An Algorithm for Generating Non-Redundant Sequential Rules for Medical Time Series Data

K. Pazhanikumar
Assistant Professor, Department of Computer Science
S.T.Hindu College
Nagercoil, India
kpk_73@yahoo.co.in

Dr. S. Arumugaperumal
Head, Department of Computer Science
S.T.Hindu College
Nagercoil, India
visvenk@yahoo.co.in

Abstract— In this paper, an algorithm for generating non-redundant sequential rules for the medical time series data is designed. This study is the continuation of my previous study titled "An Algorithm for Mining Closed Weighted Sequential Patterns with Flexing Time Interval for Medical Time Series Data" [25]. In my previous work, the sequence weight for each sequence was calculated based on the time interval between the itemsets.Subsequently, the candidate sequences were generated with flexible time intervals initially. The next step was, computation of frequent sequential patterns with the aid of proposed support measure. Next the frequent sequential patterns were subjected to closure checking process which leads to filter the closed sequential patterns with flexible time intervals. Finally, the methodology produced with necessary sequential patterns was proved. This methodology constructed closed sequential patterns which was 23.2% lesser than the sequential patterns. In this study, the sequential rules are generated based on the calculation of confidence value of the rule from the closed sequential pattern. Once the closed sequential rules are generated which are subjected to non-redundant checking process, that leads to produce the final set of non-redundant weighted closed sequential rules with flexible time intervals. This study produces non-redundant sequential rules which is 172.37% lesser than sequential rules.

Self citation

Keywords*-sequential rules; frequent itemsets; mining;*

*****

I. Introduction

The development of information technology (IT) has improved storage and retrieval problems of data, such as science data, medical data, population data, financial data, and market data. How to find useful information from those data has become the most important issue [1]**.** In early 1990s, knowledge discovery from data (KDD) term was used with the aim of knowledge extraction from database [2]. Data mining was originally considered as synonym of KDD. Data mining is the nontrivial extraction of implicit, formerly unknown, and potentially valuable information from data. Recent researches have shown that application of data mining in several fields is growing such as CRM, education, clinical medicine, financial fraud detection, intrusion detection and genetic data analyzing. The application of data mining in medicine has become a great issue. Recently, application of data mining in medicine and healthcare is most widely used by data mining developers and academic researchers compared to the other fields. The rapid growth of medical data mining in the recent years represents the kick-off medical data mining.

MEDICAL databases have accumulated large amounts of information about patients and their clinical conditions. Relationships and patterns hidden in this data can provide new medical knowledge as has been proved in a number of medical data mining applications. In the field of data mining, one of the most popular set of techniques for discovering temporal relations between events in discrete time series is sequential pattern mining, which consists of finding sequences of events that appear frequently in a sequence database**.** Several main streams of pattern mining, such as time-series mining and sequential pattern mining, have drawn much attention over the past decade. Time-series mining methods incorporate concrete notions of time in the process of finding patterns. Previous studies in this field include searching similar patterns in time-series databases**.** Mining sequential patterns from a sequence database may generate many sequential patterns especially when the support thresholds are low. In [3] they introduce the idea of data projection and develop the FreeSpan algorithm to recursively mine sequential patterns. In [4] they propose the PrefixSpan algorithm for mining long sequential patterns in large sequence databases. It continuously mines the patterns from projected databases, which speed up the candidate subsequence generation [5].

Although a complete set of frequent patterns discovered are informative, the number of these patterns may be overwhelming. The concept of mining closed patterns has been proposed to avoid unnecessary frequent patterns while preserving the same information. A frequent pattern is closed if it has no super-pattern with the same support. Generally speaking, the algorithms of mining closed patterns are more efficient than those of mining frequent patterns [6, 7]. Moreover, the closed patterns mined can be used to generate a complete set of frequent patterns. Many methods of mining frequent closed patterns have been proposed, such as A-Close [6], CLOSET [7], CLOSET+ [8], CloSpan [9], BIDE [10], and CHARM [11]. In order to meet the dynamic characteristic of online data streams [12] proposed an algorithm, called New Moment, to mine closed patterns. The NewMoment algorithm uses an effective bit-sequence representation to simplify the support calculation, and hence, results in less memory and execution time. If the patterns are discovered with flexible number of gaps between items then more interesting sequential patterns can be found

For a sequence or a sequential pattern, not only the generation order of data elements but also their generation times and time intervals are important. Therefore, for sequential pattern mining, the time-interval information of data elements can help to get more valuable sequential patterns. In [13] and [14], several sequential pattern mining algorithms have been presented which consider a time-interval between two successive items in a sequential pattern. However, they simply consider a time-interval between two successive data elements as an item, and thus they are unable to get weighed sequential patterns considering different weights of sequences in a sequence database. If the importance of sequences in a sequence database is differentiated based on the time-intervals in the sequences, more interesting sequential patterns can be found [15]. However, the patterns may be irrelevant and a sequence of events that appear frequently in a database is thus insufficient for predicting events. Therefore, the sequential rule mining problem is proposed [16-23] and sequential rules are used to allow better prediction. Sequential rules express the relationships between sequential patterns from a sequence database [21, 22] and can be considered as a natural extension of original sequential patterns, just as association rules are a natural extension of frequent itemsets. Using sequential rules, the series of events that usually occurs after a series of previous ones can be predicted. Sequential rules are rather simple, but their information has many important implications, which can be used for decision-making, management and behavior analysis. Compared with sequential patterns, sequential rules can help users better understand the chronological order of the sequences present in a sequence database. However, generating a full set of sequential rules is very costly, even for a sparse dataset. In addition, a lot of low-quality rules that are almost meaningless are generated, to solve this problem non-redundant sequential rules is constructed [a].

II. CLOSED WEIGHTED SEQUENTIAL PATTERNS WITH FLEXING TIME INTERVAL FOR MEDICAL TIME SERIES DATA

In my previous study, the time series medical data was utilized to construct the sequential patterns. Using direct database makes the mining algorithm more complex since initially, the diseases from the patient medical data were transformed into symbolic representation of sequences and the time duration of the diseases are denoted as time stamp of the sequence. Once the sequential times series data was transformed from the original medical data, the next step was to calculate the weight value of each sequence based on the strength and time interval weight. The strength of the sequence was depends on number of diseases presented in the each sequence. In another way, the candidate item sets were generated with possible intervals. The proposed support measure was calculated based on the weight of the sequence, number of occurrences of the itemsets and number of sequences has the itemsets. Once the support of the candidate itemsets were calculated the frequent weighted flexible sequential patterns (FWFSP) were filtered by the minimum support value (min-sup) subsequently the FWSP were subjected to closure checking process which leads to attain the closed weighted sequential patterns (CWFSP). The overall architecture of the proposed method is presented in the following figure 1.

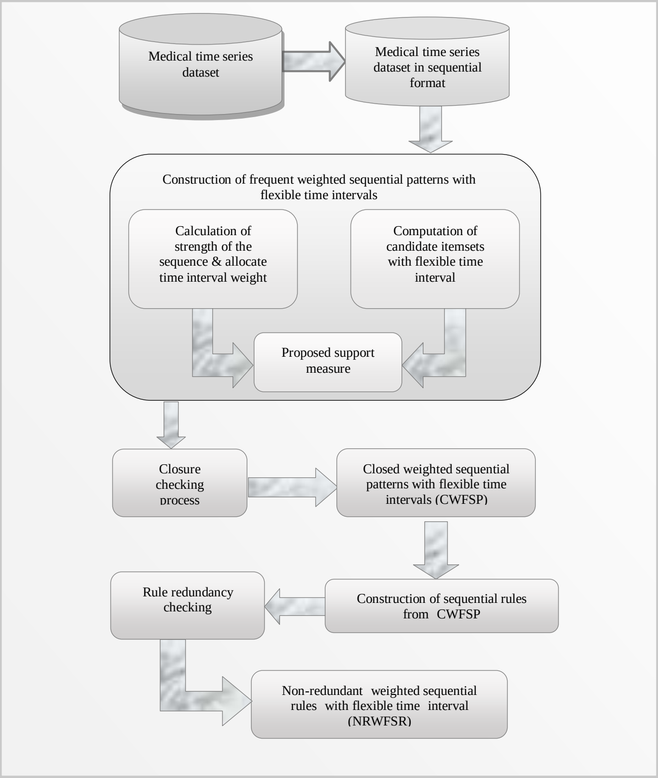
Medical time series dataset
Medical time series dataset in sequential format
Construction of frequent weighted sequential patterns with flexible time intervals
Calculation of strength of the sequence & allocate time interval weight
Computation of candidate itemsets with flexible time interval
Proposed support measure
Closure checking process
Closed weighted sequential patterns with flexible time intervals (CWFSP)
Rule redundancy checking
Construction of sequential rules from CWFSP
Non-redundant weighted sequential rules with flexible time interval (NRWFSR)

Figure 1: Overall architecture of the proposed algorithm

2.1 Patient's time series medical dataset

The medical time series database contains the set of patient id $DB = \{P_i\}$ where $1 \leq i \geq N$ each patient has set of diseases with time stamps $P_i = \{t_i\}$ where $1 \leq i \geq k$ and the value of k may be varying from one patient to another. Each time stamp t_i contains set of diseases and its status of the diseases $t_i = \{d_i^{(\pm)}\}$. The following table 1 represents the sample medical database, which contains patient id, checking date of the patients (time stamp), the set of diseases and its status of the diseases.

Patient id	Checking date	Status of the diseases
110	2/2/11	anxiety+
	4/2/11	anxiety+, biopsy+, cholera+
	5/2/11	anxiety+, biopsy-, cholera+
	6/2/11	anxiety-, cholera-, dysthymia+
120	21/4/11	anxiety+, dysthymia+
	22/4/11	anxiety-, dysthymia-, cholera+
	24/4/11	biopsy+, cholera+
	25/4/11	biopsy-, cholera-, alpha+, epilepsy+

Table 1: Sample medical time series database

2.2 Sequential time series data format

With the intention of mining the patterns from the medical dataset in the above format is take much computation complexity and requires more running time. To solve such problem, we pre-processed the medical data into set of sequences of diseases with respect to patient id. The sequences are sorting with respect to the ascending order of the time stamp of the diseases. In addition, the status of the diseases is represented with every disease. Consider the patient id 110, who affected by the disease anxiety+ while his/her first day of checking. The symbolic representation of the disease is represented by ‘a^+’. In the next checkup process revealed as “anxiety+, biopsy+, cholera+”, which indicates that two diseases are additionally affected followed by the initial disease anxiety+ which, can be represented as (a^+,b^+,c^+) in the symbolic representation. The symbol ‘+’ indicates that the diseases is in active stage and the symbol ‘-’ indicates that disease become cured. The brackets are used to present the diseases that are taken in a single day. For brevity, the brackets are omitted when the revealed disease become single in a day of checking. The following table 2 represents the pre-processed dataset of the medical dataset, which is represented in the above table 1.

Patient id	**Sequence**	**Time stamp**
110	< a^+, (a^+,b^+,c^+), (a^+,b^-,c^+), (a^-,c^-,d^+)>	(2, 4, 5, 6)
120	<(a^+,d^+), (a^-,d^-,c^+), (b^+,c^+), (b^-,c^- ,a^+,e^+)>	(21, 22, 24, 26)

Table 2: Sample medical time series sequential database

2.3 Sequence strength and weight calculation

For a sequence or a sequential pattern, not only the generation order of data elements but also their generation times and time intervals are important. By giving more importance to the time-interval information of data elements that provide more valuable sequential patterns. It is expressed in [24].

Consider the diseases sequences of the two patients (S_1, S_2) which contain the same diseases $S_1 = \{d_1^1, d_2^1, d_3^1, d_4^1\} \& S_2 = \{d_1^2, d_2^2, d_3^2, d_4^2\}$ and the order of the diseases also same in both patients but the time intervals between the diseases of the both patient are different $t_i^1 \neq t_i^2$. The sequences are looking to be the same when only considers about the order of the diseases in both sequences. In real case, both the sequences are totally differenct if we consentate on the time interval between the diseases (itemsets). However, in real world condition the importance of the minimum time intervals sequences are treated as more valuable than the maximum time intervals sequences. Since, in this paper, the importance of the minimum interval sequences are differentated by giving the more weight values to minimum time interval than the maximum time interval. This reflect the total weight value of the sequences. The weight value of the sequences are calculated based on the number of items present in the pair of itemsets and the time interval between possible pair itemsets in the sequences. The following table 3 represents the possible pairs of itemsets and its time interval between the itemsets of the sequence 110 from the above table 2.

1st itemset	2nd itemset	Time interval (TI)	Strength (ST)
a^+	(a^+,b^+,c^+)	2	(1*3)=3
a^+	(a^+,b^-,c^+)	3	(1*3)=3
a^+	(a^-,b^-,c^+)	4	(1*3)=3
(a^+,b^+,c^+)	(a^+,b^-,c^+)	1	(3*3)=9
(a^+,b^+,c^+)	(a^-,b^-,c^+)	2	(3*3)=9
(a^+,b^-,c^+)	(a^-,b^-,c^+)	1	(3*3)=9

Table 3: possible pairs of itemsets and its time interval between the itemsets of the sequence 110

The wieght value of the sequence is differentiated based on the wieght value of the time interval, the minimum time interval is more important than the maximum time interval. In this paper, the available time intervals are 1, 2, 3 and 4. The weight value of the time interval is assigned by the user based on their needs. In this paper the weight values as 0.4, 0.25, 0.2, 0.15 are allocated for the time intervals 1, 2, 3, 4 respectively.

The calcuation of weight value of a sequence is done by following equations (1) and (2) where the ST_{ij} indicates the strength of the pair of ‘ith’ itemset and ‘jth’ itemset of a sequence and TI_{ij} represents the time interval between the pair of ‘ith’ itemset to ‘jth’ itemset of a sequence.

$$W(S) = \frac{1}{N} \sum_{i=1}^{l=1} \sum_{j=i+1}^{l} w(TI_{ij}) \times ST_{ij} \qquad \dots (1)$$

$$N = \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} STij \qquad \dots (2)$$

After the calculation of the weight value of the sequence, the sequence that contains the itemsets with minimum time interval has the weight value of more than the sequence contains the itemsets with maximum time interval.

2.4 Candidate itemeset generation flexible time interval

To generate the flexible time interval sequential patterns, initially the sequential patterns are derived from the symbolic time series sequential medical database. The itemsets within the brackets considered as zero time interval. Initially, possible maximum length of 0 time interval itemsets are derived subsequently the possible time intervals are projected between the items in the itemsets.

(a) Candidate itemset generation with zero time interval

The itemsets within the brackets represents the zero time interval which indicates the symtoms or diseases are identified in a single day. The zero time interval itemsets are generated using the items with in the brackets. The maximum length of zero time interval candidate itemsets are generated. In normal candidate itemset, the possible combinations are generated based on the availabe items present in the database which generates more itemset which leads to computation complexity and time complexity. But in our proposed method, we only consider the zero time interval candidate itemset at possible length which covers all the itemsets in the database and also it avoids the unncessary candidate itemset which helps algorithm reduce the computation complexity. Consider the follwing table 4 which represents two length zero time interval candidate itemsets derived from the table 2.

(a^+,b^+)	(a^+,c^+)	(a^-,d^+)	(b^-,c^-)
(a^+,b^-)	(a^-,c^+)	(a^-,d^-)	(c^+,d^-)
(a^-,b^-)	(a^-,c^-)	(b^+,c^+)	(c^-,d^+)
(a^-,b^+)	(a^+,d^+)	(b^-,c^+)	(d^+,f^+)

Table 4: represents the sample 2 length zero time interval itemsets

(b) Projection of time intervals with zero time interval itemsets

Once the zero time interval candidate itemsets are generated, the next step is projection of possible time intervals to each itemset. After the time interval projection, the candidate itemsets are called as flexible time interval sequential patterns. in this paper, from the above table 4, the maximum time interval is four since for single itemset there are four possible patterns are generated. Consider the two length zero time interval itmeset (a^+,b^+) which is transformed in to the four patterns such as (a^+[1]b^+), (a^+[2]b^+), (a^+[3]b^+), (a^+[4]b^+). Likewise, for every itemsets, the time intervals are projected to compute the flexible time interval sequential patterns.

2.5 Support calculation

Once the flexible time interval sequential patterns are derived subsequently we calculate frequency of the through our proposed support measure which deals with the time intervals, number of occurances in the database and weight value of the sequence. Our proposed support calculation can help the sequential pattern algorithm to select the frequent fleixble time interval sequential patterns (FFSP). the FFSP has the support value which are greater than the user given minmum support (min-sup) value.

The calcuation of support of a sequence is done by following equations (3) where $cnt\left(P_i^{TI}\right)$ represents the count value of the ‘ith ’ pattern in a time interval ‘ *TI* ’ and $\max\left(cnt\left(P^{TI}\right)\right)$ represents the maximum count value of the patttern in a time interval ‘ *TI* ’. The symbol $n\left(P_i^{TI}\right)$ that indicates number of sequences that contains the ‘ith ’ pattern in a time interval ‘ *TI* ’ and $N(S)$represents the total number of sequences in the sequential database.

$$\sup\left(P_i^{TI}\right)=\frac{cnt\left(P_i^{TI}\right)}{\max\left(cnt\left(P^{TI}\right)\right)}\times\frac{n\left(P_i^{TI}\right)}{N(S)}\times\frac{\sum_{S\in P_i^{TI}}W(S)}{\sum W(S)} \quad \ldots (3)$$

once the support value of the flexible time interval sequential patterns is calculated then the frequent flexible time interval sequential pattens are filtered. The patterns which have support value greater than the user defined minimum support then that pattern is treated as frequent patterns.

2.6 Clousure checking (Clossed weighted sequential patterns with flexible time interval)

Although a complete set of frequent flexible time interval patterns discovered are informative, the number of these patterns may be overwhelming. The concept of mining closed patterns is utilized in this paper to avoid unnecessary frequent flexible time interval sequential patterns (FFTSP) while preserving the same information. A frequent pattern is closed if it has no super-pattern with the same support. In this paper, we not only consider the itemsets for closure checking process, we also mining the patterns with time intervals. According to that we have also using the following condition with the existing closure cheking process. If the FFSTP contains the same itemsets and support with differenet time intervals then we consider the mininmum time interval pattern as a closed time interval pattern.

Case I: Consider the following sequences and its support values (a^+[1]b^+)=2, (a^+[2]b^+)=2, (a^+[3]b^+)=2, (a^+[4]b^+)=2. From the above set of sequencs, we only consider the sequence (a^+[1]b^+)=2 as a closed patterns since all the other time intervals of the sequences also having the same support.

Case II: Consider the following sequences and its support values (a^+[1]b^+)=2, (a^+[2]b^+)=2, (a^+[3]b^+)=2, (a^+[4]b^+)=4. From the above set of sequencs, we need to consider the sequences (a^+[1]b^+)=2 and (a^+[4]b^+)=4as a closed patterns from the above set of sequences since both sequences represents different informations

Every sequential patterns were subjected to closure checking process, which leads to return the weighted closed flexible time interval sequential patterns (WCFSP).

2.7 Rules Construction and Redundancy Checking

However, the discovered patterns may be irrelevant and a sequence of events that appear frequently in a database is thus insufficient for predicting events. To solve this problem, in this paper we incorporate the sequential rule mining which are also used to allow better predictions. Sequential rules express the relationships between sequential patterns from a sequence database [28, 29] and can be considered as a natural extension of original sequential patterns, just as association rules are a natural extension of frequent itemsets [30].

$$conf\left(X^{TI}\rightarrow Y\right)=\frac{\sup\left(X^{TI}\rightarrow Y\right)}{\sup(X)} \quad \ldots (4)$$

$$conf\left(X^{TI}\rightarrow Y^{TI}\rightarrow Z\right)=\frac{\sup\left(X^{TI}\rightarrow Y^{TI}\rightarrow Z\right)}{\sup\left(X^{TI}\rightarrow Y^{TI}\right)} \quad \ldots (5)$$

The above equation (4) represents the confidence vlaue of two length sequential pattern and the equaiton (5) helps to calculate the confidence vlaue of three length sequential pattern.

Using sequential rules, the series of events that usually occurs after a series of previous ones can be predicted. Sequential rules has many important implications, which can be used for decision-making, management, and behavior analysis. Compared with sequential patterns, sequential rules can help users better understand the chronological order of the sequences present in a sequence database. However, generating a full set of sequential rules is very costly, even for a sparse dataset. In addition, many low-quality rules that are almost meaningless rules are generated [26]. To solve this problem, in this paper, some conditions based on the item sets in the rules and its time intervals to avoid the redundancy sequential rules are derived. The rule, which satisfies the following conditions, is considered as non-redundant sequential rules [27].

Consider the set of rules $R_X = pre_X \rightarrow post_Y$ and $R_Y = pre_Y \rightarrow post_Y$ in which the rule R_Y infers R_X if the following conditions are satisfied

(i) $pre_Y \subseteq pre_X$

(ii) $pre_X + post_X \subseteq pre_Y + post_Y$

(iii) $\sup R_X = \sup R_Y$

(iv) $conf\ R_X = conf\ R_Y$

III. RESULT AND DISCUSSION

The experimental result of the proposed technique for mining closed weighted sequential patterns with flexing time interval and non-redundant sequential rules for medical time series data is described in section. In this paper, we evaluate our proposed methodology in terms of number of sequential patterns and sequential rules generated.

3.1 Experimental design

The proposed algorithm of hybrid bi-objective optimization algorithm is programmed using JAVA version jdk1.7 with NETBEANS 7.3 IDE with cloud sim version 2.1.1. The experimentation has been carried out using the synthetic dataset with i3 processor PC machine with 4GB main memory and 32-bit version of windows 7 operating system. We generate the synthetic datasets, which consists of the following attributes such as "Patient ID", "Disease Name with its status" and "Date". In this paper, our proposed methodology is evaluated based on number of non-redundant flexible time interval sequential rules by varying the following factors such as minimum support, number of input data, minimum confidence.

3.2 Performance evaluation based on number of data

In this section, the number of sequential rules and number redundant sequential rules are evaluated by varying the number of input data by making the minimum support and minimum confidence value as constant.

The following figure 3 represents the evaluation of number of sequential rules and non-redundant sequential rules for various number of input data by making the minimum support value and minimum confidence value as 0.5. By evaluating the following figure 3, we conclude that the number of flexible time interval sequential rules and number of non-redundant flexible sequential rules also increased gradually when the number of input data increased. The number of non-redundant flexible sequential rules are always lesser than the number of sequential patterns. The number of closed sequential patterns are always lesser than the number of sequential patterns. The evaluation of follwoing figure 3 represents the maximum difference between number of sequential rules to the number of non-redundant sequential rules is happened as 185.31% at number of input data 4000 and minimum difference is obatined as 164.92%at number of input data 2000 and the overall average difference is 23.73%.

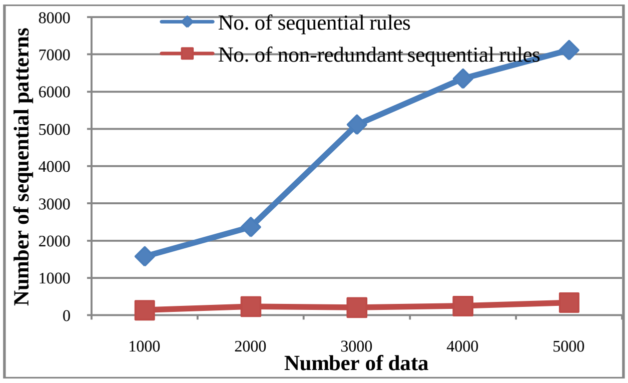

Figure 3: Evaluation of sequential rules and non-redundant sequential rules based on number of input data

3.3 Performance evluation based on thereshold vlaues

In this section, the number of sequential rules and number redundant sequential rules are evaluated by varying the minimum confidence meanwhile making the minimum support and number of input data as fixed.

In this section, the proposed methodology is evaluated based on minimum confidence. The following figure 5 represents the evaluation of number of sequential rules and non-redundant sequential rules for various number of input data by making the minimum support value and minimum confidence value as 0.5. By evaluating the following figure 5, the number of flexible time interval sequential rules and number of non-redundant flexible sequential rules also increased gradually when the number of input data increased. The number of non-redundant flexible sequential rules are always lesser than the number of sequential patterns. The evaluation of follwoing figure 5 represents the maximum difference between number of sequential rules to the number of non-redundant sequential rules is happened as 158.21% for the minimum support 0.7 and minimum difference is obatined as 167.92% for the minimum support 0.3 and the overall average difference is 168.98%.

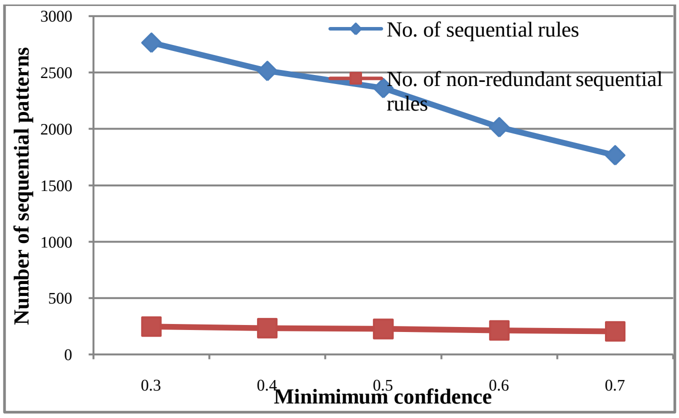

Figure 3: Evaluation of sequential rules and non-redundant sequential rules based on minimum confidence

IV. CONCLUSION

In this paper, an algorithm has been proposed for mining the non-redundant closed weighted sequential rules with flexible time intervals for the medical time series data. Initially, the sequence weight for each sequence was calculated based on the time interval between the itemsets subsequently the candidate sequences were generated with flexible time intervals. Then, computation of frequent sequential patterns was done with the aid of proposed support measure. The obtained patterns are subjected to closure checking process. Then sequential rules are derived from the closed sequential patterns subsequently the constructed rules are subjected to redundancy checking process. The final rules are named as weighted non-redundant sequential rules with flexible time intervals are revealed by the proposed methodology. Finally, the proposed methodology produces necessary sequential patterns and sequential rules, is proved. The proposed methodology produces non-redundant sequential rules which is 172.37% lesser than sequential rules.

REFERENCES

[1] R.J. Kuo, R.J. Kuo, C.Y. Liu, "Integration ofK-means algorithm and AprioriSome algorithm for fuzzy sequential pattern mining", Journal of Applied Soft Computing, vol.9, pp.85-93, 2009.

[2] Piatetsky-Shapiro, G., & Frawley, W. (1991). Knowledge discovery in databases.California: AAAI/MIT Press

[3] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M. (2000). FreeSpan: Frequent pattern-projected sequential pattern mining. InProceedings of the 6th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 355–359).

[4] Pei, J., Han, J., Mortazavi-Asl, B., & Pinto, H. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. InProceedings of the 17th international conference on data engineering(pp. 215–224).

[5] Kim, C., Lim, J., Ng, R. T., & Shim, K. (2007). SQUIRE: Sequential pattern mining with quantities.The Journal of Systems and Software, 80(10), 1726–1745.

[6] Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. InProceeding of the 7th international conference on database theory(pp.398–416).

[7] Pei, J., Han, J., & Mao, R. (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. InACM SIGMOD workshop on research issues in data mining and knowledge discovery(pp. 21–30).

[8] Wang, J., Han, J., & Pei, J. (2003). CLOSET+: Searching for the best strategies for mining frequent closed itemsets. InProceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 236–245).

[9] Yan, X., Han, J., & Afshar, R. (2003). CloSpan: Mining closed sequential patterns in large datasets. InProceedings of the SIAM international conference on data mining (pp. 166–177).

[10] Wang, J., & Han, J. (2004). BIDE: Efficient mining of frequent closed sequences. In Proceedings of the 20th international conference on data engineering(pp. 79–90).

[11] Zaki, M. J., & Hsiao, C. (2005). Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering, 17(14), 462–478.

[12] Li, H. F., Ho, C. C., & Lee, S. Y. (2009). Incremental updates of closed frequent itemsets over continuous data streams. Expert Systems with Applications, 36(2P1), 2451–2458.

[13] Y.-L. Chen, M.-C. Chiang, M.-T. Ko, Discovering fuzzy time-interval sequential patterns in sequence databases, IEEE Transactions on Systems Man and Cybernetics – Part B: Cybernetics 35 (5) (2005) 959–972.

[14] Y.-L. Chen, T.C.-H. Huang, Discovering time-interval sequential patterns in sequence databases, Expert Systems with Applications 25 (1) (2003) 343–354.

[15] Joong Hyuk Chang, "Mining weighted sequential patterns in a sequence database with a time-interval weight", Journal of Knowledge-Based Systems, vol. 24, pp. 1–9, 2011.

[16] Fournier-viger, P., Nkambou, R., Tseng, V.S., 2011. RuleGrowth: mining sequential rules common to several sequences by Pattern-Growth. In: Proceedings of the SAC'11, TaiChung, Taiwan.

[17] Fournier-viger, P., Faghihi, U., Nkambou, R., Nguifo, E.M., 2012a. CMRules: an efficient algorithm for mining sequential rules common to several sequences. Knowledge-based Syst. 25 (1), 63–76.

[18] Fournier-viger, P., Wu, C.-W., Tseng, V.S., Nkambou, R., 2012b. Mining sequential rule common to several sequences with the window size constraint. In: Proceedings of the 25th Canadian International Conference on Artificial Intelligence (AI 2012), Lecture Notes in Artificial Intelligence, vol. 7310, Springer, 299–304.

[19] Harms, S.K., Deogun, J.S., 2004. Sequential association rule mining with time lags. J. Intel. Inf. Syst. 22 (1), 7–22.

[20] Lo, D., Khoo, S.C., Wong, L., 2009. Non-redundant sequential rules-theory and algorithm. Inf. Syst. 34 (4–5), 438–453.

[21] Spiliopoulou, M., 1999. Managing interesting rules in sequence mining. In: Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, pp. 554–560.

[22] Van, T.-T., Vo, B., Le, B., 2011. Mining sequential rules based on prefix-tree. Studies in Computational Intelligence, 351. Springer, pp. 147–156.

[23] Zang, H., Xu, Y., Li, Y., 2010. Non-redundant sequential association rule mining and application in recommender systems. In: Proceedings of the 2010 IEEE/WIC/ ACM International Conference on Web Intelligence and Intelligent Agent Technology, DC, USA, vol. 3, pp. 292–295.

[24] Joong Hyuk Chang, "Mining weighted sequential patterns in a sequence database with a time-interval weight", Knowledge-Based Systems, vol.24, pp. 1–9, 2011.

[25] K.Pazhanikumar,S.Arumugaperumal,"An algorithm for mining closed weighted sequential patterns with flexing time interval for medical time series data", IEEE Xplore on ICCCS (2015) 31–35.

[26] Thi-Thiet Pham, Jiawei Luo, Tzung-Pei Hong, Bay Vo, "An efficient method for mining non-redundant sequential rules using attributed prefix-trees", Journal of Engineering Applications of Artificial Intelligence, vol. 32, pp. 88–99, 2014.

[27] David Lo, Siau-Cheng Khoo, Limsoon Wong, "Non-redundant sequential rules—Theory and algorithm", Information Systems, vol. 34, pp. 438–453, 2009.

[28] Spiliopoulou, M., 1999. Managing interesting rules in sequence mining. In: Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, pp. 554–560.

[29] Van, T.-T., Vo, B., Le, B., 2011. Mining sequential rules based on prefix-tree. Studies in Computational Intelligence, 351. Springer, pp. 147–156.

[30] Srikant, R., Agrawal, R., 1996. Mining sequential patterns: Generalizations and performance improvements. In: Proceedings of the 5th International Conference on Extending Database Technology, pp. 3–17.