

FINDING CONSTRAINED SEQUENTIAL PATTERNS FROM MEDICAL DATASETS

Mohamed Ghouse^{1*} and Fokrul A. Mazarbhuiya²

¹ *Department of Computer Science, College of Computer Science, King Khalid University, Saudi Arabia*

² *Dept. of Mathematics, School of Fundamental and Applied Sciences, Assam Don Bosco University,
Guwahati-782402*

*For correspondence. (mghoth@kku.edu.sa)

Abstract: The problem of mining sequential patterns from supermarket data is an interesting data mining problem and has received a lot of attention to researchers. Mining sequential patterns from supermarket data is actually discovering causal relationship between different itemsets or symptoms that are presents on the patients. Supermarket data contain the records about the information of the itemsets or symptoms of patients besides the patient's personal information and is ordered in accordance with the time / date of consultation of the patients in the hospital. Such data may offer us the precious information related to the cause and effect one itemset on another on the human body. Although, it gives us the ordering of the occurrence itemsets in the human body, it does not provide us the information about the time intervals within which the successive itemsets may occur in a human body. That is, we don't know after how much time the next itemset may occur in human body. In this paper, we try address the issue in detail and propose a method of extracting such sequential patterns from such medical data which occurs within a user-specified time intervals. The efficacy of our method is established by experiment conducted with a dataset collected from a private hospital of Saudi Arabia.

Keywords: Data Mining; Sequential Patterns Mining; Constraint Sequential Pattern Mining; Frequent Sequence; Maximal Sequence; Frequent Sequence within time intervals.

PACS: XXXX, YYYY, ZZZZ

1. Introduction:

Mining patterns from huge data is regarded as an innovative field of database research. This field can be identified as efficient extraction of attractive patterns from big datasets. Temporal Data Mining [1] is currently considered as a significant addition to traditional data mining. There are primarily two wide directions of temporal data mining [2]. The first direction is concerning with the finding of causal relationships among temporally oriented events. The sequences are constructed using the ordered events and the cause of an event always occurs before it. The other direction is concerning with the discovery of similar patterns inside the same time sequence or among dissimilar time sequences. The former problem is termed as sequence mining and is defined as the finding of frequent sequential patterns in the temporal datasets. In 1995, *Agrawal and Srikant* [3] discussed the method of discovering such patterns. In [4], authors have presented, a well-organized method for mining frequent sequences considering a variety of syntactic constraints like length or width limitations on the sequences, minimum or maximum gap constraints, on the consecutive sequence elements.

Pattern Mining from Supermarket data is one of the most vital Data Mining Problem and recently attracted new researchers to work in this field. In [6], works have been on extraction of temporal rules from such data. Supermarket data have all the historical records of information of itemsets of different patients besides their personal information. Every patient has a file which contains the patient's personal information along with a set of list of symptoms or itemsets from date of his first visit in the hospital. The set of list is arranged in accordance with the time of visits in the hospital.

In this paper, we propose two methods of discovering sequential patterns from Medical Datasets. Firstly, we propose an algorithm [7] for mining simple sequential patterns without having time constraints. Secondly, we propose another algorithm for mining time-constraint sequential patterns. The competencies of the two methods is established with the experimental study on a dataset collected from a private hospital of Albaha, Kingdom of Saudi Arabia and comparative studies have been made. Mining such sequential patterns from supermarket data will be helpful for a medical practitioner to recognize the causes and effects of set of symptoms or itemsets on

others. It will be also helpful for Government or Non-government agencies to formulate policies regarding the precautionary measures in a particular area before the mass outbreak of a particular itemset or itemsets.

The paper is organized as follows. In section-2, we briefly discuss about related works. In section-3, we discuss about definitions and notations. In section-4, we discuss about the proposed algorithm. In section 5, we discuss about the experimental detail of this paper and in section-6, conclusion and lines for future works is given.

2. Related Works

The sequence mining problem, an important data mining problem has been addressed in [3, 4, 8, 9, 10, 11]. *Agrawal* and *Srikant* [3] has put forwarded in 1995. In this problem, given a large set of customer transactions, having each customer transaction consisting of customer-id, transaction-time and the items present in the transaction, the problem is to find causal relationship between events where events are non-empty, disordered items. Besides introducing the problem in [3], the authors have proposed three algorithms for solving this problem. Two of these algorithms were intended to unearth only maximal sequential patterns but the third one known as A-prioriAll, unearth all patterns. The later outperforms the former two algorithms. The algorithm is a three-phase algorithm. First it finds out all frequent itemset, then transforms the database in such a manner that each transaction is substituted by the set of all frequent itemsets contained in the transaction and then finds out all sequential patterns. There are two limitations in this approach. First of all, it is computationally expensive to do the data transformation also the storage of transformed database is infeasible or unrealistic as many applications require almost double disk space. Secondly, while it is possible to extend this algorithm to handle time constraints and taxonomies, it does not appear feasible to incorporate sliding windows. In [11], an algorithm proposed which considers minimum and maximum gaps, as well as sliding windows. The algorithm [11] is known as GSP and it is empirically shown that GSP is 20 times faster than others.

Mining medical has been studied widely by distinguish researchers [5, 6, 7]. In [5], authors have presented a concise review on temporally frequent itemsets from supermarket data. A problem similar to [5], is the mining temporally frequent itemsets from supermarket datasets have been discussed in detail in [12, 13, 14]. In [6], authors have proposed a method of finding temporal rules from supermarket data. In [7], an algorithm for mining sequential patterns from supermarket data has been proposed which does consider the time constraint. The algorithm discussed in [7] is quite similar GSP [11]. In [15], authors have discussed the problem of mining medical time series data. Similar works were presented in [16, 17, 18, 19, 20]. In [21, 22], authors have proposed the methods of finding association rules from medical image data and multiple multi-dimensional time series data respectively. In [23], an efficient constraint-based sequential pattern mining is discussed which used dataset filtering techniques. In [24], methods known as pattern growth method for discovering constraint-based sequential patterns is discussed. An algorithm for mining sequential patterns with time-constraint is discussed in [25] where constraint can be pushed up and down effectively.

3. Definitions and Notations

3.1. Definitions and Notations used in Method 1

In below we discuss some of the definitions and notations related to sequence mining problem on medical data that are used in the first algorithm.

Let $\Sigma = \{s_1, s_2, \dots, s_m\}$ be a set of m distinct symptoms comprising the alphabet.

A disease contains a non-empty, disordered collection of one or more symptoms. For the sake convenience we assume that the symptoms in a disease are in some predefined order, symptoms are Boolean variables and there are no missing values which can be done using data preprocessing methods. A disease is denoted as $(s_1, s_2, \dots, s_k)$, where s_j is a symptom in Σ . In this application, we use the words disease and symptom synonymously.

A transaction contains a set of symptoms or diseases for a patient recorded when he visits hospital in a particular time. Thus each patient has a sequence of transactions which are input to the method ordered according to time of visit in a hospital. In below we present some definitions related to sequence mining of medical data.

Definition 3.1.1 Sequence

A *sequence* is an ordered list of diseases. A sequence s is denoted as $(d_1 \rightarrow d_2 \rightarrow \dots \rightarrow d_q)$, where d_i is a disease. A sequence is called k -sequence, if the sum of the cardinalities d_i is k .

A *subsequence* is a sequence within a sequence, preserving the order. A subsequence can be obtained from a sequence by deleting diseases of the sequence.

A sequence s' is said to support another sequence s if s is a subsequence of s' .

Let D be the dataset of input sequences and a sequence is a set of temporally ordered transactions i.e. set of symptoms.

Definition 3.1.2 Frequency of a sequence

The *frequency* of a sequence s with respect to this dataset D , is the total number of input sequences in D that support it.

Definition 3.1.3 Maximal Sequence

A *frequent sequence* is a sequence whose frequency exceeds some user-specified *threshold*. A *frequent sequence* is said to be maximal if it is not a subsequence of another *frequent sequence*.

The rationale behind *frequent sequences* lies in detecting precedence and causal relationships among diseases or symptoms that make them statistically remarkable.

3.2 Definition and Notations used in Method 2

In below we discuss some of the definitions and notations related to sequence mining problem on medical data that are used in the second algorithm.

Definition 3.2.1 Sequence

We mentioned earlier that a *sequence* as an ordered list of diseases. We redefine the sequence as the ordered list of 2-tuple consisting of diseases and time-stamps where each disease is associated with a time stamp which is actually the time of occurrence of the disease. A sequence s is denoted as $((d_1, t_1) \rightarrow (d_2, t_2) \rightarrow \dots \rightarrow (d_q, t_q))$, where d_i is a disease and t_i is the time of occurrence of the disease d_i . A sequence is called k -sequence, if the sum of the cardinalities d_i is k . So every sequence will be associated with a time interval. For example each interval $t_{ij} = [t_i, t_j]$ contains a sequence (d_i, d_j) means that d_i and d_j were occurred within the interval t_{ij} .

As mentioned earlier a *subsequence* is a sequence within a sequence, preserving the order. A subsequence can be obtained from a sequence by deleting some diseases of the sequence.

A sequence s' is said to contain another sequence s if s is a subsequence of s'

A sequence s' is said to support another sequence s if s is a subsequence of s' and the time gaps (time intervals) of consecutive diseases of s in s' contains that of s .

Let D be the dataset of input sequences and a sequence is a set of temporally ordered transaction where each transaction is in the form $((d_1, t_1) \rightarrow (d_2, t_2) \rightarrow \dots \rightarrow (d_q, t_q))$.

Definition 3.2.2 Frequency of a 2-sequence

For every sequence $(a \rightarrow b)$, we maintain a *counter* to count its support value which is zero initially. If an input sequence s in D contains the sequence $(a \rightarrow b)$, then the length of the corresponding time interval say $[t_{sa}, t_{sb}]$ of $(a \rightarrow b)$ in s , i.e. $|t_{sb} - t_{sa}|$ is computed. If the length of the time interval $[t_{sa}, t_{sb}]$ is found to be less than or equal to some user-specified threshold (say $minthd$), then the support value (*counter*) for $(a \rightarrow b)$ is increased. Every input sequence in D supporting $(a \rightarrow b)$ will contribute to its support value. If the total support value for $(a \rightarrow b)$ is greater than or equal to some user-specified threshold (say σ), then $(a \rightarrow b)$ is said to be frequent.

Definition 3.2.2 Frequency of an n -sequence

While finding the frequency of the n -sequence, $s = (a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_{n-1} \rightarrow b_{n-1})$, constructed from two frequent $(n-1)$ -sequences $(a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_{n-1})$ and $(b_1 \rightarrow b_2 \rightarrow \dots \rightarrow b_{n-1})$ using candidate generation method, we proceed as follows. For every n -sequence, a *counter* is kept to count its support value which is zero initially. If an input sequence s' in D contains the sequence $s = (a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_{n-1} \rightarrow b_{n-1})$, then the length of the time interval of last 2-sequence $(a_{n-1} \rightarrow b_{n-1})$ in s' , say $[t_{s'an-1}, t_{s'b_{n-1}}]$ is computed. If $|t_{s'b_{n-1}} - t_{s'an-1}|$ is less than or equal to some user-specified threshold (say $minthd$), then the support value for s is increased. Here, we do not consider the time gap (intervals) of any other 2-sequence of $s = (a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_{n-1} \rightarrow b_{n-1})$ because all its subsequence upto $(n-1)$ are already found to be frequent. Like frequent 2-sequence every input sequence supporting $s = (a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_{n-1} \rightarrow b_{n-1})$ will contribute in the support value of s . If the total support value is greater than or equal to some user-specified threshold (say σ), then we say $s = (a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_{n-1} \rightarrow b_{n-1})$ is frequent.

4 Methods Proposed

In section we give two different methods of finding sequential patterns from supermarket data.

Method 1

In this method, we do not consider the time content explicitly. We just consider the order of the symptoms in the datasets. The method is described below.

The method for mining sequential patterns (itemsets) from medical data is divided into following steps. The medical dataset (D) contains the information about the patients, their date of visits, report of investigations (containing the presence symptoms), beside other details of patients. First of all, we apply data preprocessing methods to convert the data into a form suitable for our application. Here each transaction will be a set of symptoms or itemsets. Then D will be automatically converted into dataset of patient sequences which is input to the algorithm where each transaction will be one or more symptoms. Thus the input to the algorithm is actually the symptoms sequences.

The algorithm makes multiple passes on medical data. The first pass determines the support of each symptom. At the end of pass1, we will know that which symptoms are frequent. That means which symptoms have minimum support. Each frequent symptom will give us 1-sequence i.e. sequence of size 1. It is to be mentioned here that a disease is a collection of 1 or more symptoms. The algorithm for extracting frequent 1-sequence is given below.

Algorithm 1

Algorithm to find $F1$ = the frequent sequent of size-1.

$C[1] = \{d[k]; k=1, 2, \dots, n\}$ where $d[k]$ is the k th symptomset

for ($k=1; k \leq n; k++$)

set $d_{count}[k] = 0$

for each input transaction (sequence) t in the dataset

for ($k=1; k \leq n; k++$) do

```

{ if ( $d[k] \subseteq t$ ) then
increment  $d_{count}[k]$  by 1
}
else if  $d_{count}[k] \geq \sigma$ 
Add  $d[k]$  to  $F[1]$ 

```

A count support count d_{count} is maintained for every symptomset whose value is 0 initially. If a symptomset is found in a transaction, the support count d_{count} for the corresponding symptomset is increased. For a symptomset if the value of d_{count} is found to be greater than or equal to some pre-assigned threshold then it will be called as frequent symptomset i.e. frequent 1-sequence. The process will be repeated for each symptomset present in the datasets.

After that we use candidate generation procedure to find candidate sequence of size 2 i.e. candidate 2-sequence. Then make passes thorough the dataset to find the frequency of 2-sequence and so on. The pseudo code the algorithm to find k -sequence is given below.

Algorithm 2

Algorithm to find sequence of size-k.
 $F1$ = the sequence frequent sequence of size-1 where 1-sequence is a symptomset.
 $k=2$
do while $F(k-1) \neq \text{Null}$
Generate candidate sequences C_k = set of candidate k -sequences, k -sequence is an ordered lists of symptomsets of the form $s = (d[1] \rightarrow d[2] \rightarrow \dots \rightarrow d[k])$ and each $d[i]$ is a collection of one or more symptoms.
Prune (C_k)
for all input sequence t in the dataset D
do
increment count of $s \in C_k$ if $(s \subseteq t)$
 $F_k = \{s \in C_k \text{ such that frequency}(s) \text{ exceeds the threshold}\}$

The algorithm contains two important procedures i) Candidate Generation and ii) Pruning. The candidate generation is quite similar to one discussed in [3]. To generate candidate for next level, we join any two frequent sequence for which the last symptomset are different and upto previous but last symptomset are equal. Similar for pruning we check whether all subsequence of the candidate is present in the previous level or not. If at least one of its subsequence does not belong in the previous level, it will be pruned. The candidate generation procedure and pruning are discussed below.

Candidate sequence generation with the given F_{k-1}

```

gen_candidate_sequences( $F_{k-1}$ )
 $C_k := \emptyset$ 
for all  $(k-1)$ -sequences  $d_{k-1} \in F_{k-1}$ 
for all  $(k-1)$ -sequences  $d'_{k-1} \in F_{k-1}$ 
if  $d_{k-1}[1] = d'_{k-1}[1] \rightarrow d_{k-1}[2] = d'_{k-1}[2] \rightarrow \dots \rightarrow d_{k-1}[k-2] = d'_{k-1}[k-2] \rightarrow d_{k-1}[k-1] \neq d'_{k-1}[k-1]$ 
then  $d_k = (d_{k-1}[1] \rightarrow d_{k-1}[2] \dots d_{k-1}[k-2] \rightarrow d_{k-1}[k-1] \rightarrow d'_{k-1}[k-1])$ 
and  $d'_k = (d_{k-1}[1] \rightarrow d_{k-1}[2] \dots d_{k-1}[k-2] \rightarrow d'_{k-1}[k-1] \rightarrow d_{k-1}[k-1])$ 
 $C_k := C_k \cup \{d_k, d'_k\}$ 

```

Pruning

```

prune( $C_k$ )

```

```

for all  $d_k \in C_k$ 
for all  $(k-1)$ -subsequences  $d_{k-1}$  of  $d_k$  having same starting symptomset
do
if  $d_{k-1} \notin F_{k-1}$ 
then  $C_k = C_k \setminus \{d_k\}$ 

```

The algorithm gives all the frequent sequences where each frequent sequence is an ordered list of symptomsets. Although the method gives all the sequential pattern of symptomsets available in the datasets, it has some discrepancies that is the method does not consider the gap between the cause and its effect. For example, in practice, if the gap between the frequent sequence $(A \rightarrow B)$ is too more than a reasonable time then A can be considered as cause and thus B can be its effect. So the frequent sequence $(A \rightarrow B)$ extracted the above method will be unrealistic and should be removed. To solve this problem, we incorporate the time attributes in it. The detail method is discussed below.

Method 4.2

In this method, we consider the time content explicitly. Here each symptomset will be associated with a time-stamp which is the time of occurrence of the symptomset. The method is described below.

The method for mining interval-based sequential patterns (symptomsets) from medical data is divided into following steps. The medical dataset (D) contains the information about the patients, their date of visits, report of investigations (containing the presence symptoms), beside other details of patients. First of all, we apply data preprocessing methods to convert the data into a form suitable for our application as done in the method. Here each transaction will be a set of symptoms or symptomsets along with a list of time-stamps where each symptomset associated with a time-stamp (time of occurrence of the symptomset). Then D will be automatically converted into dataset of symptomset sequences which is input to the algorithm where each transaction will be one or more symptomsets. We call it temporal sequence dataset because of its explicit time contents

The algorithm makes multiple passes on medical data. The first pass determines the support of each symptomset. At the end of pass1, we will know that which symptomsets are frequent. That means which symptomsets have minimum support. Each frequent symptomset will give us 1-sequence i.e. sequence of size 1. It is to be mentioned here that a symptomset is a collection of 1 or more symptoms. The algorithm is similar to Algorithm 1 used for finding un-constraint sequential patterns. After that we use candidate generation procedure to find candidate sequence of size 2 i.e. candidate 2-sequence. Then make passes thorough the dataset to find the frequency of 2-sequence and so on. The pseudo code the algorithm to find k -sequence is given below.

Algorithm 3

Algorithm to find sequence of size- k .

$F1$ = the sequence frequent sequence of size-1 where 1-sequence is a symptomset.

$k=2$

do while $F(k-1) \neq \text{Null}$

Generate candidate sequences C_k = set of candidate k -sequences, k -sequence is an ordered lists of symptomsets of the form $s = ((d[1], t[1]) \rightarrow (d[2], t[2]) \rightarrow \dots \rightarrow (d[k], t[k]))$ and each $d[i]$ is a collection of one or more symptoms and $t[i]$ time of occurrence of $d[i]$.

Prune (C_k)

for all input sequence t in the dataset D

do

increment count of $s \in C_k$ if $(s \subseteq t) \& \mid t[k-1] - t[k] \mid \leq \text{minthd}$

$Fk = (s \in C_k \text{ such that frequency}(s) \text{ exceeds the threshold } (\sigma))$

The algorithm gives all the frequent sequences along a set of list of time intervals where each frequent sequence will be associated with one or more lists. Like Algorithm 2, it will also have the procedures i) Candidate

Generation and ii) Pruning. To generate candidate for next level, we join any two frequent sequence for which the last symptoms are different and upto previous but last symptomset are equal. But pruning, the procedure is quite different. However, for pruning, we use two types pruning namely subset pruning and time interval pruning. Subset pruning is similar to the Algorithm 2 that is we check whether all subsequences of the candidate are present in the previous level or not. If at least one of its subsequence does not belong in the previous level, it will be pruned. However, in interval pruning, we take difference between the intervals $|t_{a(n-1)} - t_{b(n-1)}|$ of last two consecutives last 2-sequence $(a_{(n-1)} \rightarrow b_{(n-1)})$ of the candidate to check whether they are less than or equal to $minthd$. If all the differences are greater than $minthd$, we prune the candidate. Here, $t_{a(n-1)}$, the time-stamp of $a_{(n-1)}$ and $t_{b(n-1)}$, that of $b_{(n-1)}$ are collected from the sequences from which the candidate is generated. The candidate generation procedure and pruning procedure for the algorithm are explained below.

Candidate sequence generation with the given F_{k-1}

gen_candidate_sequences(F_{k-1})

$C_k := \emptyset$

for all $(k-1)$ -sequences $d_{k-1} \in F_{k-1}$

for all $(k-1)$ -sequences $d'_{k-1} \in F_{k-1}$

if $(d_{k-1}[1], t_{[k-1]}[1]) = (d'_{k-1}[1], t_{[k-1]}[1]) \rightarrow (d_{k-1}[2], t_{[k-1]}[2]) = (d'_{k-1}[2], t_{[k-1]}[2]) \rightarrow \dots \rightarrow (d_{k-1}[k-2], t_{[k-1]}[k-2]) = (d'_{k-1}[k-2], t_{[k-1]}[k-2]) \rightarrow (d_{k-1}[k-1], t_{[k-1]}[k-1]) \neq (d'_{k-1}[k-1], t_{[k-1]}[k-1])$

then $d_k = ((d_{k-1}[1], t_{[k-1]}[1]) \rightarrow (d_{k-1}[2], t_{[k-1]}[2]) \dots (d_{k-1}[k-2], t_{[k-1]}[k-2]) \rightarrow (d_{k-1}[k-1], t_{[k-1]}[k-1]) \rightarrow (d'_{k-1}[k-1], t_{[k-1]}[k-1]))$

and $d'_k = ((d_{k-1}[1], t_{[k-1]}[1]) \rightarrow (d_{k-1}[2], t_{[k-1]}[2]) \dots (d_{k-1}[k-2], t_{[k-1]}[k-2]) \rightarrow (d'_{k-1}[k-1], t_{[k-1]}[k-1]) \rightarrow (d_{k-1}[k-1], t_{[k-1]}[k-1]))$

$C_k := C_k \cup \{d_k, d'_k\}$

Pruning

prune(C_k)

for all $d_k \in C_k$

for all $(k-1)$ -subsequences d_{k-1} of d_k having same starting symptomset

do

if $d_{k-1} \notin F_{k-1}$

then $C_k := C_k \setminus \{d_k\}$

else if $|t_{[k-1]}[k-1] - t_{[k-1]}[k-1]| > minthd$

then $C_k := C_k \setminus \{d_k\}$

The algorithm supplies all the frequent sequential patterns where each frequent sequence is an ordered list of symptomsets and is associated with one or more list time intervals which is the interval of occurrence of symptomsets. For example, if $(A \rightarrow B)_{[t, t]}$ is frequent, then we say A and B occurred frequently in the interval $[t, t]$ and B follows A , Then A can be considered as cause and thus B can be its effect within the time interval $[t, t]$.

5. Experimental setting and results

For the experiment conducted in this paper, we have used a dataset collected from a private Hospital of Albaha, Saudi Arabia. We take the electronic data of total 5000 patients visiting a Private Hospital in Albaha, Saudi Arabia from the last five years for treatment and health check-up. The value of the threshold σ is taken to be 0.4 ($\theta = 0.4$) and minimum length of time interval as 15 days ($minthd = 15$ days). We built prototype application using Java programming language for experimentation. The Graphical User Interface (GUI) is built using Simplified wrapper and Interface Generator (SWIG) API. My SQL is used as back end and Java Database

Connectivity (JDBC) is used for database connectivity. Then the proposed algorithm is applied for the different sizes of datasets. The dataset sizes are determined here by the number of patient's electronic records e.g. 50 patient's records, 100 patient's records, 500 patient's records, 1000 patient's records, 1500 patient's records, 2000 patient's records, 3000 patients record and 5000 patient's records.

For different sizes of dataset, we experiment with both the methods and found that the first method extract more sequential patterns than the second one. That means some sequential patterns extracted by the former method are un-realistic and irrelevant as they have too time-gap. Again, we increase the size of intervals, the output for the former method is same but that for the later method is increased. For, too large time intervals, both methods will give almost similar results. The detail results are better explained using table1, figure1, figure2 and figure4 given below.

Table 1: Number of frequent sequence for different sizes of datasets

Dataset sizes (Number of patient's record)	Number of sequential patterns first method	Number of sequential patterns first method
0	0	0
50	1	1
100	4	3
500	6	4
1000	8	4
1500	9	5
2000	10	5
3000	10	5
5000	10	5

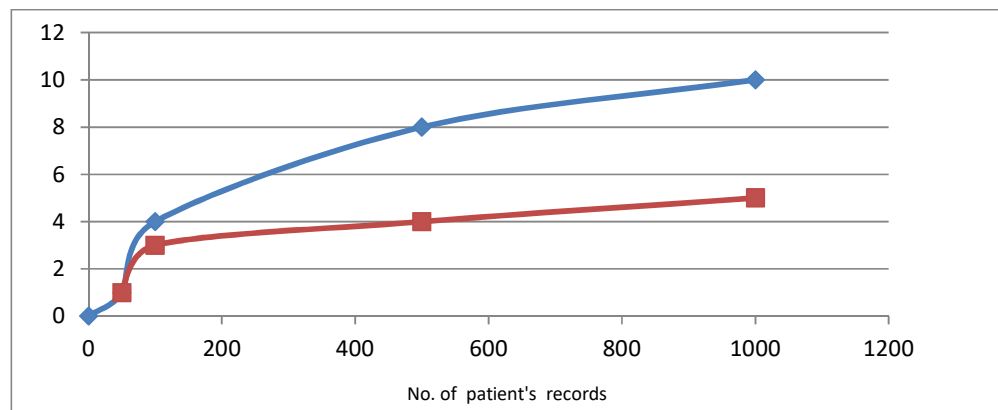


Figure 1: Frequent sequence vs sizes of datasets

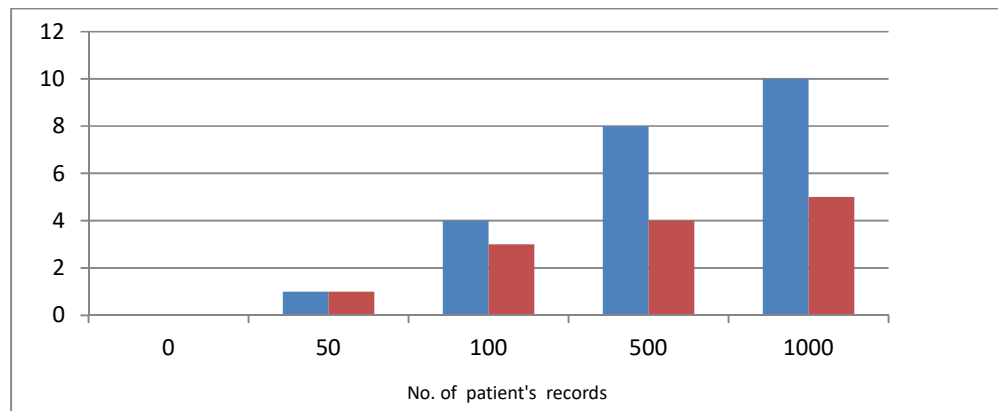


Figure 2: Frequent sequence vs sizes of datasets

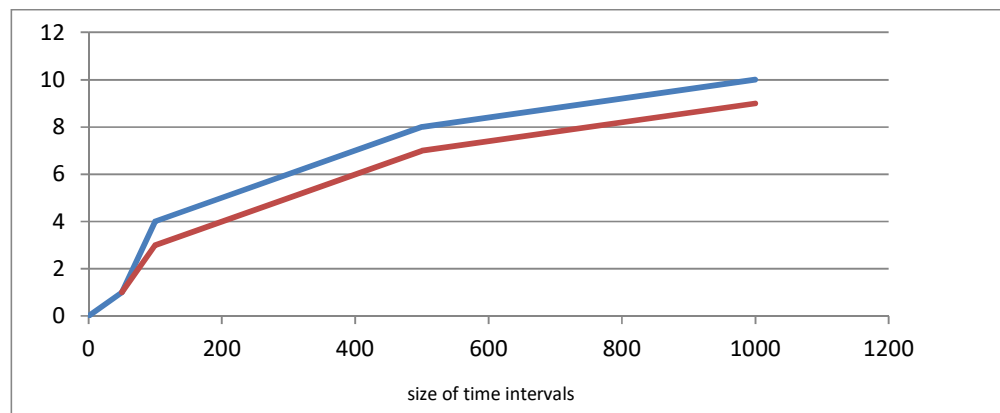


Figure 3: Frequent sequence vs sizes of time-intervals

6. Conclusion and lines for future works

In this paper, we propose two algorithms for finding sequential patterns from medical datasets. The former one is for mining un-constraint sequential patterns. The later one is for time-constraint sequential patterns. In the former, we were only interested to find the cause and effects one or more symptomsets on others irrespective of time existence. It is to be mentioned here a symptomset represents on or more diseases. In practice, it may be possible there the time gap between occurrences of two diseases is two large that they can be considered as cause and effect. To address the issue, we incorporate the explicit time contents here which times of occurrence of the diseases in the dataset and propose, another algorithm which can extract time-interval based sequential patterns of diseases. Here each pattern is associated with one or time intervals in which diseases of the patterns exist with certain order. As we mentioned earlier that medical data contains all the health related information about any patient right from the starting of his visits in the hospital and the records are arranged and kept in the order of the time of visit of the patient. The sequential patterns will tell us which disease follow which other disease in a specified number of patients within which interval or intervals. If a set of disease A follows another set of disease B in a specified number of patients are within certain time interval, then A will be considered as cause of B and B will be considered as consequence of A . The algorithms discussed here are basically level-wise algorithms. Finally, we have implemented the algorithms using a dataset collected from a Private Hospital of Albaha, Kingdom of Saudi Arabia and have tested the algorithm's efficacy by comparing the results. Although we have tried to implement our algorithms using a small dataset collected from a hospital, we are not claiming that above experimental evaluation is best. There is an ample scope in this direction.

In future we can work the following directions.

- In future we will implement our algorithms using a standard dataset to test their efficacies more accurately.
- In future we can search other approaches than level-wise or Apriori based approach.
- In future we can do clustering of patients based similarity of itemsets.
- In future we can extract association rules from set of frequent itemsets.

References:

- [1] C. M. Antunes, and A. L. Oliveira, Temporal Data Mining an overview, *Workshop on Temporal Data Mining-7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, (2001).
- [2] J. F. Roddick, and M. Spilopoulou, A Bibliography of Temporal, Spatial and Spatio-Temporal Data Mining Research, *ACM SIGKDD*, (June'1999).
- [3] R. Agrawal, and R. Srikant, Mining sequential patterns, *In Proc. of 11th Int'l Conf. on Data Engineering*, IEEE, pp.3-14, (1995).
- [4] M. J. Zaki, Efficient enumeration of frequent sequences, *In 7th Int'l Conf. on Information and Knowledge Management*, (Nov'1998).
- [5] M. A. Khaleel, S. K. Pradhan, G.N.Dash, F. A. Mazarbhuiya, A Survey of Data Mining Techniques on Supermarket data for Finding Temporally Frequent Itemsets, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol.2, Issue 12, pp. 4821-4824, (December 2013).
- [6] H. Meamarzadeh, M. R. Khayyambashi and M. H. Saraee, Extracting Temporal Rules from Supermarket data. *IEEE*, pp. 327-331, (2009).
- [7] F. A. Mazarbhuiya, and M. Y. AlZahrani (2017); An efficient method for clustering periodic patterns, Computing Conference 2017, London, U. K. (to be published shortly).
- [8] H. Manilla, H. Toivonen and I. Verkamo, Discovery of frequent episodes in event sequences, *Data Mining and Knowledge Discovery: An International Journal* 1(3), 1997, pp. 259-289.
- [9] M. J. Zaki; Sequence Mining in Categorical Domains Incorporating Constraints, *CIKM 2000*, McLean, VA, USA, pp. 422-429, (2000).
- [10] H. Manilla, H. Toivonen and I. Verkamo, Discovery of frequent episodes in event sequences, *Data Mining and Knowledge Discovery: An International Journal* 1(3), pp. 259-289, (1997).
- [11] R. Agrawal, T. Imielinski and A. N. Swami, Mining Association Rules from set of items in large databases, *In Proceedings of 1993 ACM SIGMOD Int'l Conf. on Management of Data*, Vol. 22(2) of SIGMOD Records, ACM Press, pp. 2017-216, (1997).
- [12] J. M. Ale and G. H. Rossi, An approach to discovering temporal association rules, *In Proc. of 2000 ACM symposium on Applied Computing*, Vol. 1, pp. 290-300, (2000).
- [13] A. K. Mahanta, F. A. Mazarbhuiya and H. K. Baruah, Finding Locally and Periodically Frequent Sets and Periodic Association Rules, *In Proc. of 1st Int'l Conf. on Pattern Recognition and Machine Intelligence*, LNCS 3776, 576-582, (2005).
- [14] A. K. Mahanta, F. A. Mazarbhuiya, And H. K. Baruah, Finding Calendar-based Periodic Patterns, *Pattern Recognition Letters*, Vol. 29(9), Elsevier publication, USA, 1274-1284, (2008).
- [15] C. Catley, H. Stratti and C. McGregor, Multi-Dimensional Temporal Abstraction and Data Mining of Medical Time Series Data: Trends and Challenges, *In proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 4322-4325, (February, 2008).
- [16] C. Catley, K. Smith, C. McGregor, and M. Tracy. Extending CRISP-DM to Incorporate Temporal Data Mining of Multidimensional Supermarket data Streams: A Neonatal Intensive Care Unit Case Study. *IEEE*. pp.1-5, (2009).
- [17] F. A. Mazarbhuiya, M. Shenify, Janet Kez, A. S. Wungreiphi, and Rupa Paul (2025), Correlation Coefficient of Interval-valued Fuzzy Sets with Interval-valued Reference Functions and Applications in Medical Diagnosis and Selecting Appropriate Medicines, *Contemporary Mathematics (Scopus)*, Universal wiser publisher, Vol. 6(2), 2025, Singapore, pp. 2450-2465, <https://doi.org/10.37256/cm.6220256108>.

- [18] F. A. Mazarbhuiya, M. Shenify, A.S. Wungreiphi, J. Kez, and R. Paul, Detecting Alzheimer's Disease using Rough Set-based Classification Approach, Engineered Science (**Scopus**), Vol. 37, 1736, 2025 Engineered Science Publisher, pp. 1-10. <http://dx.doi.org/10.30919/es1736>
- [19] M. Y. Alzahrani, and F. A. Mazarbhuiya; Discovering Constraint-based Sequential Patterns from Medical Datasets, International Journal Recent Technology and Engineering, Vol. 8 (4), 2019, pp 724-728, India.
- [20] M. Y. AlZahrani and F. A. Mazarbhuiya, Discovering Sequential Patterns from Medical Datasets, Proceedings of 2016 International Conference on Computational Science and Computational Intelligence, IEEE Explore, Las Vegas, USA, 2016, pp. 70-73
- [21] A. Olukunle and S. Ehikioya, A Fast Algorithm for Mining Association Rules in Medical Image Data. *IEEE*. p1-7, (2002).
- [22] G. N. Pradhan and B. Prabhakaran, Association Rule Mining In Multiple, Multidimensional Time Series Supermarket data. *IEEE*. pp.1-4, (2009).
- [23] Tadeusz Morzy, Marek Wojciechowski, and Maciej Zakrzewicz, Efficient constraint-based Sequential pattern mining using Dataset filtering techniques, Databases and Information Systems II, Fifth International Baltic Conference, Baltic DB&IS'2002 Tallinn, Estonia, pp. 297-309.
- [24] Jian Pei, Jiawei Han, and Wei Wang, Constraint-based sequential pattern mining: the pattern-growth methods, Journal of Intelligent Information Systems, Vol. 18, Issue 2, pp. 133-160, April 2003.
- [25] Anita Zala and Mehul Barot, Mining Sequential Pattern with Time-Constraint, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 7, January 2013.