Mining Heterogeneous Multidimensional Sequential Patterns

Elias Egho, Chedy Raïssi, Nicolas Jay, Amedeo Napoli

ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-279

Mining Heterogeneous Multidimensional Sequential Patterns

Elias Egho and **Chedy Raïssi** and **Nicolas Jay** and **Amedeo Napoli**[1]

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.

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. This method is rather efficient to discover rules of the type: *"customers frequently 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 [12], and music sequences matching [8].

Many efficient approaches were developed to mine ordered patterns and most of these methods are based on the *Apriori* property [1]. This property states that any super pattern of a non-frequent pattern can not be frequent. The main algorithms are *GSP* [9], *SPADE* [14], *PrefixSpan* [4] and *ClosSpan* [11]. 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 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. [5], Zhang et al. [16] and Yu et al. [13] 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. [6] 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 [10], 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 extracts patterns 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 article 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 adapting a suitable method. 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 [7]. To overcome this problem, *MMISP* mines only the most specific multidimensional sequential patterns. We report qualitative experiments with a dataset consisting of 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.

The remainder of this paper 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 Sec-

[1] LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine) BP 239, Vandoeuvre-lès-Nancy, F-54506, France, email: firstname.lastname@loria.fr

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.

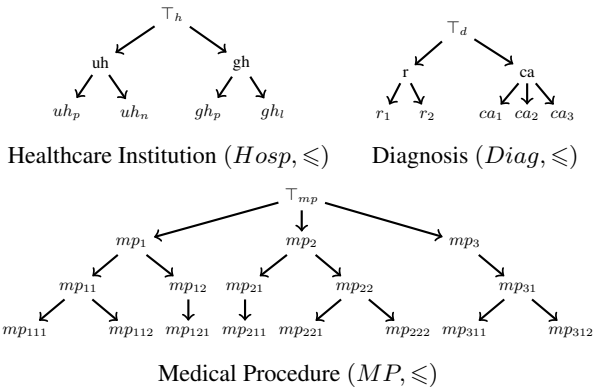

Figure 1: Taxonomies for the healthcare institution, the medical procedure and the diagnosis

tion 5 concludes the paper.

2 Problem Statement

2.1 An introductory example

This section illustrates an example to ease the understanding of our 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: (1) the hospital where a patient is admitted, (2) a reason for hospitalization and (3) set of medical procedures that a patient undergoes. 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 appear frequently in patients trajectories, by taking advantage of different levels of granularity for each element as background knowledge. An example of such patterns are very helpful to improve hospitalization planning, optimize clinical processes or detect anomalies.

2.2 Basic definitions

We assume that the domain knowledge is represented in form of taxonomies. A multidimensional sequence is an ordered set of vectors whose components are items or itemsets. Each item in the component is a node in a taxonomy. For the sake of simplicity we call a multidimensional sequence as *"md-sequence"*. More formally md-sequence is defined as follows:

Definition 2.1 *(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* $s_1 <_t s_2 <_t s_3 <_t ... <_t s_n$, *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 $e \leq e'$, iff $e_i \leq e'_i$ for all $i \leq k$. Then, 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 $s \leq s'$, if there exist a set of indices $1 \leq i_1 < i_2 < ... < i_{n_2} \leq n_1$ such that $s_j \leq s'_{i_j}$ for all $j \in \{1 \dots n_2\}$ and $n_2 \leqslant n_1$. s' is said to be ***more general*** than s.

Consider the three taxonomies $(Hosp, \leq)$, $(Diag, \leq)$ and $(MP, \leq)$ 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 md-sequence $s' = \langle(uh, ca_1, \{mp_{11}, mp_{12}\}), (gh_l, r, \{mp_{13}\})\rangle$ is more general than s, as $s_1 \leq s'_1$, $s_2 \leq s'_2$ and s_1 and s'_1 come before s_2 and s'_2 over time.

The set $\mathcal{MS}_{\mathcal{DB}} = \{s_1, s_2, ..., s_m\}$ of m md-sequences, is called a *mds-database*. The support of an elementary vector $e = (e_1, e_2, ..., e_k)$ in $\mathcal{MS}_{\mathcal{DB}}$, denoted by $supp(e, \mathcal{MS}_{\mathcal{DB}})$, is defined as follows: $supp(e, \mathcal{MS}_{\mathcal{DB}}) = |\{s_i \in \mathcal{MS}_{\mathcal{DB}};\ \exists j \leq |s_i|;\ s_{ij} \leq e\}|$. The support of a *md-sequence* s in $\mathcal{MS}_{\mathcal{DB}}$ is defined as follows:

Definition 2.2 *(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:* $supp(s, \mathcal{MS}_{\mathcal{DB}}) = |\{s_i \in \mathcal{MS}_{\mathcal{DB}}; s_i \leq s\}|$.

Given a positive integer σ as a minimal support threshold and a mds-database $\mathcal{MS}_{\mathcal{DB}}$, the elementary vector e is called frequent, iff $supp(e, \mathcal{MS}_{\mathcal{DB}}) \geq \sigma$. 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"*. Given a mds-database $\mathcal{MS}_{\mathcal{DB}}$ and a minimal support threshold, the problem of mining md-sequences is to enumerate all possible mds-patterns. For example, the md-sequence $s = \langle(uh, ca, \{mp_{11}, mp_2\}), (\top_h, \top_d, \{mp_{222}\})\rangle$ has a support equals to 3 (i.e., $supp(s, \mathcal{MS}_{\mathcal{DB}}) = 3$) in the database $\mathcal{MS}_{\mathcal{DB}}$ (see Table 1). It is a mds-pattern w.r.t the minimal support threshold $\sigma = 3$.

2.3 Mining the most specific mds-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 2.3 *(Most specific mds-pattern) Given a positive integer σ as minimal support threshold and a mds-database $\mathcal{MS}_{\mathcal{DB}}$. The md-sequence s is a most specific mds-pattern in $\mathcal{MS}_{\mathcal{DB}}$ if and only if $supp(s, \mathcal{MS}_{\mathcal{DB}}) \geq \sigma$ and there does not exist any md-sequence s' such that $supp(s', \mathcal{MS}_{\mathcal{DB}}) \geq \sigma$ and $s' \leq s$.*

In this precise setting, the frequency for md-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 mds-patterns constitute a basic for retrieving all mds-patterns. Let $\sigma = 3$ be a minimal support threshold, the md-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 md-sequence $s' = \langle(uh, ca, \{mp_{11}, mp_2\}), (gh, r, \{mp_{22}, mp_{31}\})\rangle$ is frequent and verifies that $s' \leq s$. The md-sequence s' is a most specific mds-pattern as it is frequent and there is no other md-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 mds-patterns to significantly decrease the complexity of the problem and save computational time.

3 MMISP algorithm

In this section, we present our approach for extracting the most specific mds-patterns from a mds-database $\mathcal{MS}_{\mathcal{DB}}$. The basic idea of *MMISP (Mining Multidimensional Itemsets Sequential Patterns)* 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.

Step 1: Extracting all frequent elementary vectors: The basic step in *MMISP* is extracting all frequent elementary vectors from $\mathcal{MS}_{\mathcal{DB}}$. 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., $\bot = (\bot_1, ..., \bot_k)$), is more specific than any other elementary vector. If element of the vector is an item, then we consider a node $\bot_i$ 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 $(\bot_h, \bot_d, \{mp_{111}, mp_{112}, mp_{121}, mp_{211}, mp_{221}, mp_{222}, mp_{311}, mp_{312}\})$. The set of all the elementary vectors E with the bottom vector $\bot$ is a *lattice* $(E \cup \{\bot\}, \leq)$. 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 $\{\forall(i, j);\ c_i \sqcup c'_j\}$; $i \leq n$ and $j \leq m$. The meet ($\sqcap$) of 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 $\bot_i$. 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 $c \cup c'$. Finally, we can say that $(E \cup \{\bot\}, \leq)$ is a *lattice*, while the frequent elementary vectors considers as a *join-semilattice* $(FE, \leq)$.

The main challenge now is how to efficiently build the join-semilattice $(FE, \leq)$. 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 $(FE, \leq)$ 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 $(FE, \leq)$ 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). For example, 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. 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 $(Diag, \leq)$ 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 elements 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 $i \leq m$ and c'_{m+1} is one of the right siblings of c_m and all its ancestor nodes. For example, given the taxonomy $(MP, \leq)$ 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 $\{mp_1, mp_{21}, mp_3\}$.

This way we can generate all the frequent elementary vectors in a nonredundant manner. Given the taxonomies in Figure 1, the immediate successors of $e = (\top_h, ca, \top_{mp})$ are generated by replacing 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 $(\top_h, ca_3, \top_{mp})$ and also by replacing $\top_{mp}$ 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 $(\top_h, ca, \{mp_3\})$.

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 $(FE, \leq)$. Figure 2 shows an example of generation of a part of $(FE, \leq)$ 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 both vectors $(uh, ca, \{mp_{11}\})$, $(uh, ca, \{mp_1, mp_2\})$ and the vector $(uh, ca, \{mp_{11}, mp_2\})$ 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 mds-patterns, we retain only the most specific elementary vectors *MSFEV* in $(FE, \leq)$. The most specific frequent elementary vectors constitute collection of elementary vectors in $\mathcal{MS}_{\mathcal{DB}}$ which are frequent and most specific. Table 2 shows the set of most specific frequent elementary vectors which are extracted from $(FE, \leq)$.

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 $(FE, \leq)$.

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 $s_i \leqslant e$. For example, the sequence s_4 in $\mathcal{MS}_{\mathcal{DB}}$ is transformed into $\langle\{(uh, ca, \{mp_{11}, mp_2\}), (\top_h, \top_d, \{mp_{222}\})\}, \{(gh, r, \{mp_{22}, mp_{31}\})\}\rangle$ 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 $\widehat{\mathcal{MS}}_{\mathcal{DB}}$.

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.

Step 3: Mining of mds-patterns: 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: (i) each elementary vector in *MSFEV* is assigned a unique id which is used during the mining (see Table 2) and (ii) 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. For example, 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.

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

We use CloSpan [11] 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 $\sigma = 3$.

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 $\sigma = 3$.

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

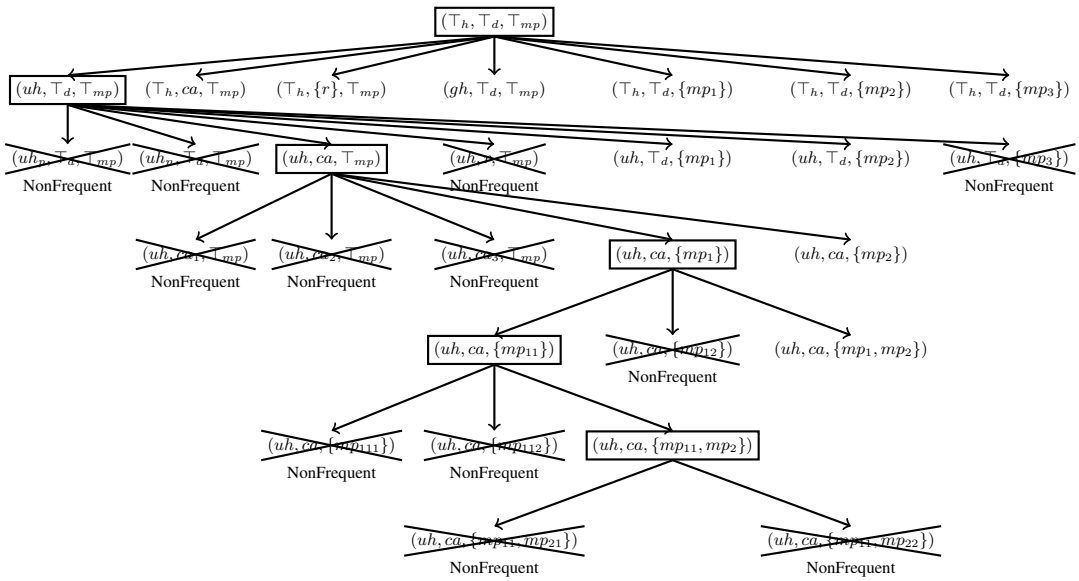

Figure 2: The steps of generating the elementary vectors in $(FE, \leq)$ with $\sigma = 3$.

4 Experiments

In this section we conduct experiments on a real dataset consisting of trajectories of cancer patients extracted from French healthcare organizations. We compare *MMISP* with two existing methods, CloSpan and M^3SP. Several other experiments have been conducted on synthetic datasets to study the scalability of *MMSIP*. The experiments on synthetic datasets and the comparison are not discussed in this paper due to lack of space. All the experiments are discussed in the associated technical repor [3].

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. The extraction of sequential patterns is based on the public C++ implementation of CloSpan algorithm [11] supplied within the IlliMine[2] toolkit. A dedicated web page to visualize data sets and interact with the experimental results is available at `http://www.loria.fr/~eegho/mmisp/`.

4.1 Mining healthcare trajectories

In order to assess the effectiveness of our approach, we run several experiments on PMSI [3], 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 100 patients suffering from lung cancer who live in the Lorraine region, of Eastern France. We reconstituted the sequence of hospitalizations of patients who have been treated over a period of one year. 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, the first level refers to the root (France) and second, third and fourth levels correspond to administrative region, administrative department and hospital respectively. For example, university hospital of Nancy (code: 540002078) is a hospital in Meurthe et Moselle, which is a department in Lorraine in the region of France. The *principal diagnosis* dimension is described within 5 levels of the 10^{th} International classification of Diseases (ICD10), while the *medical procedures* dimension is described with 5 levels of the CCAM[4] classification.

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 X-Ray (code: $ZBQK$). Then, he was hospitalized in a private clinic in Metz (code: 570023630), for a chemotherapy session (code: $Z511$) where he had a chest X-Ray and pneumonectomy (code: $GFFA$).

Patients	Trajectories
$Patient_1$	$\langle(540002078, C341, \{ZBQK\}), (570023630, Z511, \{ZBQK, GFFA\})\rangle$
$Patient_2$	$\langle(100000017, C770, \{ZBQK\}), (210780581, C770, \{ZZQK, YYYY\})\rangle$
$Patient_3$	$\langle(210780110, H259, \{YYYY\}), (210780110, H259, \{ZZQK\})\rangle$

Table 6: Care trajectories of 3 patients

In this experiment, the support value is set to 20 patients (i.e. $\sigma = 20\%$). *MMISP* generates *194 650* different frequent trajectories. Figure 3 shows the number of discovered patterns according to their length and support. With support between 30% and 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. With support threshold equals to 100%, there is only one pattern $\langle(\top_{hospital}, C34, \{ZBQK\})\rangle$ which shows that 100% of the patients had a Lung cancer (code: $C34$) and underwent chest X-Ray (code: $ZBQK$) during a visit.

Table 7 shows the items appearing in *principal diagnosis* dimension of patterns for which support is over 27%. It can be noticed that the ICD10 tree has been mined at different levels. In the neoplasm

[2] http://illimine.cs.uiuc.edu/
[3] Programme de Médicalisation des Sytèmes d'Information.
[4] Classification Commune des Actes Médicaux : the French classification of medical and surgical procedures

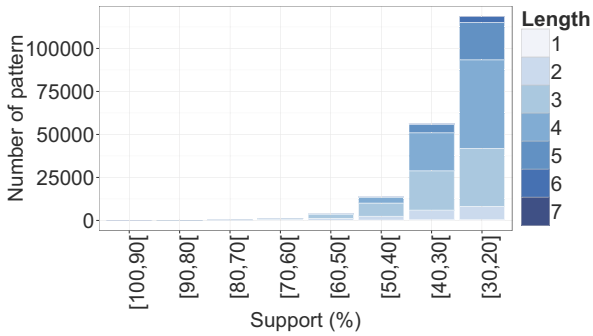

Figure 3: A cumulative description of sequential patterns by support and length

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.

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 27%)

The mds-patterns can be analyzed per se. For example, the pattern $\langle(Lorraine, C34, \{ZBQK, GFFA\})\rangle$ shows that 93% of patients had pneumonectomy (code: GFFA) and chest X-Ray (code: ZBQK) for a lung cancer (code: C34) in any hospital in Lorraine Region in France. The mds-pattern $\langle(Lorraine, \top_{Diag}, \{06.01\}), (Lorraine, C34, \{ZBQK, GFFA\}), (Lorraine, \top_{Diag}, \{06.01.03\})\rangle$ shows that 59% of patients had three hospitalizations where in the first one they started their treatment by underdoing diagnostic test of the respiratory system (code: 06.01) then having pneumonectomy and chest X-Ray 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.

5 Conclusion

This paper presents a new approach to extract sequential patterns from heterogeneous multidimensional 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 real-world dataset. For future work, we are planning to use statistical significance tests to evaluate the extracted sequential patterns and choose the most significant ones.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant, 'Mining sequential patterns', in *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pp. 3–14, Washington, DC, USA, (1995). IEEE Computer Society.

[2] C. Chothia and M. Gerstein, 'Protein evolution. how far can sequences diverge?', *Nature*, **6617**(385), 579–581, (1997).

[3] Elias Egho, Chedy Raïssi, Nicolas Jay, and Amedeo Napoli, 'Mining Heterogeneous Multidimensional Sequential Patterns', Rapport de recherche RR-8521, INRIA, (April 2014).

[4] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu, 'Mining sequential patterns by pattern-growth: The prefixspan approach', *IEEE Trans. Knowl. Data Eng.*, **16**(11), 1424–1440, (2004).

[5] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal, 'Multi-dimensional sequential pattern mining', in *CIKM*, pp. 81–88, (2001).

[6] Marc Plantevit, Anne Laurent, Dominique Laurent, Maguelonne Teisseire, and Yeow Wei Choong, 'Mining multidimensional and multilevel sequential patterns', *TKDD*, **4**(1), 1–37, (2010).

[7] Chedy Raïssi and Jian Pei, 'Towards bounding sequential patterns', in *KDD*, pp. 1379–1387, (2011).

[8] Joan Serrà, Holger Kantz, Xavier Serra, and Ralph G. Andrzejak, 'Predictability of music descriptor time series and its application to cover song detection', *IEEE Transactions on Audio, Speech & Language Processing*, **20**(2), 514–525, (2012).

[9] 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, pp. 3–17, London, UK, UK, (1996). Springer-Verlag.

[10] S.J. Wodak and J. Janin, 'Structural basis of macromolecular recognition.', *Adv Protein Chem*, **61**, 9–73, (2002).

[11] Xifeng Yan, Jiawei Han, and Ramin Afshar, 'Clospan: Mining closed sequential patterns in large datasets', in *In SDM*, pp. 166–177, (2003).

[12] Qiang Yang and Haining Henry Zhang, 'Web-log mining for predictive web caching', *IEEE Trans. on Knowl. and Data Eng.*, **15**(4), 1050–1053, (July 2003).

[13] Chung-Ching Yu and Yen-Liang Chen, 'Mining sequential patterns from multidimensional sequence data', *IEEE Transactions on Knowledge and Data Engineering*, **17**, 136–140, (2005).

[14] Mohammed J. Zaki, 'Spade: An efficient algorithm for mining frequent sequences', *Mach. Learn.*, **42**(1-2), 31–60, (January 2001).

[15] Mohammed J. Zaki and Karam Gouda, 'Fast vertical mining using Diffsets', in *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Aug 2003).

[16] 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 *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, volume 2, pp. 730–734. IEEE, (2007).