

# Time-Invariant Sequential Association Rules: Discovering Interesting Rules in Critical Care Databases

Jafar Adibi, Wei-Min Shen

Information Sciences Institute, University of Southern California  
4676 Admiralty Way, Marina del Ray, CA 90292

{adibi,shen}@isi.edu

## ABSTRACT

Discovering patterns in sequences of events has been an area of active research in Artificial Intelligence and Data Mining. Many existing techniques which generate sequential association rules have two major problems: they either produce too many rules or they cannot discover rules that have high *confidence*, but weak *support*. Both cases make manual inspection and analysis very difficult. The focus in this body of work is on discovering such rules in a recognized group of special databases, in which data are not uniformly distributed and exhibit self-similarity and fractal dimensionalities. We introduce, study and analyze a group of sequential association rules as time-invariant and self-similar association rules. We provide a formalism to discover such rules through the discovery of association rules with the high degree of *confidence* and *support*. Time-invariant and self-similar association rules has been investigated in the context of Critical Care database which has been collected during past 15 years at the King Drew Medical Center and Harbor UCLA Hospital. Even though the obtained result is in early stage but they are encouraging and we would like to apply this technique to other synthetic and real databases in the future.

## Keywords

Association Rules, Self-Similarity, Time Invariance

## 1. INTRODUCTION

Much of the existing data mining techniques have been focused on designing efficient methods to mine knowledge and patterns from databases. Sequential association rule is one of the most well known forms of extracted knowledge. Instead of statistical methods which are looking for a global model for data, association rules mainly find local patterns. An association rule is in the form  $(P \Rightarrow Q)$ , where  $P$  and  $Q$  are sets of attributes, meaning that in the rows of the database where the attributes in  $P$  have true value, also the attributes in  $Q$  tend to have true value. Association rules define with two major parameters: *support* and *confidence*. The *support*

of a given rule is the ratio of the records having true values for the attributes of  $(P \cup Q)$  to number of all records, whereas the *confidence* of that rule is the ratio of the number of records having true values for attributes of  $(P \cap Q)$  to the number of records having true values for attributes of  $P$ .

The main approach for mining association rules in general derives by Agrawal et al [4] called a-priori, which exploits the support requirements for association rules. The key observation is that if a set of attributes appears in a fraction of the tuples, then any subset of such set also appears in a fraction of the tuples. Variants and enhancements of this approach underlie essentially all known efficient algorithms for computing association rules or their variants. The general algorithm mainly works with the *support* level and *confidence* requirement plays no role in the algorithm, and is completely ignored until the end of the discovery loop when high-supported sets are screened for high *confidence*.

Many existing techniques which generate association rules are facing two major problems: they often either produce too many rules and/or they cannot find *interesting rules*. By *interesting rules* we refer to those rules which have extremely high confidence, but for which there is weak support. Both cases make manual inspection and analysis very difficult.

This work is motivated by the open question of discovering *interesting rules*. For example, in the medical treatment domain, the standard association rule algorithms may be useful for extracting patterns with high *support* such as

*“if treatment A applies to patient in first hour of admission, she recovers in less than 8 hours”*,

but are essentially useless for discovering rules such as

*“if treatment A applies to patient after 96 hours, she may recover after 4 days”*,

because there are only a few patients who received the treatment after the 96 hours following the admission procedure.

There are two possible objections to removing the support requirement from the discovery process [6]. First, this may cause an explosion in the number of rules that are produced and make it difficult for a user to distinguish the rules of interest or take huge amount of time to discover such rules. Second, it may be argued that rules of low support are uninteresting. While this might be true in the classical market-basket applications, there are many applications where it is essential to discover such rules of extremely high *confidence* without enough *support*. For many

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference 'KDD-01, August, 2001, San Francisco, CA.  
Copyright 2001 ACM 1-58113-000-0/00/0000...\$5.00.

scientific databases *interesting rules* are indeed very crucial and does not happen so often. To name a few: medical database, financial datasets including transaction databases, identifying identical or similar documents or web pages, identifying similar vectors in high-dimensional spaces and collaborative filtering [6] are examples in which a rule with low-support and high confidence is very crucial. Some of these applications consists of a sparse table and the goal is to identify column pairs that appear to be similar, without any support requirement. In addition, detecting causality is another important form in data mining, where it is important to discover associated fields, but there is no notion of support [17].

The implicit assumption in some of the studies on sequential association rules is that the data is uniformly distributed and attributes are independent from each other. This assumption basically implies that a-priori like algorithms do not use of the data structure, shape and characteristics. However, real data sets disobey these assumptions. A recognized groups of data typically are skewed and exhibit fractal dimensionalities. For instance, most of the biological systems contain self-similar structures that are made through recurrent processes. In these databases, the information and embedded complexity are hierarchical. In addition, they are self-similar and/or contain self-similar structures and/or have been generated through recurrent processes at least up to a certain level. Many physical systems contain a form of functional self-similarity that owes its richness to recursion. Human brains, economic markets, musical notes, network data also create enormously complex behavior that is much richer than the behavior of the individual component units. New findings in different branches of science and technology also show the presence of self-similarity in different domains. To name a few: medical diagnosis (physician treatment and patient response), Robot navigation (robot move and environment response to robot sensors) and Network behavior monitoring (packet transmission, switch behavior and network response) are examples of such environments.

The main goal of this paper is to provide a novel technique to address above-mentioned problems for a special form of temporal databases which shows self-similarity up to a certain degree. In general, self-similarity or long range dependence refers to observation of similar patterns when a discrete or continuous time process is scaled in time. The process in larger scale is a copy of itself in smaller time scales. We employ such idea for self-similar databases. The presence of a rule in small scale is a copy of itself in larger scale and vice versa. As regular a-priori algorithms discover rules in smaller scale with enough *user define support* we can discover rules in larger scale through the discovered rules in smaller scale even if they do not have enough support.

While there have been much effort on observing self-similar structures in scientific databases and natural structures, there is few work on using self-similar structure and fractal dimension for data mining, predictive modeling and forecasting. Among these works, using fractal dimension and self-similarity for managing the dimensionally curse [19], learning association rules [5], application in spatial joint selectivity in databases [9] and Self-Similar Layered HMM for self-similar structures [2] are considerable.

The rest of this paper is organized as follows. In section 2 we explain the problem statement along with notation and definitions. In section 3 we outline related work to this paper. In section 4, we introduce the time-invariant association rules for sequential databases, its definition and properties. In section 5 we discuss our method on discovering time-invariant sequential association rules. Section 6 shows the current result with an experimental finding in Critical Care patient database followed by the future work and conclusions in section 7.

**Table 1: Example of a Critical Care database**

Patient ID = 1						Patient ID = 2					
	Type	Time	Perception/Action				Type	Time	Perception/Action		
1	P	8	O <sub>1</sub> =101	O <sub>2</sub> =1.5	O <sub>3</sub> =0	1	P	9	O <sub>1</sub> =100	O <sub>2</sub> =1.0	O <sub>3</sub> =0
2	P	12	O <sub>1</sub> =120	O <sub>2</sub> =1.2	O <sub>3</sub> =1	2	P	13	O <sub>1</sub> =121	O <sub>2</sub> =1.1	O <sub>3</sub> =1
3	A	13	X			3	A	15	X		
4	P	24	O <sub>1</sub> =144	O <sub>2</sub> =1.5	O <sub>3</sub> =1	4	P	27	O <sub>1</sub> =143	O <sub>2</sub> =1.3	O <sub>3</sub> =1
5	A	26	X			5	A	29	X		
6	P	28	O <sub>1</sub> =106	O <sub>2</sub> =1.3	O <sub>3</sub> =-1	6	P	31	O <sub>1</sub> =116	O <sub>2</sub> =1.7	O <sub>3</sub> =-1
7	A	32	Y			7	A	36	Y		
8	P	38	O <sub>1</sub> =132	O <sub>2</sub> =1.2	O <sub>3</sub> =1	8	P	38	O <sub>1</sub> =112	O <sub>2</sub> =1.6	O <sub>3</sub> =-1
9	A	44	X			9	A	46	X		
10	P	55	O <sub>1</sub> =101	O <sub>2</sub> =1.5	O <sub>3</sub> =-1	10	P	54	O <sub>1</sub> =100	O <sub>2</sub> =1.0	O <sub>3</sub> =-1
11	P	67	O <sub>1</sub> =108	O <sub>2</sub> =1.6	O <sub>3</sub> =1	11	P	60	O <sub>1</sub> =99	O <sub>2</sub> =1.5	O <sub>3</sub> =-1
12	A	83	X			12	A	79	Y		
13	P	90	O <sub>1</sub> =144	O <sub>2</sub> =1.7	O <sub>3</sub> =1	13	P	92	O <sub>1</sub> =108	O <sub>2</sub> =1.4	O <sub>3</sub> =1
14	A	110	Y			14	P	110	O <sub>1</sub> =121	O <sub>2</sub> =1.2	O <sub>3</sub> =1
15	P	121	O <sub>1</sub> =111	O <sub>2</sub> =1.8	O <sub>3</sub> =-1	15	A	134	X		
16	A	134	X			16	P	145	O <sub>1</sub> =140	O <sub>2</sub> =0.9	O <sub>3</sub> =1
17	P	165	O <sub>1</sub> =123	O <sub>2</sub> =0.5	O <sub>3</sub> =-1	17	A	153	X		
18	A	178	X			18	P	162	O <sub>1</sub> =111	O <sub>2</sub> =0.7	O <sub>3</sub> =-1
19	A	181	X			19	A	177	Y		
20	P	182	O <sub>1</sub> =109	O <sub>2</sub> =0.6	O <sub>3</sub> =-1	20	P	190	O <sub>1</sub> =113	O <sub>2</sub> =0.8	O <sub>3</sub> =1
21	P	200	O <sub>1</sub> =115	O <sub>2</sub> =0.8	O <sub>3</sub> =1	21	P	204	O <sub>1</sub> =120	O <sub>2</sub> =1.3	O <sub>3</sub> =1
22	A	202	X			22	A	250	X		

## 2. PROBLEM STATEMENT

We are given a database of sequences  $D=\{d_1, d_2, \dots, d_n\}$ . Each sequence  $d_i$  belongs to a patient, customer or in general belongs to an entity and consists of a collection of perceptions and actions. Each item in a sequence is either a perception or an action. Each perception consists of: *entity\_id*, *perception-time* and *attribute\_set* (*attribute\_id*, *attribute\_value*). Each action also consists of *entity\_id*, *action\_time*, *action\_id*. Table 1 shows examples of 2 different patients in a given database. Perceptions  $O_1$ ,  $O_2$  and  $O_3$  has been observed for all patients across the database and treatments  $X$  and  $Y$  has been applied. Grayed rows in Table 1. refer to actions.

### 2.1 Definition

Given a database  $D$  of  $N$  data-sequences, an action set of  $C$  and Perception set  $O$ , user-specified *min-gap* and *max-gap* time constraints, the problem of mining interesting sequential patterns is to find all sequences whose support is greater than the *user-specified minimum support* or its confidence is greater than the *user-specified minimum confidence*. Each sequence represents a sequential pattern of perception and action, also called an *interesting sequence*.

Note that the notion of *min-gap* and *max-gap* are different with what Agrawal et al introduced in [3]. In their work the *min-gap* and *max-gap* basically is used for removing noise and outliers in time series matching process. However in our point of view *min-gap* and *max-gap* refers to the length of a sequential pattern in general. If we look at an association rules in form  $(P \Rightarrow Q)$ , where  $P$  and  $Q$  are sets of attributes, *min-gap* associate with  $P$  while *max-gap* associate with  $P$  and  $Q$ . The definition of *min-gap* and *max-gap* will be explained later in this chapter.

The main idea is to discover *interesting rules* by scaling the general sequential association rules (mainly those which satisfy *minimum-support*). The notion of scaling which associate with *min-gap* and *max-gap* will explain in section 4.

We do consider quantities of perceptions but do not consider the quantities for actions (for example the dosage of drug in our example): each item is a number for actions and a function of quantity for perceptions. Without losing generality and for the purpose of better understanding we map actions and perceptions to different set. We denote a sequence by  $\langle s_1, s_2, \dots, s_n \rangle$  in which  $s_i$  is an action or perception. Each  $s_i$  will be a tuple of quantities and the time (value, time) and it refers as *item*. In general we demonstrate each sequence with  $S(i)_{start, end}^{perception}$ , in which  $i$  refers

to entity *id*, *perception* stands for the index of the observed perception and *start* and *end* refers to the starting point and ending point of a given sequence. For instance, perception  $O_3$  for patient  $ID = 1$  in Table.1 will be shown as the following:

$S(1)_{1,10}^3 = \langle (0,8), (1,12), (X,13), (1,24), (X,26), (1,28), (Y,32), (1,38), (X,44), (1,50) \rangle$ . For the short note we only show the value of perception and action :  $S(1)_{1,10}^3 = \langle 0,1,X,1,X,1,Y,1,X,1 \rangle$ . For a

specific item  $j$  in sequence  $S(i)_{s,e}^p$  we use  $S(i)_j^p$ . In addition to show the time of a given item in sequence  $S(i)_j^p$  we use of

$T(S(i)_j^p)$ . For instance in previous example  $T(S(1)_2^3) = 12$ .

### 2.2 Pattern

We define a pattern as a sequence with additional capabilities to normal sequence. First, a pattern can have a general observation symbol standing for all possible observation or action. We show this symbol by  $[A]$  for actions and  $[O]$  for observations and  $[I]$  for either action or observation. A pattern also can have  $*$  in front of each *item* in sequence. A  $*$  represents the zero to infinite number of such item. We denote a pattern with  $P$  and the  $L_p = |P|$  represent the length of pattern. Patterns only represent the values and do not contain the time of sequence. Table 2 shows a full definition of pattern notation. For instance  $P_1 = \langle [A]^*, 1, X, 1, Y, [I]^*, 1 \rangle$  and  $P_2 = \langle 1, [I]^*, A, [I]^*, 1 \rangle$  are simple examples of patterns and  $S(i)_{s,e}^p = \langle 1, 0, X, Y, 1 \rangle$  and  $S(i)_{s,e}^p = \langle 1, 1, X, 0, Y, -1, 1 \rangle$  are example of sequences which satisfy  $P_2$ .

We also associate a *max-gap* ( $G_{max}$ ) and *min-gap* ( $G_{min}$ ) with a pattern.  $G_{max}$  and  $G_{min}$  basically control the length of the sequence. If we consider a sequence in the form of  $P \Rightarrow Q$ , in which  $P$  refers to  $S(i)_{1,|P-1|}^p$  and  $Q$  refers to  $S(i)_{|L|}^p$ ,  $G_{max}$  controls the length of the whole sequence and  $G_{min}$  controls the length of the  $P$  part in  $P \Rightarrow Q$ . Hence, for a given pattern with defined  $G_{max}$  and  $G_{min}$

$$T(S(i)_{|P|}^p) - T(S(i)_1^p) < G_{max}$$

$$T(S(i)_{|P-1|}^p) - T(S(i)_1^p) > G_{min}.$$

**Table 2: Notation and Definitions**

Symbol	Description	Example
item	Either Action or Perception	X,1
A	Actions	X
$O_i$	Perceptions	-1
[A]	Any kind of actions	Y
[O]	Any kind of perceptions	1
[I]	Any item (actions or perceptions)	X,1
$Y^*$	Unlimited number of perception Y	Y,Y,Y
$[A]^*$	Unlimited number of actions	X,Y,X
$[O]^*$	Unlimited number of perceptions	1,1,-1
$[I]^*$	Unlimited number of actions or perceptions	X,1,Y,-1

Rephrasing the problem statement, we are looking for all frequent patterns  $\langle s_1, s_2, \dots, s_n \rangle$  when their confidence is greater than user defined *minimum-confidence* even if their support is less than *minimum-support*, while and  $T(S(i)_{|P|}^p) - T(S(i)_1^p) < G_{max}$  and  $T(S(i)_{|P-1|}^p) - T(S(i)_1^p) > G_{min}$ . In our definition *support* is the frequency count of a sequence  $S(i)_{s,e}^p = \langle s_s, s_2, \dots, s_e \rangle$  in  $D$ , and confidence will be as: *Frequency count of*  $S(i)_{s,n}^p = \langle s_s, \dots, s_n \rangle /$

Frequency count of  $S(i)_{s,n-1}^p = \langle s_s, \dots, s_{n-1} \rangle$ . As it shows we are looking for *if-then* type of rule ( $P \Rightarrow Q$ ) in which *if* part refers to  $S(i)_{s,n-1}^p$  and *then* part refers to  $S(i)_{s,n}^p$ .

### 3. RELATED WORK

Sequential pattern mining is an important data mining problem with broad applications, including the analyses of customer purchase behavior, Web access patterns, scientific experiments, disease treatments, patient database, natural disasters, DNA sequences, Network data analysis etc. In AI a lot of work has been done for discovering patterns in sequential data [12] [8]. In the database context, where input data is usually much larger, the problem has been studied in a number of recent papers [6, 18] [14] [4]. In [14] event sequences are searched for frequent patterns of events. These patterns have a simple structure (essentially a partial order) whose total span of time is constrained by a window given by the user. The technique of generating candidate patterns from sub-patterns, together with a sliding window method, is shown to provide effective algorithms. In [4] the problem of discovering sequential patterns over large databases of customer transactions is considered. Similarly to [14], the strategy of [4] is starting with simple sub-patterns (subsequences in this case) and incrementally building longer sequence candidates for the discovery process (Apriori Algorithm). Almost all of the previously proposed methods for mining sequential patterns and other time-related frequent patterns are apriori-like, which states the fact that any super-pattern of a non-frequent pattern cannot be frequent. Based on this heuristic, a typical-like method such as GSP [18] adopts a multiple-pass, candidate-generation and test approach in sequential pattern mining.

Han et al proposed a technique to mining sequential data without candidate generation [10]. They introduced frequent pattern tree structure, which is an extended prefix tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree-based mining method, FP-growth, for mining the complete set of frequent patterns by pattern fragment growth. The work in [20] also deals with the discovery of sequential patterns. In [20] the considered patterns are in the form of specific regular expressions with a distance metrics as a dissimilarity measure in comparing two sequences. In [21] a scenario is considered where sequential patterns have previously been discovered and an update is subsequently made to the database. Das et al in their work [7] presents a new method for rule discovery from time series data. They slide a window over data and find the class of the subsequence and then find common episodes in represents.

Many researchers have studied the problem of too many rules and discovering interesting rules (e.g., Piatetsky-Shapiro & Matheus 1994; Klemetinen et al 1994; Silberschatz & Tuzhilin 1996; Liu & Hsu 1996; Padmanabhan & Tuzhilin 1998) have been proposed to help the user find interesting rules from a large number of discovered rules. The main approaches are either using some interestingness measures to filter out those uninteresting rules or using the user's domain. Cohen et al studied the finding rules with high confidence and low support. They developed a family of algorithms, employing a combination of random sampling and hashing techniques.

There are a few works, which attempt to incorporate the self-similar information in the association rule discovery. Barbara introduced using the fractal dimension to analyze how association rules occur in a dataset [5]. They developed a two tire techniques. First, as a k-itemset is under consideration, and they are scanning the dataset to compute its support, they also roll a window and compute the fractal dimension of the occurrence of this rule as the algorithm goes through the data. Secondly, if this itemset is found to have a lot of support, enough information about the fractal dimension of this rolling window will be kept to be used when processing the k+1 extensions of this itemset in the next iteration of the algorithm. Our work is different with [5] as we address scalability and *interesting rules* rather than using fractal dimension to find those rules.

### 4. TIEM INVARIANT SEQUENTIAL ASSOCIATION RULES

The notion of time-invariant sequential association rules refers to a group of discovered rules in which a change such as scaling in time constraints implies a new rules with enough *support* or *confidence*. As pattern in smaller scale repeats in larger scale, a new rules may discover by the change in time constraint of discovered rules such as  $G_{min}$  and  $G_{max}$ . Time-invariance space is broader than self-similarity space. While self-similarity has to be maintained in different scale of a given data, the notion of time-invariance does not need such requirements and may only maintains in one or two scales.

On the other hand, the notion of self-similarity in general helps to understand the concept of time invariance in a continuous time series or discrete database. In the following, we review briefly the notion of self-similarity in continuous domain and we provide a detail definition for time-invariant sequential association rules and self-similar sequential association rules.

#### 4.1 Self Similarity

The mathematical study of self-similar shapes and their relationship to natural shapes was first presented by Benoit Mandelbrot. Self-similar stochastic processes were introduced by Kolmogorov in a theoretical context and brought to the attention of probabilists and statisticians by Mandelbrot and his co-workers and have been used in hydrology, geophysics, biophysics, and biology and communication systems [13].

In general, self-similarity or long range dependence refers to observation of similar patterns when a discrete or continuous time process is scaled in time. The process in larger scale is a copy of itself in smaller time scales. In self-similar signals the key parameter is not the mean or variance, but the degree of self-similarity, defined via the Hurst parameter. The notion of self-similarity is not merely an intuitive description but a precise concept captured by the following rigorous mathematical definition. Let  $X$  be wide sense stationary process, that is; a process with constant mean and finite variance and auto correlation function  $r(k)$ . For each  $m=1,2,\dots$ , let  $X^{(m)}$  denotes a new time series obtained by averaging the original series  $X$  over non-overlapping blocks of size  $m$ . That is for each  $m=1,2,\dots$ ,

$X^{(m)}$  is given by  $X_k^{(m)} = 1/m(X_{km-m+1} + \dots + X_{km})$ , which  $X \geq 1$ . Note that for each  $m$ , the aggregated time series  $X^{(m)}$

defines a wide sense stationary process; let  $r^{(m)}$  denote the corresponding auto correlation function. The process  $X$  is called exactly H-self similar if for all  $m > 0$  it holds

$$X_k = m^{-H} \sum_{i=(k-1)m+1}^{km} X_i$$

By looking at a self-similar sequential data generated through recurrent process, a macro point of view suggests that the overall system behavior is more a trajectory among phases. As a self similar process repeats it self, a pattern will be repeated in different scales. Discrete self-similarity share the same characteristics with continues domain. In the following we define the time-invariant sequential association rules and self-similar sequential association rules.

## 4.2 Time-Invariant Association Rules

In the following, we provide definition of time invariance and self-similarity for association rules along with some examples.

**Time-invariant Sequential Association Rule:** A sequential association rule such as  $P$  define as a frequent pattern, if it satisfies user defined *support* and *confidence* with  $G_{p_{min}}$  and  $G_{p_{max}}$  ( $G_{p_{min}}$  refers to  $G_{min}$  and  $G_{p_{max}}$  refers to  $G_{max}$  for pattern  $P$ ).  $P$  is time-invariant if there is a pattern such as  $Q$ , with  $G_{q_{min}}$  and  $G_{q_{max}}$ , satisfies *minimum-support* or *minimum-confidence* when  $G_{q_{min}} = K_{min} * G_{p_{min}}$  and  $G_{q_{max}} = K_{max} * G_{p_{max}}$  ( $G_{q_{min}}$  refers to  $G_{min}$  and  $G_{q_{max}}$  refers to  $G_{max}$  for pattern  $Q$ ). Pattern  $Q$  calls interesting if it does not satisfy the *minimum-support* but satisfies the *minimum-confidence*. Note that in most of the cases  $K_{min}$  and  $K_{max}$  belong to the same order of magnitude and  $K_{min}, K_{max} \in R$ .

**Self-Similar Sequential Association Rule:** A pattern  $Q$  calls self-similar sequential association rules if  $Q$  known as time-invariant sequential association rule and if  $Q$  maintain such property for different scale of  $K_{min}$  and  $K_{max}$  in which  $K_{min}, K_{max} \in R$ .

A self-similar data exhibits fractal dimensionalities up to a certain level and has been generated through a recurrent process. The fractal dimension of a time series and self-similarity may validate through well-known algorithms such as introduced in [11].

**Example 1:** Assume  $S = \langle 1, X, Y, 1, -1, 1, 0, X, X, X, 2, Y, 1, 2, X, Y, 1, X, 0, -1, 1, X, -1, Y \rangle$ . Pattern  $P = \langle 1, [I]^*, X, [I]^*, Y \rangle$  with  $G_{min}=3$  and  $G_{max}=4$  will have *support* = 3 and *confidence* = 3/5. If we scale up this pattern with  $G_{min}=6$  and  $G_{max}=12$  the *support* and *confidence* will be 2 and 2 respectively.

**Example 2:** Table 1 shows an example of Critical Care database. There are two patients in the database with a series of perceptions and actions. No patient has more than one transaction with the same transaction-time. We do not consider quantities of given treatment applied to patient for this stage: each item is a variable representing which treatment was given to the patient or not. A treatment set is a non-empty set of treatments. A sequence is an ordered list of perceptions (signs) and actions (treatments).

For the pattern  $P = \langle 1, [I]^*, X, [I]^*, I \rangle$  with  $G_{p_{min}} = 10$  and  $G_{p_{max}} = 100$ , *support* is equal to 5 and *confidence* is 5/8. If we set the minimum support equal to 4,  $P$  pass the support filter and it is in the result. However, for the pattern  $R = \langle 1, [I]^*, X, [I]^*, I \rangle$  with  $G_{r_{min}} = 150$  and  $G_{r_{max}} = 300$  does not pass the support filter,

because even though its *confidence* is equal to 1 but its *support* is equal to 2.

The time-invariant sequential association rules provide such facilities to capture  $R$  having  $P$  with enough support. Interpretation of a sequential association rule plays an important role in rule understanding and rule scaling. For instance pattern  $P$  and  $R$  can be interpreted as:

“if treatment  $A$  applied to patient right in 10 time units, good response will be observed in less than 100 time units”.

Similar to  $P$ , Pattern  $R$  can be interpreted as :

“ If a treatment has applied to a patient in 150 time unit, patient most probably response in a longer period (300 time units)”.

## 5. METHOD

The general approach is to find the frequent sequences which satisfy the support level,  $G_{min}$  and  $G_{max}$  in first step and scale the discovered rules in second step to obtain new sequences which satisfies an acceptable level of *confidence*. In a departure of previous techniques the main contribution of this methods comes from the structure embedded in self-similar data. The self-similarity implies a rule in smaller scale may repeats in larger scale even though if it may does not satisfy the user defined *minimum-support*. In the following we explain the major steps with a simple pseudo code to capture such rules:

1. Find patterns with support greater than *minimum-support*. This part of the algorithm would be similar to the most of the existing approaches. The main difference is in the frequency count part of a pattern in which we apply the notion of *min-gap* and *max-gap*. Figure 1 shows the pseudo algorithm to find the frequency count of a pattern in a sequence considering *min-gap* and *max-gap*. As it shows the algorithm uses of a dynamic programming like algorithm to capture \* factor and gap constraints in the pattern. Similar to an a-priori like algorithm  $L$  keeps track of matches in pattern and sequence. When a match occurs,  $L(i,j)$  increases when there is at least a match in past items unless if it is a duplicate pattern.  $F$  keeps track of time difference which will be checked for  $G_{min}$  and  $G_{max}$ .  $T$  keep the exact time of a perceptions and actions. Note that the sequence scanned only once and the order of the a-priori like algorithm has not changed.
2. If there is not enough frequent pattern found in the data, change the  $G_{max}$  to a greater value. This increases the possibility of observing frequent pattern in a sequence as \* play an important role in frequency count.
3. Since data is self-similar or has shown self-similarity up to a certain degree, for all frequent patterns, scales up the rule by scaling up the  $G_{max}$  and  $G_{min}$ .
4. Scan  $D$  from the beginning and compute the frequency count of the new rule. Store new rules if their confidence is greater than user *minimum-confidence*. These rules are essentially interesting cause they are not intuitive, but they could happen only a few times for a perception-action data. As the number of occurrences of these rules is relatively low they never recognized in a-priori like algorithms, which are *support-based* algorithms.

**Figure 1: Pattern Matching Considering  $G_{min}$  and  $G_{max}$**

```

FindFrequentPattern(Pattern,Sequence,Gmax,Gmin);
initialization
Loop in Sequence i
  Loop in Pattern j
    IF Sequence(i) == Pattern(j)
      IF i==1 | j==1
        L(i,j)=1; keep track record of matches
        F(i,j)= 0; get the time difference
        T(i,j) = i-1; the exact time
      Else IF L(i-1,j-1) ~ = 0
        if window condition satisfies
          L(i,j)=L(i-1,j-1) + 1;
          F(i,j)= temp; T(i,j) = i-1;
        Else IF find all non zero members in previous column
          set index to most recent none zero
          IF it's not duplicate
            IF it satisfies window condition
              L(i,j) = L(index,j-1) + 1;
              F(i,j) = (i-1-T(index,j-1))+F(index,j-1);
              T(i,j) = i-1;
            Else no match
              L(i,j)=1; F(i,j)=0; T(i,j)=i-1;
          IF L(i,j)== length of Pattern
            count=count+1;
          ELSE
            L(i,j)=0; F(i,j)=0; T(i,j)=i-1;

```

## 5.1 Scale Factor

Scale factors ( $K_{min}$  and  $K_{max}$ ) has employed to provide new constraints as:

$$T(S(i)_{|P|}^p) - T(S(i)_1^p) < K_{max} * G_{max}$$

$$T(S(i)_{|P-1|}^p) - T(S(i)_1^p) > K_{min} * G_{min}$$

The scale factor basically is very subjective and has a strong bound with the domain knowledge, user input and user preferences. For a Medical database a scale factor is from *minute*, *15 minutes* and *an hour* up to *8 hours*, *12 hours*, *24 hours*, *96 hours*. For a Network data base scale factor is from *10 minutes*, *20 minute*, *30 minutes* and *an hour*, up to *a week*, and *a month*.

The scale factor may consider as hidden information. The lack of such knowledge is similar to hidden information such as number of clusters in a clustering problem or number of state in a Markov modeling problem. However, similar to those problems scaling factor has strong roots in the nature of the problem itself and it can provide either by user, using a heuristic or through a search process.

## 5.2 Analysis

There are two major issues in knowledge discovery loop which has to be considered. These issues are Time and Space.

**Number of scan over the time series:** a-priori like algorithm scans databases  $P$  time that is equal to number of  $L$ -length patterns satisfy the *minimum-support*. If we show the discovered rules in step 1 as  $R$  and the length  $R$  with  $L_R=|R|$ , we would scale up each rule  $C$  time depends on the fractal dimension of the data or as much as user specifies. In this case we scan database  $L_R$ .  $C$  in the worst case.

**Space needed:** If all scaled up association rules would satisfy the *minimum-confidence*, then in the worst case with an average length of  $L_A$  for all discovered rules we need  $L_A/|R|.C$  more space comparing to a-priori algorithm which is negligible comparing to the whole dataset.

## 6. RESULT

The notion of time-invariant sequential patterns has been investigated in the context of Critical Care domain. The database is a collection of two different sets of patients from King Drew Medical Center (for patient going to Intensive Care Unite mainly because of accident, gun shots and/or injuries) and Harbor UCLA Hospital (mainly for senior citizens). Our database has collected during past 15 years. We applied our test only on the selected adequately monitored patients. In addition, data has been considered only after the first surgery as the data during the surgery is not valid due to the high hemorrhage of the patient.

Our implementation is in *MATLAB* programming language and has been tested on Pentium III processor with 384 MB RAM. This study is a follow up on work by Adibi et al [1] in which a complete decision support system designed in Lisp language under Apple/McIntosh platform [15]. We do not address the feature selection problem here and we follow the guideline provided by [15].

### 6.1 Critical Care Domain

Time-invariant or self-similar sequential association rules play an important role in the context of Critical Care since time is a crucial factor in Critical Care. For instance, admission time, visit time, surgery time, treatment time etc. are examples of association of time and patient care in Critical Care unit.

Our approach is based on the well studied concept that irrespective of multiply of superficial clinical manifestations, the patient dies of physiological alternations that can be identified, and prevented. Shoemaker et al showed that the temporal patterns of postoperative survivors were found to be different from those non-survivors despite the wide variety of illness and operational. [15, 16]

The survivor and non-survivor patterns and their importance of oxygen transport pattern were confirmed by independent investigations [15, 16]. In addition, it has been showed that the increased delivered oxygen ( $Do_2$ ) and consumed oxygen ( $Vo_2$ ) patterns of early postoperative survivors are clearly separate from the relatively normal values of non-survivors [15, 16]. Similarly, in other etiologic types of shock the survivor patterns are higher than those of the non-survivors at comparable time periods. However the main question of such protocol is to find under which circumstances a patient states moves form survivor to

**Table 3: Discovered Rules in Critical Care Database**

Survivors	Pattern 1	Pattern2	Pattern 3
<b>Support</b>	188	381	358
<b>Confidence</b>	.20	.41	.39

Non-Survivors	Pattern 1	Pattern2	Pattern 3
<b>Support</b>	223	389	365
<b>Confidence</b>	.23	.40	.37

non/survivor and to find which patterns has been repeated in survivor and non-survivor patients.

The physiology of postoperative and years of study in this field [16] shows distinguishes property in  $Do2/Vo2$  diagram in first 8 hours. There are two major patterns in survivors or non-survivors plot. After the first 8 hours it would be a significant difference in survivors and non-survivors pattern.

## 6.2 Experimental Result

For the purpose of the validating of our method, we conducted a multi-step experiment on our current database as the following:

1. First we pick all adequately monitored patients from survivors and non-survivors groups.
2. We applied the a-priori like algorithm on this set. We set  $G_{min}$  to 1 hour and  $G_{max}$  to 8 hours after surgery.
3. We scaled up the discovered rules by  $K$  equal to 24 and 96 hours after surgery for  $G_{max}$ .

The result is interesting and shows rules with a low *support* and high *confidence*, which did not come up in the first step, will be discovered. The main idea is to find the effectiveness of treatment in increasing the probability of patient as being a survivor at the end of the procedure by measuring the ratio of  $Vo2/Do2$ . We show the trend of this probability with 1: increases, -1: decreased, and 0: unchanged.

The following are the list of some interesting patterns were for we found for  $G_{max} = 8$  hours and  $G_{min} = 1$  hour :

- Pattern 1.  $P = \langle [I]^*, A, [I]^*, 1 \rangle$  indicate patient response to a given treatment
- Pattern 2.  $P = \langle [I]^*, A, [I]^*, 0 \rangle$  indicate no change in patient condition after giving a treatment
- Pattern 3.  $P = \langle [I]^*, A, [I]^*, -1 \rangle$  indicate no response from the patient to a given treatment which  $A$  is the treatment.

We scale up such rule and apply to survivors and non-survivors for  $K = 3$  and 12. the observation was that if the treatment applied to patient was in the goal of bringing up the  $Do2/Vo2$  it would save patient live more probably. When any treatment, which has the capability to increase the level of delivered Oxygen to the patient, is applied late the response of the patient also has been late and sometimes also is too late to recover.

The discovered rule might not consider as hard-to-find rules or hidden rules. There are a huge set of discovered rules. However we only discuss those which are easier to interpret and they are interesting for physicians and health care providers. Even though these results are in early stage but they are encouraging and we would like to investigate more undiscovered rules and apply to other large databases.

## 7. CONCLUSION AND FUTURE WORK

Despite the broad range of research on sequential association rules, they could not easily discover rules with low *support* and high *confidence*. We refer to this series of rules as *interesting rules* that are important in a relatively broad range of application in science, technology and medicine. In this paper we provide a fairly simple but powerful formalism to extend a pool of discovered rules to capture interesting rules for a specific databases with unique characteristics. The information and complexity embedded in these collections are hierarchical, they are self-similar, contain self-similar structures and have been generated through recurrent processes. We introduced time-invariant sequential association rules as those rules which if extend in time dimension explore more knowledge from data. Since time-invariant sequential association rules are capable to discover a relatively hard-to-find association rules, they may extend to all database which shows partially self-similarity.

This research is in early stage. As future work we would like to continue our findings in Critical Care domain and extend this research to multi dimensional sequential association rules. In addition we are in the process of discovering rules in strongly self-similar time series such as synthetic data or network databases. The next step for Critical Care domain is considering the dosage of in discovery loop. In addition we would like to extend this work when the model shows self-similar structure only in a limited range of structure scale.

## 8. ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation under grant: NSF-IDM 9529615. We would like to thank Dr. William C. Shoemaker for his tremendous effort on data analysis, rule interpretation and his advise on Critical Care domain.

## 9. REFERENCE

- [1] Adibi, J., Patil, R. S., and Shoemaker W. C. *A Perception-Action model for Critical Care*. in *American Medical Informatics Association*. (1997). Nashville, Tennessee.
- [2] Adibi, J., Shen, W-M. *Self Similar Layered Hidden Markov Model*. in *5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)*. (2001). Freiburg, Germany.
- [3] Agrawal, R., Lin, K. I., Sawheny, H. R., and Shim, K. *Fast similarity search in the presence of noise, scaling and translation in time series databases*. in *VLDB*. (1995). Zurich, Switzerland.

- [4] Agrawal, R., Srikant, R. *Mining sequential patterns. in the Int'l Conference on Data Engineering (ICDE)*. (1995). Taipei, Taiwan.
- [5] Barbara, D. *Chaotic Mining: Knowledge discovery using the fractal dimension. in ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*. (1999). Philadelphia, USA,.
- [6] Cohen, E., Datar, D., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J. and Yang, C. *Finding interesting associations without support pruning. in 16th Annual IEEE Conference on Data Engineering (ICDE)*. (2000).
- [7] Das, G., Lin, K., Mannila, H., Renganathan, G. and Smyth, P. *Rule discovery for time series. in Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95)*. (1995). New York City, New York.
- [8] Dietterich, T.G., Michalski, R. S., *Discovering patterns in sequences of events. Artificial Intelligence*, (1985). **25**: p. 187-232.
- [9] Faloutsos, C., Seeger, B., Traina, A. and Traina Jr., C. *Spatial Join selectivity using power law. in SIGMOD*. (2000). Dallas, TX.
- [10] Han, J., Pei, J. and Yin, Y. *Mining frequent patterns without candidate generation. in ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00)*. (2000). Dallas, TX.
- [11] Higuchi, T., *Approach to an irregular time series on the basis of the fractal theory. Physica D*, (1998). **31**.
- [12] Laird, P. *Identifying and using patterns in sequential data. in Algorithmic Learning theory, 4th International Workshop*. (1993): Springer-Verlag.
- [13] Mandelbrot, B., Van Ness, J. W., *Brownian motion fractional noises and applications. SIAM review*, (1968). **422**(437).
- [14] Mannila, H., Toivonen, H., and Verkamo, A. I. *Discovering generalized episodes using minimal occurrences. in Second International Conference on Knowledge Discovery and Data Mining*. (1996). Portland, Oregon.
- [15] Patil, R.S., Adibi, J., and Shoemaker W. C., *Application of an Artificial Intelligence program to therapy of high risk surgical patients. New Horizons The Science and Practice of Acute Medicine*, (1996). **4**(4): p. 541-550.
- [16] Shoemaker, W.C.S., Patil, R. S., Adibi, J. et al., *Early physiologic patterns in acute illness and accidents: Toward a concept of circulatory dysfunction and shock based on invasive and noninvasive homodynamic monitoring. New Horizons The Science and Practice of Acute Medicine*, (1996). **4**(4): p. 395-412.
- [17] Silverstein, C., Brin, S., Motwani, R., and Ullman, J.D. *Scalable Techniques for Mining Causal Structures. In Proceedings of the. in 24th International Conference on Very Large Data Bases*. (1998).
- [18] Srikant, R., Agrawal, R. *Mining sequential patterns: Generalizations and performance improvements. in Fifth Int'l Conference on Extending Database Technology (EDBT)*. (1996). Avignon, France.
- [19] Traina, C., Traina, A., Wu, L., and Faloutsos, C. *Fast feature selection using the fractal dimension. in XV Brazilian Symposium on Databases (SBBD)*. (2000). Paraiba, Brazil.
- [20] Tsong-Li Wang, J., Chim, G., Marr, T.G., Shapiro, B. A., Shasha, D., and Zhang, K. *Combinatorial pattern discovery for scientific data: Some preliminary results. in SIGMOD*. (1994): ACM Press.
- [21] Wang, T., Tan, J. *Incremental discovery of sequential patterns. in Workshop on Research Issues on Data Mining and Knowledge Discovery, in cooperation with ACM-SIGMOD*. (1996). Montreal, Canada.