International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 32399-32406

http://www.ripublication.com

STM: A General Model for Mining Long and Closed Sequence in Time Series Databases

M. Parimala

Senior Assistant Professor, Department of Computer Applications,
M. Kumarasamy college of Engineering,
Karur, Tamilnadu, India.
Email: pariguns@gmail.com

Dr. N. Karthikeyan

Professor & Head, Science & Humanities,
SNS College of Engineering,
Coimbatore, Tamilnadu, India.

Dr. S. Sathiyabama

Assistant Professor of Computer Science,
Thiruvalluvar Govt Arts and Science College,
Rasipuram, Tamilnadu, India.

ABSTRACT

One of the major tasks involved in temporal data mining is mining of long and closed sequence which has received a greater deal of attention in numerous areas. One of the problems faced is the forecasting of natural calamities like the forecasting of earthquake. In this case, it is highly necessary to recognize all patterns of different sorts of documented events (i.e., the calamity involving with or without earthquake) which potentially designate the chance of an earthquake condition and can be easily differentiated also. The proposed work Sequential Task Miner (STM) is applied on a huge amount of temporal data that has successfully generated the sequence for all time sensitive data. STM did a repeated process of candidate-generation which mines Long and Closed Sequential Mining Variants in order to improve the efficiency of the mining process. Two boundaries min_iteration and max_interference are used to denote the minimum number of iterations that are needed for each segment and the maximum allowed interferences

between any two consecutive valid segments. Once these two metrics are satisfied, the longest and closed valid subsequence of a pattern is determined. The experiment done shows the good performance and scalability in large time series dataset.

Key Words: Long Sequence, Closed Sequence, Time series Database, Sequential Task Miner

1. INTRODUCTION

For mining long and closed sequences, a significant analysis not only made to discovers all the recurrent patterns but also measures the closed sequences for the reason that the application of closed sequences results in a more compact and efficient form of representation. Though, different methods have been presented for mining recurrent item sets, more efforts have not been put forth for mining recurrent item sets using long and closed sequential patterns. It is mainly because of the difficulty faced during the mining of closed sequences.

Different types of existing long and closed sequence mining algorithms preserve the results of the previously mined long and closed sequence candidates that could be used to prune the search space and ensure if a newly acknowledged sequence is capable of closing. In the current scenario, most of the closed sequence pattern mining algorithms have to retain the results of the existing mined closed sequences in memory. Certain other operations that have to be performed include closed sequence sub-pattern checking and closed sequence super-pattern checking.

The closed sequence sub-pattern checking sees to that if the newly identified closed pattern matches with an existing mined closed pattern whereas the task of closed sequence super-pattern checking identifies whether the newly identified closed pattern matches with an already mined closed pattern. It consumes more memory and requires huge search space while examining new patterns, which is generally the case when the support margin is small or the patterns grow to be long.

2. PROBLEM DEFINITION

In this section Sequential pattern mining algorithms mine frequent sub sequences, satisfying a mini_sup margin in a sequence database. However common long sequence includes a combinatorial integer of frequent subsequences. The results of mining therefore generate an unequal number of frequent subsequences for long patterns, which results in the utilization of both time and space.

One of the equivalent but more powerful solutions instead of mining the absolute set of frequent subsequences is presented that mines frequently closed sub sequences only i.e., persons surround no super sequence with equivalent support (i.e., occurrence frequency).

By exploring Pattern Miner, the proposed STM is developed for mining long and closed sequential patterns using time series dataset including 60000 records at Periodic Time Stamps. With this the STM, by applying long and closed sequences, produces less number of sequences being discovered when compared to the

conventional methods of mining. This is because not only the group of frequent subsequences but also their supports are derived easily from the results of mining.

The process involved in STM includes two phases namely,

Event phase - Candidate pruning approach.

Sequence phase - Integrate the process involved in both the intra-event and inter-event constraints.

2.1 Design considerations of STM

In STM, Item represents the value which is going to be assigned to an attribute. The value of the item includes an item like a letter of the alphabet i.e., a, b,..., z. Let A = $\{a_1, a_2, ...a_m\}$ be the set of all probable items included in the area of consideration in STM. Nonempty set represents an event that occurs at an indistinguishable time and it indicates an event. An event holding 'l' item is known as an l-event.

Without the generalization failure, if the items in an event are prearranged lexicographically, results in containing a radix type of arrangements between various events. For instance, if an event (i.e., first event) is a proper superset of another event (i.e., second event) then, the second event is radix ordered before the first event.

A timestamp for STM is exclusive for every event in progression which is used as an identifier for an event. The frequency of a succession's', denoted as supp(s), in a sequence database DAT is the numeral of sequences in 'DAT' that contain 's'. In this scenario, given a minimum amount of support or mini_sup, such that, 's' is frequent if supp(s) ≥ mini_sup. Our mining task is to identify all the frequent subsequences in the given sequence database for a user specified support value.

2.2 STM Constraints

The proposed STM using long and closed sequences includes two categories of constraints. The intra-event constraints refers that, it is not related to time and inter-events constraints, which are in a way highly related to the temporal aspect of the data.

For the experiments conducted in the literature study and in agreement with the domain, two constraints namely inter event and intra events constraints are included in the proposed work. Intra events are constraints that include Singletons and Maximum Event Length whereas the inter events constraints are Maximum Gap and Maximum Sequence Length.

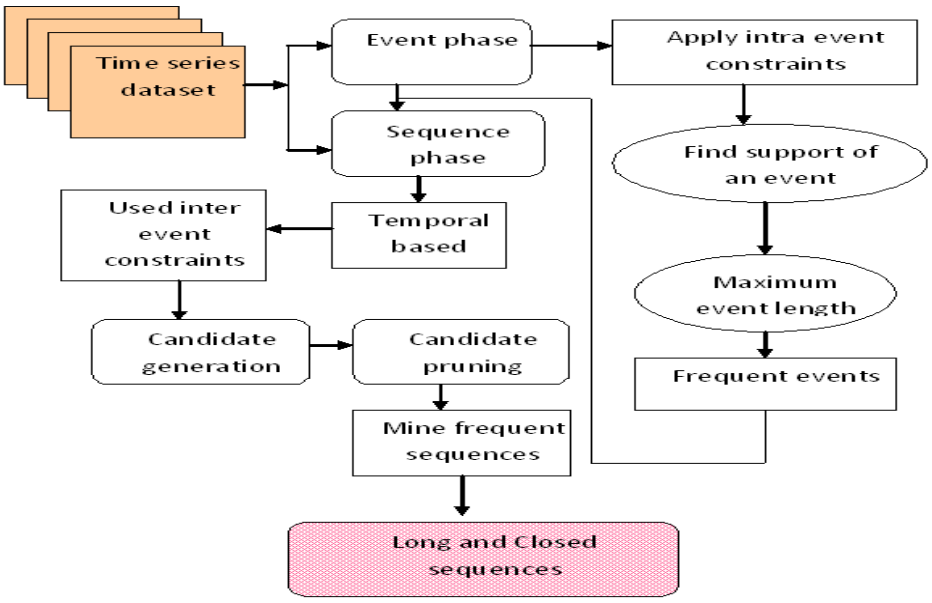

Figure 1: Process of Sequential Task Miner

The time series dataset holds the event and sequence operations in STM using long and closed sequence pattern mining. During the event phase, the algorithm not given the attention for the sequential order of data and considers the data as location of events. The output of sequence phase is a register of frequent sequences satisfying the inter-events constraints. With the application of time stamp, long and closed sequences are obtained.

3. EXPERIMENTAL EVALUATION

To evaluate the performance of STM, Sequential Task Miner, a detailed experimental result on time series data set with 60000 records are performed. The experiments are performed on 2.5 GHz CPU clock rate and 2 GB of main memory using WEKA data mining tool. In order to measure the effectiveness of Sequential Task Miner (STM) periodic time stamp, CPU execution time, mining efficiency and memory consumption boundaries are evaluated using time series dataset.

The time series dataset includes four example observations where change of circumstances occurs at some point in the series. Dense data sets consisting of 60,000 records are used on the basis of time which further evaluates SMCA, CAMLS and STM. The impact of mining characteristics using SP in SMCA, CAMLS and STM are studied on datasets comprising of time series obtained from UCI repository of 60,000 records taking into account the entire enumerated characteristic attributes

The performance evaluation of SMCA, CAMLS and STM are performed. During the process of algorithm, main memories are used to store the periodic time stamp. In order to perform the analysis over a dense data set including different characteristics, time series from Weka tool Machine Learning repository is used.

Long and Close Sequential Data Patterns conducted an extensive set of experiments to compare the approach with other representative algorithms. The car and bank dataset from UCI repository is taken for experiments. Long and closed sequential patterns are measured in which the whole car and bank datasets are too large to fit in the memory space.

For the purpose of performance evaluation, time series dataset is used which is extracted from the UCI repository consisting number of events and time instants. A set of periodic complex patterns is generated as follows. First, the period length from ordinary distribution with normal length is decided. Then, Pattern positions are chosen which are smaller than the average pattern length for nonempty event sets.

To test the STM algorithm, experiments are conducted to

- Determine the performance gain in terms of CPU runtime using STM method in comparison with SMCA and CAMLS method for long and closed sequences for large databases.
- Determine Number of long and closed sequences mined using STM method.

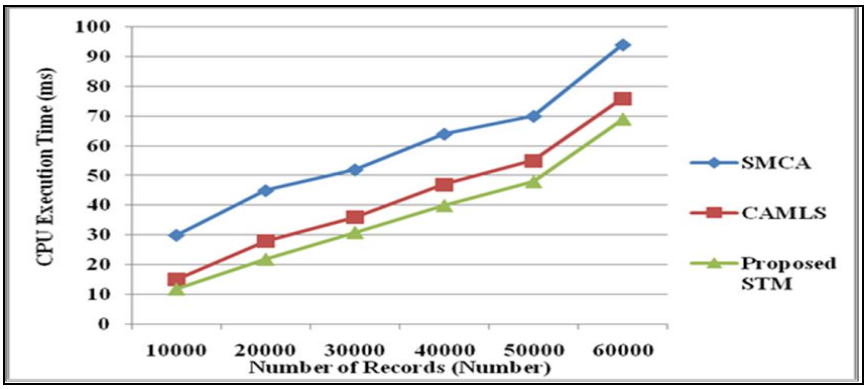

Figure 2: CPU Execution Time Using Time Series Dataset

Figure 2 shows the performance measure of CPU execution time with respect to the number of records using time series dataset. For the experimental purpose 60,000 records are taken where the CPU execution time is measured in terms of milliseconds (ms). From the figure it is illustrative that the CPU execution time observed is minimum in using STM when compared to the existing methods namely, Single Multi Complex Asynchronous (SMCA) and CAMLS (Constraint-Based Apriori Algorithm for Mining Long Sequences) using time series dataset. This is because the STM method uses the event and sequence phase separately resulting in less execution time.

The STM method provides 26 – 60 % improvement when compared to SMCA which processes long sequences at low minimum support when many frequent patterns are present. Further, STM method provides an improvement of 10 – 27 %

when compared to CAMLS, because, with the application of long and closed sequences separately, the CPU execution time is minimized at an extensive rate.

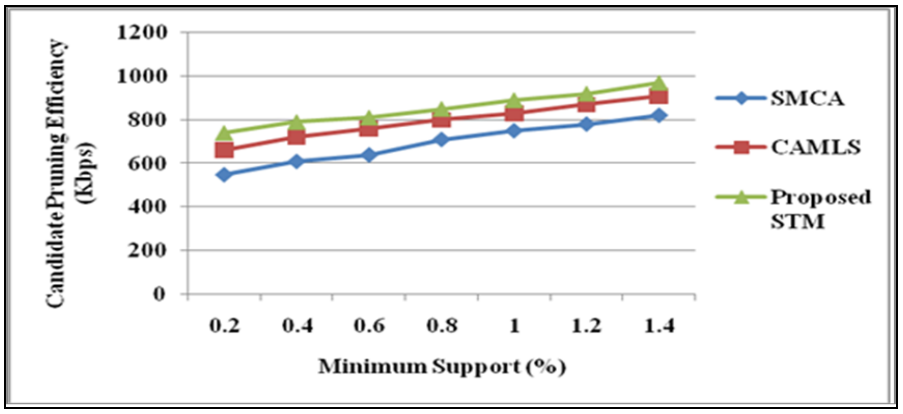

Figure 3: Measure of Candidate Pruning Efficiency Using Time Series Dataset

Figure 3 shows the measure of candidate pruning efficiency using time series dataset with respect to the minimum support value measured in terms of percentage (%). The candidate pruning efficiency is measured in terms of kilo bits per second. For experimental purpose, the value of support is taken from the range 0.2 to 1.4.

As illustrated in the above figure, with the increase in the support value, the candidate pruning efficiency also gets increased in all the three methods. But comparatively, using STM, the candidate pruning efficiency is improved by 15-25 % when compared to SMCA and 5 – 10 % when compared to CAMLS. This is because of the application of STM algorithm on periodic timestamp using intra-event constraints and inter-event constraints separately. Using STM, both the temporal aspect of data and data not related to time are taken into consideration, resulting in the improvement of candidate pruning efficiency.

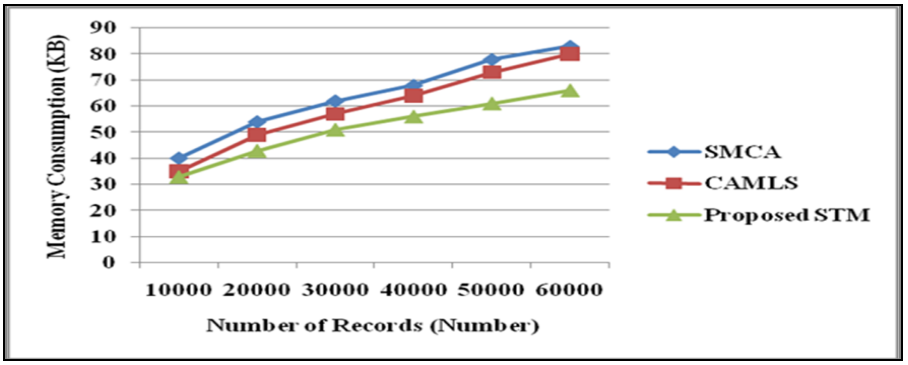

Figure 4: Measure of Memory Consumption using Time series dataset

The memory consumption for STM using time series dataset is displayed in Figure 4 respectively. For time series dataset of 10000 to 60000 records, the graph rises gradually high. From the figure, the memory consumption using STM method is comparatively less by 20 – 28 % when compared to SMCA and 6 – 21 % when compared to CAMLS. This is because of the fact that in STM method, temporal based data are used as candidate generation that uses inter-event constraints in order to mine frequent sequences by minimizing the memory consumption. The graph of 60000 records using time series dataset has an optimal consumption of memory.

4. CONCLUSION

STM, Sequential Task Miner technique is implemented for efficiently mining very long and closed sequences. It avoids the curse of the candidate maintenance-and-test paradigm, and checks the pattern closure in a more efficient way while consuming much less memory. It does not need to maintain the set of historical closed patterns. Thus it scales very well in the number of frequent long and closed patterns. Experimental results showed that the performance of STM using different minimum support margins and varying data sizes calculated the residual candidates' support and removed the non-frequent ones in an efficient manner. The STM method is designed to identify the long and closed patterns of time series dataset comprising of 60,000 records using event phase and sequence phase and analyze the performance of SMCA, CAMLS and STM. From the results it is clear that the method STM outperforms SMCA and CAMLS in terms of CPU execution time, mining efficiency and memory consumption.

The experiment conducted proves the efficiency of the STM method that utilizes lesser CPU time while pruning than the CAMLS (nearly 10% to 27%). Subsequently in terms of memory consumption using STM, consumption is less than the (6 – 21%) SMCA and candidate pruning efficiency are increased by 5 – 10% when compared to CAMLS.

REFERENCES

[1] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc.11th Int'l Conf. Data Eng. (ICDE '95), pp. 3-14, 1995

[2] Eric Hsueh-Chan Lu, Vincent S. Tseng, "Mining Cluster-Based Temporal Mobile Sequential Patterns in Location-Based Service Environments," IEEE Transl. On Knowledge And Data Engineering, Vol. 23, No. 6, February 2011

[3] M. Garofalakis, R. Rastogi, and K. Shim, SPIRIT: Sequential PAttern Mining with regular expression constraints. In VLDB'99, San Francisco, CA, Sept. 1999.

[4] Jae-Gil Lee, Member, IEEE, Jiawei Han, Fellow, IEEE, Xiaolei Li, and Hong Cheng, "Mining Discriminative Patterns for Classifying Trajectories on Road

Networks," IEEE Transl. On Knowledge And Data Engineering, Vol. 23, No. 5, May 2011.

[5] Kuo-Yu Huang and Chia-Hui Chang, "SMCA: A General Model for Mining Asynchronous Periodic Patterns in Temporal Databases," IEEE Transl. On Knowledge And Data Engineering, Vol. 17, No. 3, May 2005.

[6] Hsiao-Ping Tsai, Member, IEEE, De-Nian Yang, and Ming-Syan Chen, Fellow, IEEE," Mining Group Movement Patterns for Tracking Moving Objects Efficiently," IEEE Transl. On Knowledge And Data Engineering, Vol. 23, No. 2, February 2011.

[7] Jinlin Chen, Member, IEEE, "An UpDown Directed Acyclic Graph Approach for Sequential Pattern Mining," IEEE Transl. On Knowledge And Data Engineering, Vol. 22, No. 7, July 2010.