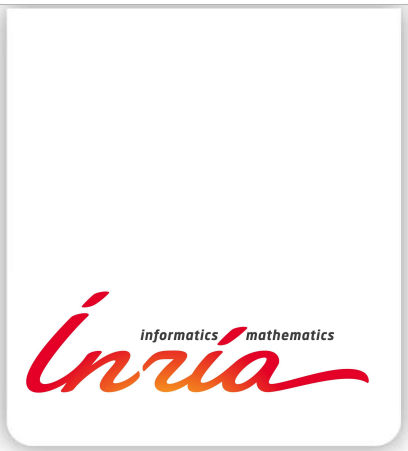
Mining Heterogeneous Multidimensional Sequential Patterns

Elias Egho , Chedy Raïssi , Nicolas Jay , Amedeo Napoli

RESEARCH REPORT
N° 8521
April 2014
Project-Teams Orpailleur

ISSN 0249-6399 ISRN INRIA/RR--8521--FR+ENG



Mining Heterogeneous Multidimensional Sequential Patterns

Elias Egho *, Chedy Raïssi *, Nicolas Jay *, Amedeo Napoli *

Project-Teams Orpailleur

Research Report n° 8521 — April 2014 — 25 pages

Abstract: All domains of science and technology produce large and heterogeneous data. Although much work has been done in this area, mining such data is still a challenge. No previous research targets the mining of heterogeneous multidimensional sequential data. In this work, we present a new approach to extract heterogeneous multidimensional sequential patterns with different levels of granularity by relying on external taxonomies. We show the efficiency and interest of our approach with the analysis of trajectories of care for colorectal cancer using data from the French casemix information system.

Key-words: multidimensional sequential patterns, data mining, heterogeneous sequential data

* LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine) Vandoeuvre-lès-Nancy, France

RESEARCH CENTRE NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Extraction des Motifs Séquentiels dans des Données Séquentielles Multidimensionnelles et Hétérogènes

Résumé : Tous les domaines de la science et de la technologie produisent gros volumes de données hétérogènes. L'exploration de tels volumes de données reste toujours un défi. Peu de travaux ciblent l'exploration et l'analyse de données séquentielles multidimensionnelles et hétérogènes. Dans ce travail, nous présentons une nouvelle approche appelée *MMISP* pour extraire des motifs séquentiels à partir de données séquentielles multidimensionnelles et hétérogènes, selon différents niveaux de granularité dépendant de connaissances extérnes. L'approache *MMISP* a été appliquée à l'analyser de trajectoires de soins de santé de patients en oncologie. Les données proviennent d'une base de données médico-administrative incluant toutes les informations sur les hospitalisations des patients dans la région Lorraine (France).

Mots-clés : motifs séquentiels multidimensionnels, fouille de données, données hétérogènes séquentielles

1 Introduction

Sequential pattern mining, introduced by Agrawal et al [1], is a popular approach to discover patterns in ordered data. Frequent sequence mining can be seen as an extension of the well known itemset mining problem where the input data is modeled as sequences elements. This method is rather efficient to discover rules of the type: *"customers frequently first buy DVDs of seasons I, II of Sherlock, then buy within 6 months season III of the same crime drama series"*. Sequential pattern mining has been successfully used so far in various domains : amino-acids protein sequence analysis [2], web log analysis [3], and music sequences matching [4].

Many efficient approaches were developed to mine patterns depending on time or order where most of them are based on the *Apriori* property [1], which states that any super pattern of a non-frequent pattern cannot be frequent. The main algorithms are *GSP* [5], *SPADE* [6], *PrefixSpan* [7] and *ClosSpan* [8]. However, these techniques and algorithms focus solely on one-dimensional aspect of sequential databases and do not deal with the multidimensional aspect where items can be of different types and described over different levels of granularity. For instance, in a real-world retail company, a database or data warehouse holds much more complex information such as article prices, gender of the customers, geolocation of the stores and so on. In addition, articles are usually represented following a hierarchical taxonomy: apples can be either described as fruits, fresh food or food. Pinto et al. [9], Zhang et al. [10] and Yu et al. [11] introduced the notion of multidimensionality in a sequence and proposed several algorithms to mine this type of data without taking into account the different levels of granularity for each dimension. Plantevit et al. [12] introduced M^3SP, an algorithm able to incorporate several dimensions described over different levels of granularity within the sequential pattern mining process. These approaches focus on homogeneous multidimensional sequence where its elements are described simply as vectors of items. By contrast, in modern life sciences [13], a multidimensional sequential data set is often represented as sequences of vectors with elements having different types (i.e., *item* and *itemset*). This special feature is in itself a challenge and multidimensionality in sequence mining needs to be carefully taken into account when devising new efficient algorithms.

In our approach, we aim at providing an approach that extract rules such as: *"After buying an article from the fruit category from supermarket A, a customer will buy two articles from the Egg and Dairy products and Beverage categories from supermarket B"*. This example not only combines two dimensions (supermarket and products) which are ordered over time and are represented with different levels of granularity, but it also characterizes them in a different way as a magazine can be considered as an element taking one value (*"item"*), while a product can take several values (*"itemset"*). This example shows that each dimension has to be managed in a proper and suitable way. We believe that our work is the first to present a full framework and algorithm to mine such multidimensional sequential patterns from heterogeneous multidimensional sequential database.

The main contribution of this report is to generalize the concept of multidimensional sequence by considering heterogeneous multidimensional sequences. The event in a sequence is considered as a vector of items and itemsets, Such multidimensional and heterogeneous patterns have to be mined by adapted and suitable methods. Accordingly, we propose a new method *MMISP (Mining Multidimensional Itemsets Sequential Patterns)* to extract sequential patterns from heterogeneous multidimensional sequential database. In addition, the approach is able to take into account background knowledge lying in taxonomies existing for each dimension. As often with enumeration algorithms, mining all possible sequential patterns from a multidimensional sequential database results in a huge amount of patterns which is difficult to be analyzed [14]. To overcome this problem, *MMISP* mines only the most specific patterns. We report an extensive empirical study on synthetic datasets and qualitative experiments with a dataset consisting of

Patients	Trajectories
s_1	$\langle(uh_p, ca_1, \{mp_{111}, mp_{221}\}), (uh_p, ca_1, \{mp_{222}\}), (gh_l, r_1, \{mp_{221}, mp_{311}\})\rangle$
s_2	$\langle(uh_n, ca_1, \{mp_{111}\}), (uh_n, ca_2, \{mp_{111}, mp_{211}\}), (gh_l, r_1, \{mp_{222}, mp_{312}\})\rangle$
s_3	$\langle(uh_n, ca_3, \{mp_{112}, mp_{211}\}), (gh_l, r_2, \{mp_{222}, mp_{311}\})\rangle$
s_4	$\langle(uh_p, ca_2, \{mp_{112}, mp_{222}\}), (gh_p, r_2, \{mp_{221}, mp_{312}\}), (gh_p, r_2, \{mp_{221}, mp_{312}\})\rangle$

Table 1: An example of a database of patient trajectories.

trajectories of cancer patients extracted from French healthcare organizations. The successive hospitalizations of a patient can be expressed as a sequence of heterogeneous multidimensional attributes such as healthcare institution, diagnosis and set of medical procedures. Our goal is to be able to extract patterns describing patient stays along with combinations of procedures over time. This type of pattern is very useful to healthcare professionals to better understand the global behaviour of patients over time and the associated supports.

The remainder of this report is organized as follows, Section 2 introduces the problem statement as well as a running example and briefly reviews the preliminaries needed in our development. *MMISP* method is described in Section 3. Section 4 presents experimental results from both quantitative and qualitative point of views and Section 5 concludes the paper.

2 Problem Statement

2.1 An introductory example

In this section, we propose an example to illustrate and ease the understanding of our proposed approach. The example focus on mining patient trajectory in a healthcare system. A patient trajectory can be considered as a sequence of hospitalizations ordered over time, where each time stamp corresponds to one hospitalization. This hospitalization is represented as a vector of 3 elements. Each vector represents specific information about one stay of a patient in a hospital:

- The hospital where a patient is admitted.
- A reason for hospitalization.
- Set of medical procedures that a patient undergoe

Table 1 describes four patient trajectories. For example, s_1 is a patient trajectory with three hospitalizations and the vector $(uh_p, ca_1, \{mp_{111}, mp_{221}\})$ fully describes the first hospitalization of patient s_1 who was admitted to the hospital uh_p for treatment of lung cancer ca_1, and underwent procedures mp_{111} and mp_{221}. In a patient trajectory, background knowledge is usually available in form of taxonomies, classification or concept hierarchies. Each element in the hospitalization can be represented at different levels of granularity, by using a taxonomy (see Figure 1).

Our goal is to find specific patterns that appears frequently in patients trajectories', by taking advantage of different levels of granularity for each element as background knowledge. Such patterns are very helpful to improve hospitalization planning, optimize clinical processes or detect anomalies.

2.2 Basic definitions

We assume in the following that domain knowledge always exists in the form of taxonomies. A sequence is an ordered set of vectors whose components are items or itemsets. Each item is a node

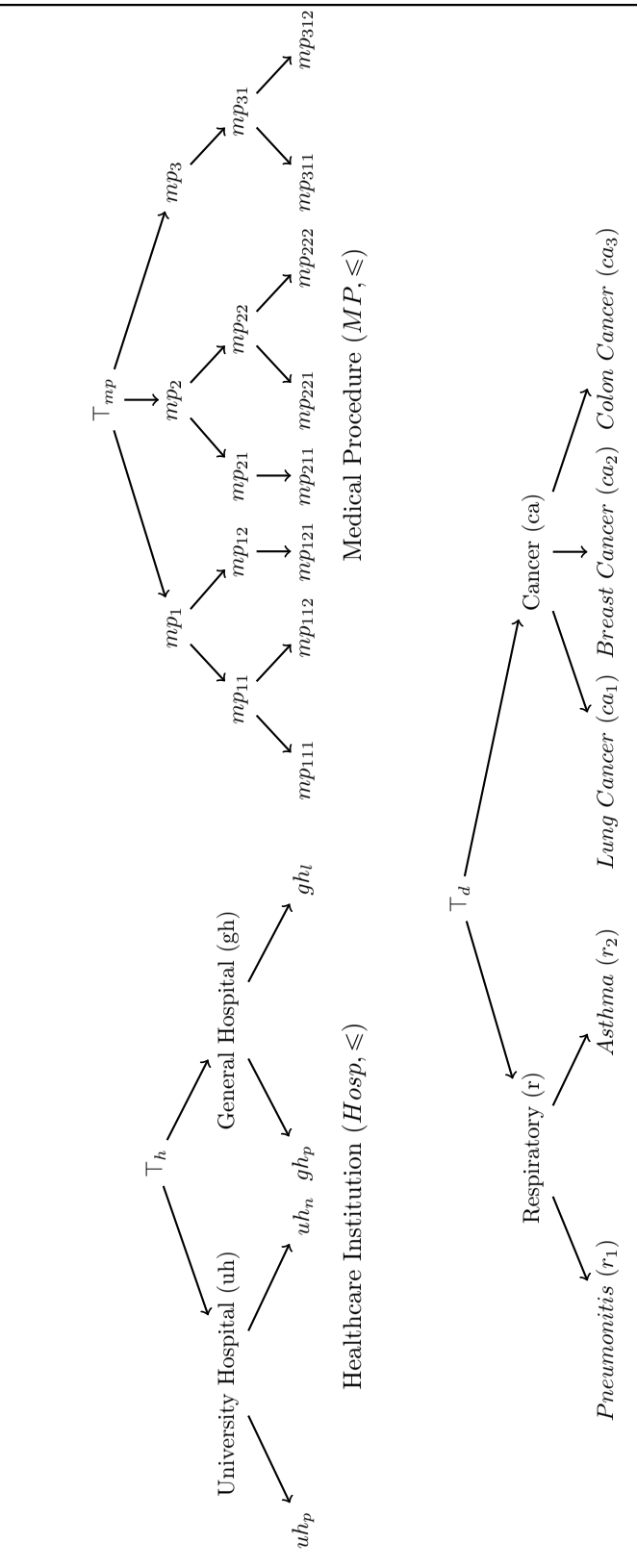


Figure 1: Taxonomies for the healthcare institution, the medical procedure and the diagnosis

in a taxonomy. For the sake of simplicity we call a multidimensional sequence a *"md-sequence"* and we define it as follows:

Definition 1. *(md-sequence)*
A md-sequence $s{=}\langle s_1, s_2, ..., s_n\rangle$ *is defined as set of elementary vectors* $s_i = (e_1, ..., e_k)$ *ordered by the temporal order relation* $<_t$ *such as* $s_1 <_t s_2 <_t s_3 ... <_t s_n$*, where* n *is called the size of the sequence* s*; i.e.,* $|s| = n$*.*

The vector $e = (e_1, ..., e_k)$ is ***more specific*** than $e' = (e'_1, ..., e'_k)$, denoted by $e \leqslant e'$, iff $e_i \leqslant e'_i$ for all $i \leqslant k$. Then, ordering over md-sequence is defined as follows

Definition 2. *(Ordering of Multidimensional Sequences)*
Given two md-sequences $s = \langle s_1, s_2, ..., s_{n_1}\rangle$ *and* $s' = \langle s'_1, s'_2, ..., s'_{n_2}\rangle$*, the sequence* $s = \langle s_1, s_2, ..., s_{n_1}\rangle$ *is* ***more specific*** *than* $s' = \langle s'_1, s'_2, ..., s'_{n_2}\rangle$*, denoted by* $s \leqslant s'$*, if there exist a set of indices* $1 \leqslant i_1 < i_2 < ... < i_{n_2} \leqslant n_1$ *such that* $s_j \leqslant s'_{i_j}$ *for all* $j \in \{1 \dots n_2\}$ *and* $n_2 \leqslant n_1$*.* s' *is said to be* ***more general*** *than* s*.*

Example 1. *Consider the three taxonomies* $(Hosp, \leqslant)$*,* $(Diag, \leqslant)$ *and* $(MP, \leqslant)$ *in Figure 1. The elementary vector* $e = (uh_p, ca_1, \{mp_{111}, mp_{121}\})$ *is more specific than* $e' = (uh, ca, \{mp_{11}, mp_{12}\})$*, as* $uh_p \leqslant uh$*,* $ca_1 \leqslant ca$ *and* $\{mp_{111}, mp_{121}\} \leqslant \{mp_{11}, mp_{12}\}$*. The sequence* $s = \langle (uh_p, ca_1, \{mp_{111}, mp_{121}\}), (gh_l, r_2, \{mp_{121}, mp_{131}\})\rangle$ *is a md-sequence with two elementary vectors* $s_1 = (uh_p, ca_1, \{mp_{111}, mp_{121}\})$ *and* $s_2 = (gh_l, r_2, \{mp_{121}, mp_{131}\})$ *where* s_1 *comes before* s_2 *over time. The sequence* $s' = \langle (uh, ca_1, \{mp_{11}, mp_{12}\}), (gh_l, r, \{mp_{13}\})\rangle$ *is more general than* s*, as* $s_1 \leqslant s'_1$*,* $s_2 \leqslant s'_2$ *and* s_1 *and* s'_1 *come before* s_2 *and* s'_2 *over time.*

Given a set $\mathcal{MS}_{\mathcal{DB}} = \{s_1, s_2, ..., s_m\}$ of m md-sequences. The set $\mathcal{MS}_{\mathcal{DB}}$ is called a *mds-database*. The support of an elementary vector in a mds-database is defined as follows:

Definition 3. *(Support of an elementary vector)*
Let $\mathcal{MS}_{\mathcal{DB}}$ *be a mds-database and let* $e = (e_1, e_2, ..., e_k)$ *be an elementary vector. The support of the elementary vector* e *in* $\mathcal{MS}_{\mathcal{DB}}$*, denoted by* $supp(e, \mathcal{MS}_{\mathcal{DB}})$*, is defined as follows:*

$$supp(e, \mathcal{MS}_{\mathcal{DB}}) = |\{s_i \in \mathcal{MS}_{\mathcal{DB}};\ \exists j \leqslant |s_i|;\ e \leqslant s_{ij}\}|$$

While the support of a sequence s in $\mathcal{MS}_{\mathcal{DB}}$ is defined as follows:

Definition 4. *(Support of md-sequence)*
Let $\mathcal{MS}_{\mathcal{DB}}$ *be a mds-database and let* s *be a md-sequence. The support of* s*, denoted by* $supp(s, \mathcal{MS}_{\mathcal{DB}})$ *is defined as follows:*

$$supp(s, \mathcal{MS}_{\mathcal{DB}}) = |\{s_i \in \mathcal{MS}_{\mathcal{DB}}; s_i \leqslant s\}|$$

Given a positive integer σ as minimal support threshold, and a mds-database $\mathcal{MS}_{\mathcal{DB}}$, the elementary vector e is called frequent, iff $supp(e, \mathcal{MS}_{\mathcal{DB}}) \geqslant \sigma$. A md-sequence is *frequent* in $\mathcal{MS}_{\mathcal{DB}}$ if its support in $\mathcal{MS}_{\mathcal{DB}}$ exceeds the minimal support threshold σ. A frequent md-sequence is called a *"mds-pattern"*. The problem of mining md-sequences is to enumerate all possible mds-patterns, given a mds-database $\mathcal{MS}_{\mathcal{DB}}$ and a minimal support threshold.

Example 2. *The sequence* $s = \langle (uh, ca, \{mp_{11}, mp_2\}), (\top_h, \top_d, \{mp_{222}\})\rangle$ *has a support which is equal to 3 (i.e.,* $supp(s, \mathcal{MS}_{\mathcal{DB}}) = 3$*) in the database* $\mathcal{MS}_{\mathcal{DB}}$ *(see Table 1) and is a mds-pattern w.r.t to the minimal support (i.e.,* $\sigma = 3$*).*

2.3 Most specific multidimensional sequential patterns

In this section, we present the problem of mining *the most specific mds-patterns*. Mining all possible mds-patterns from $\mathcal{MS}_{\mathcal{DB}}$ results in a huge amount of patterns that is difficult to manage. Thus, we extract a set of mds-patterns that are not only frequent but also the most specific. This second constraint allows the reduction of the number of returned sequences by discarding patterns that are *"too general"*. The most specific mds-pattern is defined as follows:

Definition 5. *(Most specific mds-pattern)*
Given a positive integer σ as minimal support threshold and a mds-database $\mathcal{MS}_{\mathcal{DB}}$. The sequence s is a most specific mds-pattern in $\mathcal{MS}_{\mathcal{DB}}$ if and only if $supp(s, \mathcal{MS}_{\mathcal{DB}}) \geqslant \sigma$ and there does not exist any sequence s' such that $supp(s', \mathcal{MS}_{\mathcal{DB}}) \geqslant \sigma$ and $s' \leqslant s$.

In this precise setting, the frequency for sequences is *monotone*; i.e., whenever s is frequent, any generalization of s is also frequent. For example, if $s = \langle (uh_p,\ ca_1, \{mp_{111}, mp_{112}\})\rangle$ is frequent in $\mathcal{MS}_{\mathcal{DB}}$ then $s' = \langle (uh, ca, \{mp_1\})\rangle$ which is more general than s is also frequent. Thus, the most specific sequential patterns constitute a basic for retrieving all sequential patterns. Let $\sigma = 3$ be a minimal support threshold, the sequence $s = \langle (uh,\ \top_d, \{mp_1\}), (\top_h, \top_d, \{mp_2\})\rangle$ is frequent, but is not the most specific one as the sequence $s' = \langle (uh, ca,\ \{mp_{11}, mp_2\}), (gh, r,\ \{mp_{22}, mp_{31}\})\rangle$ is frequent and verifies that $s' \leqslant s$. The sequence s' is a most specific mds-pattern as it is frequent and there is no other sequence in $\mathcal{MS}_{\mathcal{DB}}$ which is frequent and more specific.

The problem of mining mds-patterns is reduced to discover only the most specific patterns to significantly decrease the complexity of the problem and save computational time.

3 MMISP algorithm

In this section, we present an approach for extracting the most specific mds-patterns from a mds-database $\mathcal{MS}_{\mathcal{DB}}$. Our approach is called ***MMISP (Mining Multidimensional Itemsets Sequential Patterns)***. The basic idea of *MMISP* consists in transforming the mds-data into a *"classical form"* (i.e., sequence of itemsets) and then applying a standard algorithm for sequential pattern mining. *MMISP* is based on three steps:

1. ***Extraction of frequent elementary vectors***
   The algorithm searches for the frequent and specific elementary vectors.

2. ***Transformation***
   In this step, all frequent elementary vectors are mapped into an alternate representation, then, the mds-database is encoded by using this new representation.

3. ***mds-patterns mining***
   In this step, a standard sequential algorithm is applied to the sequential database produced at the preceding step.

3.1 Step 1: Extracting all frequent elementary vectors:

The basic step in *MMISP* is extracting all frequent elementary vectors from $\mathcal{MS}_{\mathcal{DB}}$ w.r.t. the ordering relation existing between their elements.

If the elementary vector is infrequent, then neither it nor its specifications will appear in mds-patterns. Thus, *MMISP* extracts only the frequent elementary vectors from $\mathcal{MS}_{\mathcal{DB}}$ to find all the most specific mds-patterns. The main challenge in this step is *how to efficiently mine the frequent elementary vectors.*

Assume that we have an elementary vector, composed of k bottom elements (i.e., $\perp = (\perp_1, ..., \perp_k)$), is more specific than any other elementary vector. If element of the vector is an item, then we consider a node $\perp_i$ connected by edges to all leaves of taxonomy as a bottom node. Otherwise if the element is an itemset, then the bottom is a set of all the leaf nodes. In our running example, the bottom elementary vector is $(\perp_h, \perp_d, \{mp_{111}, mp_{112}, mp_{121}, mp_{211}, mp_{221}, mp_{222}, mp_{311}, mp_{312}\})$. The set of all the elementary vectors E with the bottom vector $\perp$ is a *lattice* $(E \cup \{\perp\}, \leqslant)$. Given two elementary vectors $e = (e_1, ..., e_k)$ and $e' = (e'_1, ..., e'_k)$ in E, the join $(\sqcup)$ of e and e' is defined as the join of i^{th} element in e and e'; i.e., $e \sqcup e' = (e_1 \sqcup e'_1, ..., e_k \sqcup e'_k)$. The join of two nodes in a taxonomy is the lowest common ancestor of these nodes, while the join of two set of nodes $c = \{c_1, ..., c_n\}$ and $c' = \{c'_1, ..., c'_m\}$ is the most specific values from the set $\{\forall(i,j);\ c_i \sqcup c'_j\}$; $i \leqslant n$ and $j \leqslant m$.

Example 3. *Consider the three taxonomies* $(Hosp, \leqslant)$, $(Diag, \leqslant)$ *and* $(MP, \leqslant)$ *in Figure 1, the two elementary vectors* $e = (uh_p, ca_2, \{mp_{111}, mp_2\})$ *and* $e' = (uh_n, ca, \{mp_{121}, mp_{21}\})$. *The join of* e *and* e', $e \sqcup e'$, *is* $(uh_p \sqcup uh_n, ca_2 \sqcup ca, \{mp_{111}, mp_2\} \sqcup \{mp_{121}, mp_{21}\})$, *where the join of* uh_p *and* uh_n *is* uh , ca_2 *and* ca *is* ca *and the join of* $\{mp_{111}, mp_2\}$ *and* $\{mp_{121}, mp_{21}\}$ *is the set* $\{mp_1, mp_2\}$. *Thus, the The join of* e *and* e' *is the elementary vector* $(uh, ca, \{mp_1, mp_2\})$.

The meet $(\sqcap)$ of two elementary vector e and e' is $e \sqcap e' = (e_1 \sqcap e'_1, ..., e_k \sqcap e'_k)$. The meet between two nodes in a taxonomy is the most specific one if they are comparable, otherwise it is the bottom node $\perp_i$. The meet of two set of nodes $c = \{c_1, ..., c_n\}$ and $c' = \{c'_1, ..., c'_m\}$ is the most specific values from the set of the ancestors of each item in the two sets c and c'.

Example 4. *Consider the three taxonomies* $(Hosp, \leqslant)$, $(Diag, \leqslant)$ *and* $(MP, \leqslant)$ *in Figure 1, the two elementary vectors* $e = (uh_p, ca_2, \{mp_{111}, mp_2\})$ *and* $e' = (uh_n, ca, \{mp_{121}, mp_{21}\})$. *The meet of* e *and* e', $e \sqcap e'$, *is* $(uh_p \sqcap uh_n, ca_2 \sqcap ca, \{mp_{111}, mp_2\} \sqcap \{mp_{121}, mp_{21}\})$, *where the meet of* uh_p *and* uh_n *is* $\perp_h$, ca_2 *and* ca *is* ca_2 *and the meet of* $\{mp_{111}, mp_2\}$ *and* $\{mp_{121}, mp_{21}\}$ *is the set* $\{mp_{111}, mp_{121}, mp_{21}\}$. *Thus, the The meet of* e *and* e' *is the elementary vector* $(\perp_h, ca_2, \{mp_{111}, mp_{121}, mp_{21}\})$.

Finally, we can say that $(E \cup \{\perp\}, \leqslant)$ is a *lattice*, while the frequent elementary vectors considers as a *join-semilattice* $(FE, \leqslant)$. The main challenge now is how to efficiently build the join-semilattice $(FE, \leqslant)$. This task is achieved through a *depth-first*, and from *left-to-right* traversal [15], starting from the most general elementary vector $\top = (\top_1, ..., \top_k)$, we consider it as frequent elementary vector. Then, for each frequent elementary vector e, we recursively generate all the immediate successors of e, and for each of them, we compute its support in $\mathcal{MS}_{\mathcal{DB}}$ and we keep the frequent one.

We need to define an effective and nonredundant way to characterize the immediate successors in $(FE, \leqslant)$ of a given elementary vector $e = (e_1, ..., e_k)$ as follows. Firstly, we will assume that elements of the elementary vector are ordered according to a fixed total ordering. We will define an index z over the elementary vector to generate its immediate successors without redundancy and to build $(FE, \leqslant)$ from left to right. This index is defined as the position of the element in e which is more specific than $\top_i$ and all the elements after this one until end of e are $\top_i$ (in the case of e is $(\top_1, ..., \top_k)$, the index z equals to 1).

Example 5. *Given the elementary vector* $e = (\top_h, ca, \top_{mp})$, *then the index* z *equals to 2 as the second element is not* $\top_d$; *i.e.,* $e_2 = ca$, *and all the elements after* ca *until end of* e *are top; i.e.,* $e_3 = \top_{mp}$.

To generate the immediate successors of an elementary vector e, we substitute each element in e, which has its position greater than or equal to the index z, with one of its immediate successors and the rest of the elements are kept as it is.

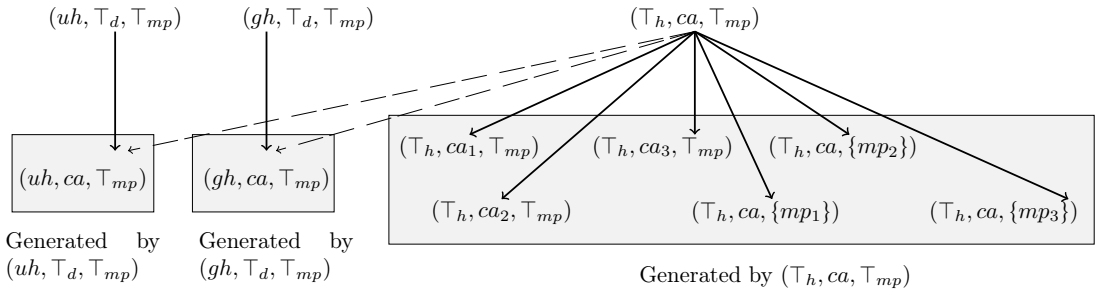


Figure 2: The immediate successors of $(\top_h, ca, \top_{mp})$

Example 6. *Given the same previous example $e = (\top_h, ca, \top_{mp})$, as we see that the index z in e is equal to 2, then its immediate successors are consisted of two sets. The first one contains all the elementary vectors which are generated by substituting the second element ca with one of its immediate successors and keeping the first and the third element; i.e., $\top_h$ and $\top_{mp}$ respectively. While the second set contains all the elementary vectors which are generated by substituting the third element $\top_{mp}$ with one of its immediate successors and keeping the first and the second element; i.e., $\top_h$ and ca respectively.*

The immediate successors of an element depend on its type, if an element is an item then we follow the standard definition of immediate successor in the taxonomy. For example, given the taxonomy $(Diag, \leqslant)$ in Figure 1, the immediate successors of ca are ca_1, ca_2 and ca_3.

In case, the element is an itemset $c = \{c_1, c_2, ..., c_m\}$, then to generate nonredundant immediate successors of c, we assume that its items are ordered according to a fixed total ordering. The immediate successors of c are splitted into two sets. The first one is generated by substituting the last item in c; i.e., c_m, with one of its immediate successors and the rest of the items are kept as it is; i.e., the first set of the immediate successors of c contains an itemset $c' = \{c'_1, ..., c'_m\}$ where $c'_i = c_i$ for all $i < m$ and c'_m is one of the immediate successors of c_m. The second set is generated by adding new item c_{m+1} to end of c, where c_{m+1} is one of the right siblings of c_m and all its ancestor nodes; i.e., the second set of the immediate successors of c contains an itemset $c' = \{c'_1, ..., c'_m, c'_{m+1}\}$ where $c'_i = c_i$ for all $i \leqslant m$ and c'_{m+1} is one of the right siblings of c_m and all its ancestor nodes.

Example 7. *Given the taxonomy $(MP, \leqslant)$ in Figure 1, the immediate successors of $c = \{mp_1, mp_{21}\}$ are generated by substituting the last item mp_{21} in c with its immediate successors; i.e., $\{mp_1, mp_{211}\}$, and by adding the right siblings of mp_{21} and all its ancestor nodes; i.e., mp_{22} and mp_3, to the end of c; i.e., $\{mp_1, mp_{21}, mp_{22}\}$ and $\{mp_1, mp_{21}, mp_3\}$.*

Given the taxonomies in Figure 1, the immediate successors of $e = (\top_h, ca, \top_{mp})$ are $(uh, ca, \top_{mp})$, $(gh, ca, \top_{mp})$, $(\top_h, ca_1, \top_{mp})$, $(\top_h, ca_2, \top_{mp})$, $(\top_h, ca_3, \top_{mp})$, $(\top_h, ca, \{mp_1\})$, $(\top_h, ca, \{mp_2\})$ and $(\top_h, ca, \{mp_3\})$. The first two are generated from $(uh, ca, \top_{mp})$ and $(gh, ca, \top_{mp})$ respectively, while the rest are generated from $(\top_h, ca, \top_{mp})$. First by replacing only ca with one of ca_1, ca_2 and ca_3 and keeping $\top_h$ and $\top_{mp}$; i.e., $(\top_h, ca_1, \top_{mp})$, $(\top_h, ca_2, \top_{mp})$ and $(\top_h, ca_3, \top_{mp})$. Then, by replacing only $\top_{mp}$ with one of $\{mp_1\}$, $\{mp_2\}$ and $\{mp_3\}$ and keeping $\top_h$ and ca; i.e., $(\top_h, ca, \{mp_1\})$, $(\top_h, ca, \{mp_2\})$ and $(\top_h, ca, \{mp_3\})$ (see Figure 2).

The frequency of an elementary vector is monotone, the specialization of a non-frequent elementary vector is also non-frequent. We use this monotonicity to prune the enumeration space and efficiently build the semilattice $(FE, \leqslant)$. Figure 3 shows an example of generation of a part

id	Elementary Vector
1	$(uh, ca, \{mp_{11}, mp_2\})$
2	$(gh, r, \{mp_{22}, mp_{31}\})$
3	$(\top_h, \top_d, \{mp_{222}\})$

Table 2: The most specific frequent elementary vectors extracted from $(FE, \leqslant)$.

of $(FE, \leqslant)$ with $\sigma = 3$ which is detailed as follows. As the first step we consider the most general elementary vector $(\top_h, \top_d, \top_{mp})$, from which seven new frequent elementary vectors, $(uh, \top_d, \top_{mp})$, $(gh, \top_d, \top_{mp})$, $(\top_h, r, \top_{mp})$, $(\top_h, ca, \top_{mp})$, $(\top_h, \top_d, \{mp_1\})$, $(\top_h, \top_d, \{mp_2\})$ and $(\top_h, \top_d, \{mp_3\})$, are generated. Let us consider the first elementary vector, $(uh, \top_d, \top_{mp})$, the immediate successors generate by *MMISP* are $(uh, ca, \top_{mp})$, $(uh, \top_d, \{mp_1\})$ and $(uh, \top_d, \{mp_2\})$. Now, for the vector $(uh, ca, \top_{mp})$ obtained in the previous level, further immediate successors $(uh, ca, \{mp_1\})$ and $(uh, ca, \{mp_2\})$ are generated. Similarly we obtain the following two vectors $(uh, ca, \{mp_{11}\})$ and $(uh, ca, \{mp_1, mp_2\})$ and $(uh, ca, \{mp_{11}, mp_2\})$ from $(uh, ca, \{mp_1\})$ and $(uh, ca, \{mp_{11}\})$ respectively. Finally, for the vector $(uh, ca, \{mp_{11}, mp_2\})$, no any new frequent elementary vectors can be found, thus the generation stops.

As the objective of *MMISP* is extracting the most specific sequential patterns, we retain only the most specific elementary vectors *MSFEV* in $(FE, \leqslant)$. The most specific frequent elementary vectors constitute collection of elementary vectors in $\mathcal{MS}_{\mathcal{DB}}$ with respect to σ which are frequent and most specific. Table 2 shows the set of most specific frequent elementary vectors which are extracted from $(FE, \leqslant)$.

The pseudocode of the first step in *MMISP* is presented in Algorithm 1. The idea is simple: Lines 2-3 generate the most general elementary vector $e = (\top_1, ..., \top_m)$, while Line 6 adds e to $(FE, \leqslant)$. Then, Line 7 recursively calls the function get_rec_msfev starting from e to build the *join-semilattice* $(FE, \leqslant)$ and to get all the most specific frequent elementary vectors $MSFEV$.

Algorithm 2 presents the pseudocode of the function get_rec_msfev. Lines 2-5 get the index z in e, Line 6 generates the immediate successor of e based on the index z, while Line 7 keeps only the frequent ones. When there are no frequent immediate successors of e, then e considers as one of the most specific frequent elementary vectors (see Lines 8-9). Otherwise, we call recursively the function get_rec_msfev for each frequent immediate successors of e (Lines 10-13).

Algorithm 3 shows how to generate immediate successors of an elementary vector. Line 2 shows that generation starts from the z^{th} element. Lines 3-9 show that if an element is an item, we substitute it with one of its immediate successors and the rest of the elements are kept as it is. While, if an element is an itemset c, we substitute it with another itemset. This itemset is generated by substituting the last item in c with one of its immediate successors (Lines 13-17). Then, by appending the right sibling of last item in c at its end (Lines 18-22). Finally, by appending the right siblings of all the ancestors of last item in c at its end (Lines 23-29).

3.2 Step 2: Transformation of mds-database:

We now study the temporal relation between the extracted specific frequent elementary vectors as follow. Firstly, we replace each elementary vector in each md-sequence of $\mathcal{MS}_{\mathcal{DB}}$ with all its generalizations from *MSFEV* set. Given a sequence $s = \langle s_1, ..., s_n \rangle$ in $\mathcal{MS}_{\mathcal{DB}}$ the replacement consists in substituting each elementary vector s_i in s by several elementary vectors $e \in MSFEV$ such that $s_i \leqslant e$.

Example 8. *Given the sequence s_4 in $\mathcal{MS}_{\mathcal{DB}}$ (see Table 1). The sequence s_4 is transformed*

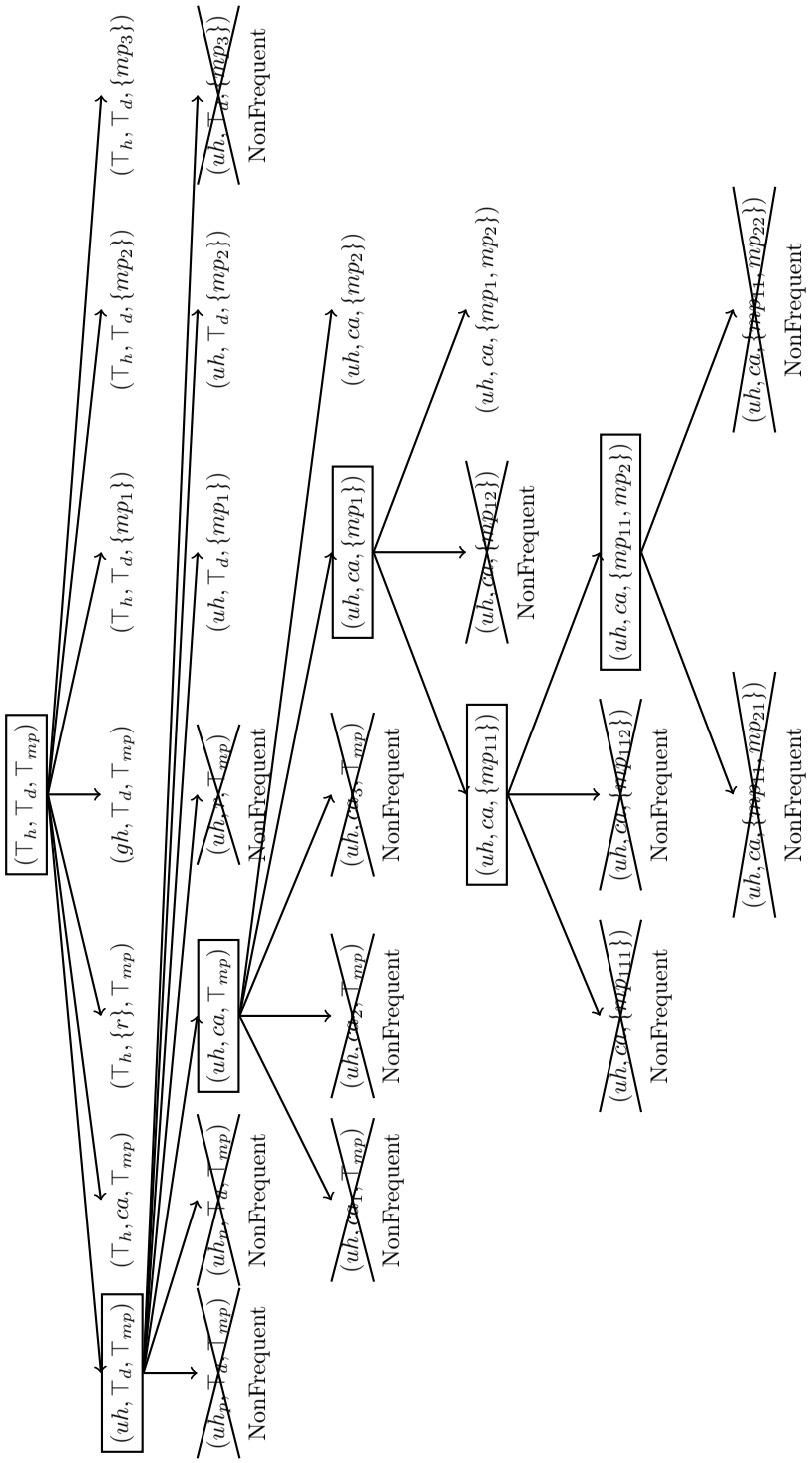


Figure 3: The steps of generating the elementary vectors in $(FE, \leqslant)$ with $\sigma = 3$.

Algorithm 1: Mining All The Most Specific Frequent Elementary Vector

input : mds-database $\mathcal{MS}_{\mathcal{DB}}$, Minimum support threshold σ, Set of m taxonomies $Tax = \{Tax_1, ..., Tax_m\}$.

output: The set MSFEV of all most specific frequent elementary vectors, The join semi-lattice FE.

begin

```
/* Generating the most general elementary vectors (⊤1, ⊤2, ..., ⊤m)   */
```

for $i \leftarrow 1$ **to** m **do**

  $e_i = \top_i$;

$MSFEV \leftarrow \varnothing$;

$FE \leftarrow \varnothing$;

$FE \leftarrow FE \cup \{ e \}$;

```
/* The recursive method to build the join-semilattice (FE, ⩽) and to get
   all the most specific frequent elementary vectors MSFEV.            */
```

call $get\_rec\_msfev(\mathcal{MS}_{\mathcal{DB}}, e, \sigma, Tax, FE, MSFEV)$;

Algorithm 2: Routine get_rec_msfev

input : mds-database $\mathcal{MS}_{\mathcal{DB}}$, Elementary vector e, Minimum support threshold σ, Set of m taxonomies $Tax = \{Tax_1, ..., Tax_m\}$, Set of frequent elementary vectors FE, Set of most specific elementary vectors $MSFEV$.

begin

```
/* Computing the index z                                               */
```

$z = 1$;

for $i \leftarrow 1$ **to** m **do**

  if $e_i \neq \top_i$ **then**

    $z =$ i;

$Cand \leftarrow immediate\_successor\_elementary\_vector(e, z, Tax)$;

$Freq \leftarrow \{ e' \in Cand \, ; supp(e', \mathcal{MS}_{\mathcal{DB}}) \geqslant \sigma \}$;

if $Freq = \varnothing$ **then**

```
  /* e is a most specific elementary vector                            */
```

  $MSFEV \leftarrow MSFEV \cup \{ e \}$;

else

```
  /* Recursive call get_rec_msfev for each new frequent elementary
     vector generated                                                  */
```

  foreach $e' \in Freq$ **do**

    $FE \leftarrow FE \cup \{ e' \}$;

    call $get\_rec\_msfev(\mathcal{MS}_{\mathcal{DB}}, e, \sigma, Tax, FE, MSFEV)$;

Algorithm 3: Routine *immediate_successor_elementary_vector*

input : Elementary vector e, Index z, Set of m taxonomies $Tax = \{Tax_1, ..., Tax_m\}$.
output: A set of immediate successors of e.
begin
  for $i \leftarrow z$ **to** m **do**
    /* In cas the element e_i is an item */
    if e_i *is an item* **then**
      foreach $x \in$ *immediate_successor* (e_i, Tax_i) **do**
        $e' = e$;
        e'_i=x ;
        $Output \leftarrow Output \cup \{ e' \}$;
      end
    end
    /* In cas the element e_i is an itemset */
    else if e_i *is an itemset* **then**
      $c = e_i$;
      $k = |c|$;
      /* Replacing the last item in c with its immediate successor */
      foreach $x' \in$ *immediate_successor* (c_k, Tax_i) **do**
        $e' = e$;
        $e'_i = append\_at\_end(c \backslash c_k, x')$;
        $Output \leftarrow Output \cup \{ e' \}$;
      end
      /* Appending the right sibling of c_k at the end of c */
      foreach $x' \in right\_sibling(c_k, Tax_i)$ **do**
        $e' = e$;
        $e'_i = append\_at\_end(c, x')$;
        $Output \leftarrow Output \cup \{ e' \}$;
      end
      /* Appending the right sibling of all the ancestors of c_k at the end of c */
      foreach $x \in ancestors(c_k, Tax_i)$ **do**
        foreach $x' \in right\_sibling(x, Tax_i)$ **do**
          $e' = e$;
          $e'_i = append\_at\_end(c, x')$;
          $Output \leftarrow Output \cup \{ e' \}$;
        end
      end
  end
  return *Output* ;
end

Patients	Trajectories
$\widehat{s}_1$	$\langle\{(uh, ca, \{mp_{11}, mp_2\})\}, \{(\top_h, \top_d, \{mp_{222}\})\}, \{(gh, r, \{mp_{22}, mp_{31}\})\}\rangle$
$\widehat{s}_2$	$\langle\{(uh, ca, \{mp_{11}, mp_2\})\}, \{(\top_h, \top_d, \{mp_{222}\}), (gh, r, \{mp_{22}, mp_{31}\})\}\rangle$
$\widehat{s}_3$	$\langle\{(uh, ca, \{mp_{11}, mp_2\})\}, \{(\top_h, \top_d, \{mp_{222}\}), (gh, r, \{mp_{22}, mp_{31}\})\}\rangle$
$\widehat{s}_4$	$\langle\{(uh, ca, \{mp_{11}, mp_2\}), (\top_h, \top_d, \{mp_{222}\})\}, \{(gh, r, \{mp_{22}, mp_{31}\})\}, \{(gh, r, \{mp_{22}, mp_{31}\})\}\rangle$

Table 3: A mds-database $\widehat{\mathcal{MS}}_{\mathcal{DB}}$ which is the transformation of the patient trajectories in Table 1 by using the set of all most specific frequent elementary vector in Table 2.

Patients	Trajectories
$\widehat{s}_1$	$\langle\{1\}, \{3\}, \{2\}\rangle$
$\widehat{s}_2$	$\langle\{1\}, \{2, 3\}\rangle$
$\widehat{s}_3$	$\langle\{1\}, \{2, 3\}\rangle$
$\widehat{s}_4$	$\langle\{1, 3\}, \{2\}, \{2\}\rangle$

Table 4: Transformed database in Table 3

into $\langle\{(uh, ca, \{mp_{11}, mp_2\}), (\top_h, \top_d, \{mp_{222}\})\}, \{(gh, r, \{mp_{22},\ mp_{31}\})\}, \{(gh, r, \{mp_{22}, mp_{31}\})\}\rangle$ *where:*

- *The elementary vector* s_{41}, $(uh_p, ca_2, \{mp_{112}, mp_{222}\})$, *is replaced by* $(uh, ca, \{mp_{11},\ mp_2\})$ *and* $(\top_h,\ \top_d, \{mp_{222}\})$ *from* MSFEV *set in Table 2, with* $s_{41} \leqslant (uh,\ ca, \{mp_{11},\ mp_2\})$ *and* $s_{41} \leqslant (\top_h,\ \top_d, \{mp_{222}\})$.
- *The elementary vector* s_{42} *and* s_{43}, $(gh_p, r_2, \{mp_{221}, mp_{312}\})$, *are replaced by* $(gh, r, \{mp_{22},\ mp_{31}\})$ *from* MSFEV *set.*

Table 3 shows the transformation of $\mathcal{MS}_{\mathcal{DB}}$ in Table 1 based on the set of all most specific frequent elementary vectors *MSFEV* in Table 2. Transformation of $\mathcal{MS}_{\mathcal{DB}}$ denoted by $\widehat{\mathcal{MS}}_{\mathcal{DB}}$.

3.3 Step 3: mds-patterns mining:

In a classical sequential pattern mining algorithm, the sequential database to be mined should be represented as a set of pairs (sid, s) where sid is a unique sequence identifier and s is a sequence of itemsets. To apply this algorithms on $\widehat{\mathcal{MS}}_{\mathcal{DB}}$, we transformed it as follows:

- Each elementary vector in *MSFEV* is assigned a unique *id* which is used during the mining (see Table 2).
- For each sequence $\widehat{s}_i$ in $\widehat{\mathcal{MS}}_{\mathcal{DB}}$ and for each elementary vector e in $\widehat{s}_{ij}$ where $\widehat{s}_{ij} \in \widehat{s}_i$, e is replaced by its *id*.

Example 9. *The sequence* $\widehat{s}_4 = \langle\{(uh, ca, \{mp_{11}, mp_2\}), (\top_h, \top_d, \{mp_{222}\})\}, \{(gh, r, \{mp_{22}, mp_{31}\})\}, \{(gh, r, \{mp_{22}, mp_{31}\})\}\rangle$ *in* $\widehat{\mathcal{MS}}_{\mathcal{DB}}$ *(see Table 3) is transformed into* $\langle\{1, 3\}, \{2\}, \{2\}\rangle$ *as:* $(\{uh\}, \{ca\}, \{mp_{11}, mp_2\})$, $(\{gh\}, \{r\}, \{mp_{22}, mp_{31}\})$ *and* $(\top_h, \top_d, \{mp_{222}\})$ *in* $\widehat{s}_4$ *has id 1, 2 and 3 respectively in Table 2.*

Table 4 shows the transformation of the database $\widehat{\mathcal{MS}}_{\mathcal{DB}}$ in Table 3 by using the identifiers of all most specific frequent elementary vectors *MSFEV* in Table 2.

sequential patterns	mds-patterns	support
$\langle\{3\}\rangle$	$\langle(\top_h, \top_d, \{mp_{222}\})\rangle$	4
$\langle\{1\},\{2\}\rangle$	$\langle(uh, ca, \{mp_{11}, mp_2\}),(gh, r, \{mp_{22}, mp_{31}\})\rangle$	4
$\langle\{1\},\{3\}\rangle$	$\langle(uh, ca, \{mp_{11}, mp_2\}),(\top_h, \top_d, \{mp_{222}\})\rangle$	3

Table 5: All the most specific sequential patterns extracted from $\mathcal{MS}_{\mathcal{DB}}$ in Table 1 with $\sigma = 3$.

We use CloSpan [8] as a sequential pattern mining algorithm to extract sequential patterns form Table 4. Table 5 displays all sequential patterns in their transformed format and the frequent patient trajectories in which identifiers are replaced with their actual values, with $\sigma = 3$.

These 3 steps allow us to extract from heterogeneous multidimensional sequential database patterns that include elements with different levels of granularity.

4 Implementation and Experimental Validation

We conduct experiments on both real and synthetic datasets. The MMISP algorithm is implemented in Java and the experiments are carried out on a MacBook Pro with a 2.5GHz Intel Core i5, 4GB of RAM Memory running OS X 10.6.8. Extraction of sequential patterns is based on the public implementation of CloSpan algorithm [8] supplied by the IlliMine[1] toolkit.

4.1 Healthcare Trajectory

4.1.1 Mining healthcare trajectories

In order to assess the effectiveness of our approach, we run several experiments on PMSI [2], which is a French national information system for managing hospital activity with both economical and medical points of view. This section describes the results obtained after applying MMISP on a set of 2600 patients suffering from lung cancer who live in the Lorraine region, of Eastern France. We reconstituted the sequence of hospitalizations of patients who have treatment between 2006 and 2010. Each event in a sequence was characterized by the following dimensions : *hospital*, *principal diagnosis*, *medical procedures* delivered during the stay.

The *hospital* dimension was associated with a geographical taxonomy of 4 levels, first level refers to the root (France) and second, third and fourth level correspond to administrative region, administrative department and hospital respectively. Figure 4 illustrates University Hospital of Nancy (code: 540002078) as a hospital in Meurthe et Moselle, which is a department in Lorrain in the Region of France. There are 151 nodes in this taxonomy.

The *principal diagnosis* dimension could be described at 5 levels of the 10^{th} International classification of Diseases (ICD10): root , chapter, block, 3-character, 4-character, terminal nodes. Figure 5 depicts chapters such as $Neoplasms$ have specializations: block $C30-C39$ is a malignant neoplasms of respiratory, block $C50 - C50$ which is a malignant neoplasms of breast etc. The block $C30 - C39$ has specializations: malignant neoplasm of larynx (code: $C34$), malignant neoplasm of bronchus and lung (code: $C32$), etc. $C34$ (Lung cancer) has specializations: $C340$ is a cancer of the main bronchus, $C341$ is a cancer of upper lobe etc. The number of nodes in the disease taxonomy is 990 nodes.

[1] http://illimine.cs.uiuc.edu/

[2] Programme de Médicalisation des Sytèmes d'Information.

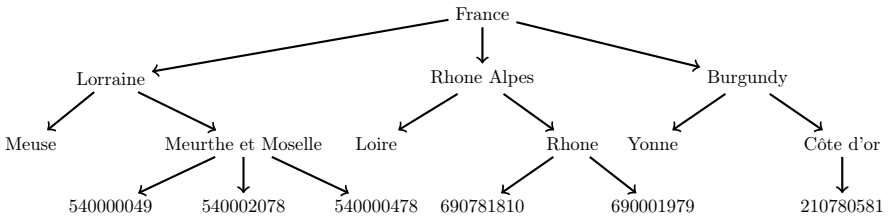


Figure 4: A geographical poset of the healthcare institution

Patients	Trajectories
$Patient_1$	$\langle(540002078, C341, \{ZBQK\}), (570023630, Z51, \{ZBQK, GFFA\}), \ldots\rangle$
$Patient_2$	$\langle(100000017, C770, \{ZBQK\}), (210780581, C770, \{ZZQK, YYYY\}), \ldots\rangle$
$Patient_3$	$\langle(210780110, H259, \{YYYY\}), (210780110, H259, \{ZZQK\}), \ldots\rangle$

Table 6: Care trajectories of 3 patients

The *medical procedures* dimension can have 5 levels of CCAM[3] which is a French nomenclature for coding procedures performed by physicians. For example, $ZBQK$ is a chest radiography for respiratory system, which is one of the procedures applied for respiratory diseases. These procedures are a part of the respiratory procedure which in turn is included in the medical procedure.

Table 6 shows an example of care trajectories for 3 patients. For example, $Patient_1$ has two hospitalizations, the first was in the University Hospital of Nancy (code: 540002078) for lung cancer (code: $C341$) where he underwent a chest radiography (code: $ZBQK$). Then, he was hospitalized in a private clinic in Metz (code: 570023630), for a chemotherapy session (code: $Z51$) where he had a chest radiography and pneumonectomy (code: $GFFA$).

In this experiment, the support value is set to 500 patients (i.e. $\sigma = 20\%$). *MMISP* generates *194 650* different frequent trajectories. Figure 6.a shows the number of discovered patterns at different thresholds according to their length. With support threshold equals 20%, the high number of length 3 and 4 patterns is explained by a combinatorial effect resulting from a high number of sequences of length 5-11 in the database. These frequent sequences correspond to the patients who underwent chemotherapy and usually had around 3 and 6 stays for 1 cycle. Figure 6.b shows the percentage discovered patterns at different thresholds according to their length. Here we can observe that 80% of patterns for which support is over 20% have at most 4 hospitalizations which correspond to the case where in each hospitalization the patient underwent chemotherapy. With support threshold equals to 100%, there is only one pattern $\langle(France, C_{34}, \{ZBQK\})\rangle$ which shows that 100% of the patients in France whose had a Lung cancer they underwent chemotherapy during their visit.

Table 7 shows the items appearing in *principal diagnosis* dimension of patterns for which support is over 40%. It can be noticed that the ICD10 tree has been mined at different levels. In the neoplasm branch, the most specific observed item is of depth 3, *" malignant neoplasm of bronchus and lung"*. In the branch of *"factors influencing health status and contact with health services"*, items of depth 4 (*"chemotherapy session for neoplasm"*) have been extracted. Children of *"Malignant neoplasm of bronchus and lung"* are not frequent enough to be extracted, but *"chemotherapy session"* appears in a sufficient proportion of trajectories to be seen. Such results cannot be obtained by representing items at an arbitrary pre-determined level.

[3]Classification Commune des Actes Médicaux : the French classification of medical and surgical procedures

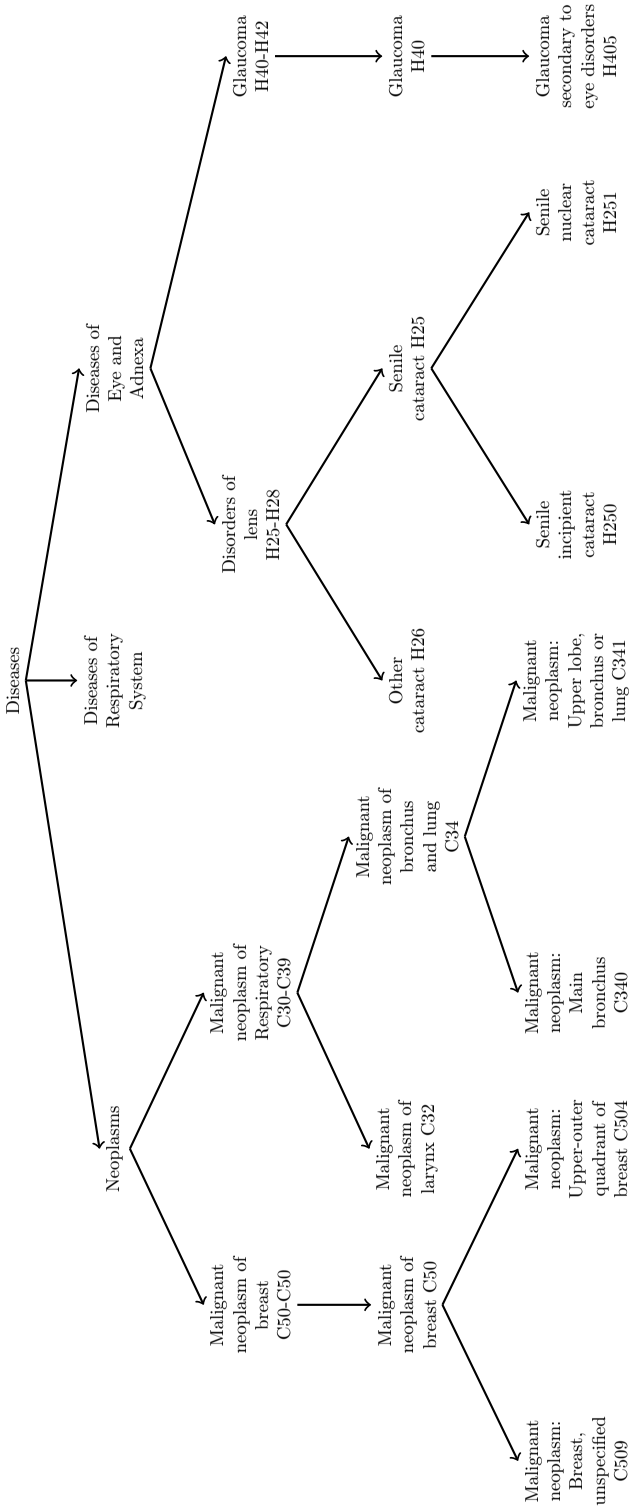
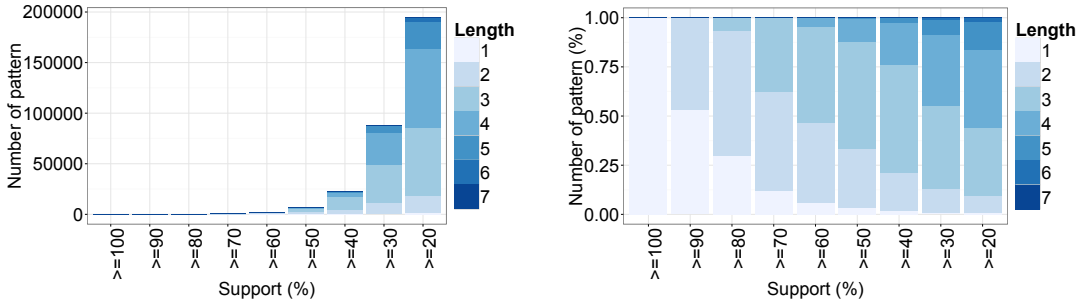


Figure 5: A disease taxonomy



(a) Number of sequential patterns by support and length.

(b) Percent of sequential patterns by support and length.

Figure 6: Distribution of sequential patterns by support and length

ICD10 level – Diagnosis Taxonomy
0– Root
1– Neoplasms
2– Malignant neoplasms of respiratory and intrathoracic organs (C30–C39)
3– Malignant neoplasm of bronchus and lung (C34)
1– Factors influencing health status and contact with health services
2– Persons encountering health services for specific procedures and health care (Z40-Z54)
3– Other medical care (Z51)
4– Chemotherapy session for neoplasm (Z511)

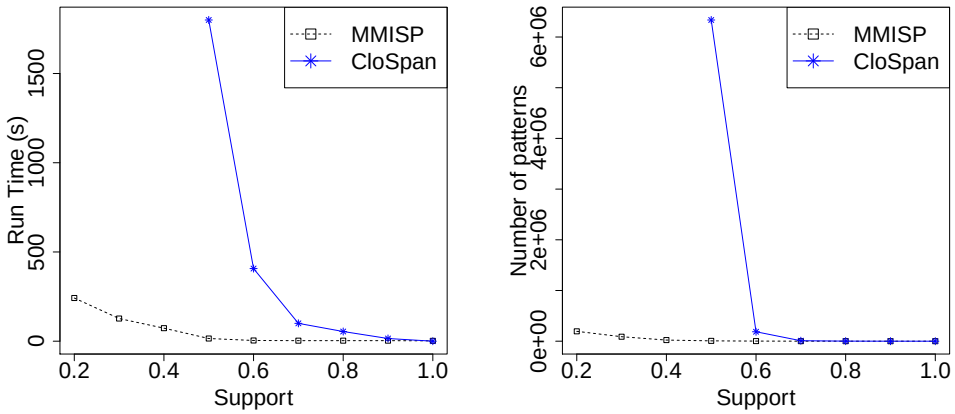
Table 7: Items extracted in the Principal Diagnosis dimension, (minimal support equals to 40%)

The mds-patterns can be analyzed per se. For example, the pattern $\langle(Lorraine, C34, \{Chemo, Pneumo\})\rangle$ shows that 93% of patients had pneumonectomy and chemotherapy for a lung cancer in any hospital in Lorraine Region in France. The pattern $\langle(Lorraine, T_{Diag}, \{Z40 - Z54\}), (Lorraine, C34, \{Chemo, Pneum\}), (Lorraine, T_{Diag}, \{Z51\})\rangle$ shows that 59% of patients had three hospitalizations where in the first one they started their treatment by undergoing chemotherapy then having pneumonectomy and chemotherapy for a lung cancer and a subsequent stay in the Lorraine Region for complementary treatments and follow-up.

This kind of information helps healthcare managers and deciders in planning and organizing healthcare resources at a regional level. Besides, sequential patterns can be seen as a condensed representation of care trajectories. As such, patterns can be reused as new variables to distinguish subgroups of patients in subsequent analysis.

4.1.2 *MMISP* versus Standard sequential pattern mining method

In this section, we compare *MMISP* with a standard sequential pattern mining method such as *CloSpan* [8]. All standard sequential pattern mining algorithms require that the dataset to be mined is composed of pairs of the form (id, seq), where id is a sequence identifier and seq is a sequence of itemsets. We transfer each sequence s_i in $\mathcal{MS}_{DB}$ with an *extended-sequence* $\widehat{s}_i$. Each elementary vector of a sequence s_i is transformed into a single itemset by replacing its elements with all its ancestors. For example, the elementary vector $(uh_p, ca_1, \{mp_{111}, mp_{221}\})$ would be replaced with $\{\top_h, uh, uh_p, \top_d, ca, ca_1, \top_{mp}, mp_1, mp_2, mp_{11}, mp_{22}, mp_{111}, mp_{221}\}$ as $uh_h \leqslant uh \leqslant \top_h$, $ca_1 \leqslant ca \leqslant \top_d$, $mp_{111} \leqslant mp_{11} \leqslant mp_1 \leqslant \top_{mp}$ and $mp_{221} \leqslant mp_{22} \leqslant mp_2 \leqslant \top_{mp}$. Then, we apply *CloSpan* as a sequential pattern mining algorithm on the *extended-sequential*



(a) Runtime sequences over frequency threshold (b) Number patterns over frequency threshold

Figure 7: *MMISP* versus *CloSpan*

database. This way of managing hierarchies has been used in *GSP* which is proposed by [5].

Our main goal is to evaluate the quality of the frequent trajectories mined with *MMISP* and its performance compared to naive approach using *CloSpan*. We firstly transform the 2400 patients trajectories, then we apply the two approaches using minimum supports threshold ranging from 100% to 20%. Figure 7 reports the execution time and the number of frequent trajectories according to different values of support threshold for both *CloSpan* and *MMISP*.

Actually, *CloSpan* cannot finish its calculations for support threshold less than 50% because the transformation increases number of items in itemset and generates a large number of density similar sequences. Whereas, *MMISP* runs in acceptable time for support as low as 20%.

MMISP is able to extract condensed frequent trajectories w.r.t. the ones mined by *CloSpan*. For example, the trajectory $\langle\{\top_h, uh, \top_d, ca, \top_{mp}, mp_1, mp_2, mp_{11}\}\ \{\top_h, gh, \top_d, r, \top_{mp}, mp_2, mp_3, mp_{22}, mp_{31}\}\rangle$ generated by *CloSpan* contains redundant information as a hospitalization containing mp_{22} will also contain $\top_{mp}$ and mp_2. *MMISP* is not affected by this pattern because it extracts just the most specific frequent elementary vector in the first step of the algorithm.

CloSpan extracts all the frequent trajectories (i.e., the general and specific ones) while *MMISP* generates just the more specific ones. For example, if $\langle(\top_h, \top_d, \{mp_{222}\})\rangle$ is a frequent trajectory. *MMISP* does not extract the trajectories which are more general like $\langle(\top_h, \top_d, \{mp_{22}\})\rangle$ while *CloSpan* extracts both the general and specific ones. Figure 7 shows the difference between the number of frequent trajectories extracted by *CloSpan* and *MMISP*. For example, with a support threshold of 70 %, *MMISP* extracts *565 frequent trajectories* while *CloSpan* extracts *6 335 683 frequent trajectories*.

Finally, we may conclude that:

- *MMISP* is more efficient than *CloSpan* over *extended-sequential* database with low support threshold.
- The trajectories extracted by *CloSpan* require post processing while this is not the case with *MMISP*.

- *MMISP* extracts just the most specific frequent trajectories while *CloSpan* extracts both general and specific ones. This means that *CloSpan* extracts a huge number of patterns. Analyzing all of them is not an easy task for healthcare managers and decision makers.

4.1.3 $MMISP$ versus M^3SP

Another experiment is carried out for comparing M^3SP with *MMISP*. Our main goal is to evaluate the effectiveness of sequential patterns mined by *MMISP* compared to the ones extracted by M^3SP. For this purpose, we applied M^3SP with hospital, diagnosis and medical procedures as analysis dimensions. The support value is set to 500 patients (i.e. $\sigma = 20\%$). Table 8 reports an example of the extracted patterns with M^3SP and *MMISP*.

Firstly, we observe that *MMISP* is able to extract condensed trajectories w.r.t. the ones mined by M^3SP. For example, *48 trajectories*, *Pattern #1*,..., *Pattern #48*, generated by M^3SP are summarized by *3 ones*, *Pattern #50*, *Pattern #51* and *Pattern #52*, extracted by *MMISP* (see Table 8). This shows that the rigid structure of multidimensional item assumed by M^3SP limits the expressivity of the results.

Besides that, in M^3SP, several dimensions can be repeated at the same hospitalization. For example, in M^3SP, *Pattern #48* represents one hospitalization including 9 multidimensional items. Each multidimensional item is associated with the same value of hospital and diagnosis (540002078 and C341) and different values of medical procedures. In *MMISP*, *Pattern #50* (extracted by *MMISP*) represents the same trajectory as *Pattern #48*. *Pattern #50* has one elementary vector with three elements: hospital 540002078, diagnosis $C341$ and a set of medical procedure $\{ZBQK, DEQP, GFFA, GLLD, GELD,\ ZZQK, GELE, FCFA,\ AGLB\}$. *Pattern #50* is much more compact and informative than *Pattern #48*.

Given a minsup threshold, *MMISP* extracts sequential patterns that are not found by M^3SP. For instance, *Pattern #53* extracted by *MMISP* is not found by M^3SP. This is due to the fact that *MMISP* extracts new frequent hospitalizations not extracted by M^3SP. For instance e=*(Lorraine,* $\top_{Diseases}$*, {GEQE, ACQH, ZCQH})* and e'=*(Lorraine,Diseases of the Respiratory, {ACQH})* are extracted by *MMISP*. As e and e' are frequent and not comparable (i.e. $e \not\leqslant e'$ and $e' \not\leqslant e$), M^3SP extracts only *(Lorraine, Diseases of the Respiratory, ACQH)* and not *(Lorraine,* $\top_{Diseases}$*, ACQH)* as *(Lorraine, Diseases of the respiratory, ACQH)* is more specific than *(Lorraine,* $\top_{Diseases}$*, ACQH)*.

From a quantitative point of view, *MMISP* extracts *419 frequent hospitalizations* with *194 650 frequent trajectories* while M^3SP extracts *102 multidimensional items* with *1 242 frequent trajectories*. The execution time of M^3SP is about 82 *seconds* while *MMISP* takes about 242 *seconds*.

Finally, we may conclude that:

- Several frequent trajectories generated by M^3SP can be summarized by only one mined by *MMISP*.
- Several multidimensional items generated by M^3SP can be summarized by only one elementary vector in *MMISP*.
- One elementary vector in *MMISP* represents one hospitalization in the trajectory while one multidimensional item in M^3SP represents only a part of hospitalization in the trajectory.
- Some frequent trajectories can be extracted by *MMISP* while they can not be extracted by M^3SP.

Methods	id	Trajectory Patterns
M^3SP	1	$\langle\{(540002078, C341, GFFA)(540002078, C341, ZZQK)\}\rangle$
	2	$\langle\{(540002078, C341, DEQP)(540002078, C341, GFFA)(540002078, C341, ZZQK)\}\rangle$
		$\vdots$
	48	$\langle\{(540002078, C341, ZBQK)(540002078, C341, DEQP)(540002078, C341, GFFA)$ $(540002078, C341, GLLD)(540002078, C341, GELD)(540002078, C341, ZZQK)$ $(540002078, C341, GELE)(540002078, C341, FCFA)(540002078, C341, , AGLB)\}\rangle$
	49	$\langle(Lorraine, Diseases\ of\ the\ respiratory, ACQH)\rangle$
$MMISP$	50	$\langle\{(540002078, C341, \{ZBQK, DEQP, GFFA, GLLD, GELD, ZZQK, GELE, FCFA, AGLB\}\rangle$
	51	$\langle(540002078, C341, \{DEQP, GELD, GELE, ZZQK, AGLB, GLLD, GFFA\})\rangle$
	52	$\langle(540002078, C341, \{ZBQK, DEQP, GELD, GELE, ZZQK, GLLD, GFFA\})\rangle$
	53	$\langle(Lorraine, \top_{Diseases}, \{GEQE, ACQH, ZCQH\})\rangle$

Table 8: Some patterns obtained by M^3SP and MMISP.

4.2 Experiments on Synthetic Datasets

In this experiment, we study the scalability of the MMISP approach. We consider the number of extracted patterns and the running time with respect several parameters:

- number of elements in each elementary vector.
- depth of the taxonomy of each component.
- number of elementary vectors in each sequence (i.e. sequence length).
- number of sequences in a sequential database.

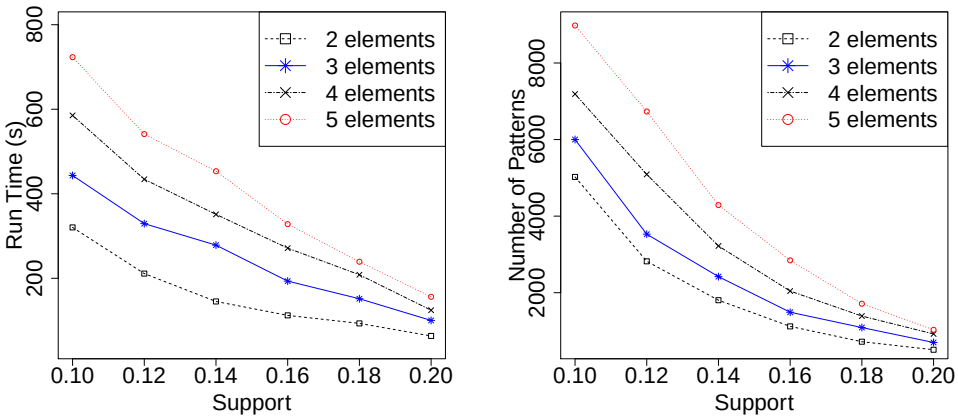
The first batch of synthetic data generated contains 1000 sequences defined over 2, 3, 4 and 5 elements in the elementary vector for each sequence. Each sequence contains 15 elementary vectors. Each taxonomy is defined over 3 levels of granularity between its items. Figure 8 reports the results according to different values of support threshold for different numbers of elements in the elementary vector. The running time increases for each newly added component.

In Figure 9, we study the performance of *MMISP* by considering several levels of granularity for each taxonomy. We generated 1000 sequences defined over 15 elementary vectors. Each elementary vector has 3 components. Each taxonomy of the element is defined over 3, 4, 5, 6 levels of granularity. The number of extracted patterns does not change with each newly added level as *MMISP* extracts only the most specific sequential patterns. The execution time increases with the increase in the number of levels in a taxonomy because of the complexity of the product of taxonomies in the first step of *MMISP* which generates all the frequent elementary vectors.

We study the performance of *MMISP* and the number of extracted patterns with respect the number of sequences in a sequential database and the length of each sequence. Figure 10 shows the execution time and the number of patterns extracted for 1000 sequences with 3 dimensions associated with taxonomy with 3 levels of granularity and with varying sequence length. The execution time increases with the increase in length of a sequence. This is due to the third step of *MMISP* which uses *CloSpan* to mine the transformed sequences.

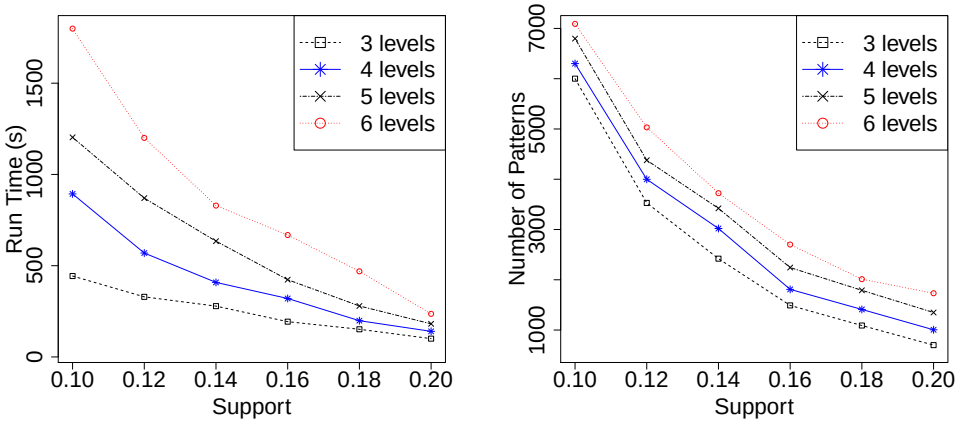
Figure 11 shows the running time and the number of patterns extracted for several number of sequences (1000, 2000, 3000, 4000 and 5000 sequences) also with 3 dimensions associated with taxonomy with 3 levels of granularity. In these experiments, the sequence length is roughly 15.

Figures 8 - 11 highlight the fact that MMISP is efficient in terms of runtime for a large panel of sequences with varying different parameters.



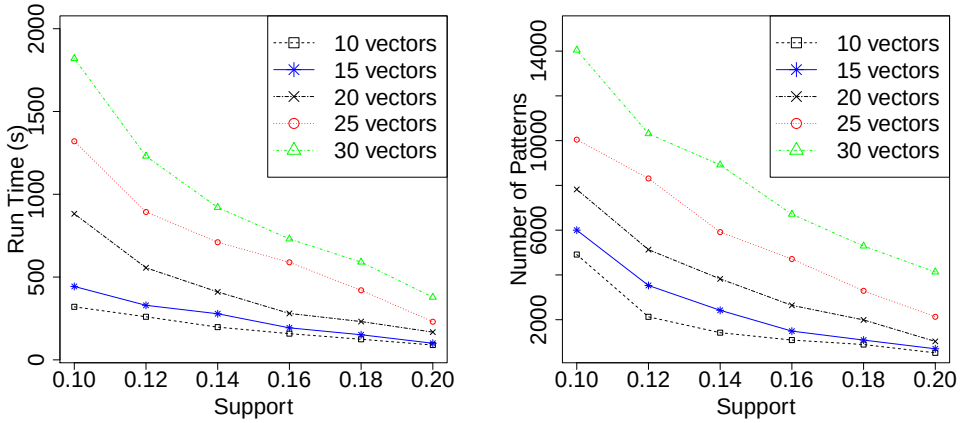
(a) Runtime sequences over frequency threshold. (b) Number patterns over frequency threshold.

Figure 8: Number of sequential pattern extracted and Running time obtained by *MMISP* with varying in the length of elementary vectors.



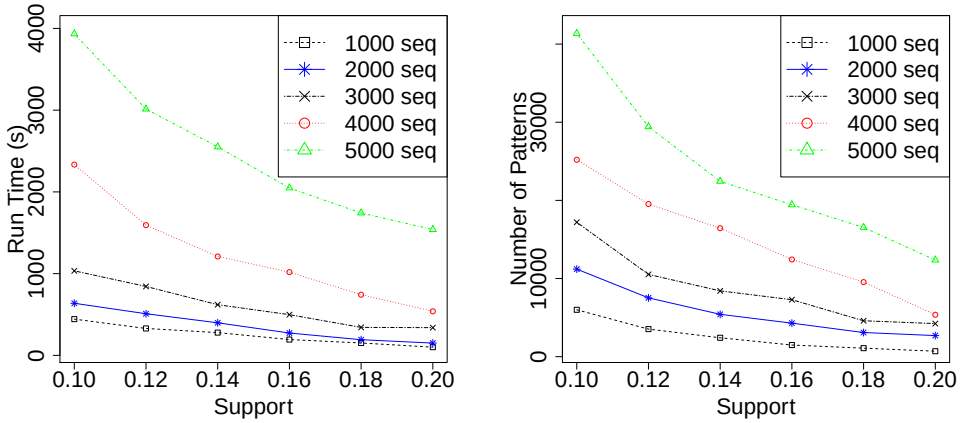
(a) Runtime sequences over frequency threshold. (b) Number patterns over frequency threshold.

Figure 9: Number of extracted pattern (right) and Running Time (left) obtained by MMISP with varying over the levels of granularity between items of the taxonomies.



(a) Runtime sequences over frequency threshold. (b) Number patterns over frequency threshold.

Figure 10: Number of sequential pattern extracted and Running time for a large panel of sequences when varying over sequence length.



(a) Runtime sequences over frequency threshold. (b) Number patterns over frequency threshold.

Figure 11: Number of sequential pattern extracted and Running time for a large panel of sequences when varying over several number of sequences.

5 Conclusion

This report presents a new approach to extract sequential patterns from heterogeneous multi-dimensional sequential database. We provide formal definitions and propose a new algorithm *MMISP* to mine this kind of data. This method mined the database which are often represented as a sequence of vector of heterogeneous elements with different types (i.e, item and itemset) takes into account background knowledge lying in term taxonomies for each dimension. We conduct experiments on both real-world and synthetic datasets. The method is applied on real-world data where the problem is to mine healthcare patients trajectories and gives potential interesting patterns for healthcare specialists. For future work, we are planning to use statistical significance tests to evaluate the sequential patterns extracted and choose the most significant ones. On the other hand, proposing a graphical interface to visualize and query the sequential patterns.

References

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, ser. ICDE '95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 3–14. [Online]. Available: http://dl.acm.org/citation.cfm?id=645480.655281

[2] C. Chothia and M. Gerstein, "Protein evolution. how far can sequences diverge?" *Nature*, vol. 6617, no. 385, pp. 579–581, 1997.

[3] Q. Yang and H. H. Zhang, "Web-log mining for predictive web caching," *IEEE Trans. on Knowl. and Data Eng.*, vol. 15, no. 4, pp. 1050–1053, Jul. 2003.

[4] J. Serrà, H. Kantz, X. Serra, and R. G. Andrzejak, "Predictability of music descriptor time series and its application to cover song detection," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 2, pp. 514–525, 2012.

[5] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, ser. EDBT '96. London, UK, UK: Springer-Verlag, 1996, pp. 3–17. [Online]. Available: http://dl.acm.org/citation.cfm?id=645337.650382

[6] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Mach. Learn.*, vol. 42, no. 1-2, pp. 31–60, Jan. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1007652502315

[7] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Mining sequential patterns by pattern-growth: The prefixspan approach," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1424–1440, 2004.

[8] X. Yan, J. Han, and R. Afshar, "Clospan: Mining closed sequential patterns in large datasets," in *In SDM*, 2003, pp. 166–177.

[9] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal, "Multi-dimensional sequential pattern mining," in *CIKM*, 2001, pp. 81–88.

[10] C. Zhang, K. Hu, Z. Chen, L. Chen, and Y. Dong, "Approxmgmsp: A scalable method of mining approximate multidimensional sequential patterns on distributed system," in *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, vol. 2. IEEE, 2007, pp. 730–734.

[11] C.-C. Yu and Y.-L. Chen, "Mining sequential patterns from multidimensional sequence data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 136–140, 2005.

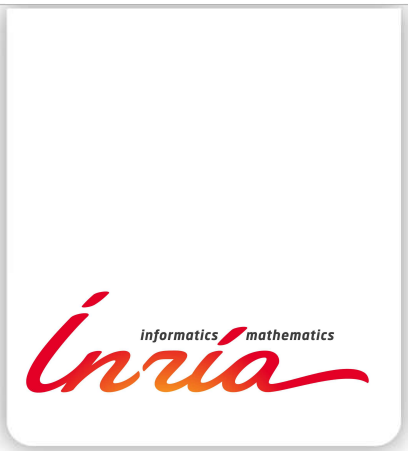
[12] M. Plantevit, A. Laurent, D. Laurent, M. Teisseire, and Y. W. Choong, "Mining multidimensional and multilevel sequential patterns," *TKDD*, vol. 4, no. 1, pp. 1–37, 2010.

[13] S. Wodak and J. Janin, "Structural basis of macromolecular recognition." *Adv Protein Chem*, vol. 61, pp. 9–73, 2002.

[14] C. Raïssi and J. Pei, "Towards bounding sequential patterns," in *KDD*, 2011, pp. 1379–1387.

[15] M. J. Zaki and K. Gouda, "Fast vertical mining using Diffsets," in *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2003.

Contents



RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399