A contribution to the discovery of multidimensional patterns in healthcare trajectories

Elias Egho[1], Nicolas Jay[1], Chedy Raïssi[1], Dino Ienco[2,3], Pascal Poncelet[2,3],
Maguelonne Teisseire[2,3] and Amedeo Napoli[1]

[1] LORIA(CNRS - Université de Lorraine)/Inria Nancy Grand Est
{firstname.lastname}@inria.fr
[2] Irstea, UMR TETIS, F-34093 Montpellier
{firstname.lastname}@teledetection.fr
[3] LIRMM, Univ. Montpellier 2, Montpellier
{firstname.lastname}@lirmm.fr

Abstract. Sequential pattern mining is aimed at extracting correlations among temporal data. Many different methods were proposed to either enumerate sequences of set valued data (i.e., itemsets) or sequences containing dimensional items. However, in real-world scenarios, data sequences are described as combination of both multidimensional items and itemsets. These heterogeneous descriptions cannot be handled by traditional approaches. In this paper we propose a new approach called MMISP (*Mining Multidimensional Itemset Sequential Patterns*) to extract patterns from complex sequential database including both multidimensional items and itemsets. The novelties of the proposal lies in: (i) the way in which the data are efficiently compressed; (ii) the ability to reuse and adopt sequential pattern mining algorithms and (iii) the extraction of new kind of patterns. We introduce a case-study on real-world data from a regional healthcare system and we point out the usefulness of the extracted patterns. Additional experiments on synthetic data highlights the efficiency and scalability of the approach *MMISP*.

Keywords: complex sequential patterns, multidimensional sequential patterns, data mining, complex data

1 Introduction

Real-world databases can be viewed as large and complex sources of information that can be analyzed for discovering new knowledge units or for decision making [15]. When the temporal dimension is also considered, every bit of information or event is associated with a timeline describing a total order over events. This total ordering introduces complexity in the extraction process. Many efficient approaches were developed to mine patterns depending on time or order, such as PrefixSpan [7], SPADE [16], ClosSpan [12],...etc. These approaches focus on a single dimensional sequence dataset. However, there are many situations in which a database can be multidimensional, i.e. several characteristics of data can be ordered over time. Pinto et al. [8], Zhang et al. [17] and Yu et al. [14] introduced the notion of multidimensionality in a sequence and proposed several algorithms to mine this type of data. In data warehouse environments, a background knowledge is usually available in form of taxonomies, classification or concept hierarchies. Based on that, Plantevit et al. introduced M^3SP [9], an algorithm able to incorporate several dimensions and the possible associated posets within the sequential pattern mining process.

The above approaches focus on sequences of homogeneous items. They do not pay attention to real-world complex data described by a vector of heterogeneous elements with different types, i.e. item or itemset. For example, in the healthcare domain, a patient trajectory is defined as a sequence of hospitalizations, where each hospitalization is defined as a vector of three heterogeneous elements: (i) healthcare institution, (ii) diagnosis and (iii) set of medical procedures. The healthcare institution and the diagnosis can be encoded as variables taking values, where values are organized within posets, while medical procedures are not comparable. This example shows that each dimension in data has to be managed in a proper and suitable way.

In this paper, we present an approach for mining multidimensional and heterogeneous patterns from medical patient trajectories, i.e. the trajectory of a patient visiting several hospitals. Our objective is to discover interesting patterns able to characterize patient stays and associated medical procedures. Such patterns can be interpreted by healthcare professionals to better understand patient pathways and improve the organization of care. Such multidimensional and heterogeneous patterns have to be mined by adapted and suitable methods. Accordingly, in this paper, we propose a new method to extract sequential patterns from databases including sequences of heterogeneous vectors. In addition, the approach is able to take into account background knowledge lying in term posets. The approach is original and efficient. An adapted algorithms is proposed which shows a very good behaviour on real-world medical data.

The remainder of this paper is organized as follows, Section 2 describes related work in classical and multidimensional sequential patterns. Section 3 introduces the problem statement and a running example. The algorithm for extracting complex frequent patterns is described in Section 4. Section 5 presents experimental results from both quantitative and qualitative points of views and Section 6 concludes the paper.

2 Related Work

Agrawal and Srikant [1] introduced the problem of mining sequential patterns over large sequential databases. Formally, given a set of sequences, where each sequence is a list of transactions ordered by time and each transaction is a set of items, the problem amounts to find all frequent subsequences that appear a sufficient number of times with a user-specified minimum support threshold (*minsup*). Following the work of Agrawal and Srikant many studies have contributed to the efficient mining of sequential patterns [6]. Most of them are based on the *Apriori* property, which states that any super pattern of a non-frequent pattern cannot be frequent. The main algorithms are PrefixSpan [7], SPADE [16], SPAM [2], PSP [5], DISC [3], PAID [13] and FAST [10]. All these algorithms aim at discovering sequential patterns from a set of sequences of itemsets such as customers who frequently buy DVDs of episodes I, II and III of Stars Wars, then buy within 6 months episodes IV, V, VI of the same famous epic space opera.

Many studies about sequential patterns discovery focus on single-dimensional sequences. However, in many situations, the database is multidimensional in the sense that items can be of different nature. For example, a consumer database can hold information such as article price, gender of the customer, location of the store and so on. Pinto et al [8] proposed the first work for mining multidimensional sequential patterns. In this work, a *multidimensional sequential database* is defined as a schema $(ID, D_1, ..., D_m, S)$, where ID is a unique customer identifier, $D_1, ..., D_m$ are dimensions describing the data and S is the sequence of itemsets. A *multidimensional sequence* is defined as a vector $\langle\{d_1, d_2, ..., d_m\}, S_1, S_2, ..., S_l\rangle$ where $d_i \in D_i$ for $(i \leqslant m)$ and $S_1, S_2, ..., S_l$, are the itemsets of sequence S. For instance, $\langle\{Paris, Male\}, \{mp_1, mp_2\}, \{mp_3\}\rangle$ describes a male patient who underwent procedures mp_1 and mp_2 in Paris and then underwent mp_3 also in Paris. Here, dimensions remain constant over time, such as the location of the treatment. This means that it is not possible to have a pattern indicating that when the patient underwent procedures mp_1 and mp_2 in Paris then he underwent mp_3 in Nancy. Among other proposals, Yu et al [14] proposed two methods AprioriMD and PrefixMDSpan for mining multidimensional sequential patterns in the web domain. This study considers pages, sessions and days as dimensions. Actually, these three different dimensions can be projected into a single dimension corresponding to web pages, gathering web pages visited during a same session and ordering sessions w.r.t the day as order.

In real world applications, each dimension can be represented at different levels of granularity, by using a poset. For example, apples in a market basket analysis can be either described as fruits, fresh food or food. The interest lies in the capacity of extracting more or less general/specific multidimensional sequential patterns and overcome problems of excessive granularity and low support. Srikant and Agrawal [11] proposed GSP which uses posets for extracting sequential patterns. The basic approach is based on replacing every item with all the ancestors in the poset and then the frequent sequences are generated. This approach is not scalable in a multidimensional context because the size of the database becomes the product of maximum height of the posets and number of dimensions. Plantevit et al [9] defined a *multidimensional sequence* as an ordered list of multidimensional items, where a *multidimensional item* is a tuple $(d_1, ..., d_m)$ where d_i is an item associated with the i^{th} dimension. They proposed M^3SP, an approach taking both aspects into account where each dimension is represented at different levels of granularity, by using a poset. M^3SP is able to search for sequential patterns with the most appropriate level of granularity. Their approach is based on the extraction of the most specific frequent multidimensional items, which are then used as alphabet to rephrase the original database. Then, M^3SP uses a standard sequential pattern mining algorithm to extract multidimensional sequential patterns. However, M^3SP is not adapted to mine sequential databases, where sequences are defined over a combination of sets of items and items lying in a poset. Then it is not possible to have a pattern indicating that when the patient went to uh_p for a problem of cancer ca, where he underwent procedures mp_1 and mp_2, then he went to gh_l for the same medical problem ca, where he underwent mp_3 (i.e, $\langle(uh_p, ca, \{mp_1, mp_2\}), (gh_l, ca, \{mp_3\})\rangle$).

Compared to M^3SP, the main contribution of this article is to generalize the concept of multidimensional sequence by considering multidimensional itemsets instead of multidimensional items. In our approach, an event in a sequence can be seen as a vector of itemsets, whenever M^3SP represents an event as a tuple of atomic elements. This restriction prevents M^3SP from extracting condensed and heterogeneous patterns.

3 Problem Statement

3.1 An introductory example

Firstly we propose an example to illustrate the present approach. The French healthcare system called PMSI [4] is a national information system used in France to manage hospital activity with both an economical and a medical points of view [4]. In this system, each patient's stay is a standardized record of administrative and clinical data. Accordingly, each hospitalization can be formalized along three dimensions: (i) healthcare institution, (ii) diagnosis and (iii) medical procedures. The first two dimensions, i.e. healthcare institutions and diagnosis, are considered as variables whose values are lying in a poset. The last dimension is about medical procedures and can be considered as a variable which is set-valued. The basic sets of the healthcare institutions H, the diagnosis DG and the medical procedures MP, are the following:

- H = $\{T_h, uh, gh, uh_p, uh_n, gh_p, gh_l\}$.
- DG = $\{T_d, ca, r, r_1, r_2, ca_1, ca_2, ca_3\}$.
- MP = $\{mp_1, mp_2, mp_3, mp_4\}$.

Hospital and diagnosis can be described at different levels of granularity through two posets $(H, \leqslant)$and $(DG, \leqslant)$ which are defined in Figure 1.

In the poset $(H, \leqslant)$, uh_p and uh_n denote university hospitals, uh, with $uh_p \leqslant uh$ and $uh_n \leqslant uh$; gh_p and gh_l are general hospitals, gh, with $gh_p \leqslant gh$ and $gh_l \leqslant gh$.

The hospitalization of a patient is defined as a vector with three components, (h, dg, mp). The component h refers to institution and its value lies in the

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

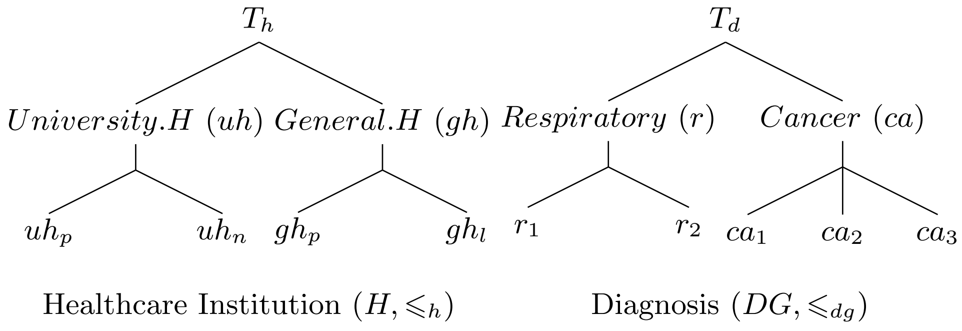


Fig. 1: The posets for the healthcare institution set H and the diagnosis set DG

partially ordered set $(H, \leqslant)$. The same thing applies to dg with $(DG, \leqslant)$. The component mp refers to medical procedures and is a set-valued.

Then, $(uh_p, ca_1, \{mp_1, mp_2\})$ describes an hospitalisation, where $uh_p \in H, ca_1 \in DG$ and $\{mp_1, mp_2\} \subseteq MP$.

3.2 Basic definitions

Definition 1. *(Elementary sequence)*

An elementary sequence $e = (e_1, e_2, ..., e_n)$ is defined as a vector of n elements where an element can be either:

- *an atomic element taken from a partially order set.*
- *a subset of a non ordered set.*

Definition 2. *(Ordering of elementary sequences)*

Given two elementary sequences $e = (e_1, e_2, ..., e_n)$ and $e' = (e_1^{'}, e_2^{'}, ..., e_n^{'})$, e is more specific than e', denoted by $e \leq_e e'$, if for all $i = 1...n$ we have:
$e_i \leqslant e_i'$ *where e_i, e_i' are elements of a poset.*
$e_i' \subseteq e_i$ *where e_i, e_i' are sets.*

Example 1. $e = (uh_p, ca_1, \{mp_1, mp_2\})$ is an elementary sequence more specific than $e' = (uh, T_d, \{mp_1\})$ as:

- $uh_p \leqslant uh;\ uh_p, uh \in H.$
- $ca_1 \leqslant T_d;\ ca_1, T_d \in DG.$
- $\{mp_1\} \subseteq \{mp_1, mp_2\};\ \{mp_1\}, \{mp_1, mp_2\} \subseteq MP.$

Example 2. A patient healthcare trajectory can be considered as a set of hospitalisations ordered over time. $\langle(uh_p, ca_1, \{mp_1, mp_2\}), (gh_l, r_1, \{mp_2\})\rangle$ represents a patient trajectory with two hospitalizations. The patient was first admitted in the university hospital uh_p for a lung cancer ca_1, and underwent procedures mp_1 and mp_2. Then he visited the general hospital gh_l for a pneumonitis r_1 where he underwent procedure mp_2.

Definition 3. *(Sequence)*

A sequence is represented as $S = \langle S_1, S_2, ..., S_l \rangle$ *is a set of elementary sequences* S_i *ordered by the temporal order relation* $<_t$*, such as* $S_1 <_t S_2 <_t S_3 ... <_t S_l$.

Definition 4. *(Ordering of sequences)*

Given two sequences $S = \langle S_1, S_2, ..., S_l \rangle$ *and* $T = \langle T_1, T_2, ..., T_{l'} \rangle$*,* S *is more specific than* T*, denoted by* $S \leq_s T$*, if there exist a set of indices* $1 \leq i_1 < i_2 < ... < i_{l'} \leq l$ *such that* $S_j \leq_e T_{i_j}$ *for all* $j = 1 \ldots l'$ *and* $l' \leqslant l$*. The most specific sequence is also the longest sequence as* $l' \leqslant l$.

Example 3. $S = \langle (uh_p, ca_1, \{mp_1, mp_2\}), (gh_l, r_1, \{mp_2\}) \rangle$ is a sequence with two elementary sequences $S_1 = (uh_p, ca_1, \{mp_1, mp_2\})$ and $S_2 = (gh_l, r_1, \{mp_2\})$ where $S_1 <_t S_2$. The sequence S is more specific than $T = \langle (uh, T_d, \{mp_1\}) \rangle$, $S \leq_s T$, as $(uh_p, ca_1, \{mp_1, mp_2\}) \leq_e (uh, T_d, \{mp_1\})$.

A set of m sequences $S_{DB} = \{S^1, S^2, ..., S^m\}$ is called hereafter a *"sequential database"*. An example is given in Table 1 where there are four sequences describing four patient trajectories.

Definition 5. *(Support of sequence S)*

Let $S_{DB} = \{S^1, ..., S^k\}$ *a sequential database. The support of a sequence* S*, denoted by* $support_s(S)$ *is defined as follows:*

$$support_s(S) = |\{S^i \in S_{DB}; S^i \leq_s S\}|$$

Definition 6. *(Sequential pattern)*

Given a positive integer σ *as a minsup threshold and a sequential database* S_{DB}*, the sequence* S *is called a sequential pattern in* S_{DB} *iff* $support_s(S) \geq \sigma$.

Example 4. The sequence $S = \langle (uh_p, ca, \{mp_1, mp_2\}), (gh_l, r, \{\}) \rangle$ has a support equals to 3 (i.e, $support_s(S) = 3$) in the sequential database S_{DB} (see Table 1). It is a sequential pattern according to minsup threshold equals to 3 (i.e, $\sigma = 3$).

Patients	Trajectories
S^1	$\langle (uh_p, ca_1, \{mp_1, mp_2\}), (uh_p, ca_1, \{mp_1\}), (gh_l, r_1, \{mp_3\}) \rangle$
S^2	$\langle (uh_n, ca_1, \{mp_4\}), (uh_p, ca_2, \{mp_1, mp_2\}), (gh_l, r_1, \{mp_2\}) \rangle$
S^3	$\langle (uh_n, ca_3, \{mp_4\}), (gh_l, r_2, \{mp_3\}) \rangle$
S^4	$\langle (uh_p, ca_2, \{mp_1, mp_2\}), (gh_p, r_2, \{mp_3\}), (gh_l, r_2, \{mp_2\}) \rangle$

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

4 Mining Sequential Patterns

4.1 Most specific sequential patterns

In this section, we present the problem of mining *the most specific sequential patterns.* Given a sequential database, mining all possible frequent patterns results in a huge amount of information that is difficult to manage from the analyst point of view. To overcome this problem, we extract a set of sequential patterns that are not only frequent but also as specific as possible. This second constraint allows to reduce the volume of the final result discarding redundant patterns. An extracted pattern is called a *"most specific sequential pattern"*.

Definition 7. *(Most Specific Sequential Pattern (MSSP))*
Given a positive integer σ as minsup threshold and a sequential database S_{DB}, a sequence S is a most specific sequential pattern in S_{DB} or MSSP if and only if $support_s(S) \geq \sigma$ and there does not exist any sequence T such that $T \leq_S S$ with $support_s(S) = support_s(T) \geq \sigma$

Actually, frequency is monotone (i.e whenever S is frequent, any generalization of S is also frequent). For example, if $S = \langle(uh_p, c, \{mp_1, mp_2\})\rangle$ is frequent then $T = \langle(uh, c, \{mp_1\})\rangle$ which is more general than S is also frequent. Thus, the most specific sequential patterns are sufficient to retrieve all sequential patterns.

Example 5. Let $\sigma = 3$ (i.e. a sequence is frequent if it appears at least three times in S_{DB}). The sequence $S = \langle(uh, ca, \{mp_1\})\rangle$ is frequent but is not the most specific one because the sequence $T = \langle(uh_p, ca, \{mp_1, mp_2\})\rangle$ is frequent and verifies $T \leq_s S$ and $support_s(S) = support_s(T)$. The sequence $T = \langle(uh_p, ca, \{mp_1, mp_2\})\rangle$ is a most specific frequent sequence in S_{DB}: (i) T is frequent ($support_s(S) \geq \sigma$) and (ii) there is no other sequence in S_{DB} which is frequent, more specific than T and has the same support of T.

The objective of our approach is to extract a set of frequent sequential patterns that are as specific as possible. In the next section, we present the algorithm *MMISP* for finding most specific sequential patterns. The basic idea of *MMISP* consists in transforming the sequential database into an *"adapted form"* and then to apply a standard algorithm for sequential mining.

4.2 The MMISP algorithm

MMISP is based on three steps:

First step (Extraction of frequent elementary sequences)
The algorithm searches for the frequent and specific elementary sequences. It extracts these frequent elementary sequences w.r.t. partial ordering existing between the elements (posets).

Second step (Transformation)

In this step, all frequent elementary sequences extracted in the previous step are mapped to an alternate representation. Then, the sequential database is encoded by using this new representation.

Third step (Mining)

In this step, a standard sequential algorithm is applied on the sequential database produced at the preceding step.

4.2.1 Extracting all frequent elementary sequences

Firstly, *MMISP* considers the elementary sequences in all sequences of S_{DB}. Actually if the elementary sequence is not frequent in a sequential database S_{DB} it does not belong to an extracted sequential pattern. For example, If we have two sequences:

S^1	$\langle\{s_2\}\{s_1\}\{s_3\}\rangle$
S^2	$\langle\{s_2\}\{s_4\}\{s_3\}\rangle$

Given a support threshold $\sigma = 2$, pattern $\langle\{s_2\}\{s_3\}\rangle$ is frequent. In this example, s_2 and s_3 are frequent items while s_1 and s_4 are not frequent items. In the following, we only consider sequential patterns composed of frequent items.

MMISP extracts all the frequent elementary sequences in a sequential database by taking into account the partial order relation between their elements. The support of an elementary sequence s is defined as follows:

Definition 8. *(Support of an elementary sequence e)*

Let $S_{DB} = \{S^1, ..., S^m\}$ a sequential database. The support of an elementary sequence $e = (e_1, e_2, ..., e_n)$, denoted by $support_e(e)$, is defined as follows:

$$support_e(e) = |\{S^i \in S_{DB}; S^i = \langle S^i_1, ..., S^i_l\rangle \text{ and } \exists j \in [1, .., l];\ S^i_j \leq_e e\}|$$

Example 6. The support of $(gh, r, \{mp_3\})$ is 3 in S_{DB} as:

- $S^1_3 = (gh_l, r_1, \{mp_3\}) \leq_e (gh, r, \{mp_3\})$.
- $S^3_2 = (gh_l, r_2, \{mp_3\}) \leq_e (gh, r, \{mp_3\})$.
- $S^4_2 = (gh_p, r_2, \{mp_3\}) \leq_e (gh, r, \{mp_3\})$.

The frequent elementary sequences are ordered over a poset L, denoted by $(L, \leq_e)$, as follows. Firstly, we generate the most general elementary sequences. In the running example, we consider triples of the form (h, dg, mp) and the most general triple is $(T_h, T_d, \{\})$ where T_h and T_d denote the most general items in the posets H and D respectively, and the empty set $\{\}$ denotes the most general item in the set MP. Then, we recursively generate new elementary sequences by starting from the most general one. This generation is done by replacing each element $e_1, e_2, ..., e_n \in e$ with all of its direct specializations.

The set of all direct specializations of an element e_i, denoted by $desc(e_i)$, is defined as follows:

Definition 9. *(Direct specializations of an element e_i)*

Let e_i be an element in the set D_i. The direct specializations of e_i, denoted by $desc(e_i)$, is defined by:

$$desc(e_i) = \begin{cases} \{u \in D_i; u \leq e_i \text{ and } \nexists w \in D_i;\ u \leq w \text{ and } w \leq e_i\} & \text{if } D_i \text{ is a poset.} \\ \{e_i \cup \{u\}; u \in D_i \setminus e_i\} & \text{if } D_i \text{ is a set.} \end{cases}$$

The set of all direct specializations of an elementary sequence e, denoted by $desc(e)$, is defined as follows:

Definition 10. *(Direct specializations of an elementary sequence e)*

Let $e = (e_1, e_2, ..., e_n)$ be an elementary sequence. The direct specializations of an elementary sequence e, denoted by desc(s), is defined by:

$$desc(e) = \Big\{(e_1^{'}, e_2^{'}, ..., e_n^{'}) \mid \exists i \in \{1, ..., n\}\Big((e_i^{'} \in desc(e_i)) \text{ and } (\forall j \neq i)(e_j^{'} = e_j)\Big)\Big\}.$$

Example 7. Given the most general elementary sequence s=$(T_h, T_d, \{\})$, the direct specializations of T_h are uh and gh, the direct specializations of T_d are r and c and the direct specializations of $\{\}$ are $\{mp_1\}$, $\{mp_2\}$, $\{mp_3\}$ and $\{mp_4\}$. Thus, the direct specializations of $(T_h, T_d, \{\})$ are $(uh, T_d, \{\})$, $(gh, T_d, \{\})$, $(T_h, r, \{\})$, $(T_h, ca, \{\})$, $(T_h, T_d, \{mp_1\})$, $(T_h, T_d, \{mp_2\})$, $(T_h, T_d, \{mp_3\})$ and $(T_h, T_d, \{mp_4\})$.

The frequency of an elementary sequence is anti-monotone w.r.t the specificity of elementary sequence, i.e., whenever an elementary sequence e is not frequent, all the specializations of e are also not frequent. For example, $(T_h, T_d, \{mp_4\})$ is not frequent when minsup threshold is equal to 3, then all the specializations of $(T_h, T_d, \{mp_4\})$ such as $(gh, T_d, \{mp_4\})$ or $(gh, r, \{mp_1, mp_4\})$... are also not frequent. We use this anti-monotonicity to prune the enumeration space and efficiently build the poset $(L, \leq_e)$.

Specialization is applied recursively for each new frequent elementary generated sequence. Figure 2 shows an example of generation of a poset of $(L, \leq_e)$ which is detailed below:

- Considering the most general elementary sequence $(T_h, T_d, \{\})$, our approach generates seven new frequent elementary sequences, which are: $(uh, T_d, \{\})$, $(gh, T_d, \{\})$, $(T_h, r, \{\})$, $(T_h, ca, \{\})$, $(T_h, T_d, \{mp_1\})$, $(T_h, T_d, \{mp_2\})$ *and* $(T_h, T_d, \{mp_3\})$.
- Based on $(uh, T_d, \{\})$, *MMISP* generates $(uh_p, T_d, \{\})$, $(uh, ca, \{\})$, $(uh, T_d, \{mp_1\})$ and $(uh, T_d, \{mp_2\})$.
- Based on $(uh_p, T_d, \{\})$, *MMISP* generates $(uh_p, ca, \{\})$, $(uh_p, T_d, \{mp_1\})$ and $(uh_p, T_d, \{mp_2\})$.
- Based on $(uh_p, ca, \{\})$, *MMISP* generates $(uh_p, ca, \{mp_1\})$ and $(uh_p, ca, \{mp_2\})$.
- Based on $(uh_p, ca, \{mp_1\})$, *MMISP* generates $(uh_p, ca, \{mp_1, mp_2\})$.
- By using $(uh_p, ca, \{mp_1, mp_2\})$, no any new frequent elementary sequences can be found, thus the generation stops.

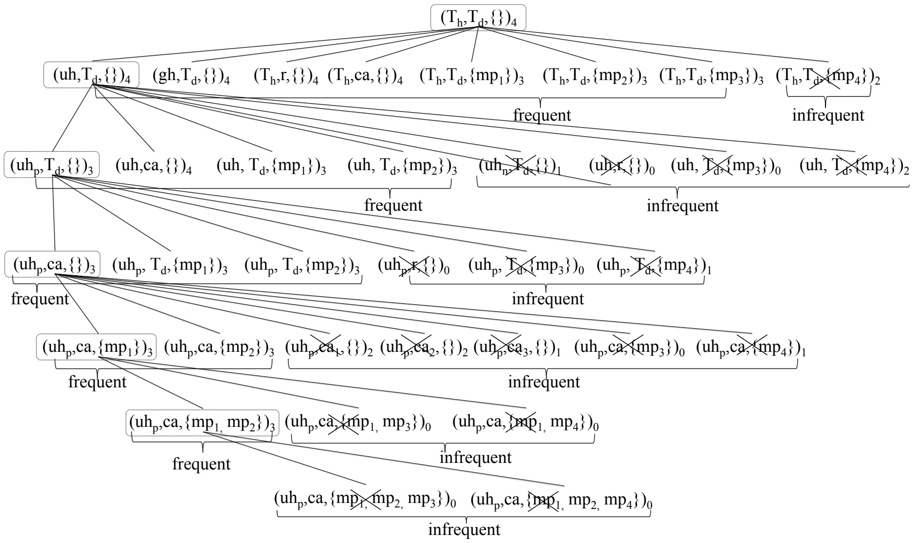


Fig. 2: The steps of generating the elementary sequences in $(L, \leq_e)$ with minsup= 3.

Figure 3 shows the poset $(L, \leq_e)$ generated with $\sigma = 3$.

As the objective of *MMISP* is to extract specific sequential patterns, we retain only the most specific elementary sequences in $(L, \leq_e)$. The most specific frequent elementary sequences is defined as follows:

Definition 11. *(Most Specific Frequent Elementary Sequence (MSFES))*

Given a positive integer σ as minsup threshold and a sequential database S_{DB}, an elementary sequence e is a most specific frequent elementary sequence in S_{DB} or MSFES if and only if $support_e(e) \geq \sigma$ and there does not exist any elementary sequence $e^{'}$ such that $support_e(e^{'}) \geq \sigma$ and $e^{'} \leq_e e$.

Table 2 shows the set of most specific frequent elementary sequences which are extracted from $(L, \leq_e)$. Algorithm 1 and 2 describe the two steps for extracting all the frequent elementary sequences and the most specific ones.

id	*MSFES*
1	$(uh_p, ca, \{mp_1, mp_2\})$
2	$(gh_l, r, \{\})$
3	$(gh, r, \{mp_3\})$

Table 2: The most specific frequent elementary sequences extracted from $(L, \leq_e)$.

Algorithm 1: Mining All The Most Specific Frequent Elementary Sequence

```
input  : Sequential Database $S_{DB}$, the minimum support threshold $\sigma$, ground
         sets $D_1$,...,$D_n$
output: The set MSFES of all most specific frequent elementary sequences,
         The poset L
begin
    /* Step1: Generating the most general elementary sequence
       $(T_1, T_2, ..., T_k, \{\}, \{\}, ...., \{\})$                            */
    for $i \leftarrow 1$ to $n$ do
        $e\ .add(Top(D_i))$;
    $MSFES \leftarrow\ \emptyset$ ;
    $L \leftarrow\ \emptyset$ ;
    $L \leftarrow L \cup e$;
    /* Step2: The recursive generation of the new elementary sequence
       */
    call $get\_rec\_msfes$($e$,$\sigma$);
```

Algorithm 2: Routine *get_rec_msfes*

```
input  : Elementary sequence $e$, the minimum support threshold $\sigma$
begin
    $Cand \leftarrow \{v^{'} \in desc(e) \mid supp(v^{'}) \geqslant \sigma\}$ ;
    if  $Cand = \emptyset$ then
        $MSFES \leftarrow MSFES \cup e$;
    else
        foreach $e \in Cand$ do
            if $e \notin L$ then
                $L \leftarrow L \cup e$;
                call $get\_rec\_msfes$($e$,$\sigma$);
```

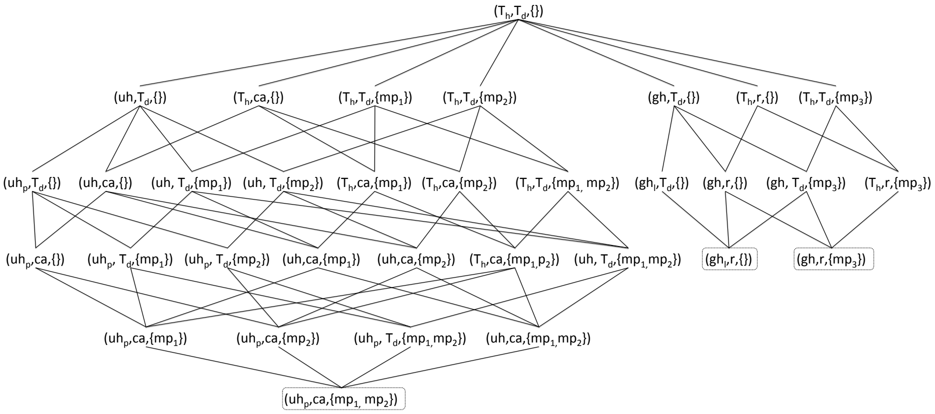


Fig. 3: The poset $(L, \leq_e)$ is generated by taking into account: (i) the sequential database S_{DB} in Table 1 and (ii) the two posets H and DG in Figure 1 and the set $MP = \{mp_1, mp_2, mp_3, mp_4\}$ with minsup threshold equals to 3.

4.2.2 Transformation and mining sequences

We now study the temporal relation between the extracted specific frequent elementary sequences as follow. Firstly, we replace each elementary sequence in each sequence of S_{DB} with all its generalizations from $MSFES$ set. Given a sequence $S = \langle S_1, ..., S_n\rangle$ in S_{DB} the replacement consists in substituting each elementary sequence S_i in S by several elementary sequences $e \in MSFES$ such that $S_i \leqslant_e e$.

Example 8. The sequence S^1 in S_{DB}, $\langle(uh_p, ca_1, \{mp_1, mp_2\}), (uh_p, ca_1, \{mp_1\}), (gh_l, r_1, \{mp_3\})\rangle$ is transformed into $\langle\{(uh_p, ca, \{mp_1, mp_2\})\}, \{(gh_l, r, \{\}), (gh, r, \{mp_3\})\}\rangle$:

- $S_1^1 = (uh_p, ca_2, \{mp_1, mp_2\})$ is replaced by $(uh_p, ca, \{mp_1, mp_2\})$ from *MSFES* set in Table 2, with $(uh_p, ca_2, \{mp_1, mp_2\}) \leqslant_e (uh_p, ca, \{mp_1, mp_2\})$.
- $S_2^1 = (uh_p, ca_1, \{mp_1\})$, none of the elementary sequences in *MSFES* set are more general than S_2^1 .
- $S_3^1 = (gh_l, r_1, \{mp_3\})$ is replaced by $(gh_l, r, \{\})$ and $(gh, r, \{mp_3\})$ from *MSFES* set.

Table 3 shows the transformation of S_{DB} in Table 1 based on the set of all most specific frequent elementary sequences *MSFES* in Table 2.

In a classical sequential pattern mining algorithm, the sequential database to be mined should be a set of pairs (sid, s) where sid is a unique sequence identifier and s is a sequence of itemsets. Thus S_{DB} in Table 3 is transformed as follows:

- Each elementary sequence in the *MSFES* is assigned a unique id which is used during the mining (see Table 2) .

Patients	Trajectories
S^1	$\langle\{(uh_p, ca, \{mp_1, mp_2\})\}, \{(gh_l, r, \{\}), (gh, r, \{mp_3\})\}\rangle$
S^2	$\langle\{(uh_p, ca, \{mp_1, mp_2\})\}, \{(gh_l, r, \{\})\}\rangle$
S^3	$\langle\{(gh_l, r, \{\}), (gh, r, \{mp_3\})\}\rangle$
S^4	$\langle\{(uh_p, ca, \{mp_1, mp_2\})\}, \{(gh, r, \{mp_3\})\}, \{(gh_l, r, \{\})\}\rangle$

Table 3: Transforming the patient trajectories in Table 1 by using the set of all most specific frequent elementary sequences in Table 2.

- For each sequence S^i in the transformed database (see Table 3) and for each elementary sequence T in S^i_j; $S^i_j \in S^i$, T is replaced by its id.

Example 9. The sequence $S^1 = \langle\{(uh_p, ca, \{mp_1, mp_2\})\}, \{(gh_l, r, \{\}), (gh, r, \{mp_3\})\}\rangle$ in Table 3 is transformed into $\langle\{1\}, \{2, 3\}\rangle$ such as:

- $(uh_p, ca, \{mp_1, mp_2\})$ in S^1 has id 1 in Table 2.
- $(gh_l, r, \{\})$ in S^1 has id 2 in Table 2.
- $(gh, r, \{mp_3\})$ in S^1 has id 3 in Table 2.

Table 4 shows the transformation sequential database of Table 3 by using the identifiers of all most specific frequent elementary sequences *MSFES* in Table 2.

Patients	Trajectories
S^1	$\langle\{1\}\{2, 3\}\rangle$
S^2	$\langle\{1\}\{2\}\rangle$
S^3	$\langle\{2, 3\}\rangle$
S^4	$\langle\{1\}\{3\}\{2\}\rangle$

Table 4: Transformed database in Table 3

Then in MMISP, we use CloSpan [12] as the sequential pattern mining algorithm. 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 $\sigma = 3$.

5 Experiments

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 [12] supplied by the IlliMine[5] toolkit.

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

Frequent sequential patterns	Frequent patient trajectory patterns	Support
$\langle\{2\}\rangle$	$\langle(gh_l, r, \{\})\rangle$	4
$\langle\{3\}\rangle$	$\langle(gh, r, \{mp_3\})\rangle$	3
$\langle\{1\}\{2\}\rangle$	$\langle(uh_p, ca, \{mp_1, mp_2\}), (gh_l, r, \{\})\rangle$	3

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

5.1 Healthcare Trajectory

5.1.1 Mining healthcare trajectories

In order to assess the effectiveness of our approach, we run several experiments on the PMSI system for describing and analyzing patient trajectories. In PMSI, each hospitalization is characterized by the following dimensions: hospital, principal diagnosis and procedures delivered during the stay.

The hospital dimension is associated with a geographical poset of 4 levels: root (France), administrative region, administrative departement and hospital. As illustrated in Figure 4, University Hospital of Nancy (coded as 540002078) is a hospital in Meurthe-et-Moselle and Meurthe-et-Moselle is a department in Lorraine which is a region of France. The number of nodes in this taxonomy is 151 nodes.

Principal Diagnosis could be described at 5 levels of the 10^{th} International classification of Diseases (ICD10): root, chapter, block, 3 characters and 4 characters. As illustrated in Figure 5, chapters such as $Neoplasms$ have specializations: block $C30-C39$ which 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 (coded as $C34$), malignant neoplasm of bronchus and lung (coded as $C32$), 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 1543 nodes.

Procedures were represented by their first CCAM[6] code. For example, $ZBQK$ is a chest radiography, $GFFA$ is a pneumonectomy etc.

Our dataset contains 828 patients suffering from lung cancer and living in Lorraine Region, in the East of France. Table 6 shows an example of care trajectories for 3 patients. For example, $Patient_1$ has two hospitalizations. He was admitted in the University Hospital of Nancy (coded as 540002078), for a Lung cancer (coded as $C341$), and underwent a chest radiography (coded as $ZBQK$). Then, he was hospitalized in a private clinic in Metz (coded as 570023630), for a chemotherapy session (coded as $Z51$) where he had a chest radiography and pneumonectomy (coded as $GFFA$).

[6] “ Classification Commune des Actes Médicaux ”: the French classification of medical and surgical procedures.

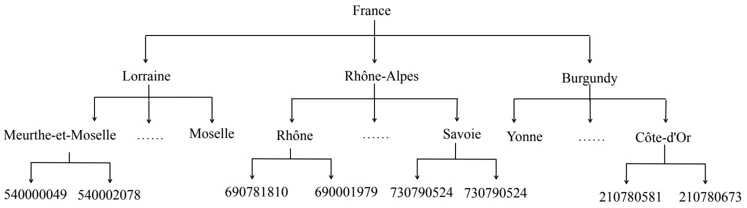


Fig. 4: A geographical poset of the healthcare institution

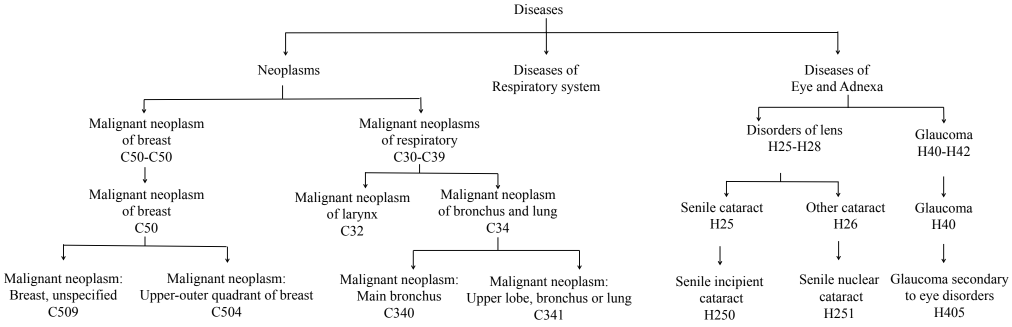


Fig. 5: A disease poset

As a characterization of the care trajectory database, Figure 6 shows the distribution of the length of healthcare trajectories in our dataset, the median length is of 11 stays.

Patients	Trajectories
$Patient^1$	$\langle (C341, 540002078, \{ZBQK\}), (Z51, 570023630, \{ZBQK, GFFA\}), \ldots \rangle$
$Patient^2$	$\langle (C770, 100000017, \{ZBQK\}), (C770, 210780581, \{ZZQK, YYYY\}), \ldots \rangle$
$Patient^3$	$\langle (H259, 210780110, \{YYYY\}), (H259, 210780110, \{ZZQK\}), \ldots \rangle$

Table 6: Care trajectories of 3 patients

The support value is set to 40 patients (i.e. $\sigma = 5\%$) for this experiment . *MMISP* generates *615* different frequent trajectories. Table 7 shows some extracted patterns. *Pattern #1* can be interpreted as follows: 5% of patients had a hospitalization in Meurthe et Moselle department for any kind of lung cancer (coded as C34). They underwent three medical procedures: chest radiography (coded as ZBQK) with an electrocardiography (coded as DEQP) and a therapeutic procedure on blood (coded as FELF). *Pattern #2* shows that 8% of patients had a hospitalization in Lorraine, because of poisoning.

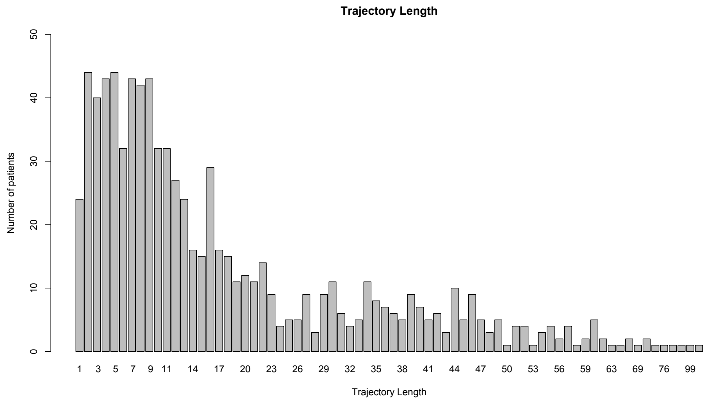


Fig. 6: Distribution length of the patient trajectories

This kind of information helps healthcare managers and decision makers in planning and organizing healthcare resources at regional level. Besides, sequential patterns can be seen as condensed representations of the care trajectories.

Pattern	Trajectories	support
$Pattern^1$	$\langle(Meurthe\ et\ Moselle, C34, \{ZBQK, FELF, DEQP\})\rangle$	5%
$Pattern^2$	$\langle(Lorraine, Poisoning)\rangle$	8%
$Pattern^3$	$\langle(540003019, Z510, \{ZZNL\}), (540003019, Z510, \{ZZNL\})\rangle$	14%

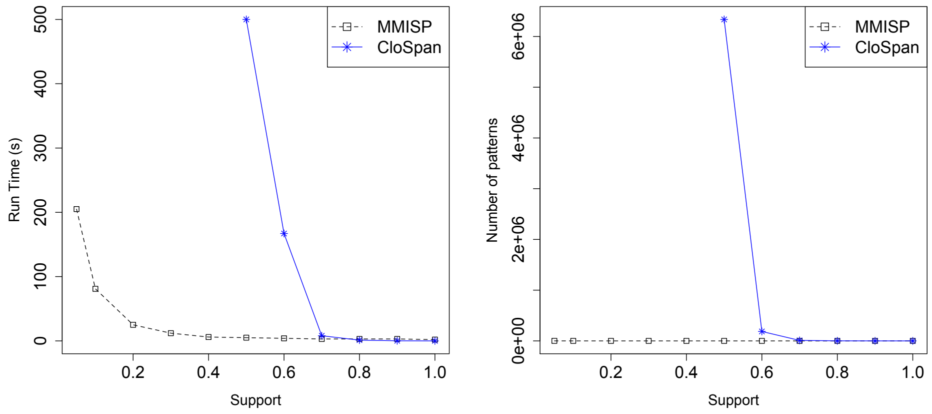
Table 7: 3-Care trajectories extracted by *MMSIP*

5.1.2 *MMISP* versus Standard sequential pattern mining method

In this section, we compare *MMISP* with a standard sequential pattern mining method such as *CloSpan*. All standard sequential pattern mining algorithms requires the mined 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. So, we replace each sequence S^i in S_{DB} with an *extended-sequence* $S^{'i}$. Each elementary sequence of a sequence S^i is transformed into a single itemset by replacing each element in the elementary sequence with all its ancestors. For example, with the posets shown in Figure 1, an elementary sequence $(uh_p, ca_1, \{mp_1, mp_2\})$ would be replaced with $\{T_h,\ uh,\ uh_p,\ T_d,\ ca,\ ca_1,\ mp_1,\ mp_2\}$. Table 8 shows the *extended-sequential* database $S^{'}_{DB}$ of the sequential database S_{DB} in Table 1. Then, we apply *CloSpan* as a sequential pattern mining algorithm on the *extended-sequential* database. This way of managing taxonomies has been used in *GSP* which is proposed by Srikant and Agrawal [11].

Patients	Extended sequence
S^1	$\langle\{T_h, uh, uh_p, T_d, ca, ca_1, mp_1, mp_2\}, \{T_h, uh, uh_p, T_d, ca, ca_1, mp_1\}, \{T_h, gh, gh_l, T_d, r, r_1, mp_3\}\rangle$
S^2	$\langle\{T_h, uh, uh_n, T_d, ca, ca_1, mp_4\}, \{T_h, uh, uh_p, T_d, ca, ca_2, mp_1, mp_2\}, \{T_d, gh, gh_l, T_d, r, r_1, mp_2\}\rangle$
S^3	$\langle\{T_h, uh, uh_n, T_d, ca, ca_3, mp_4\}, \{T_h, gh, gh_l, T_d, r, r_2, mp_3\}\rangle$
S^4	$\langle\{T_h, uh, uh_p, T_d, ca, ca_2, mp_1, mp_2\}, \{T_h, gh, gh_p, T_d, r, r_2, mp_3\}, \{T_h, gh, gh_l, T_d, r, r_2, mp_2\}\rangle$

Table 8: An extended sequential database of patient trajectories in Table 1.



(a) Runtime sequences over frequency threshold

(b) Number patterns over frequency threshold

Fig. 7: *MMISP* versus *CloSpan*

Our main goal is to evaluate the quality of the patterns mined with *MMISP* and its performance compared to the “naive” approach using *CloSpan*. So, we transform the 828 patients trajectories, then we apply the two approaches using several minimum support threshold ranging from 100 % to 5%. Figure 7 reports the running time and the number of pattens 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 itemsets and generates a large number of similar sequences. By contrast, *MMISP* runs in acceptable time for support as low as 5 %.

MMISP is able to extract condensed patterns w.r.t. the ones mined by *CloSpan*. For example, the sequential pattern $\langle\{T_h, uh, uh_p, T_d, ca, mp_1, mp_2\}$ $\{T_h, gh, gh_l, T_d, r\}\rangle$ generated by *CloSpan* contains redundant information as a hospitalization containing uh_p will also contain T_h and uh. *MMISP* is not concerned by this pattern because it extracts just the most specific frequent elementary sequence in the first step of the algorithm.

CloSpan extracts all the sequential patterns while *MMISP* generates just the more specific ones. For example, if $\langle(uh_p, ca, \{mp_1, mp_2\})\rangle$ is a sequential pattern. *MMISP* does not extract the patterns which are more general such as $\langle(uh, T_d, \{mp_1\})\rangle$ while *CloSpan* extracts both the general and specific ones. Figure 7 shows the differences between the number of sequential patterns extracted by *CloSpan* and *MMISP*. For example, with a support threshold of 50 %, *MMISP* extracts *150 sequential patterns* while *CloSpan* extracts *6335683 sequential patterns*.

Finally, we may conclude that:

- *MMISP* is more efficient than *CloSpan* over *extended-sequential* database with low support threshold.
- The sequential patterns extracted by *CloSpan* require post processing while this is not the case with *MMISP*.
- *MMISP* extracts just the most specific sequential patterns while *CloSpan* extracts both general and specific ones. This means that *CloSpan* extracts a huge number of sequential patterns. Analyzing all these sequential patterns is not an easy task for healthcare managers and decision makers.

5.1.3 $MMISP$ versus M^3SP

Another experiment was 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 40 patients (i.e. $\sigma = 5\%$). Table 9 reports an example of the extracted patterns with M^3SP and *MMISP*.

Firstly, we observe that *MMISP* is able to extract condensed patterns w.r.t. the ones mined by M^3SP. For example, *48 sequential patterns*, *Pattern #1*,..., *Pattern #48*, generated by M^3SP are summarized by *3 sequential patterns*, *Pattern #50*, *Pattern #51* and *Pattern #52*, extracted by *MMISP* (see Table 9). 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 in 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 (570000588 and C341) and different values of medical procedures. By contrast, *Pattern #50* extracted by *MMISP* represents the same trajectory as *Pattern #48*. *Pattern #50* has one elementary sequence with three elements: hospital 570000588, 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 elementary sequences not extracted by M^3SP. For instance e_1*=(Lorraine, Diseases,* $\{$*GEQE, ACQH, ZCQH*$\}$*)* and e_2*=(Lorraine,Diseases of the respiratory,* $\{$*ACQH*$\}$*)* are extracted by *MMISP*. As e_1 and e_2 are frequent and not comparable (i.e. $e_1 \nleq_e e_2$ and $e_2 \nleq_e e_1$), M^3SP extracts only *(Lorraine, Diseases of the respiratory, ACQH)* and not *(Lorraine, Diseases, ACQH)* as *(Lorraine, Diseases of the respiratory, ACQH)* is more specific than *(Lorraine, Diseases, ACQH)*.

From a quantitative point of view, *MMISP* extracts 803 *frequent elementary sequences* with 615 *sequential patterns* while M^3SP extracts 331 *multidimensional items* with 470 *multidimensional sequential patterns*. The execution time of M^3SP is about 82 *seconds* while *MMISP* takes about 98 *seconds*.

Finally, we may conclude that:

- Several sequential patterns generated by M^3SP can be summarized by only one sequential pattern mined by *MMISP*.
- Several multidimensional items generated by M^3SP can be summarized by only one elementary sequence in *MMISP*.
- One elementary sequence 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 sequential patterns can be extracted by *MMISP* while they cannot be extracted by M^3SP.

Methods	id	Trajectory Patterns
M^3SP	1	$\langle\{(570000588, C341, GFFA)(570000588, C341, ZZQK)\}\rangle$
	2	$\langle\{(570000588, C341, DEQP)(570000588, C341, GFFA)(570000588, C341, ZZQK)\}\rangle$
		$\vdots$
	48	$\langle\{(570000588, C341, ZBQK)(570000588, C341, DEQP)(570000588, C341, GFFA)$ $(570000588, C341, GLLD)(570000588, C341, GELD)(570000588, C341, ZZQK)$ $(570000588, C341, GELE)(570000588, C341, FCFA)(570000588, C341, , AGLB)\}\rangle$
	49	$\langle(Lorraine, Diseases\ of\ the\ respiratory, ACQH)\rangle$
MMISP	50	$\langle\{(570000588, C341, \{ZBQK, DEQP, GFFA, GLLD, GELD, ZZQK, GELE, FCFA, AGLB\}\rangle$
	51	$\langle(570000588, C341, \{DEQP, GELD, GELE, ZZQK, AGLB, GLLD, GFFA\})\rangle$
	52	$\langle(570000588, C341, \{ZBQK, DEQP, GELD, GELE, ZZQK, GLLD, GFFA\})\rangle$
	53	$\langle(Lorraine, Diseases, \{GEQE, ACQH, ZCQH\})\rangle$

Table 9: Some patterns obtained by M^3SP and MMISP.

5.1.4 $MMISP^+$

In our approach, the fundamental step is the first one which is *"extraction of frequent elementary sequences"*, because it provides all elements that will occur in sequences to be mined. In this step, *MMISP* extracts only the most specific frequent elementary sequences from the poset (L, $\leq_e$). As a result, it is impossible to find a sequential pattern that contains an elementary sequence which is comparable with another elementary sequence in the same pattern or in the another one. Formally, the sequential patterns extracted by *MMISP* have the following property: for any two sequential patterns $SP^1 = \langle SP^1_1, SP^1_2, ..., SP^1_{k^1}\rangle$ and $SP^2 = \langle SP^2_1, SP^2_2, ..., SP^2_{k^2}\rangle$, we have :

$$\nexists(i,j);\ SP^1_i <_e SP^2_j$$

For example, in Table 1 the pattern $\langle(uh, ca, \{\}), (gh_l, r, \{\})\rangle$ is frequent according to minsup threshold equals 3. This pattern does not appear in the results of *MMISP* because in the first step *MMISP* extracts $(uh_p, ca, \{mp_1, mp_2\})$ which is more specific than $(uh, ca, \{\})$ and still frequent. As a result, *MMISP* does not extract any pattern which includes elementary sequence more general than $(uh_p, ca, \{mp_1, mp_2\})$.

To solve this problem, we propose an extension to our approach which is called $MMISP^+$. In $MMISP^+$, instead of choosing just the most specific elementary sequence in $(L, \leq_e)$, we choose all the elementary sequences in $(L, \leq_e)$. Thus, we replace each elementary sequence in each sequence of S_{DB} with all of its generalizations from $(L, \leq_e)$ and not just with the most specific ones. For example, the elementary sequence $(gh_p, r_2, \{mp_3\})$ in the sequence S^4 in Table 1 is replaced by $(gh, r, \{mp_3\})$, $(gh, r, \{\})$, $(gh, T_d, \{mp_3\})$, $(T_h, r, \{mp_3\})$,$(gh, T_d, \{\})$, $(T_h, r, \{\})$, $(T_h, T_d, \{mp_3\})$ and $(T_h, T_d, \{\})$ from the $(L, \leq_e)$ set in Figure 3. Then, we use *CloSpan* as a sequential pattern mining algorithm on the transformed sequential database.

Frequent patient trajectory patterns	Support
$\langle(uh, ca, \{\}), (gh_l, r, \{\})\rangle$	4
$\langle(uh, ca, \{\}), (T_h, T_d, \{\}), (gh_l, r, \{\})\rangle$	3
$\langle(uh, ca, \{\}), (gh, r, \{mp_3\})\rangle$	3
$\langle(uh_p, ca, \{mp_1, mp_2\}), (gh_l, r, \{\})\rangle$	3

Table 10: The sequential patterns extracted from S_{DB} with support threshold equals 3.

Applying $MMISP^+$ generates non condensed patterns w.r.t *MMISP*. For example, the pattern $\langle\{(T_h, T_d, \{\}),\ (uh, T_d, \{\}),\ (T_h, ca, \{\}),\ (uh, ca, \{\})\}\{(T_h, T_d, \{\}),\ (gh, T_d, \{\}),\ (T_h, r, \{\}),\ (gh_l, T_d, \{\}),\ (gh, r, \{\}),\ (gh_l, r, \{\})\}\rangle$ generated by $MMISP^+$ contains redundant information. as a patient having the hospitalization $(uh, ca, \{\})$ will also have $(T_h, T_d, \{\})$, $(uh, T_d, \{\})$ and $(T_h, ca, \{\})$. Redundancy can be removed by post-processing. Table 10 shows all the sequential patterns extracted from S_{DB} in Table 1 by applying $MMISP^+$. These patterns have been post-processed to remove redundant information.

Figure 8 reports the number of patterns extracted from our dataset (i.e, 828 patient trajectories) according to different values of *minsup* threshold for applying the two solutions: *MMISP* and $MMISP^+$. Actually, $MMISP^+$ cannot finish its calculations for support threshold less than 50 %. This happens because the *CloSpan* algorithm (i.e, the third step in $MMISP^+$) cannot process with support threshold less than 50 %.

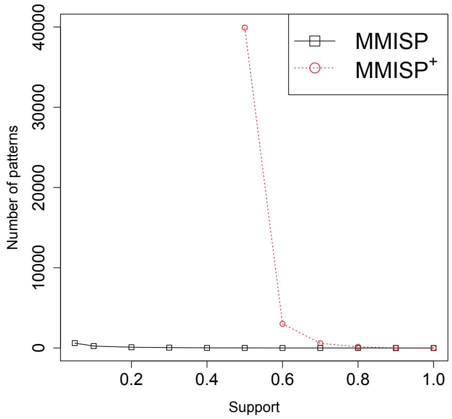


Fig. 8: Number of sequential pattern extracted according to different values of support threshold for both *MMISP* and $MMISP^+$

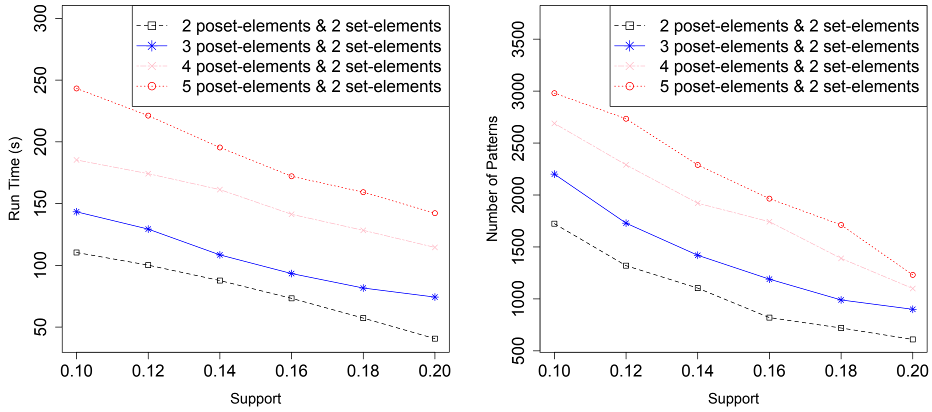
5.2 Experiments on Synthetic Datasets

In the second 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 dimensions with an associated subsumption relation.
- number of dimensions without any associated subsumption relation.
- number of elementary sequences in each sequence (i.e. sequence length).
- depth of the poset of elements with an associated subsumption relation.
- number of sequences in a sequential database.

In the following, we use the term *"set-element"* for an element lying in a set and the term *"poset-element"* for an element lying in a partially ordered set.

The first batch of synthetic data generated contains 1000 sequences defined over 2, 3, 4 and 5 *poset-elements* and 2 *set-elements*. Each sequence contains 15 elementary sequences. Each poset is defined over 3 levels of granularity between its elements. Figure 9 reports the results according to different values of support threshold for different numbers of *poset-elements* in the elementary sequence. The running time increases for each newly added dimension.



(a) Runtime sequences over frequency threshold.

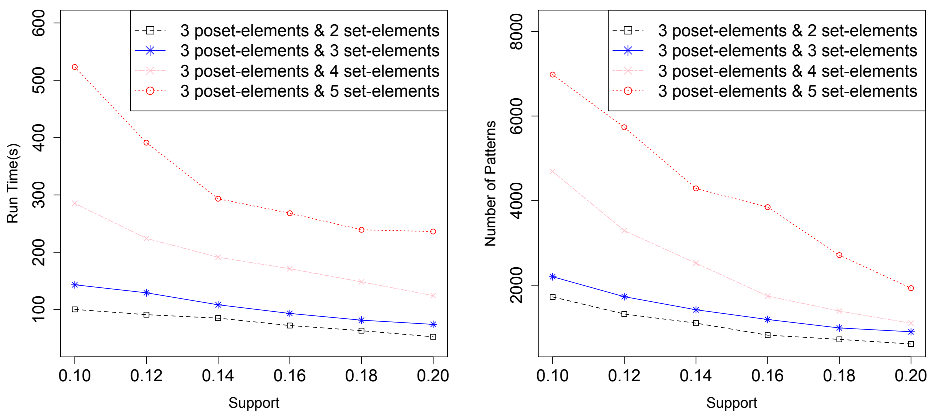
(b) Number patterns over frequency threshold.

Fig. 9: Number of sequential pattern extracted and Running time obtained by MMISP with varying in the number of poset-elements.

The second batch of generated synthetic data contains 1000 sequences with varying number of *set-elements* (2, 3, 4 and 5 elements). The sequences have three *poset-elements* with 3-level of granularity. Figure 10 reports the results according to different values of support threshold for different number of *set-elements* in the elementary sequence.

In Figure 11, we study the performance of *MMISP* by considering several levels of granularity within the posets. We generated 1000 sequences defined over 3 *poset-elements* and 3 *set-elements*. Each poset is defined over 3, 4, 5, 6 levels of granularity. Each sequence contains 15 elementary sequences. The number of extracted sequential patterns does not change with each newly added level as *MMISP* extracts only the most specific sequential patterns.

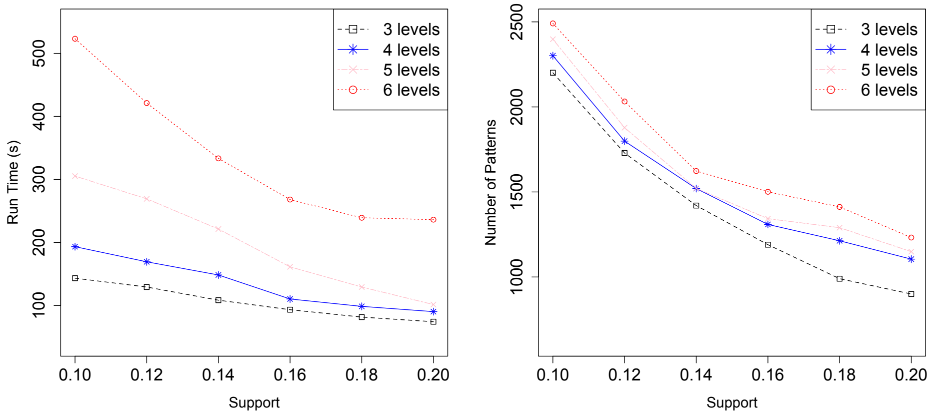
We study the performance of *MMISP* and the number of extracted sequential patterns with respect the number of sequences in a sequential database and the length of each sequence. Figure 12 shows the running time and the number of



(a) Runtime sequences over frequency threshold.

(b) Number patterns over frequency threshold.

Fig. 10: Number of sequential pattern extracted and Running time for a large panel of sequences when varying over the number of set-elements.

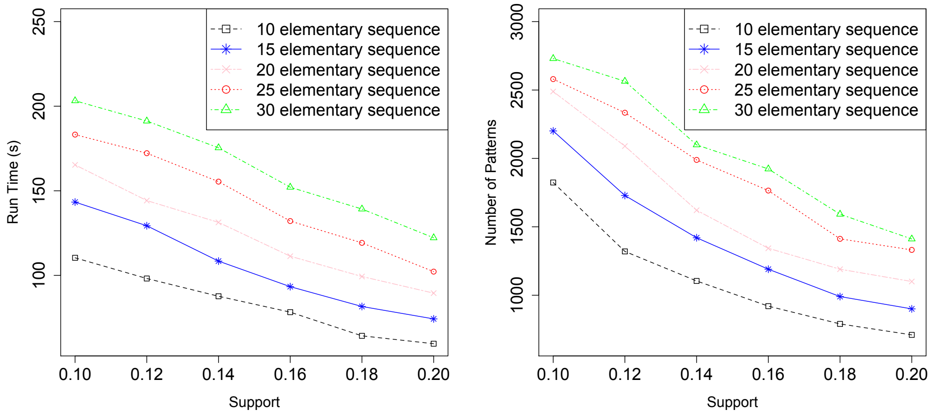


(a) Runtime sequences over frequency threshold.

(b) Number patterns over frequency threshold.

Fig. 11: Number of sequential pattern extracted and Running Time obtained for a large panel of sequences when varying over the levels of granularity between poset-elements.

patterns extracted for 1000 sequences with 3 *poset-elements* and 3 *set-elements* with varying sequence length. Figure 13 shows the running time and the number of sequential patterns extracted for several number of sequences (1000, 2000, 3000, 4000 and 5000 sequences) also with 3 *poset-elements* and 3 *set-elements*. In this experiments, the sequence length is roughly equal to 15.



(a) Runtime sequences over frequency threshold.

(b) Number patterns over frequency threshold.

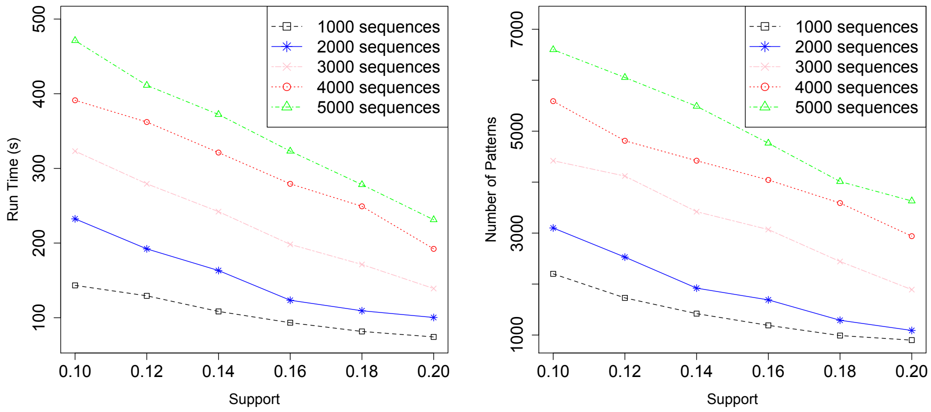
Fig. 12: Number of sequential pattern extracted and Running time for a large panel of sequences when varying over sequence length.

Figures 9 - 13 highlight the fact that *MMISP* is efficient in terms of runtime for a large panel of sequences when varying different parameters.

6 Conclusion

In this paper, we propose a new approach to mine a sequential database of heterogeneous sequences. We provide formal definitions and propose a new algorithm *MMISP* to mine this kind of sequences. The *MMISP* algorithm relies on external posets to improve the mining process and produces results with appropriate levels of granularity. 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



(a) Runtime sequences over frequency threshold.

(b) Number patterns over frequency threshold.

Fig. 13: Number of sequential pattern extracted and Running time for a large panel of sequences when varying over several number of sequences.

the other hand, proposing a graphical interface to visualize and query the sequential patterns. We are also interested in generalizing our method by considering sequences all elements of which are lying in partially ordered set.

Finally, we are aware that choosing the most specific frequent elementary sequence to mine the sequential patterns prevents us from extracting all of the most specific sequential patterns. Coping with this issue is another interesting extension of the present work that we plan to investigate in the future.

7 Acknowledgments

The authors would like to thank the French institution “Institut National du Cancer (INCa)” for its financial support.

References

1. Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE ’95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.
2. Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *KDD*, pages 429–435, 2002.
3. Ding-Ying Chiu, Yi-Hung Wu, and Arbee L. P. Chen. An efficient algorithm for mining frequent sequences by a new strategy without support counting. In *ICDE*, pages 375–386, 2004.

4. RB Fetter, Y Shin, JL Freeman, RF Averill, and JD Thompson. Case mix definition by diagnosis-related groups. *Med Care*, 18(2):1–53, Feb 1980.
5. Florent Masseglia, Fabienne Cathala, and Pascal Poncelet. The PSP approach for mining sequential patterns. In *PKDD*, pages 176–184, 1998.
6. Carl H. Mooney and John F. Roddick. Sequential pattern mining – approaches and algorithms. *ACM Comput. Surv.*, 45(2):19:1–19:39, March 2013.
7. Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. Prefixspan: Mining sequential patterns by prefix-projected growth. In *ICDE*, pages 215–224, 2001.
8. Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal. Multi-dimensional sequential pattern mining. In *CIKM*, pages 81–88, 2001.
9. Marc Plantevit, Anne Laurent, Dominique Laurent, Maguelonne Teisseire, and Yeow Wei Choong. Mining multidimensional and multilevel sequential patterns. *TKDD*, 4(1):1–37, 2010.
10. Eliana Salvemini, Fabio Fumarola, Donato Malerba, and Jiawei Han. Fast sequence mining based on sparse id-lists. In *Proceedings of the 19th international conference on Foundations of intelligent systems*, ISMIS'11, pages 316–325, Berlin, Heidelberg, 2011. Springer-Verlag.
11. Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag.
12. Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan: Mining closed sequential patterns in large datasets. In *In SDM*, pages 166–177, 2003.
13. Zhenglu Yang, Masaru Kitsuregawa, and Yitong Wang. Paid: Mining sequential patterns by passed item deduction in large databases. In *IDEAS*, pages 113–120, 2006.
14. Chung-Ching Yu and Yen-Liang Chen. Mining sequential patterns from multidimensional sequence data. *IEEE Trans. Knowl. Data Eng.*, 17(1):136–140, 2005.
15. Cong Yu and H. V. Jagadish. Querying complex structured databases. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 1010–1021. VLDB Endowment, 2007.
16. Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Mach. Learn.*, 42(1-2):31–60, January 2001.
17. Changhai Zhang, Kongfa Hu, Zhuxi Chen, Ling Chen, and Yisheng Dong. Approxmgmsp: A scalable method of mining approximate multidimensional sequential patterns on distributed system. In *FSKD (2)*, pages 730–734, 2007.