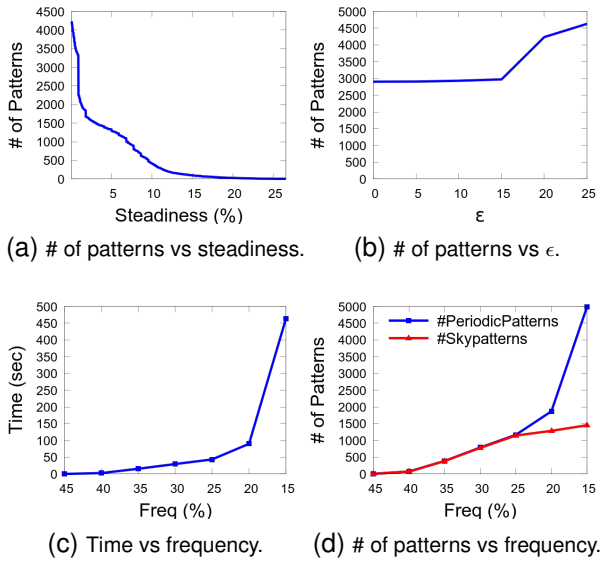


Fig. 4. Quantitative analysis (Computing time and number of patterns).

- *Number of Interruptions:* The maximal number of interruptions that occur between the beginning and the end of a pattern `x` in any of the occurrences of `x`.
This measure is to maximize because a pattern `x` with a high value of `#Int(x)` is a pattern having events that trigger a lot of interruptions consuming CPU-time and probably hinders the usual working of other events.
- *Delay between two events:* The average delay (i.e. gap) between two given events appearing in a pattern `x` for all its occurrences. In this case, e_1=I-140(0) and e_2=I-141(1402) because these interruptions for vertical synchronization in both screens are not supposed to happen independently.
This measure is to maximize because a pattern `x` with a high value of $\texttt{delay}(\texttt{x}, e_1, e_2)$ is a pattern where these two events e_1, e_2 occur with a big time gap meaning that two events supposed to occur sequentially have an unusual behavior.

3) Quantitative analysis: Figure 4a shows the evolution of number of patterns by varying the steadiness threshold. The number of steady patterns decreases rapidly as soon as steadiness threshold increases meaning that there are a few steady patterns. Figure 4b shows the evolution of number of patterns by varying ϵ (i.e. neighborhood size for thick-skypatterns). The number of thick-skypatterns does not increase dramatically even for values over 20%. Figure 4c shows the evolution of computing time by varying the frequency threshold (in terms of percentage). Our approach remains effective even with low frequency thresholds. Figure 4d shows the evolution of number of patterns by varying the frequency threshold (in terms of percentage). The number of skypatterns is much lower than the number of periodic patterns making them easier to handle.

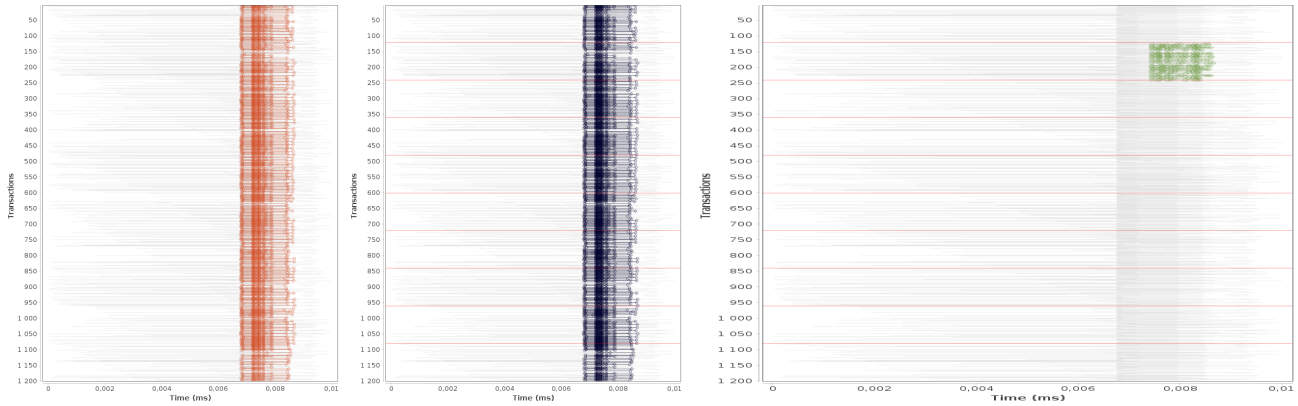
4) Qualitative analysis: In this section we analyze the results of (un)steady patterns in HDMI blackout trace.
a) Global Steady Skypatterns In this section we apply the global steadiness (see Definition 6). Figure 5a shows the distribution over the dataset of the most steady pattern:

p923 (C0-1577 C0-2272 C1403-1402 C1577-0 C72-0 I-138(0) I-140(0) I-141(1402)). Each grey horizontal line represents events in a time window, and the occurrences of this pattern are the orange lines.
This pattern is interesting for analysis because it contains events that are directly linked to display. For instance, it contains the display-related interruptions 138, 140 and 141 and switches between their corresponding handler process 1577, 1402 and 1403. There are many other patterns that contain partially these events but this one is the largest. Also, this pattern has peaks of `%Rupture` which means that even if this events are expected to happen periodically together in this kind of trace, sometimes they are cut or replaced by other switches events which breaks the periodicity of the pattern in some regions. Finally, this pattern has a peak of `delay`, meaning that both events (I-140(0) and I-141(1402)) occur too separately when they were not supposed to.
b) Local-aware Steady Skypatterns In this section we apply the local-aware steadiness (see Definition 9). Figure 5b shows the distribution over the dataset (and the 10 subdatasets divided with red lines) of the most local-aware steady pattern: **p954 (C0-1402 C0-1577 C0-2272 C1403-1402 C1577-0 C72-0 I-138(0) I-140(0) I-141(1402))**. The occurrences of this pattern (blue lines) in the whole dataset (i.e. all the trace).
Similarly to the most global steady pattern p923, this pattern also contain events that are directly linked to display but adding some more switches. Unlike p923, we can also study the nature of this pattern according to the regions. For instance, in region 5 this pattern has no peak for `delay` meaning that for this region there was no problems for vertical sync.
c) Local-aware Unsteady Skypatterns For the least local-aware steady skypattern we have a tie between various patterns which are thick-skypatterns for a measure set in a specific region. In order to illustrate, we only take one **p5084 (C1402-18 C1402-2276 C1402-2278 C1403-1402 C1403-72 C18-0 C2275-1402 C2278-2275 I-141(1402))** which is a thick-skypattern in region 2. Figure 5c shows the distribution over the dataset of the least steady pattern by showing their occurrences of each pattern (color lines) in the whole dataset (i.e. all the trace), and the 10 subdatasets are divided with red lines (corresponding to the time windows).
This pattern is a skypattern for $\{\texttt{size}, \texttt{freq}\}$ is mostly composed by switches from/to process 1402 (**irq/140-vsync0**) meaning that in this region this interruption has many interaction with other processes.

C. Steady skypatterns for Weather

In this section, we report an experimental evaluation for weather data. The dataset has been obtained from *Canadian historical weather data*(http://climate.weather.gc.ca), and we use the data for Montreal in year 2012 giving the weather state hourly for the whole year.

1) Measures: Now we describe the **knowledge-based measures** commonly used in weather analysis, in addition to the other measures described in Section VI-A2.



(a) p923 (Global) (b) p954 (Local-aware) (c) p5084 (Local-aware)

Fig. 5. Dataset view of the most (un)steady patterns.

For some measures we are going to use a relative gap to an optimal value, now we define this gap.

$$\texttt{rel\_gap}(v,o,m,M) = \begin{cases} \frac{v-m}{o-m} & \text{if } v < o \\ \frac{M-v}{M-o} & \text{otherwise} \end{cases}$$

where v is an input value, o an optimal value for v, m minimal possible value for v and M maximal possible value for v.

- *% of Humidity:* The percentage of average optimal humidity for an event in a pattern `x`. Formally, `%Hum(x)`=$\texttt{rel\_gap}(\texttt{avg}(\texttt{x}.hum), 40, 100, 18)$ where v is the average humidity for `x`, the optimal value for humidity (o) is set to 40% because the good conditions of humidity for life are about between 30% and 50%, the minimal (resp. maximal) humidity (m) (resp. M) registered in Montreal in 2012 is 18% (resp. 100%). This measure is to minimize because a pattern `x` with a low value of `%Hum(x)` is a pattern having an humidity far from the optimal living conditions.

- *% of Temperature:* The percentage of average optimal temperature for an event in a pattern `x`. Formally, `%Temp(x)`=$\texttt{rel\_gap}(\texttt{avg}(\texttt{x}.temp), 25, 33, -23.3)$ where v is the average temperature for `x`, the optimal value for temperature (o) is set to $25°C$ because the good conditions of temperature for living are about between $23°C$ and $27°C$, the minimal temperature (m) registered in Montreal in 2012 is $-23.3°C$ and the maximal temperature (M) registered in Montreal in 2012 is $33°C$ This measure is to minimize because a pattern `x` with a low value of `%Temp(x)` its a pattern having values of temperature far from the optimal living conditions.

- *Wind Speed:* The average wind speed of a pattern `x`. Formally, `wd-spd(x)`=$\texttt{avg}(\texttt{x}.windspeed)$. This measure is to maximize because a pattern `x` with a high value of `wd-spd(x)` contains high-risk events (e.g. tornadoes).

- *Visibility:* The average visibility of a pattern `x`. Formally, `vis(x)` = $\texttt{avg}(\texttt{x}.visibility)$. This measure is to maximize because a pattern `x` with a high value of `vis(x)` contains dangerous events in the city (e.g. thunderstorms).

2) Quantitative analysis: Figure 7a shows the evolution of number of patterns by varying the steadiness threshold. Similarly, the number of steady skypatterns is quickly decreases as soon as steadiness threshold increases. Figure 7b shows the evolution of number of mined patterns by varying the frequency threshold (in terms of percentage). This dataset is very particular because it contains a low number of periodic patterns which could lead to a negligible difference between the number of periodic patterns and the number of skypatterns.

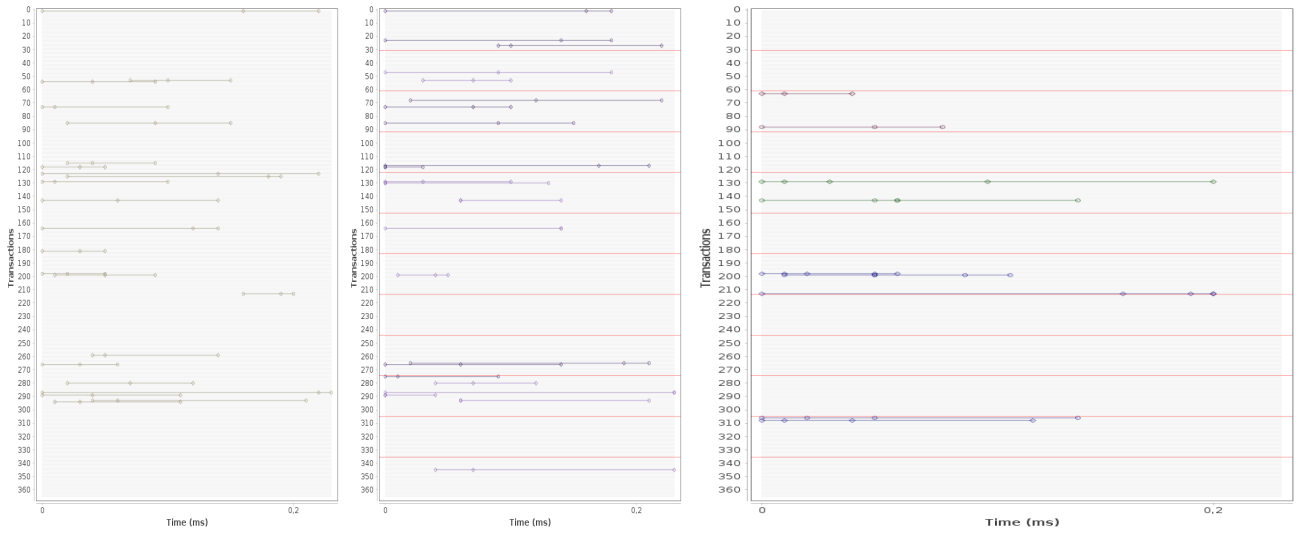
3) Qualitative analysis: In this section we analyze the results of (un)steady patterns in weather log data.

a) Global Steady Skypatterns: In this section we apply the global steadiness (see Definition 6). Figure 6a shows the distribution over the dataset of the most steady pattern: **p7 (Fog Mosty-Cloudy Rain-Showers)** which is a skypattern for $\{\texttt{size}, \texttt{freq}, \texttt{\%Temp}, \texttt{\%Hum}, \texttt{vis}\}$ and $\{\texttt{freq}, \texttt{\%Temp}, \texttt{\%Hum}, \texttt{vis}, \texttt{\%Rupture}\}$. When this pattern occur it's periodic but with interval according to the month of the year. And as expected Fog and Rain-Showers have high humidity, low temperature and low visibility.

b) Local-aware Steady Skypatterns: In this section we apply the local-aware steadiness (see Definition 9). Figure 6b shows the distribution over the dataset of the most steady pattern **p4 (Fog Mosty-Cloudy Rain)** and its occurrences (blue lines) in the whole year, and the 12 months. Some occurrences have a darker tone meaning that this pattern is a local skypattern for that corresponding subdataset. For example, this pattern is a skypattern for $\{\texttt{freq}, \texttt{\%Hum}, \texttt{\%Rupture}\}$ and $\{\texttt{\%Temp}, \texttt{\%Hum}, \texttt{wd-spd}, \texttt{vis}\}$ globally but locally is a skypattern in January (for $\{\texttt{\%Hum}, \texttt{vis}\}$), April (for $\{\texttt{\%Temp}, \texttt{wd-spd}, \texttt{vis}\}$) and September (for $\{\texttt{\%Temp}, \texttt{wd-spd}, \texttt{\%Rupture}\}$). Meaning that a mostly-cloudy rainy fog have peaks of low visibility and unusual humidity in January, this same effect with unusual temperatures in April and September.

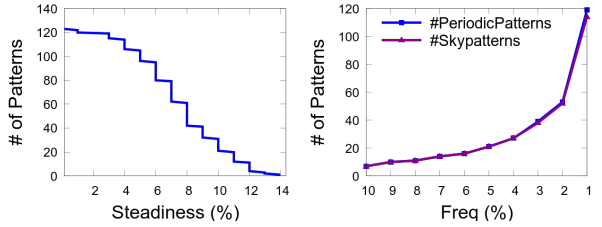
c) Local-Aware Unsteady Skypatterns: Figure 6c shows the distribution over the dataset of the least steady patterns by showing their occurrences (color lines) in the whole year, and the 12 months of the year. In this case we can clearly see that these are patterns that are optimal locally to a subdataset.

1) March p139 (Cloudy Ice-Pellets Snow): A skypattern for $\{\texttt{\%Temp}\}$ and $\{\texttt{wd-spd}\}$. This was expected because March



(a) p7 (Global) (b) p4 (Local-aware) (c) p159 (Local-aware)

Fig. 6. Dataset view of the most (un)steady patterns.



(a) # of patterns vs steadiness. (b) # of patterns vs frequency.

Fig. 7. Quantitative analysis (Computing time and number of patterns).

can reach very low temperatures but extraordinarily in 2012 overnight lows for the first time in winter broke $-20°C$. Which partially explains Ice-Pellets instead of Snow Storms.

2) May p145 (Drizzle Fog Mosty-Cloudy Rain Rain-Showers): A skypattern for $\{\texttt{\%Hum}, \texttt{wd-spd}\}$ and $\{\texttt{\%Hum}, \texttt{vis}\}$. In May 29, after two days of rain unleashing 120 mm of water, the streets turned into mud-choked waterfalls and damaged hundreds of art works at museum.

3) July p155 (Fog Mainly-Clear Mosty-Cloudy Rain-Showers Thunderstorms): A skypattern for $\{\texttt{\%Temp}, \texttt{wd-spd}, \texttt{\%Rupture}\}$ and $\{\texttt{\%Temp}, \texttt{vis}, \texttt{\%Rupture}\}$. Meaning that mostly-cloudy, thunderstorm, fog and rain showers weathers may occur when there are peaks of temperature and wind speed or visibility but happen repeatedly in separated periods of July.

4) November p159 (Drizzle Mosty-Cloudy Rain Rain-Showers): A skypattern for $\{\texttt{\%Temp}, \texttt{\%Hum}, \texttt{\%Rupture}\}$ and $\{\texttt{\%Hum}, \texttt{vis}\}$. Meaning that mostly-cloudy, rain-showers, drizzle weathers occur when there are peaks of humidity or temperature but happen repeatedly in separated periods.

Conclusion: Skypatterns have proved to be an efficient way to discover patterns of interest when given a set of measures. In order to study the impact of the different measure combinations, the skypatterns can be explored through a skypattern cube. In this paper, we went one step further and proposed the notion of steady patterns, which can be seen as "salient points" of the skypattern cube. Their limited number allows to get a quick understanding of the main features of the data according to the given measures. We also extended this notion to datasets that can be partitioned, providing a new way to characterize the partitions based on the measures. The relevance and the effectiveness of our approach have been highlighted through experiments that show that (un)steady patterns can provide interesting insights. In the future, we would like to investigate some promising perspectives. The first one is applicative: our study is focused on temporal datasets, but studying other kinds of datasets, such as medical studies, which have many available measures and possible partitionings. A second one is algorithmic: currently our approach requires to post-process all the patterns of the skypattern cube. For sake of efficiency it could be interesting to directly compute the steady patterns.

References

[1] P. López-Cueva, A. Bertaux, A. Termier, J. Méhaut, and M. Santana, "Debugging embedded multimedia application traces through periodic pattern mining," in *EMSOFT*. ACM, 2012, pp. 13–22.

[2] A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data," in *PKDD*, 2000.

[3] A. Soulet, C. Raïssi, M. Plantevit, and B. Crémilleux, "Mining dominant patterns in the sky," in *ICDM*. IEEE Comp. Society, 2011, pp. 655–664.

[4] W. Ugarte, P. Boizumault, S. Loudni, and B. Crémilleux, "Computing skypattern cubes," in *ECAI*, vol. 263. IOS Press, 2014, pp. 903–908.

[5] H. Mannila and H. Toivonen, "Levelwise search and borders of theories in knowledge discovery," *DMKD.*, vol. 1, no. 3, pp. 241–258, 1997.

[6] W. Ugarte, P. Boizumault, S. Loudni, B. Crémilleux, and A. Lepailleur, "Mining (soft-) skypatterns using dynamic CSP," in *CPAIOR*, 2014.

[7] W. Jin, J. Han, and M. Ester, "Mining thick skylines over large databases," in *PKDD*, vol. 3202. Springer, 2004, pp. 255–266.

[8] W. Ugarte, P. Boizumault, S. Loudni, and B. Crémilleux, "Computing skypattern cubes using relaxation," in *ICTAI*, 2014, pp. 859–866.

[9] C. Raïssi, J. Pei, and T. Kister, "Computing closed skycubes," *PVLDB*, vol. 3, no. 1, pp. 838–847, 2010.