# Up and Down: Mining Multidimensional Sequential Patterns Using Hierarchies

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

LIRMM, Univ. Montpellier 2, CNRS ; 161 rue Ada, 34392 Montpellier, France
{plantevi,laurent,teisseire}@lirmm.fr

**Abstract.** Data warehouses contain large volumes of time-variant data stored to help analysis. Despite the evolution of OLAP analysis tools and methods, it is still impossible for decision makers to find data mining tools taking the specificity of the data (e.g. multidimensionality, hierarchies, time-variant) into account. In this paper, we propose an original method to automatically extract sequential patterns with respect to hierarchies. This method extracts patterns that describe the inner trends by displaying patterns that either go from precise knowledge to general knowledge or go from general knowledge to precise knowledge. For instance, one rule exhibited could be *data contain first many sales of coke in Paris and lemonade in London for the same date, followed by a large number of sales of soft drinks in Europe*, which is said to be *divergent* (as precise results like coke precede general ones like soft drinks). On the opposite, rules like *data contain first many sales of soft drinks in Europe and chips in London for the same date, followed by a large number of sales of coke in Paris* are said to be *convergent*. In this paper, we define the concepts related to this original method as well as the associated algorithms. The experiments which we carried out show the interest of our proposal.

## 1 Introduction

Data warehouses collect large volume of data through time for decision making purpose. As soon as data are described through time, sequential pattern mining is well adapted [1]. Indeed, sequential patterns aim at describing the main trends from a database based on correlations between events through time. However, sequential patterns are mined among only one dimension whereas databases can contain several dimensions. Therefore, they have recently been extended to multidimensional sequential patterns in order to handle this multidimensionality [7, 8, 10]. Even if multidimensional sequential pattern mining provides a better view of source data for decision support, these methods cannot totally take advantage of the framework of multidimensional databases. In particular, they do not consider hierarchies. Note that mining rules at very high levels of granularity leads to trivial rules whereas mining rules at very low levels of granularity is not always possible because the support value would be too low. The algorithm HYPE [9] defined to take hierarchies into account in the extraction of multidimensional sequential patterns has some drawbacks. This approach does not allow to discover

patterns such as *"When coke sales increase in U.K, soft drink sales increase one month later in E.U"*. Indeed, the two multidimensional items of the sequence $(U.K, Coke)$ and $(Europe, Soft\ drink)$ are comparable (*i.e.* $(U.K, Coke)$ is more specific than $(Europe, Soft\ drink)$). Note that mining all possible combinations of items is impracticable because of the size of the search space. To the best of our knowledge, there is no sequence mining approach that proposes to take hierarchy into account in a multidimensional framework such that *comparable* items can appear in the discovered sequences.

In this paper, we propose the concepts of *convergent* and *divergent* multidimensional sequences. They provide a more complete knowledge extraction that is better adapted to the main specificity of multidimensional frameworks. Thus, the generation of patterns is either from general items to specific items or from specific items to general ones in order to limit the number of candidate patterns. These new kinds of multidimensional sequences allow to mine longer sequences by modulating the degree of precision/generalization among them. A convergent sequence goes from general knowledge to precise knowledge. As an example, *"when soft drink sales increase in USA, coke sales increase on the west coast whereas lemonade sales increase on the east cost"* is a convergent sequence. A divergent sequence goes from precise knowledge to general knowledge. For instance, *"many sales of beer in London and of wine in France are followed by many sales of alcoholic drinks in Europe"* is a divergent sequence.

The rest of this paper is organized as follows. Preliminary concepts and related work are described in Section 2. We define the convergent and divergent multidimensional sequences and algorithms that allow their discovery in Section 3. Some experiments carried out on synthetic and real data are reported in Section 4. In the last Section, we give some conclusions and perspectives for future researches.

## 2 Related Work: Multidimensional Sequential Patterns and Hierarchies

Combining several analysis dimensions allows to extract knowledge that well describe the data. [7] was the first work dealing with multidimensional sequential pattern mining. The purchased products are not only described by *date_id* and *customer_id* as in classic sequential pattern mining, but according to a set of dimensions such as *Cust-Grp, City, Cust-Age, etc.*. This approach mines sequences that are defined among only one dimension (*product*). These sequences are described by a multidimensional pattern. Thus, it is impossible to mine combinations of multidimensional pattern through time.

[8] proposes to mine such *inter pattern* multidimensional sequences. Discovered patterns do not only combine several analysis dimensions. These dimensions are combined through temporal dimensions (e.g. time). As an example, in the pattern *"lemonade sales increase in N.Y. then coke sales increase in L.A."*, $NY$ appears before $LA$ and *lemonade* before *coke*.

In [10], the authors mine for sequential patterns in the specific framework of Web Usage Mining. Even though they consider three dimensions (pages, sessions, days), these dimensions are very particular since they belong to a single hierarchized dimension.

This approach provides a better time management but does not fit to multidimensional framework.

Few approaches handle both hierarchy and multidimensionality in sequential pattern mining. In [10] , dimensions are just used to represent time, so multidimensionality is not really handled. HYPE ([9]) allows the mining of sequences that are defined among different levels of hierarchy. HYPE provides the discovery of rules as *"when drink sales increase in Europe, carbonated water exports increase in France whereas soft drink exports increase in USA"* where different levels of hierarchy are present in the multidimensional sequence. However, this proposal cannot extract sequences with items that are defined on the same dimensions but with different granularities such as $(London, Coke)$ and $(Europe, Soft\ drink)$. Indeed, this approach mines multidimensional sequential patterns from the most specific items in order to preserve its scalability.

## 3 CD_M2S Convergent or Divergent Multidimensional Sequential Patterns

In this section, we introduce an original concept. Indeed, human mind often thinks in two different and symmetrical ways. Thinking runs from general to specific or from specific to general. We try to replicate these types of reasoning in the knowledge that we want to extract. We introduce the concept of convergent and divergent sequences. First, we present the preliminary definitions associated to multidimensional sequential patterns and hierarchies. We then describe the convergent and divergent patterns and the associated algorithms.

### 3.1 Preliminary Definitions

Let $SDB$ be a set of *multidimensional data sequences*. Each element of data sequences is defined on a set of $m$ *analysis dimensions* denoted by $D_A$. Each dimension $D_i \in D_A$ is associated with a domain of values, denoted by $Dom(D_i)$. For every dimension $D_i$, we assume that $Dom(D_i)$ contains a specific value denoted by $ALL_i$.

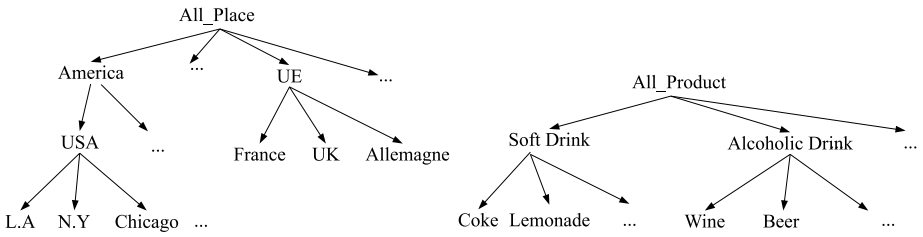
We assume that each dimension $D_i \in D_A$ is associated with a *hierarchy* $H_i$. Every hierarchy $H_i$ is a direct acyclic graph (DAG) whose nodes are elements of $Dom(D_i)$ and whose *root* is $ALL_i$. As usual, the edges of such a DAG can be seen as *is-a* relationships. The *specialization* relation corresponds to a top-down path in $T_i$, *i.e.* a path connecting two nodes when scanning $T_i$ from the root to the leaves. We note that when no hierarchy is defined for a dimension $D_i$, we consider $H_i$ as being a tree whose root is $ALL_i$ and whose leaves are all the elements of $Dom(D_i) \setminus \{ALL_i\}$.

Every element (item) $e_i$ of a multidimensional data sequence is a tuple $t = (d_1, \ldots, d_m)$ such that for every $i = 1, \ldots, m$, $d_i \in Dom(D_i)$ and $d_i$ is a leaf in $H_i$. In other words, data sequences are defined at the finest levels of the hierarchies associated to $D_A$.

The multidimensional sequence database in Table 1 is used to illustrate the different concepts and definitions. It describes the purchases of products carried out in various cities of the world for three different companies identified by an $S_{ID}$. Items of data sequences are defined on two dimensions: *Place* and *Product*. Dimension *Place* is associated to hierarchy $H_{Place}$ whose root is $ALL_{Place}$. Element of dimension *Place* are defined through several levels of hierarchy: $ALL_{Place} >$ *Continent* $>$ *Country* $>$ *City*. Dimension *Product* is associated to hierarchy $H_{Product}$ whose root is $ALL_{Product}$ and $Coke$ and $Wine$ are leaves. Part of the hierarchies is illustrated Fig. 3.1.

| $S_{ID}$ | Multidimensional data sequences |
|---|---|
| $S_1$ | $\langle\{(Paris, Coke)\}\{(Paris, Coke)\}\{(London, Coke)\}\{(Tokyo, Coke)\}\rangle$ |
| $S_2$ | $\langle\{(Paris, Coke)\}\{(Lyon, Wine)(Paris, Coke)\}\{(Turin, Coke)\}\{(N.Y, Coke)\}\rangle$ |
| $S_3$ | $\langle\{(N.Y, Wine)\}\{(L.A, Coke)\}\{(Paris, Wine), (London, Wine)\}\rangle$ |

**Table 1.** Set of multidimensional data sequences $SDB$




(a) Cities (b) Drinks

**Fig. 1.** Part of Hierarchies $ALL_{Place}$ and $H_{Product}$

A *multidimensional item* $a = (d_1, \ldots, d_m)$ is a tuple defined on $D_A$ such that $\exists d_i \neq ALL_i$. It is important to note that a multidimensional item can be defined with any value at any level of the hierarchies associated to the analysis dimensions. For instance, $(Europe, Coke)$ and $(N.Y, Wine)$ are two multidimensional items.

Since multidimensional items are defined at different levels of hierarchies, it is possible to compare them using a specificity relation. Let $a = (d_1, \ldots, d_m)$ and $a' = (d'_1, \ldots, d'_m)$ be two multidimensional: (i) $e$ is said to be *more general* than $a'$ ($a' \subseteq a$) if $\forall d_i$, $d_i$ is an ancestor of $d_{i'}$ in $H_i$ or $d_i = d'_i$; (ii) $a$ is said to be *more specific* than $a'$ ($a \subseteq a'$) if $\forall d_i$, $d_i$ is a descendant of $d'_i$ in $H_i$ or $d_i = d'_i$; (iii) $a$ and $a'$ are said to be *incomparable* if there is no relation between them ($a \nsubseteq a'$ and $a' \nsubseteq a$).

A *multidimensional itemset* $i = \{a_1, \ldots, a_k\}$ is a non-empty set of multidimensional items such that for all distinct $i$, $j$ in $\{1 \ldots k\}$, $a_i$ and $a_j$ are incomparable. For instance, $\{(France, Wine), (U.K, Wine)\}$ is a multidimensional itemset. $\{(Europe, ALL_{Product}), (London, Wine)\}$ is not a multidimensional itemset because $(London, Wine) \subseteq (Europe, ALL_{Product})$.

A *multidimensional sequence* $s = \langle i_1, \ldots, i_l \rangle$ is an ordered list of multidimensional itemsets. $\langle \{(USA, Wine)\}\{(France, Wine)(U.K, Wine)\}\rangle$ is a multidimensional sequence associated to the database $SDB$ Table 1.

A multidimensional data sequence $S$ *supports* a multidimensional sequence $s = \langle i_1, \ldots, i_l \rangle$ if for every item $a_i$ of every itemset $i_j$, there exists an item $a'_i$ in $S$ such that $a'_i \subseteq a_i$ with respect to the ordered relation (itemset $i_1$ must be discovered before itemset $i_2$, etc.). According to Table 1, data sequence $S_2$ supports the sequence $s = \langle \{(France, Coke)\}\{(Europe, Wine)\}\{(USA, Coke)\}\rangle$.

The support of a sequence $s$ is the number of data sequences of $SDB$ that support $s$. Given a user-defined minimum support threshold *minsup*, a sequence is said to be *frequent* if its support is greater than or equal to *minsup*.

Given a set of multidimensional data sequences $SDB$ that are defined on a set of dimension $D_A$, the problem of mining *multidimensional sequential patterns* is to discover all multidimensional sequences that have a support greater than or equal to the user specified minimum support threshold *minsup*.

### 3.2 Convergent and Divergent Multidimensional Sequences

So far, we have defined the problem of mining multidimensional sequential patterns. We have also noticed that taking hierarchies into account provides relations between multidimensional items. Now, we can introduce the concept of convergent and divergent sequences.

**Definition 1 (Divergent Sequence )** *A sequence $s = \langle i_1, \ldots, i_k \rangle$ is said to be a divergent sequences if for every item $e_j \in i_k$, $\nexists e'_{j'} \in i_{k'}$ such that $k' < k$ and $e_j \subset e'_{j'}$.*

In other words, for each item $e$ of the sequence, there does not exist more general item contained before $e$ in the sequence. The sequence $\langle \{(Paris, Coke)\}$,-$\{(France, Coke)(U.K, Coke)\}, \{(Europe, Coke)\}\{(ALL_{Place}, Coke)\}\rangle$ is a divergent sequence.

**Definition 2 (Convergent Sequence)** *A sequence $s = \langle i_1, \ldots, i_k \rangle$ is said to be a convergent sequence if for every item $e_j \in i_k$, $\nexists e'_{j'} \in i_{k'}$ such that $k' < k$ and $e'_{j'} \subset e_j$.*

For each item $e$ of the sequence, there does not exist more specific item contained before $e$ in the sequence. The sequence $\langle \{(ALL_{Place}, Wine)\}, \{(Europe, Wine)\}$, $\{(Italy, Wine)(France, Wine)\}\rangle$ is a convergent sequence.

### 3.3 Algorithm

Ordering the items in the itemsets of the sequences is a fundamental step to avoid the already examined cases. Existing methods that are based on different paradigms (*pattern growth* ([6]), *Apriori*([1, 5, 11, 2])), are not directly applicable in a multidimensional framework. Indeed, items, that are not defined with the finest level of hierarchy, are not explicitly present in the database. Such items are retrieved by inference since there is no associated tuple in the database.

We then introduce functions to locally handle all items and not only items that are present in data sequences. An itemset is said to be *extended* if it is equal to its closure according to the relation of specialization ($\subseteq$). This notion allows to take all items into account. In order to enhance the management of items, we introduce a *lexicographicospecific order* (lgs) that is an alpha-numeric order according to the precision degree of the item. Thus, the most specific items are the first to be handled. We have to define a function LGS-Closure that transforms an itemset of a data sequence into its extended itemset that contain all the items that can be inferred. As an example, $LGS\text{-}Closure(\{(Paris, Coke)\})$ = $\{(Paris, Coke)$,$(Paris, ALL_{Product}$, $(France, Coke)$, $(France$, $ALL_{Product})$, $(Europe, Coke)$,$(Europe, ALL_{Product}), (ALL_{Place}, Coke)\}$. We note that the tuple ($ALL_{Place}$, $ALL_{Product}$) is not considered by definition of multidimensional item.

The extraction of frequent items can be done on each extended itemset. In pattern growth approaches, sequences are extracted by greedily adding a frequent item to a frequent sequence. It is thus necessary to define an efficient way for extending sequences from the last itemset of the sequences. For this purpose, we define the function $LGS\text{-}Closure_X$ where $X$ is an itemset that contains "forbidden items" (*i.e.* every items $e_i$ of the last itemset of the prefix sequence and all items comparable to $e_i$). As an example, $\{(ALL_{Place}, Wine)\}$ = $LGS\text{-}Closure_{\{(Europe, ALL_{Product})\}}(\{(Paris, Wine), (London, Wine)\})$.

Divergent sequences are discovered thanks to algorithm $CD\_M2S$. To mine all divergent sequences on $SDB$, routine$CD_M2S(\langle\rangle, SDB, \emptyset, minsup)$ is called.

This algorithm is pattern growth based [6]. Instead of scanning the whole database, level by level as Apriori based methods, the database is projected according to the *prefix sequence*. This data projection projection is quite different from [6]. Indeed, we have to handle all possible items, so the projection has to take the itemsets of the data sequences that contain the discovered item, and not only the item as in [6].

Two kinds of items can be extracted from the projected database:

1. Items that are added in a new itemset of the prefix sequence $\alpha$. These items are mined thanks to LGS-Closure.
2. Items (denoted by _e) that are added in the last itemset of the prefix sequence $\alpha$. In this case, we use the function LGS-Closure$_X$ where $X$ is the last itemset of the sequence $\alpha$.

This algorithm allows the extraction of divergent sequences. To mine convergent sequences, it is necessary to use the same algorithms but on an i**nverted database**. Indeed, beginning by the end (invert the ordered relation) of the data

**Algorithm 1:** CD_M2S

**Data**: Prefix sequence $\alpha$, projected database $SDB|_\alpha$, set of current frequent sequences $FS$ , minimum support threshold $minsup$

**Result**: Set of divergent frequent sequences with prefix $\alpha$

**begin**

  **if** $\alpha \neq \langle\rangle$ **then**

    $FS \leftarrow FS \cup \{\alpha\}$;

  $LF \leftarrow \{e \ s.t. \ support(e, SDB|_\alpha) \geq minsup$ and $\nexists e' \ s.t. \ support(e', SDB|_\alpha) \geq minsup$ and $e' \subset e\}$;

  **foreach** *items* $e \in LF$ **do**

    $\alpha' \leftarrow \alpha.e$;

    Call $CD\_M2S(\alpha', SDB|_{\alpha'}, FS, minsup)$;

**end**

sequences and re-invert the discovered patterns allows a discovery from general to particular case.

## 4 Experiments

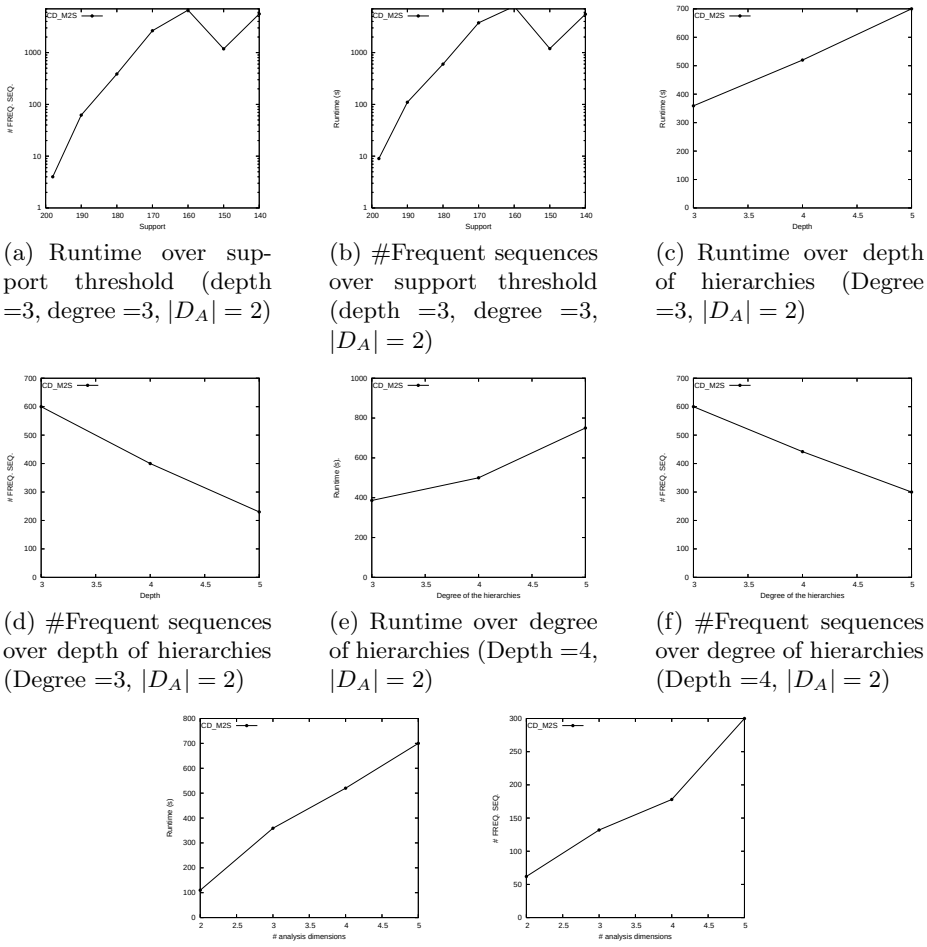
In this Section, we report experiments on both synthetic and real data.

**Synthetic Data:**

Experiments were carried out on a synthetic database. This database contains $10,000$ data sequences (with an average of 47 itemsets) over 5 analysis dimensions. Some hierarchical relations are defined between elements of each analysis dimension. In this paper, we report the behavior of our approach (number of patterns, runtime) according to several parameters (support threshold, $|D_A|$, degree and depth of hierarchies).

Figures Fig. 2(a) and Fig. 2(b) respectively report the runtime and the number of frequent sequences according to the minimum support threshold. The number of sequences tends to increase when the support decreases. The runtime follows the same behavior. However, it is possible that the number of frequent sequences decreases when the support decreases. Indeed, some more specific items can appear. Furthermore, a more general item is faster inferred in a data sequence than a more specific one. Therefore, we can obtain a smaller number of frequent sequences.

Figures Fig. 2(c) and Fig. 2(d) report the runtime and the number of frequent sequences according to the depth of the hierarchies on the analysis dimension. Adding one level in the hierarchies provides more precise data (*soda* becomes *pepsi* or *coca*). There is thus more different values in the database. $CD\_M2S$ is robust front of the specialization phenomena. Indeed, even if the data become very specific (5 different levels in the hierarchy), our approach allows to extract some sequences that are described among several level of hierarchy. We can notice that the runtime increases when the number of levels of the hierarchies increases. This is due to the number of potentially frequent items that increases.




(a) Runtime over support threshold (depth =3, degree =3, $|D_A| = 2$)

(b) #Frequent sequences over support threshold (depth =3, degree =3, $|D_A| = 2$)

(c) Runtime over depth of hierarchies (Degree =3, $|D_A| = 2$)

(d) #Frequent sequences over depth of hierarchies (Degree =3, $|D_A| = 2$)

(e) Runtime over degree of hierarchies (Depth =4, $|D_A| = 2$)

(f) #Frequent sequences over degree of hierarchies (Depth =4, $|D_A| = 2$)

(g) Runtime over $|D_A|$ (Depth =3, Degree =3)

(h) #Frequent sequences over $|D_A|$ (Depth =3, Degree =3)

**Fig. 2.** Experiments carried out on synthetic data

Figures Fig. 2(e) and Fig. 2(f) report the behavior (runtime and number of sequences) of our approach according to the degree of the hierarchies. Increasing the degree of a hierarchy provides more specific data (adding a son or an instance). Our approach allows to discovery knowledge while hierarchies become more specific. However, the cost of the extraction (runtime) is more expensive.

Figures Fig. 2(g) and Fig. 2(h) report the runtime and the number of sequences according to the number of analysis dimensions ($|D_A|$). Adding some

analysis dimensions generates an increase of the number of frequent sequences and the runtime.

These experiments on synthetic data show the robustness of our approach according to the diversity of the data ($D_A$, degree and depth of the hierarchies, etc.). Considering a more diverse source data leads to a more important extraction cost that stays acceptable.

**Real Data:**

We report experiments on logs of toy game [3] which is an *Eleusis* based card game. This game was created in order to simulate the activity of the scientific discovering (publications, refutations, experiments). The problem in Robert Abbot's Eleusis card game [4] is to find a secret law hidden from the players and determining the valid sequences of cards that can be played during the game. [3] proposes a new version in which humans are helped by machines to produce a theory. Players have to discover a rule (for instance a rule is *"two successive cards must have two different colors"* and a positive example according to this rule is *"ace of hearts followed by king of spades"*). A player wins points if he produces a positive example or he publishes his theory. He also earns points if he refutes a theory published by another player. He loses points if his theory is disproved by another player.

The hidden rules are card sequences. Each sequence contains a left part and a right part. Each part can contain several cards. We have described this problem according to several analysis dimensions: one dimension for the card values ( king, queen, ..., ace); one dimension for the card colors (heart, diamond, spade and clubs); one dimension for the position of the card in the sequence(right or left) and one dimension for the oracle answer (true, wrong).

We can obtain convergent and divergent rules. A divergent sequence is: "*For the secret rule water lily, players frequently propose the following cards: three of spade, ace of spade, a odd card of spade and finally a black odd card.*". A convergent sequence is: "*For the secret rule lily, players frequently propose a red card, a card of heart, a numbered card of heart.*". We notice that these rules are relevant for the expert and they cannot be extracted with classical algorithm.

## 5 Conclusion

We proposed an original method to extract multidimensional sequences that are defined on several levels of hierarchy according to different points of view: from general to particular (convergent) or from particular to general (divergent). We thus defined the concepts of convergent and divergent multidimensional sequences. We also introduced the algorithm $CD\_M2S$ that is pattern growth based. Some experiments on synthetic and real data show the interest of our approach. Note that this proposal is totally different from [9]. Indeed, in this paper, we focus on mining for special sequences: divergent or convergent sequences. Such sequences mean that comparable items can appear together within a convergent or divergent sequence whereas they cannot in $HYPE$. Furthermore, $HYPE$ algorithm is APriori based whereas algorithm $CD\_M2S$ is pattern-growth based.

This work offers several perspectives. First, divergent sequence can model special behaviours like buzz or the appearance of a seminal paper that leads to lot of publications and applications. Convergent sequences can model behaviors that become specialized through time like the appearance of new scientific topics or marketing products. Therefore, It would be very interesting to focus on the discovery and the prediction of such behaviors. Second, the efficiency of the extraction can be enhanced thanks to condensed representations (closed patterns, etc.) that provide some properties to efficiently prune the search space. Furthermore, other propositions can be done on the management of the hierarchies. We can imagine a modular management of hierarchies where some levels of the hierarchies would be more important (minimal and maximal levels on some hierarchies in order to be not too general or too specific) to fit user needs and to preserve the scalability of the extraction.

## References


1. R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. L. P. Chen, editors, *ICDE'95*, pages 3–14. IEEE Computer Society, 1995.
2. J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *KDD*, pages 429–435. 2002.
3. C. Dartnell and J. Sallantin. Assisting scientific discovery with an adaptive problem solver. In *Discovery Science*, pages 99–112. 2005.
4. M. Gardner. *Mathematical games*. Scientific American, 1959.
5. F. Masseglia, F. Cathala, and P. Poncelet. The psp approach for mining sequential patterns. In J. M. Zytkow and M. Quafafou, editors, *PKDD '98*, volume 1510 of *Lecture Notes in Computer Science*, pages 176–184. Springer, 1998.
6. J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(10), 2004.
7. H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *CIKM'01*, pages 81–88. ACM, 2001.
8. M. Plantevit, Y. W. Choong, A. Laurent, D. Laurent, and M. Teisseire. $M^2$SP: Mining Sequential Patterns Among Several Dimensions. In *PKDD*, pages 205–216, 2005.
9. M. Plantevit, A. Laurent, and M. Teisseire. Hype: mining hierarchical sequential patterns. In *DOLAP*, pages 19–26, 2006.
10. C.-C. Yu and Y.-L. Chen. Mining sequential patterns from multidimensional sequence data. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):pp. 136–140, 2005.
11. M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.