Multimed Tools Appl (2019) 78:4217–4265
https://doi.org/10.1007/s11042-017-5280-y

ASTRA - A Novel interest measure for unearthing latent temporal associations and trends through extending basic gaussian membership function

Vangipuram Radhakrishna[1] · **Shadi A. Aljawarneh**[2] · **Puligadda Veereswara Kumar**[3,4] · **Vinjamuri Janaki**[5]

Received: 9 June 2017 / Revised: 25 September 2017 / Accepted: 4 October 2017 /
Published online: 6 December 2017

Abstract Time profiled association mining is one of the important and challenging research problems that is relatively less addressed. Time profiled association mining has two main challenges that must be addressed. These include addressing i) dissimilarity measure that also holds monotonicity property and can efficiently prune itemset associations ii) approaches for estimating prevalence values of itemset associations over time. The pioneering research that addressed time profiled association mining is by J.S. Yoo using Euclidean distance. It is widely known fact that this distance measure suffers from high dimensionality. Given a time stamped transaction database, time profiled association mining refers to the discovery of underlying and hidden time profiled itemset associations whose true prevalence variations are similar as the user query sequence under subset constraints that include i) allowable dissimilarity value ii) a reference query time sequence iii) dissimilarity function that can find degree of similarity

✉ Vangipuram Radhakrishna
vrkrishna@acm.org; radhakrishna_v@vnrvjiet.in

Shadi A. Aljawarneh
saaljawarneh@just.edu.jo

Puligadda Veereswara Kumar
pvkumar58@gmail.com

Vinjamuri Janaki
janakicse@yahoo.com

[1] Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology (Autonomous), Hyderabad, Telangana 500090, India

[2] Department of Software Engineering, Jordan University of Science and Technology, Irbid, Jordan

[3] Department of Computer Science and Engineering, Acharya Institute of Technology, Bangalore, India

[4] University College of Engineering, Osmania University, Hyderabad, India

[5] Department of Computer Science and Engineering, Vaagdevi College of Engineering (Autonomous), Warangal, India

between a temporal itemset and reference. In this paper, we propose a novel dissimilarity measure whose design is a function of product based gaussian membership function through extending the similarity function proposed in our earlier research (G-Spamine). Our approach, MASTER (Mining of Similar Temporal Associations) which is primarily inspired from SPAMINE uses the dissimilarity measure proposed in this paper and support bound estimation approach proposed in our earlier research. Expression for computation of distance bounds of temporal patterns are designed considering the proposed measure and support estimation approach. Experiments are performed by considering naïve, sequential, Spamine and G-Spamine approaches under various test case considerations that study the scalability and computational performance of the proposed approach. Experimental results prove the scalability and efficiency of the proposed approach. The correctness and completeness of proposed approach is also proved analytically.

Keywords Temporal association pattern · Prevalence time sequence · Dissimilarity · Time profiled · Transaction database · Temporal databases

1 Introduction

Temporal data mining may be defined as any data mining task that is associated with some dimension of time. Some of the temporal data mining tasks comprise pattern search and retrieval, evolutionary clustering, trajectory clustering, spatio-temporal data mining, classification, temporal association rules. Of all these temporal data mining tasks that exist, the problem of temporal association pattern analysis has extensively influenced data mining research community both from the academia and industry. Conventional association rule mining (ARM) (http://www.westbrookstevens.com/continuous.htm) [2] targets at retrieving the set of all hidden rules which satisfy user defined constraints such as support and confidence. Also traditional association rules do not consider the time and are limited to discovery of unordered correlations between transaction items of a given transaction database. Discovery of such interesting hidden rules is extended to temporal databases by associating the concept of time. These association rules are also called as temporal association rules. Some of the related works on temporal pattern mining includes “event analysis” [5, 12, 14] such as exploring regularities in event occurrence and finding the hidden temporal dependence between events. Given an interval with specified size, [52] discusses frequent episode mining. A review on sequential pattern discovery [3] is carried by Srikant and Agrawal. Bettini [14 , 14] consider patterns that are much more complex than those considered in [3, 5, 12, 52]. All these works do not consider itemset lifespan for discovering temporal patterns. Juan and Gustavo [6] extend the ARM algorithm [2] proposed for non-temporal databases for temporal context. The temporal association rule mining approach proposed in [6] eliminates the need to specify the time interval or calendars by introducing the concept of “itemset lifespan”. Such a temporal association rule discovers ordered correlations between transaction items. An algorithm that discovers temporal association rules is called the temporal association rule mining algorithm (TARM).

Jin Soung Yoo [87] observed that previous research related to temporal data mining are not suitable for discovering special temporal regulation patterns such as “seasonal temporal associations”, “emerging temporal associations” and “diminishing temporal associations”. This limitation is addressed initially by Jin Soung Yoo and Shashi Sekhar [84–86] in their

research. Mining time-profiled temporal associations is the pioneering work by Jin Soung Yoo and Shashi Shekhar [84–87]. But, these studies [84–87] are only limited to the use of the Euclidean distance function for the discovery of similarity-based time profiled associations. This possible space for extending their research inspired us to devise new methods and approaches for estimating temporal association pattern support bounds, distance bounds and to propose a novel temporal dissimilarity measure that fits support estimation expressions and facilitates estimating distance bounds to perform early pattern pruning [8, 21, 58–64, 69]. We now state the time profiled temporal association pattern mining problem.

Given a time stamped transaction database, time profiled association mining aims at "Discovery of time profiled temporal associations that are similar to a given reference query time sequence such that their support variations of association item sets vary similar to given query time sequence" [87]. Time-profiled temporal association mining has implicitly two important computational challenges. They are i) approaches that can substantially minimize the total number of true support computations and ii) dissimilarity measure that can accurately find the similarity between time profiled temporal associations. This work extends our previous research [8, 21, 59–64] by proposing a novel dissimilarity measure for retrieving all the possible and valid time profiled temporal association patterns from time stamped transaction databases.

This paper is outlined as follows: Section 2 explores some of the important and significant studies related to frequent pattern mining, association rule mining, temporal association rule mining and time profiled association mining respectively. The proposed prevalence estimation approach and dissimilarity measure is addressed in section-3 and section-4 respectively. Section-5 outlines the algorithm design and the time profiled association mining algorithm. A working example is discussed in Section-6. Experimental results are discussed in Section-7 w.r.t algorithm scalability and performance. Section-8 concludes this paper.

2 Literature review

Historically, summaries of temporal database research that were carried at various research labs and those that are addressed at various symposiums and workshops were first published in ACM SIGMOD record in the year 1982 [57]. The importance of temporal databases has become evident with IEEE Data Engineering devoting a complete special issue in the year 1988. Following this, two research papers contributing to survey of temporal databases [57] are published in the year 1990 and 1992. Some of the significant algorithms that addressed the discovery of itemset associations in a transaction database are discussed in this subsection.

2.1 Mining frequent itemsets and association rules in transaction databases

Rakesh Agrawal [4] proposed association rule discovery in transaction database by introducing AIS and SETM algorithms. Apriori [4] and aprioriTid [2] algorithms that are addressed for mining frequent itemsets and hidden association rules in a transaction database are the pioneering work in frequent pattern mining. These algorithms only consider itemsets that are found to be large in the previous pass and generate candidate

itemsets in the next pass with out scanning the database. The limitation of apriori (or aprioriTid) algorithm is that it does not consider the structural properties of frequent itemsets. Normally, frequent item sets are used to generate association rules. Applying, apriori [2, 4] generates massive set of association rules eventually consuming more computational space. Nicolas Pasquier et al. (in year 1999) introduced A-Close which is based on the closed itemset lattice framework. A-CLOSE [55] reduces the problem of mining association rules to "finding closed frequent items sets". Using closed frequent item sets to generate association rules instead of using frequent item sets [2, 4] generates reduced set of association rules without any information loss.

Algorithms inspired from Apriori have degrading performance when data is not sparse and generate long frequent patterns. Apriori algorithm utilizes downward closure property [2] and applies BFS technique that itemizes each and every possible single frequent itemset [55]. The number of database scans performed is equal to length of the longest possible frequent itemset and this leads to huge I/O overhead. Also, for a given frequent itemset of length, L there exists (2^L-2) additional frequent itemsets. For a sufficiently large value of L, these algorithms grow into CPU bound. Vertical mining algorithms proposed for mining association rules have scalability issue when intermediate vertical transaction ids (Vertical TID) occupy large part of memory. (Zaki, 2003) proposed a fast-vertical mining algorithm called dCharm for closed frequent pattern mining [55]. (Grahne and Zhu, 2005) [26] extends FP-growth algorithm proposed by Han [32] to compute maximal frequent itemsets, closed frequent itemsets by proposing a new technique called FP-array. The objective of [26] is to reduce the complexity involved in traversing the FP-tree.

A scalable algorithm for association rule mining called as Eclat algorithm is proposed by Zaki [88] (in year, 2000) that considers structural properties of frequent items. Eclat aims at I/O cost minimization and uses lattice traversal technique for discovering frequent itemsets. Following this, Zaki (in year, 2001) extended Eclat by introducing a vertical mining based approach for frequent itemset mining called Dclat algorithm [89]. Dclat algorithm uses novel vertical data representation called Diffset. Cohen [23] addressed correlation based association rule mining that is applicable for several interesting applications such as clustering web data, finding similar web documents, colloborative filtering and other data mining related applications.

(Han, 2004) [32] proposed a compressed tree based approach for finding frequent patterns called FP-tree approach. Advantages of FP-tree approach are i) Generating highly compact FP-tree that is substantially smaller than original transaction database ii) it overcomes the computationally costly candidate generation and test process by concatenating the frequent-1 itemsets present in conditional FP-trees iii) the partitioning based divide and conquer approach reduces the size of conditional patterns. The FP-tree approach of frequent pattern mining [32] is extended in [15] which applies recursive elimination principle. The advantage of this approach is simple tree structure. (F. Zhu et al., 2007) propose Pattern-Fusion [90] approach that gives an interesting and efficient way to retrieve colossal frequent itemsets. The approach proposed in [90] discuss ways to overcome dis-advantages of Apriori and FP-tree based frequent pattern mining algorithms.

Following research of [2] several studies on association rule mining have been addressed that includes generalized association rule mining [73], multiple-level association rule mining [30], quantitative association rule mining [74], high-dimensional association rule mining [83], constraint based and multiple minimum support based ARM [51, 77] incremental association rule mining [22, 40], parallel association rule mining [1, 54], Collosal Frequent pattern mining [72] for high dimensioanal datasets, emerging patterns [24], partial periodic patterns [31] but are not exhaustive.

2.2 Mining temporal association rules

Mining temporal association rules has gained important attention during the last decade. Chen and Petrounias[18] outlines typical issues in mining temporal association rules.(Juan and Gustavo, 2000) proposed discovery of temporal association rules by extending apriori algorithm and support concept of apriori to temporal support [6]. The main idea is to introduce notion of time to frequent items and eliminate the need for specifying intervals that must be specified for other approaches [53]. Applying conventional association rule mining (ARM) approach, it is only possible to discover correlation between data items of transactions in a transaction database without considering ordering among data items. If ordering constraint is imposed on these transaction data items then the resulting patterns are termed "temporal patterns". Temporal database facilitates to record periodic behavior. Calendar based association rules (CBTAR), Cyclic association rules (CAR) and Periodic association rules (PAR) are various periodic temporal patterns. CBTAR are multi-granular temporal patterns where as CAR and PAR are single time granular. The first contribution for discovering CBTARs is by [45, 46] that uses level-based apriori. (Wan-Jui Lee et al., 2004) [42]extends the CBTAR mining approach proposed in [45] by proposing an approach that performs database scan only twice.

FP-tree [32] and constraint based approaches [51, 77] are not suitable to discover interesting rules from publication databases. To facilitate this, an approach for discovery of temporal association rules from publication databases and causal relationalship between itemsets (that are actually infrequent) called "progressive partioned miner" [41] is proposed. Ozden, Ramaswamy, and Silberschatz introduces cyclic association rules [53] that satisfy periodicity. i.e. if a rule does not hold true for a time instance then, for all subsequent cycles it also does not hold true. Conversely, if an association rule satisfies at a given time point, then this rule also holds good for all other cycles at that particular time instant. Most real life patterns are actually not perfect and the objective is to discover and retrieve all imperfect temporal patterns. Also, all the above discussed research contributions do not consider multiple time granularities. Hence even a query of the form, "second holiday of every year" cannot atleast be addressed. Given a time stamped transaction dataset, the problem of mining association rules in calendar schema is addressed in [44, 45]. (Tansel and Imberman, 2007) uses enumeration operation of the temporal relation algebra to generate association rules [76].

Given, an interval-based data [82], (Edi Winarko, 2007) proposed an approach called ARMADA, that is based on maximum gap-constraint for discovering interval based temporal patterns. ARMADA overcomes limitations of sequential patter mining [75], memory indexing based sequential pattern mining [47], Prefix-Scan [56]. A time-indexing based sequential pattern mining called METISP [48] proposed by (Ming-Yen Lin, 2008) considers time constraints such as maximum-gap, sliding window, minimum-gap, exact-gap, and duration. METISP [48] builds time-index sets for improving processing efficiency. Mining frequent patterns in time series databases and transaction databases have been studied extensively in data mining and most of the previous research use candidate set generation and test such as apriori which is computationally expensive. Mining temporal patterns from interval databases is addressed in [19] that proposed Gaussian based similarity measure. Summary, detailed information and implementation of various data mining algorithms and respective synthetic and real time datasets for sequential pattern mining, sequential rule mining, sequence prediction, frequent itemset mining, periodic itemset mining, high utility pattern mining, association rule mining, time series mining[38, 81], clustering and classification is available as open source (http://www.philippe-fournier-viger.com/spmf). Other important contributions that are based on temporal association rules includes temporal association rules based on temporal interval data using Allens theory in [43], ITARM (Incremental temporal association rule mining) [25], TSET-Miner (event-based sequence mining based on tree data structure) [27],

TSET^{FUZZY}-MINER (based on fuzzy counting) [28], fuzzy based TARs [20, 39] that are based on itemset life span, discovering calendric association rules [72], temporal associations in multi-variate time series [91], weighted periodic pattern mining in time series databases [17].

Although, these studies have considered transaction data that is implicitly related to time, these did not address approaches that can discover special regulation patterns such as "emerging temporal patterns", "seasonal temporal patterns" or diminishing patterns which consider "actual prevalence similarity". Most of these studies did not address time profiled temporal association mining that has various applications in stock market exchange, analyzing sales trend in market-basket, climate measurement (such as temperature, moisture, precipitation etc) to mention some of them.

Temporal association mining that is based on the similarity function called similarity-based temporal association mining (SPAMINE) is one of the important research problems with pioneering contribution by Jin Soung Yoo and Shashi Shekhar [84–86]. Although SPAMINE [84–86] addressed the problem of "similarity-based temporal association mining" but it was restricted to the use of Euclidean distance measure. Following research by Yoo [84–86], there exists no important studies that subsequently addressed this problem except our previous studies [8, 21, 59–64]. In [59], a dissimilarity measure for mining temporal association patterns is proposed. However, it has a limitation [59], as it is not addressed how deviation value must be computed. It was simply fixed equal to threshold value. This is overcome in our subsequent contributions by proposing an expression for computing deviation [11, 65, 68, 78–80] and for choosing proper threshold value corresponding to the deviation. Studies [21, 66, 67, 69, 70] propose approaches for estimating supports of temporal association patterns. All these similarity functions may also be applied to different applications that are related to [7, 9, 10, 33]. Application of similarity measures for dimensionality reduction and process transformation are discussed in [29, 35, 36, 37].

2.3 Motivation and research scope

It is worth to mention that our research is inspired from [34, 49, 84–87]. Studies [60, 66, 67, 84–87] that have addressed solutions for mining time profiled temporal associations considered the widely known L_p norm distance metric. Support time sequences in time stamped transaction databases are implicitly high dimensional. It is a well-known fact that this L_p norm distance measure (i.e Euclidean distance for $p = 2$) falls prey to high dimensionality and is hence not suitable for time profiled association mining. Some of our previous research [8, 11, 63–65, 68, 78–80] proposed distance measures for finding similarity between temporal trends and patterns and this research extends our previous research by proposing a new dissimilarity measure. The difference between the proposed dissimilarity measure and the previous measures is that the membership function is product based in contrast to previous measures that are summation based. The similarity function is designed considering gaussian membership function.

3 Support bound estimation of time-profiled temporal associations

We discuss the proposed support bound estimation procedure in this section. Support estimation of time profiled associations facilitates for early elimination of invalid pattern combinations. It also minimizes computational space and processing time consumed by the pattern mining algorithm. Table 1 describes various notations that are used in designing expressions for estimating supports of temporal associations and their associated meaning.

3.1 Support bound estimation of temporal association

Our approach for finding support limits of itemset associations uses positive and negative supports of temporal items or itemsets. To estimate support values of time profiled associations, the set of all temporal pattern combinations possible are divided into two classes. The first class comprises temporal patterns of size equal to two (i.e |S| = 2). The second comprises all temporal patterns of size greater than two (i.e |S| > 2).

3.1.1 Generalized expressions for estimating support values at t^{th} time slot

Let, T_P and T_Q are any two chosen temporal patterns (positive singleton items) then, the corresponding temporal association pattern generated from these two temporal patterns, T_P and T_Q is denoted using the notation, T_{PQ} . Let representations, T_{P_t} and T_{Q_t} denote support value for temporal patterns at t^{th} time slot then, T_{PQ_t} represents support value of the itemset association, PQ at t^{th} time slot. Also, notations, $\bar{T}_{P_t}$ and $\bar{T}_{Q_t}$ each represent the support value of negative temporal patterns $\bar{T}_P$ and $\bar{T}_Q$ at t^{th} time slot respectively.

The necessary expression to compute the maximum prevalence bound at t^{th} time slot is given by Eq. (1)

$$T_{PQ}^{max} = T_{P_t} - \max\left\{\left(1 - \bar{T}_{P_t} - T_{Q_t}\right), 0\right\} \tag{1}$$

The necessary expression to compute the minimum prevalence bound at t^{th} time slot is given by Eq. (2)

$$T_{PQ}^{min} = \max\left(1 - \bar{T}_{P_t} - \bar{T}_{Q_t}, 0\right) \tag{2}$$

Table 1 Description of notations

Notation	Description of notation used
I	finite set of transaction items
N	total number of transaction items
n	total number of time slots
I	itemset or pattern
T_I	temporal itemset or temporal pattern support
S	size of itemset
$\bar{T}_I$	Negative temporal pattern
T_{I_t}	Support of temporal itemset, I at t^{th} time slot
P	itemset combination of size, (S-1)
Q	singleton item of size equal to one
PQ	itemset association of size, |S| generated from sizes (S-1) and 1
$\overrightarrow{T_{PQ}}$	Support time sequence of temporal association pattern, T_{PQ}
T_{PQ_t}	Support of temporal association PQ at t^{th} time slot
$T_{PQ_t}^{max}$	Maximum support value of temporal association PQ at t^{th} time slot
$T_{PQ_t}^{min}$	Minimum support value of temporal association PQ at t^{th} time slot
$\overrightarrow{T_{PQ}^{max}}$	Maximum possible support time sequence of temporal association pattern, T_{PQ} for 'n' time slots
$\overrightarrow{T_{PQ}^{min}}$	Minimum possible support time sequence of temporal association pattern, T_{PQ} for 'n'time slots

Equations (1) and (2) together can be represented as Eq. (3)

$$support\_bounds\left(T_{PQ_t}\right) = \begin{cases} T^{max}_{PQ_t} \\ T^{min}_{PQ_t} \end{cases} = \begin{cases} T_{P_t} - max\left\{\left(1 - \bar{T}_{P_t} - T_{Q_t}\right), 0\right\} \\ max\left(1 - \bar{T}_{P_t} - \bar{T}_{Q_t}, 0\right) \end{cases} \quad (3)$$

Equation (3) gives the generalized expression to find the maximum possible and the minimum possible support values of the temporal association pattern (T_{PQ}) for a single time slot (say t^{th} time slot).

3.1.2 Estimation of support time sequence of Level-2 temporal association pattern (Size, S = 2)

This subsection describes the computation of the maximum possible and the minimum possible support time sequences of temporal associations. Let, T_P and T_Q are any two singleton temporal patterns and their respective true support time sequences over 'n' time slots are denoted using $\overrightarrow{T_P} = (T_{p_1}, T_{p_2}, T_{p_3}, \ldots\ldots\ldots\ldots, T_{p_n})$ and $\overrightarrow{T_Q} = (T_{Q_1}, T_{Q_2}, T_{Q_3}, \ldots\ldots\ldots\ldots, T_{Q_n})$. The maximum possible and minimum possible support time sequences of a temporal association for 'n' time slots can be obtained by extending Eqs. (1) and (2) to 'n' time slots as given by Eqs. (4) and (5).

Maximum possible support time sequence $\left(\overrightarrow{T^{max}_{PQ}}\right)$ The maximum possible support time sequence of T_{PQ},for 'n' disjoint time slots is denoted by $\overrightarrow{T^{max}_{PQ}}$ and can be obtained by applying Eq. (4).

$$\overrightarrow{T^{max}_{PQ}} = \left(T^{max}_{PQ_1}, T^{max}_{PQ_2}, T^{max}_{PQ_3}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, T^{max}_{PQ_n} \right).$$

where

$$T^{max}_{PQ_t} = T_{P_t} - \max\left\{\left(1 - \bar{T}_{P_t} - T_{Q_t}\right), 0\right\} \text{ and } 1 \le t \le n \quad (4)$$

Minimum possible support time sequence $\left(\overrightarrow{T^{min}_{PQ}}\right)$ The minimum possible support time sequence over 'n' disjoint time slots for T_{PQ}, is denoted by $\overrightarrow{T^{min}_{PQ}}$ and is given by Eq. (5),

$$\overrightarrow{T^{min}_{PQ}} = \left(T^{min}_{PQ_1}, T^{min}_{PQ_2}, T^{min}_{PQ_3}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, T^{min}_{PQ_n} \right)$$

where

$$T^{min}_{PQ_t} = \max\left(1 - \bar{T}_{P_t} - \bar{T}_{Q_t}, 0\right) \text{ and } 1 \le t \le n \quad (5)$$

3.1.3 Prevalence time sequence bounds (Size, S > 2)

Let, T_P and T_Q each denote temporal pattern and their respective support time sequences over 'n' time slots are denoted by $\overrightarrow{T_P} = (T_{P_1}, T_{P_2}, T_{P_3}, \ldots\ldots\ldots\ldots, T_{P_n})$ and $\overrightarrow{T_Q} = (T_{Q_1}, T_{Q_2}, T_{Q_3}, \ldots\ldots\ldots\ldots, T_{Q_n})$ and size of T_P and T_Q is equal to (|S|-1) and 1 respectively. A temporal association pattern, T_{PQ} is generated from itemset association PQ (or from

temporal patterns, T_P and T_Q). Let, Ss(PQ) be the subset itemset of size equal to (|S|-1) and S(PQ) denotes the singleton item of size equal to 1. It must be noted that, patterns represented by Ss(PQ) and S(PQ) together form the itemset PQ and the corresponding temporal pattern is denoted using T_{PQ}, i.e. for some randomly chosen itemset association PQ, we have Ss(PQ) ≡ P of size is equal to (|S|-1) and S(PQ) ≡ Q of size equal to 1 respectively such that Ss(PQ) ∩S(PQ) = ∅ . Here, Ss(PQ) and S(PQ) represents all viable subset combinations possible at level (l-1) and level-1 using which superset itemset combination PQ at level 'l' can be generated.

In general, notations, $T_{\mathrm{Ss}(PQ)_t}$ and $T_{\mathrm{S}(PQ)_t}$ are used to represent support value of subset temporal patterns, $T_{\mathrm{Ss}(PQ)}$ and $T_{\mathrm{S}(PQ)}$ at t^{th} time slot. The support time sequence bounds (maximum possible and minimum possible) for such temporal associations of size greater than two are obtained for each time slot by considering every possible subset of size equal to (S-1), and 1 as discussed below.

Minimum possible support time sequence $\left(\overrightarrow{T_{PQ}^{min}}\right)$ The minimum possible support time sequence of temporal association pattern, T_{PQ} of size equal to |S| (i.e. at level 'l') over 'n' time slots is obtained by considering each possible k^{th} subset (i.e. $Ss^k(PQ)$) of size, |S|-1 at previous level, i.e. (l-1) and singleton item, S(PQ) at level-1 such that $\mathrm{Ss}^k(PQ)$ ∩S(PQ) = ∅. For example, consider the itemset XYZ of size equal to 3, then all viable size-2 itemsets are XY, XZ and YZ and corresponding size-1 itemsets are Z, Y and X. Itemset XYZ may be obtained by considering any of these three possible combinations. It can be easily verified that {X, Y} ∩ {Z} = ∅ where $\mathrm{Ss}^2(XYZ)$ ≡ XY and S(XYZ)≡ Z. Equivalently, {X, Z} ∩ {Y} = ∅ and {Y, Z} ∩ {X} = ∅. To find support bounds for temporal itemset XYZ, i.e. T_{XYZ} we must consider all these viable subset combinations.

Equation (6) represents the support time sequence of temporal association pattern, T_{PQ}(i.e those obtained from k^{th} possible subset denoted by $Ss^k(PQ)$ of size equal to |S|-1) and singleton pattern S(PQ)

$$\overrightarrow{\mathrm{T}_{\mathrm{PQ}}^{\mathrm{k}}} = \left(T_{PQ_1}^k, T_{PQ_2}^k, T_{PQ_3}^k, \ldots\ldots\ldots\ldots, T_{PQ_n}^k\right)$$

where

$$T_{PQ_t}^k = \text{maximum}\left\{\left(1 - T_{\overline{Ss^k(PQ)_t}} - T_{\overline{S(PQ)_t}}\right), 0\right\} \text{and } 1 \le \mathrm{t} \le \mathrm{n} \tag{6}$$

Support time sequences are obtained by considering individual subset itemset associations denoted by $Ss^k(PQ)$ and $S(PQ)$, utilizing Eq. (6). From these support sequences, the minimum support time sequence is obtained by considering maximum support value at respective time slot over all viable subsets of itemset association, PQ.

The minimum support time sequence, $\overrightarrow{T_{PQ}^{min}}$ of temporal association pattern, T_{PQ} is given by Eq. (7)

$$\overrightarrow{T_{PQ}^{min}} = \left(T_{PQ_1}^{min}, T_{PQ_2}^{min}, T_{PQ_3}^{min} \ldots\ldots\ldots\ldots, T_{PQ_n}^{min}\right) \tag{7}$$

where $T_{PQ_t}^{min} = \max\left\{T_{PQ_t}^1, T_{PQ_t}^2, \ldots., T_{PQ_t}^k\right\}$ and $1 \le \mathrm{t} \le \mathrm{n}$

Maximum support time sequence, $\left(\overrightarrow{T_{PQ}^{max}}\right)$ The maximum possible support time sequence of temporal association pattern, T_{PQ} of size equal to |S| (i.e at level 'l') for 'n' time slots is

obtained by considering each possible k[th] subset (i.e. $Ss^k(PQ)$) of size |S|-1 at previous level, i.e. (l-1) and singleton item, S(PQ) at level-1 such that $Ss^k(PQ) \cap S(PQ) = \varnothing$.

Equation (8) represents the support time sequence of temporal association pattern T_{PQ} obtained by considering the k[th] possible subset denoted by $Ss^k(PQ)$ of size equal to |S|-1 and the singleton pattern S(PQ)

$$\overrightarrow{T^{k}_{PQ}} = \left(T^k_{PQ_1}, T^k_{PQ_2}, T^k_{PQ_3}, \ldots\ldots\ldots\ldots\ldots, T^k_{PQ_n} \right) \tag{8}$$

where $T^k_{PQ_t} = \left(T_{Ss^k(PQ)_t} - \max \left\{ \left(1 - T_{\bar{Ss}^k(PQ)_t} - T_{S(PQ)_t} \right), 0 \right\} \right)$ and $1 \le t \le n$

Support time sequences are obtained by considering individual subset itemset associations denoted by $Ss^k(PQ)$ and $S(PQ)$, utilizing Eq. (8). From these support sequences, the maximum support time sequence is obtained by considering minimum support value at respective time slot over all viable subsets of itemset association, PQ.

The maximum support time sequence, $\overrightarrow{T_{PQ}{}^{max}}$ is given by Eq. (9)

$$\overrightarrow{T_{PQ}{}^{max}} = \left(T^{max}_{PQ_1}, T^{max}_{PQ_2}, T^{max}_{PQ_3}, \ldots\ldots\ldots\ldots\ldots, T^{max}_{PQ_n} \right) \tag{9}$$

where $T^{max}_{PQ_t} = \text{minimum} \left\{ T^1_{PQ_t}, T^2_{PQ_t}, \ldots\ldots, T^k_{PQ_t} \right\}$ and $1 \le t \le n$

In Eq. (9), the representation $T^k_{PQ_t}$ denotes the support value obtained by considering the k[th] possible subset itemset combination at t[th] time slot and $T^{max}_{PQ_t}$ denotes the maximum possible support value of temporal association pattern, T_{PQ} at t[th] time slot.

3.2 Case study

This section explains our approach discussed in section 3.1 for estimating prevalence time sequence bounds of temporal association patterns. For demonstrating the computation procedure, a time stamped transaction database generated using IBM data generator [87] as depicted in the Fig. 1 is considered.

The database in Fig.1 is defined over two-time slots (denoted by T2) and consists of ten transactions per each time slot (denoted by TD10). The total number of transactions is 20 (D20). The total number of items in finite itemset is three (I3) with average transaction size equal to two (L2). The dataset can hence be denoted as TD10-D20-I3-L2-T2. As depicted in Fig. 1, possible transaction items are A, B and C.

Figure 2 shows prevalence values at two-time slots t1 and t2 for level-1 (singleton) temporal patterns. Notations, T_A, T_B, T_C represent positive temporal pattern and $\bar{T}_A$, $\bar{T}_B$ and $\bar{T}_C$ are negative temporal pattern. T_{A_1}, T_{B_1} are positive supports of patterns at time slots t_1 and T_{A_2}, T_{B_2} are positive supports of patterns at time slots t_2. Similarly, $\bar{T}_{A_1}$, $\bar{T}_{B_1}$ are negative supports at time slots t_1 and $\bar{T}_{A_2}$, $\bar{T}_{B_2}$ are negative supports at time slots t_2.

In subsections 3.2.1 to 3.2.4, our approach for estimating support bounds of temporal associations is explained for itemset associations AB, AC, BC and ABC.

D_1		D_2	
Time slot-1: (t_1)		Time slot-2 : (t_2)	
time	transaction items	time	transaction items
T1	{A, B, C }	T11	{C}
T2	{A, B, C }	T12	{A, B, C }
T3	{A}	T13	{B, C }
T4	{ A }	T14	{B}
T5	{C}	T15	{A, C }
T6	{A, B, C }	T16	{A, B, C }
T7	{C}	T17	{B, C }
T8	{A ,C }	T18	{B}
T9	{ C }	T19	{A, B, C }
T10	{C}	T20	{C}

Fig. 1 Example dataset

3.2.1 Prevalence time sequence bounds for temporal pattern, T_{AB}

Consider the temporal itemset, T_{AB}. The computation of prevalence sequence bounds of temporal pattern can be obtained by applying Eqs. (4) and (5) respectively. Figure 3 shows the maximum possible support sequence and minimum possible support sequence for temporal pattern, T_{AB}.

Maximum support time sequence of T_{AB}, ($\overrightarrow{T_{AB}^{max}}$) The maximum support time sequence of temporal itemset, T_{AB} is denoted by $\overrightarrow{T_{AB}^{max}}$ and can be computed using $\overrightarrow{T_{AB}^{max}} = \left(T_{AB_1}^{max}, T_{AB_2}^{max} \right)$ where $T_{AB_1}^{max} = T_{A_1} - \max(1 - \bar{T_{A_1}} - T_{B_1}, 0)$ and $T_{AB_2}^{max} = T_{A_2} - \max(1 - \bar{T_{A_2}} - T_{B_2}, 0)$. In our case, T_{A_1} =0.6, T_{A_2} = 0.4, T_{B_1} =0.3, T_{B_2} = 0.7, $\bar{T_{A_1}}$ = 0.4, $\bar{T_{A_2}}$ = 0.6, $\bar{T_{B_1}}$ = 0.7, $\bar{T_{B_2}}$ = 0.3.

So, $T_{AB_1}^{max} = T_{A_1} - \max(1 - \bar{T_{A_1}} - T_{B_1}, 0)$ = 0.6 – maximum (1–0.4-0.3, 0) = 0.6- maximum (0.3, 0) = 0.6–0.3 = 0.3. Similarly, $T_{AB_2}^{max} = T_{A_2} - \max(1 - \bar{T_{A_2}} - T_{B_2}, 0)$ = 0.4 – maximum (1–0.6 – 0.7, 0) = 0.4 – maximum (−0.3, 0) = 0.4–0 = 0.4. Hence, $\overrightarrow{T_{AB}^{max}} = (0.3, 0.4)$

Item (I)	Positive Temporal pattern (T_I) (Level-1)	Prevalence at t1 T_{I_1}	Prevalence at t2 T_{I_2}	Negative Temporal pattern ($\overline{T_I}$) (Level-1)	Prevalence at t1 ($\overline{T_{I_1}}$)	Prevalence at t2 ($\overline{T_{I_2}}$)
A	T_A	0.6	0.4	$\overline{T_A}$	0.4	0.6
B	T_B	0.3	0.7	$\overline{T_B}$	0.7	0.3
C	T_C	0.8	0.8	$\overline{T_C}$	0.2	0.2

Fig. 2 Support values of singleton temporal pattern

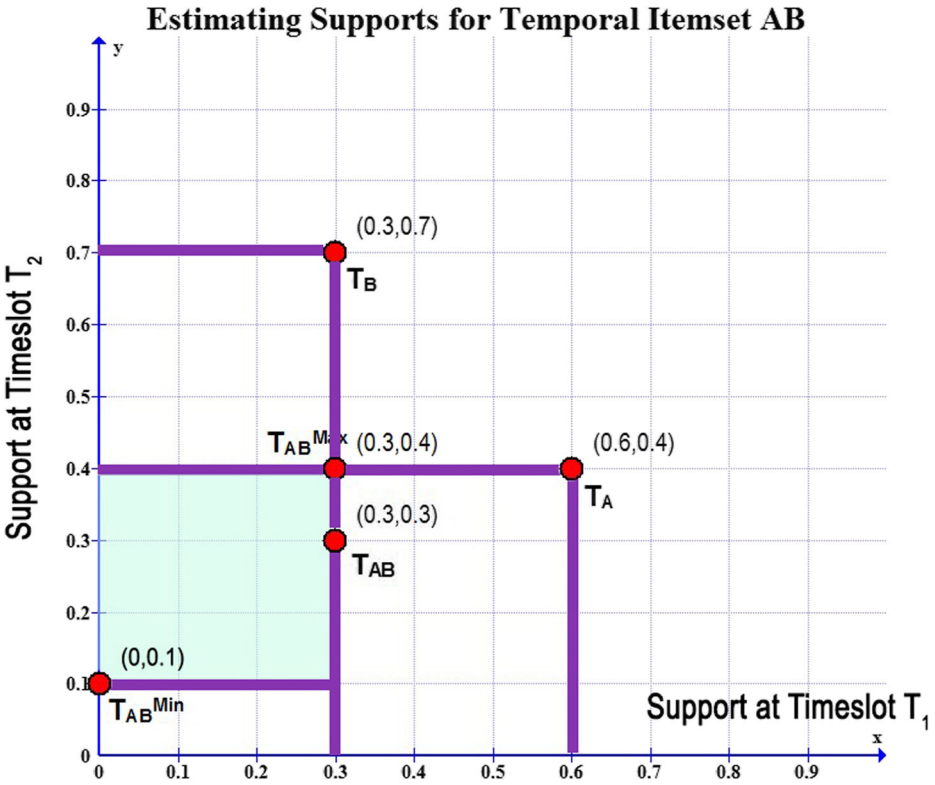

Fig. 3 Support bounds for temporal pattern, T_{AB}

Minimum support time sequence of T_{AB}, ($\overrightarrow{T_{AB}^{min}}$) The minimum support time sequence of temporal itemset, T_{AB} is denoted by $\overrightarrow{T_{AB}^{min}}$ and can be computed using $\overrightarrow{T_{AB}^{min}} = \left(T_{AB_1}^{min}, T_{AB_2}^{min} \right)$ where $T_{AB_1}^{min} = \max(1-\bar{T}_{A_1}-\bar{T}_{B_1}, 0)$ and $T_{AB_2}^{min} = \max(1-\bar{T}_{A_2}-\bar{T}_{B_2}, 0)$. In the present case, we have $\bar{T}_{A_1} = 0.4$, $\bar{T}_{A_2} = 0.6$, $\bar{T}_{B_1} = 0.7$, $\bar{T}_{B_2} = 0.3$. So, $T_{AB_1}^{min} = maximum(1-\bar{T}_{A_1}-\bar{T}_{B_1}, 0) = \max$ (1–0.4-0.7, 0) = max (−0.1, 0) = 0. Similarly, $T_{AB_2}^{min} = maximum(1-\bar{T}_{A_2}-\bar{T}_{B_2}, 0) = \max$ (1–0.6 – 0.3, 0) = max (0.1, 0) = 0.1. Hence, $\overrightarrow{T_{AB}^{min}} = (\ 0.0, 0.1\ )$.

It can be verified from Fig. 3, that the true support sequence of temporal itemset, T_{AB} lies between the maximum possible support sequence ($\overrightarrow{T_{AB}^{max}}$) and minimum possible support sequence ($\overrightarrow{T_{AB}^{min}}$) as represented by the shaded region. The shaded region in Fig. 3 is used to represent the fact that the true support of temporal pattern, T_{AB} can only belong to this region.

3.2.2 Prevalence time sequence bounds for temporal pattern, T_{AC}

The maximum and minimum bound support sequences for temporal pattern, T_{AC} can be obtained by applying Eqs. (4) and (5) as discussed below.

Maximum support time sequence of T_{AC},($\overrightarrow{T_{AC}^{max}}$) The maximum temporal support sequence of temporal itemset, T_{AC} is denoted by $\overrightarrow{T_{AC}^{max}}$ and is computed using $\overrightarrow{T_{AC}^{max}} = \left(T_{AC_1}^{max}, T_{AC_2}^{max} \right)$ where

$T^{max}_{AC_1} = T_{A_1} - \max(1 - \bar{T}_{A_1} - T_{C_1}, 0)$ and $T^{max}_{AC_2} = T_{A_2} - \max(1 - \bar{T}_{A_2} - T_{C_2}, 0)$. Here, T_{A_1} =0.6, T_{A_2} = 0.4, T_{C_1} =0.8, T_{C_2} = 0.8, $\bar{T}_{A_1}$ = 0.4, $\bar{T}_{A_2}$ = 0.6, $\bar{T}_{C_1}$ = 0.2, $\bar{T}_{C_2}$ = 0.2. So, $T^{max}_{AC_1} = T_{A_1} - \max(1 - \bar{T}_{A_1} - T_{C_1}, 0)$ = 0.6 – maximum (1–0.4-0.8, 0) = 0.6- maximum (−0.2, 0) = 0.6. Similarly, $T^{max}_{AC_2} = T_{A_2} - \max(1 - \bar{T}_{A_2} - T_{C_2}, 0)$ = 0.4 – maximum (1–0.6 – 0.8, 0) = 0.4 – maximum (−0.4, 0) = 0.4–0 = 0.4. Hence, the maximum possible support sequence of temporal pattern, T_{AC} is given by $\overrightarrow{T^{max}_{AC}} = (0.6, 0.4)$.

Minimum support time sequence of T_{AC}, ($\overrightarrow{T^{min}_{AC}}$) The minimum temporal support sequence of temporal itemset, T_{AC} is denoted by $\overrightarrow{T^{min}_{AC}}$ and can be computed using $\overrightarrow{T^{min}_{AC}} = \left(T^{min}_{AC_1}, T^{min}_{AC_2} \right)$ where $T^{min}_{AC_1} = \max(1 - \bar{T}_{A_1} - \bar{T}_{C_1}, 0)$ and $T^{min}_{AC_2} = \max(1 - \bar{T}_{A_2} - \bar{T}_{C_2}, 0)$. In the present example, we have $\bar{T}_{A_1} = 0.4$, $\bar{T}_{A_2} = 0.6$, $\bar{T}_{C_1} = 0.2$, $\bar{T}_{C_2} = 0.2$. So, $T^{min}_{AC_1} = maximum\ (1 - \bar{T}_{A_1} - \bar{T}_{C_1}, 0)$ = max (1–0.4-0.2, 0) = max (0.4, 0) = 0.4. Similarly, $T^{min}_{AC_2} = maximum\ (1 - \bar{T}_{A_2} - \bar{T}_{c_2}, 0)$ = max (1–0.6 – 0.2, 0) = max (0.2, 0) = 0.2.

Hence, $\overrightarrow{T^{min}_{AC}} = (0.4, 0.2)$. Figure 4, depicts that the true support sequence of temporal itemset, T_{AC} lies between the maximum possible support sequence ($\overrightarrow{T^{max}_{AC}}$) and minimum possible support sequence ($\overrightarrow{T^{min}_{AC}}$) as represented by the shaded region.

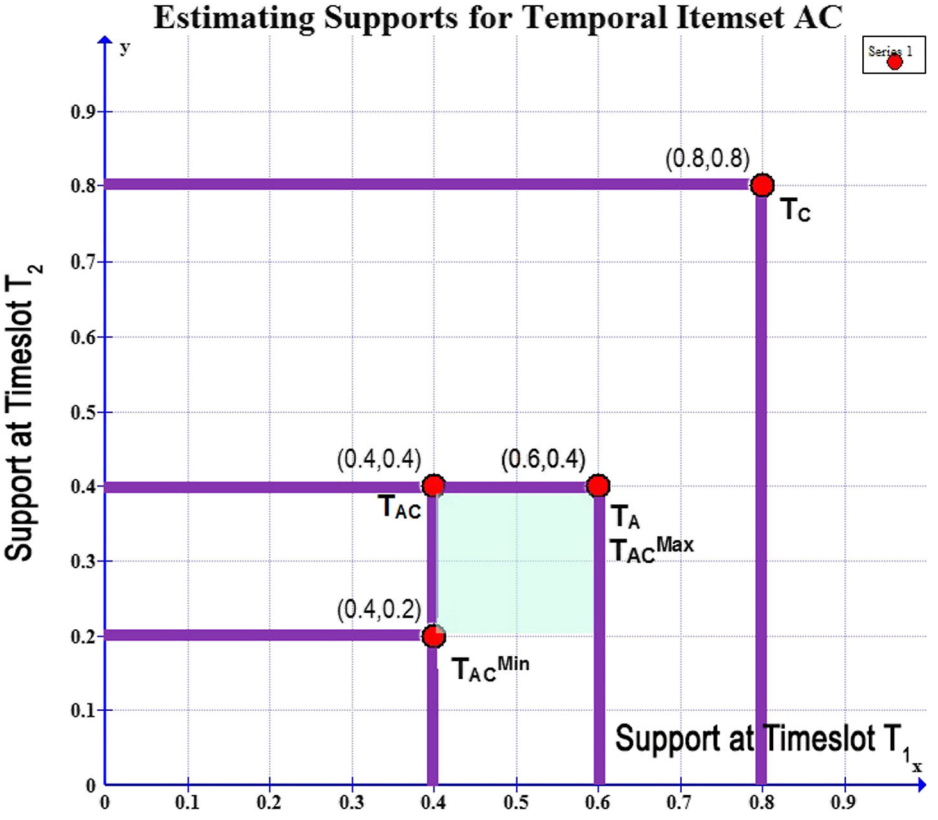

Fig. 4 Maximum Support Bound for temporal pattern, T_{AC}

3.2.3 Prevalence time sequence bounds for temporal pattern, T_{BC}

The prevalence sequence bounds of temporal pattern, T_{BC} can be obtained by applying Eqs. (4) and (5) as explained below.

Maximum support time sequence of T_{BC}, ($\overrightarrow{T^{max}_{BC}}$) The temporal support sequence of temporal itemset, T_{BC} is denoted by $\overrightarrow{T^{max}_{BC}}$ and can be computed using $\overrightarrow{T^{max}_{BC}} = \left(T^{max}_{BC_1}, T^{max}_{BC_2} \right)$ where $T^{max}_{BC_1} = T_{B_1} - \max(1 - \bar{T}_{B_1} - T_{C_1}, 0)$ and $T^{max}_{BC_2} = T_{B_2} - \max(1 - \bar{T}_{B_2} - T_{C_2}, 0)$. In our case, $T_{B_1} = 0.3$, $T_{B_2} = 0.7$, $T_{C_1} = 0.8$, $T_{C_2} = 0.8$, $\bar{T}_{B_1} = 0.7$, $\bar{T}_{B_2} = 0.3$, $\bar{T}_{C_1} = 0.2$, $\bar{T}_{C_2} = 0.2$. So, $T^{max}_{BC_1} = T_{B_1} - \max(1 - \bar{T}_{B_1} - T_{C_1}, 0) = 0.3 -$ maximum (1–0.7-0.8, 0) = 0.3- maximum (−0.5, 0) = 0.3–0 = 0.3. Also, $T^{max}_{BC_2} = T_{B_2} - \max(1 - \bar{T}_{B_2} - T_{C_2}, 0) = 0.7 -$ maximum (1–0.3 – 0.8, 0) = 0.7 – maximum (−0.1, 0) = 0.7–0 = 0.7. Hence, the maximum possible support sequence of temporal pattern, T_{BC} is given by $\overrightarrow{T^{max}_{BC}} = (0.3, 0.7)$.

Minimum support time sequence of T_{BC}, ($\overrightarrow{T^{min}_{BC}}$) The minimum temporal support sequence of temporal itemset, T_{BC} is denoted by $\overrightarrow{T^{min}_{BC}}$ and can be computed using $\overrightarrow{T^{min}_{BC}} = \left(T^{min}_{BC_1}, T^{min}_{BC_2} \right)$ where $T^{min}_{BC_1} = \max(1 - \bar{T}_{B_1} - \bar{T}_{C_1}, 0)$ and $T^{min}_{BC_2} = \max(1 - \bar{T}_{B_2} - \bar{T}_{C_2}, 0)$. Here, $\bar{T}_{B_1} = 0.7$, $\bar{T}_{B_2} = 0.3$, $\bar{T}_{C_1} = 0.2$, $\bar{T}_{C_2} = 0.2$. So, $T^{min}_{BC_1} = maximum(1 - \bar{T}_{B_1} - \bar{T}_{C_1}, 0) =$ max (1–0.7-0.2, 0) = max (0.1, 0) = 0.1. Similarly, $T^{min}_{BC_2} = maximum(1 - \bar{T}_{B_2} - \bar{T}_{C_2}, 0) =$ max (1–0.3 – 0.2, 0) = max (0.5, 0) = 0.5. The minimum support time sequence is hence given by $\overrightarrow{T^{min}_{BC}} = (0.1, 0.5)$. The shaded region in Fig. 5 is used to represent the fact that the true support of temporal pattern, T_{BC} can only belong to this region.

3.2.4 Prevalence time sequence bounds for temporal pattern, T_{ABC}

Consider the temporal itemset, T_{ABC}. The prevalence sequence bounds of temporal association pattern, T_{ABC} can be obtained by applying Eqs. (6) to (9). Figure 6 shows the maximum possible support sequence and minimum possible support sequence for temporal association pattern, T_{ABC}.

Maximum support time sequence of T_{ABC}, $\left(T^{max}_{ABC}\right)$ The maximum support sequence of temporal association pattern, T_{ABC} at level-3 is computed by considering all possible size-2 subset patterns of level-2 and singleton patterns at level-1. This gives us three cases.

Case-1: k = 1, $Ss^1(ABC) = AB$, $S(ABC) = C$ i.e. $T_{Ss^1(ABC)} = T_{AB}$ and $T_{S(ABC)} = T_C$

$$\begin{aligned}\overrightarrow{T^1_{ABC}} &= \left(T^1_{ABC_1}, T^1_{ABC_2} \right) \\ &= \left(T_{AB_1} - \max\left\{ \left(1 - \overline{T_{AB_1}} - T_{C_1} \right), 0 \right\}, T^1_{AB_2} - \max\left\{ \left(1 - \overline{T_{AB_2}} - T_{C_2} \right), 0 \right\} \right) \\ &= (0.3 - \max\{(1 - 0.7 - 0.8), 0\}, 0.3 - \max\{(1 - 0.7 - 0.8), 0\}) = (0.3, 0.3)\end{aligned}$$

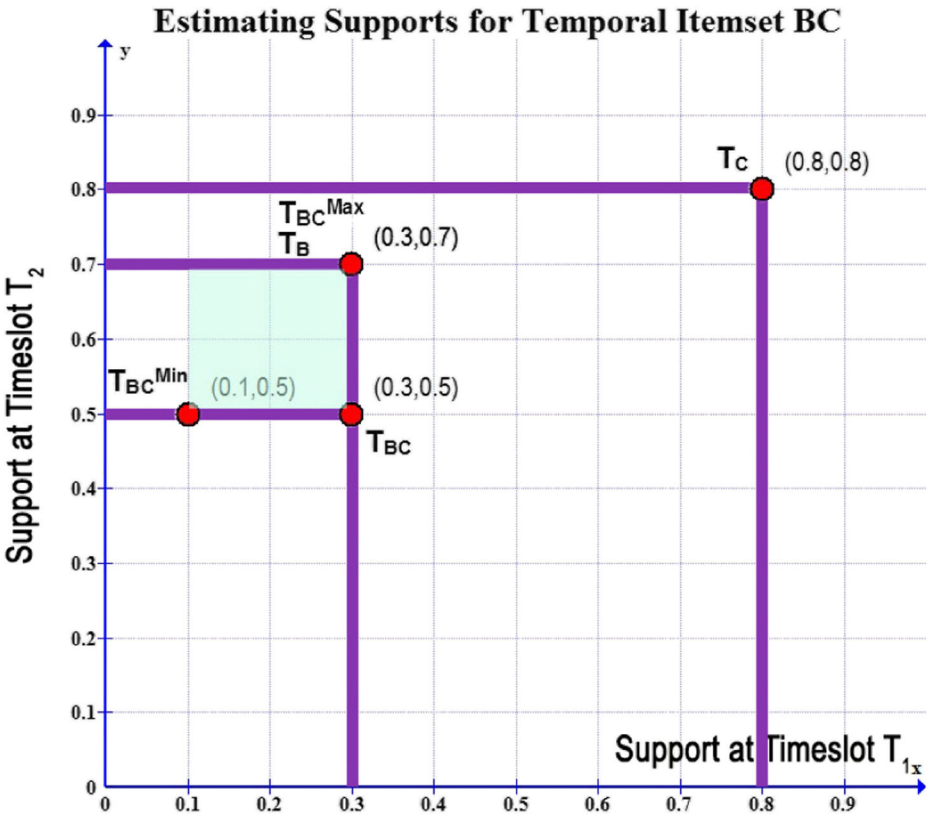

Fig. 5 Support Bound for temporal pattern, T_{BC}

Case-2: k = 2, $Ss^2(ABC) = AC$, $S(ABC) = B$ i.e. $T_{Ss^2(ABC)} = T_{AC}$ and $T_{S(ABC)} = T_B$

$$\overrightarrow{T^2_{ABC}} = \left(T^2_{ABC_1}, T^2_{ABC_2}\right)$$
$$= \left(T_{AC_1} - \max\left\{\left(1 - \overline{T_{AC_1}} - T_{B_1}\right), 0\right\}, T_{AC_2} - \max\left\{\left(1 - \overline{T_{AC_2}} - T_{B_2}\right), 0\right\}\right)$$
$$= (0.4 - \max\{(1 - 0.6 - 0.3), 0\}, 0.4 - \max\{(1 - 0.6 - 0.7), 0\}) = (0.3, 0.4)$$

Case-3: k = 3, $Ss^3(ABC) = BC$, $S(ABC) = A$ i.e. $T_{Ss^3(ABC)} = T_{BC}$ and $T_{S(ABC)} = T_A$

$$\overrightarrow{T^3_{ABC}} = \left(T^3_{ABC_1}, T^3_{ABC_2}\right)$$
$$= \left(T_{BC_1} - \max\left\{\left(1 - \overline{T_{BC_1}} - T_{A_1}\right), 0\right\}, T_{BC_2} - \max\left\{\left(1 - \overline{T_{BC_2}} - T_{A_2}\right), 0\right\}\right)$$
$$= (0.3 - \max\{(1 - 0.7 - 0.6), 0\}, 0.5 - \max\{(1 - 0.5 - 0.4), 0\}) = (0.3, 0.4)$$
$$\overrightarrow{T_{ABC}^{max}} = \left(T^{max}_{ABC_1}, T^{max}_{ABC_2}\right) = (\min(0.3, 0.3.0.3), \min(0.3, 0.4.0.4)) = (0.3, 0.3)$$

$$So, \quad \overrightarrow{T_{ABC}^{max}} = \left(T^{max}_{ABC_1}, T^{max}_{ABC_2}\right) = (0.3, 0.3)$$

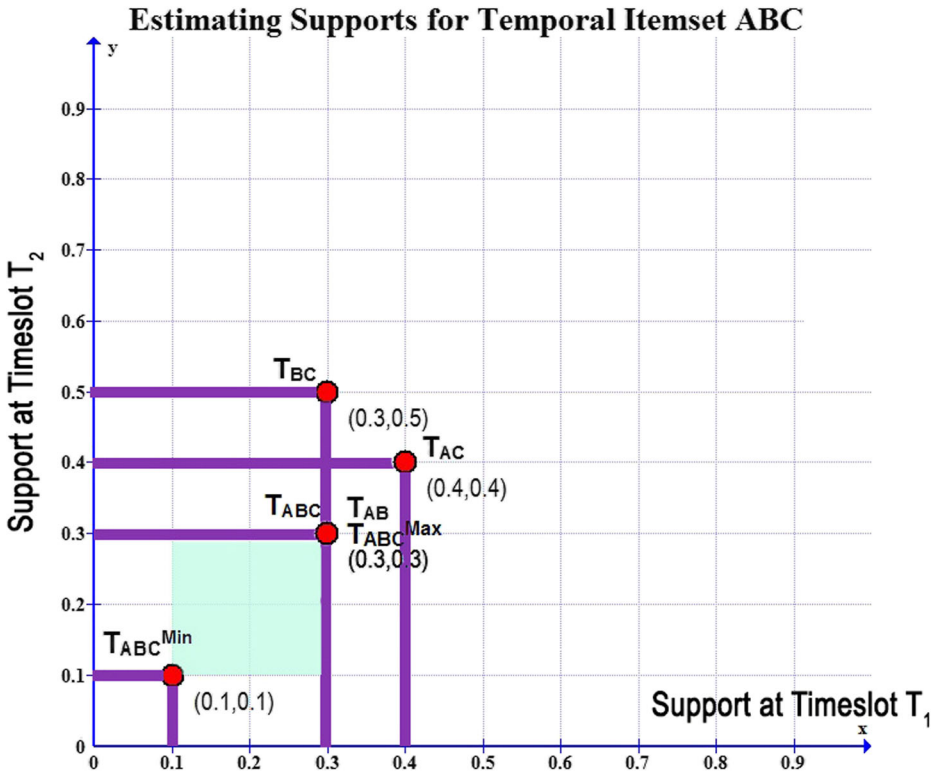

Fig. 6 Support bounds for temporal pattern, T_{ABC}

Minimum support time sequence of T_{ABC}, $(\overrightarrow{T^{min}_{ABC}})$ The minimum support sequence bound of temporal association pattern, T_{ABC} at level-3 is computed by considering all possible size-2 subset patterns of level-2 and singleton patterns at level-1. This gives us three cases

Case-1: k = 1, $Ss^1(ABC) = AB$, $S(ABC) = C$ i.e. $T_{Ss^1(ABC)} = T_{AB}$ and $T_{S(ABC)} = T_C$

$$\begin{aligned}\overrightarrow{T^1_{ABC}} &= \left(T^1_{ABC_1}, T^1_{ABC_2}\right)\\ &= \left(\max\left\{\left(1-\overline{T_{AB_1}}-\bar{T}_{C_1}\right),0\right\}, T_{AB_2}-\max\left\{\left(1-\overline{T_{AB_2}}-\bar{T}_{C_2}\right),0\right\}\right)\\ &= (\max\{(1-0.7-0.2),0\}, \max\{(1-0.7-0.2),0\}) = (0.1, 0.1)\end{aligned}$$

Case-2: k = 2, $Ss^2(ABC) = AC$, $S(ABC) = B$ i.e. $T_{Ss^2(ABC)} = T_{AC}$ and $T_{S(ABC)} = T_B$

$$\begin{aligned}\overrightarrow{T^2_{ABC}} &= \left(T^2_{ABC_1}, T^2_{ABC_2}\right)\\ &= \left(\max\left\{\left(1-\overline{T_{AC_1}}-\bar{T}_{B_1}\right),0\right\}, \max\left\{\left(1-\overline{T_{AC_2}}-\bar{T}_{B_2}\right),0\right\}\right)\\ &= (\max\{(1-0.6-0.7),0\}, \max\{(1-0.6-0.3),0\}) = (0.0, 0.1)\end{aligned}$$

Case-3: k = 3, $Ss^3(ABC) = BC$, $S(ABC) = A$ i.e. $T_{Ss^3(ABC)} = T_{BC}$ and $T_{S(ABC)} = T_A$

$$\overrightarrow{\mathrm{T}^3_{\mathrm{ABC}}} = \left(T^3_{ABC_1}, T^3_{ABC_2}\right)$$
$$= \left(\max\left\{\left(1-\overline{T_{BC_1}}-\bar{T}_{A_1}\right), 0\right\}, \max\left\{\left(1-\overline{T_{BC_2}}-\bar{T}_{A_2}\right), 0\right\}\right)$$
$$= (\max\{(1-0.7-0.4), 0\}, \max\{(1-0.5-0.6), 0\}) = (0.0, 0.0)$$
$$\overrightarrow{T_{ABC}{}^{min}} = \left(T^{min}_{ABC_1}, T^{min}_{ABC_2}\right) = (\max(0.1, 0.0, 0.0), \max(0.1, 0.1, 0)) = (0.1, 0.1)$$

So, $\overrightarrow{T_{ABC}{}^{min}} = \left(T^{min}_{ABC_1}, T^{min}_{ABC_2}\right)$ =(0.1, 0.1). Thus, the minimum and maximum possible support sequences of temporal association pattern, $\mathrm{T_{ABC}}$ are $\overrightarrow{T_{ABC}{}^{min}}$ = (0.1, 0.1) and $\overrightarrow{T_{ABC}{}^{max}}$ = (0.3, 0.3). The true support of temporal pattern, $\mathrm{T_{ABC}}$ is (0.3, 0.3).

4 Temporal dissimilarity measure

Problem Definition: Given $'\Delta'$, T_P, and T_q then, two temporal patterns T_P, T_q are considered as similar, if the dissimilarity value denoted by $D^{true}_{T_P,T_q}$ does not exceed, Δ^g.

$$\text{i.e}\, D^{true}_{T_p,T_q} \leq \Delta^g \tag{10}$$

4.1 Proposed dissimilarity measure (ASTRA)

Let $\mathrm{T_p}$ and $\mathrm{R_r}$ be the temporal and reference pattern and their respective prevalence values at $\mathrm{k^{th}}$ time slot are denoted by T_{P_k}, R_{r_k}. Their corresponding prevalence time sequences over 'm' time slots are represented using $\overrightarrow{T_P}$ = (T_{P_1}, T_{P_2}, T_{P_3}, ……., T_{P_m}) and $\overrightarrow{R_r}$ = (R_{r_1}, R_{r_2}, R_{r_3}, ……., R_{r_m}). The dissimilarity measure introduced in this section is motivated from the basic Gaussian membership function [34, 50] and is extended using [11, 65, 68, 78–80]. The dissimilarity measure is defined using Eq. (11),

$$D^{true}_{T_P,R_r} = \frac{1-\mathcal{M}^{T_p}_{R_r}}{1.3679} \tag{11}$$

where, $\mathcal{M}^{T_p}_{R_r}$ is the membership function given by Eq. (12)

$$\mathcal{M}^{T_p}_{R_r} = \prod_{k=1}^{k=m} e^{-\frac{1}{m}\left(\frac{T_{p_k}-R_{r_k}}{\sigma^g}\right)^2} = e^{-\frac{1}{m}*\left(\frac{\Delta}{\sigma^g}\right)^2} = e^{-\left(\frac{\Delta_n}{\sigma^g}\right)^2} \tag{12}$$

In Eq. (12), σ^g, Δ and Δ_n denote standard deviation, Euclidean distance and normalized Euclidean distance between temporal patterns which can be obtained by applying Eqs. (13) and (14).

$$\sigma^g = \frac{\Delta}{\sqrt{ln_e\left(\frac{1}{abs\left(1-1.3679^*\Delta\right)}\right)}} \tag{13}$$

$$\Delta_n = \frac{\sqrt{\left(T_{p_1}-R_{r_1}\right)^2+\left(T_{p_2}-R_{r_2}\right)^2+\ldots\ldots\ldots+\left(T_{p_m}-R_{r_m}\right)^2}}{\sqrt{m}} \cong \frac{\Delta}{\sqrt{m}} \tag{14}$$

Let $'\Delta'$ be the threshold specified in Euclidean space which represents allowable dissimilarity limit between temporal pattern and reference pattern. To find the similarity profiled temporal patterns, the value of $'\Delta'$ in Euclidean space is projected to a new space whose value is computed using expression for Δ^g given by Eq. (15).

$$\Delta^g = \frac{1-e^{-\left(\frac{\Delta}{\sigma^g}\right)^2}}{1.3679} \tag{15}$$

4.2 Threshold and deviation

4.2.1 Threshold in Gaussian Space

Let Δ is the dissimilarity threshold specified by the user and σ^g be the standard deviation. The transformed threshold for new space can be obtained directly from Eq. (11) and is denoted by, Δ^g given by Eq. (16).

$$\Delta^g = \frac{1-e^{-\left(\frac{\Delta}{\sigma^g}\right)^2}}{1.3679} \tag{16}$$

4.2.2 Standard deviation

The Euclidean distance between temporal pattern, T_p and reference, R_r considering support at k^{th} time slot is given by Eq. (17),

$$\Delta = \sqrt{\left(T_{p_k}-R_{r_k}\right)^2} = \left(T_{p_k}-R_{r_k}\right) \tag{17}$$

The Euclidean distance between temporal pattern and reference pattern considering supports for 'm' time slots is given by Eq. (18) and its normalized distance (Δ_n) value is given by Eq. (19). The normalized Euclidean distance always lies between 0 and 1.

$$\Delta = \sqrt{\left(T_{p_1}-R_{r_1}\right)^2+\left(T_{p_2}-R_{r_2}\right)^2+\ldots\ldots\ldots+\left(T_{p_m}-R_m\right)^2} \tag{18}$$

$$\Delta_n = \frac{\Delta}{\sqrt{m}} \tag{19}$$

Equation (20) gives the true dissimilarity value between temporal and reference pattern considering support values for 'm' time slots using the proposed measure

$$D^{true}_{T_p, R_r} = \frac{1 - \mathcal{M}^{T_p}_{R_r}}{1.3679} \cong \frac{1 - e^{-\left(\frac{\Delta_n}{\sigma^g}\right)^2}}{1.3679} \tag{20}$$

Equating (19) and (20), we have Eq. (21)

$$\frac{1-e^{-\left(\frac{\Delta_n}{\sigma^g}\right)^2}}{1.3679} = \Delta_n \tag{21}$$

Solving Eq. (21), the expression for deviation is given by Eq. (22),

$$\sigma^g = \frac{\Delta_n}{\sqrt{log_e\left(\frac{1}{1-1.3679^{*}\Delta_n}\right)}} \tag{22}$$

5 Algorithm for time profiled association mining (MASTER)

5.1 Algorithm design concept

Two major challenges that must be addressed when devising the algorithm for time profiled temporal association mining are i) Computational space (pattern search space) and ii) Computational cost (execution time). The first issue is due to the enormous number of itemset associations that must be considered for mining similar time-profiled associations. The second issue is since the execution time becomes intractable in the process of validating itemset associations for similarity. Our algorithm for time profiled temporal association mining addresses these challenges by

a) Introducing approach for estimating support limits of temporal itemset support time sequences
b) Reducing search space of temporal associations by devising a similarity function that holds the monotonicity property.

Approaches for estimation of support time sequences of itemset associations in a time stamped transaction database are also proposed by Jin Soung Yoo [84–86], Calders [16]. Inspired from the work of Jin Soung Yoo [84–87] some of our previous research [11, 68, 78, 79] have addressed estimation of prevalence values. In this paper, we use the support estimation approach discussed in section-3.

5.1.1 Limits of support time sequences

Computing limits or covers of support time sequences of temporal pattern is addressed in section-3 supported by a detailed case study.

5.1.2 Lower bounding distance

The computation cost of temporal pattern mining process can be reduced if we can somehow prune all the invalid temporal association patterns (i.e those temporal patterns whose dissimilarity value to reference exceeds user threshold) much ahead in the pattern mining process. This objective is achieved through computing the minimum dissimilarity bound value in MASTER (pattern mining algorithm) that uses the proposed measure. The basic idea is to find the value of minimum dissimilarity bound for a given temporal pattern (w.r.t reference) and if this value exceeds the threshold limit, then the temporal pattern is pruned. This is because whenever the minimum bound dissimilarity value exceeds the dissimilarity threshold then, its true dissimilarity also exceeds the threshold limit.

Definition-1 Given a reference support sequence, $\overrightarrow{R} = (R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots\ldots, R_{r_m})$ and the maximum possible prevalence sequence of an item set, $\overrightarrow{T_I^{max}} = (T_{I_1}^{max}, T_{I_2}^{max}, T_{I_3}^{max}, \ldots\ldots\ldots\ldots\ldots\ldots, T_{I_m}^{max})$. let $\overrightarrow{R^U} = (R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots\ldots, R_{r_w})$ and $\overrightarrow{T_I^L} = (T_{I_1}^{max}, T_{I_2}^{max}, \ldots\ldots\ldots\ldots\ldots\ldots, T_{I_w}^{max})$ be the subsequences of $\overrightarrow{R}$ and $\overrightarrow{T_I^{max}}$ respectively, where $R_{r_t} > T_{I_t}^{max}$; $1 \le t \le w$. The maximum possible minimum dissimilarity value between temporal patterns, $\overrightarrow{R}$ and $\overrightarrow{T_I^{max}}$, $D^{ulb}\left(\overrightarrow{T_I^{max}}, \overrightarrow{R}\right)$ is defined as $D\left(\overrightarrow{T_I^L}, \overrightarrow{R^U}\right)$.

Explanation: Let $D\left(\overrightarrow{T_I}, \overrightarrow{R}\right)$ denote the true distance between temporal pattern and reference sequence. For example, when the dissimilarity function of section 4.1 is used then,

$$D\left(\overrightarrow{T_I}, \vec{R}\right) = \frac{1-\prod_{k=1}^{k=m} \mathcal{M}_{R r_k}^{T_{I_k}}}{1.3679} \tag{23}$$

In similar lines, the maximum possible minimum dissimilarity between true support sequence of temporal pattern and reference is

$$D^{ulb}\left(\overrightarrow{T_I^{max}}, \vec{R}\right) = D\left(\overrightarrow{T_I^L}, \overrightarrow{R^U}\right) = \frac{1-\prod_{k=1}^{k=m} \mathcal{M}_{R r_k}^{T_{I_k}{}^{max}}}{1.3679} = \frac{1-\mathcal{M}_{R r_k}^{T_{I_k}{}^{ULB}}}{1.3679} \tag{24}$$

where $\mathcal{M}_{R r_k}^{T_{I_k}{}^{max}} = exp^{-\left(\frac{R r_k - T_{I_k}{}^{max}}{\sigma g}\right)^2}$ if $R_{r_k} > T_{I_k}{}^{max}$ and is equal to 1 otherwise. The membership function for upper-lower distance bound can hence be considered as $\mathcal{M}_{R r_k}^{T_{I_k}{}^{ULB}}$.

Definition-2 Given a reference support sequence, $\overrightarrow{R} = (R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots\ldots, R_{r_m})$ and the minimum possible prevalence sequence of an item set, $\overrightarrow{T_I^{min}} = (T_{I_1}^{min}, T_{I_2}^{min}, T_{I_3}^{min}, \ldots\ldots\ldots\ldots\ldots\ldots, T_{I_m}^{min})$. let $\overrightarrow{R^L} = (R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots\ldots, R_{r_w})$ and $\overrightarrow{T_I^U} = (T_{I_1}^{min}, T_{I_2}^{min}, \ldots\ldots\ldots\ldots\ldots\ldots, T_{I_w}^{min})$ be the subsequences of $\overrightarrow{R}$ and $\overrightarrow{T_I^{min}}$ respectively, where $R_{r_t} < T_{I_t}^{min}$; $1 \le t \le w$. The minimum possible minimum dissimilarity value $D^{llb}\left(\overrightarrow{T_I^{min}}, \overrightarrow{R}\right)$ between temporal patterns, $\overrightarrow{R}$ and $\overrightarrow{T_I^{min}}$ is defined as $D\left(\overrightarrow{T_I^U}, \overrightarrow{R^L}\right)$.

Explanation: Let $D\left(\overrightarrow{T_I}, \overrightarrow{R}\right)$ denote the distance between temporal pattern and reference sequence. For example, when the dissimilarity function of section 4.1 is used then, true distance is given by

$$D\left(\overrightarrow{T_I}, \vec{R}\right) = \frac{1-\prod_{k=1}^{k=m} \mathcal{M}_{Rr_k}^{T_{I_k}}}{1.3679} \tag{25}$$

On similar lines, the lower-lower distance bound (or minimum possible minimum dissimilarity) is given by

$$D^{llb}\left(\overrightarrow{T_I^{min}}, \vec{R}\right) = D\left(\overrightarrow{T_I^{U}}, \overrightarrow{R^{L}}\right) = \frac{1-\mathcal{M}_{Rr_k}^{T_{I_k}\,LLB}}{1.3679} = \frac{1-\prod_{k=1}^{k=m} \mathcal{M}_{Rr_k}^{T_{I_k}\,min}}{1.3679} \tag{26}$$

where $\mathcal{M}_{R_{r_k}}^{T_{I_k}\,min} = exp^{-\left(\frac{R_{r_k}-T_{I_k}^{\,min}}{\sigma g}\right)^2}$ if $R_{r_k} < T_{I_k}^{\,min}$ and is equal to 1 otherwise.

The membership function for minimum possible minimum dissimilarity (or lower-lower bound) can hence be considered as $\mathcal{M}_{Rr_k}^{T_{I_k}\,LLB}$.

Definition-3 Given a reference support sequence, $\overrightarrow{R}$ = ($R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots\ldots, R_{r_m}$), maximum possible prevalence sequence of an item set, $\overrightarrow{T_I^{max}}$ = ($T_{I_1}^{max}, T_{I_2}^{max}, T_{I_3}^{max}, \ldots\ldots\ldots\ldots\ldots\ldots, T_{I_m}^{max}$), and minimum possible prevalence sequence of an item set, $\overrightarrow{T_I^{min}}$ = ($T_{I_1}^{min}, T_{I_2}^{min}, T_{I_3}^{min}, \ldots\ldots\ldots\ldots\ldots\ldots, T_{I_m}^{min}$), the minimum dissimilarity bound is defined by considering the resultant membership function obtained by considering the product of membership functions obtained when considering maximum possible minimum (upper-lower bound) dissimilarity and minimum possible minimum dissimilarity (lower-lower bound) bound computations. The minimum dissimilarity bound is formally represented as

$$D^{LB}\left(\overrightarrow{T_I^{max}}, \overrightarrow{T_I^{min}}, \overrightarrow{R_r}\right) = \frac{1-\mathcal{M}_{Rr_k}^{T_{I_k}\,LB}}{1.3679} = \frac{1-\mathcal{M}_{Rr_k}^{T_{I_k}\,ULB} \;X\; \mathcal{M}_{Rr_k}^{T_{I_k}\,LLB}}{1.3679} \tag{27}$$

Explanation: When computing the true distance, we consider support values of temporal and reference pattern for all 'm' time slots. However, the computation of lower bounding distance is a function of upper-lower and lower-lower distance bounds. When computing upper-lower distance bound, only those support values which satisfy $R_{r_m} > T_{I_m}^{max}$ at 'mth' time slot is considered and the distance is computed. Similarly, for lower-lower bound only those pattern support values which satisfy $R_{r_m} < T_{I_m}^{min}$ are considered and the distance is found. The minimum distance bound is computed by considering membership functions of both these distance bounds. We have, from definition-1 the membership function used as part of upper-lower distance bound computation given by

$$\mathcal{M}_{R_{r_k}}^{T_{I_k}\,ULB} = \prod_{k=1}^{k=m} \mathcal{M}_{Rr_k}^{T_{I_k}\,max} = \prod_{k=1}^{k=m} \begin{cases} exp^{-\left(\frac{R_{r_k}-T_{I_k}^{\,max}}{\sigma g}\right)^2} & ;\ R_{r_k} > T_{I_k}^{\,max} \\ 1 & ;\ R_{r_k} \le T_{I_k}^{\,max} \end{cases} \tag{28}$$

From definition-2, the membership function used as part of lower-lower distance bound computation given by

$$\mathcal{M}_{R_{r_k}}^{T_{I_k}^{LLB}} = \prod_{k=1}^{k=m} \mathcal{M}_{R_{r_k}}^{T_{I_k}^{min}} = \prod_{k=1}^{k=m} \begin{cases} exp^{-\left(\frac{R_{r_k} - T_{I_k}^{min}}{\sigma g}\right)^2} ; R_{r_k} < T_{I_k}^{min} \\ 1 \qquad\qquad\qquad\quad ; R_{r_k} \geq T_{I_k}^{min} \end{cases} \tag{29}$$

The membership function for minimum dissimilarity bound is given by Eq. (30)

$$\mathcal{M}_{R_{r_k}}^{T_{I_k}^{LB}} = \mathcal{M}_{R_{r_k}}^{T_{I_k}^{ULB}} \; X \quad \mathcal{M}_{R_{r_k}}^{T_{I_k}^{LLB}} \tag{30}$$

The expression for minimum dissimilarity bound is hence given by Eq. (31)

$$D^{LB}\left(\overrightarrow{T_I^{max}}, \overrightarrow{T_I^{min}}, \overrightarrow{R_r}\right) = \frac{1 - \mathcal{M}_{R_{r_k}}^{T_{I_k}^{LB}}}{1.3679} \tag{31}$$

Lemma-1 Given the maximum possible prevalence sequence, $\overrightarrow{T_I^{max}} = (\, T_{I_1}^{max}, T_{I_2}^{max}, T_{I_3}^{max}, \ldots\ldots\ldots\ldots\ldots, T_{I_m}^{max}\,)$, minimum possible prevalence sequence $\overrightarrow{T_I^{min}} = (\, T_{I_1}^{min}, T_{I_2}^{min}, T_{I_3}^{min}, \ldots\ldots\ldots\ldots\ldots, T_{I_m}^{min}\,)$, true support sequence, $\overrightarrow{T_I} = (T_{I_1}, T_{I_2}, T_{I_3}, \ldots\ldots\ldots\ldots, T_{I_m})$ of temporal pattern T_I and a reference temporal pattern, $\overrightarrow{R} = (R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots\ldots, R_{r_m})$. The lower bounding distance and true distance holds the inequality, $D^{LB}\left(\overrightarrow{T_I^{max}}, \overrightarrow{T_I^{min}}, \overrightarrow{R_r}\right) \leq D^{true}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right)$, if the proposed dissimilarity measure in section 4.1 is used as a similarity function.

Proof:

According to definition of lower-bounding distance using proposed dissimilarity measures, it is known that

$$D^{LB}\left(\overrightarrow{T_I^{max}}, \overrightarrow{T_I^{min}}, \overrightarrow{R_r}\right) = \frac{1 - \mathcal{M}_{R_{r_k}}^{T_{I_k}^{LB}}}{1.3679} = \frac{1 - \mathcal{M}_{R_{r_k}}^{T_{I_k}^{ULB}} \; X \; \mathcal{M}_{R_{r_k}}^{T_{I_k}^{LLB}}}{1.3679}$$
$$= \frac{1 - \prod_{k=1}^{k=m} \begin{cases} exp^{-\left(\frac{R_{r_k} - T_{I_k}^{max}}{\sigma g}\right)^2} ; R_{r_k} > T_{I_k}^{max} \\ 1 \qquad\qquad\qquad\quad ; R_{r_k} \leq T_{I_k}^{max} \end{cases} \; X \; \prod_{k=1}^{k=m} \begin{cases} exp^{-\left(\frac{R_{r_k} - T_{I_k}^{min}}{\sigma g}\right)^2} ; R_{r_k} < T_{I_k}^{min} \\ 1 \qquad\qquad\qquad\quad ; R_{r_k} \geq T_{I_k}^{min} \end{cases}}{1.3679} \tag{32}$$

$$\text{i.e} D^{LB}\left(\overrightarrow{T_I^{max}}, \overrightarrow{T_I^{min}}, \overrightarrow{R_r}\right) = \frac{1 - \prod_{k=1}^{k=m} \begin{cases} exp^{-\left(\frac{R_{r_k} - T_{I_k}^{max}}{\sigma g}\right)^2} ; R_{r_k} > T_{I_k}^{max} \\ 1 \qquad\qquad\qquad\quad ; R_{r_k} \leq T_{I_k}^{max} \end{cases} \; X \; \prod_{k=1}^{k=m} \begin{cases} exp^{-\left(\frac{R_{r_k} - T_{I_k}^{min}}{\sigma g}\right)^2} ; R_{r_k} < T_{I_k}^{min} \\ 1 \qquad\qquad\qquad\quad ; R_{r_k} \geq T_{I_k}^{min} \end{cases}}{1.3679}$$
$$\leq \frac{1 - \prod_{t=1}^{t=m} \begin{cases} exp^{-\left(\frac{R_{r_t} - T_{I_t}}{\sigma g}\right)^2} ; R_{r_k} < T_{I_k}^{min} \\ 1 \qquad\qquad\qquad\quad ; else \end{cases} \; X \; \prod_{t=1}^{t=m} \begin{cases} exp^{-\left(\frac{R_{r_t} - T_{I_t}}{\sigma g}\right)^2} ; R_{r_t} < T_{I_t} \\ 1 \qquad\qquad\qquad\quad ; else \end{cases}}{1.3679}$$
$$= \frac{1 - \prod_{t=1}^{t=m} exp^{-\left(\frac{R_{r_t} - T_{I_t}}{\sigma g}\right)^2}}{1.3679} = D^{true}\left(\overrightarrow{T_p}, \overrightarrow{R_r}\right) \tag{33}$$

Thus, the above lemma also holds good for the proposed dissimilarity measure.

5.1.3 Monotonicity property of maximum-minimum dissimilarity

Monotonicity property of the support (or prevalence) measure is the most popular technique which is used to reduce the search space of itemset [11, 68, 84–87]. The support values of all possible superset temporal itemset (or patterns) of a given itemset cannot be greater than item set's support values. Hence, according to monotonicity property of support measure, if a temporal itemset does not satisfy support threshold, then all its superset temporal itemset can also be pruned [11, 86]. If we can come up with an interest measure (or dissimilarity measure) which has the property that is similar to monotonicity then, the search space of temporal itemset can be reduced, thus achieving computational efficiency. The supporting argument or proof is discussed below.

Lemma-2 The prevalence time sequence of temporal patterns (or association patterns) decreases with size of the temporal pattern at each disjoint time slot. i.e. the prevalence value is monotonically non-increasing.

Proof: The prevalence sequence of a temporal pattern is obtained by considering the prevalence values obtained from disjoint set of transactions for each time slot. As the size of temporal pattern increases, the prevalence value of a temporal pattern (or itemset) decreases w.r.t each time slot. Prevalence sequences obtained for all possible temporal patterns hold this property. For example, if T_I and T_J are two temporal patterns such that $J \subseteq I$, then prevalence $(T_I) \leq$ prevalence (T_J).

Lemma-3 The upper-lower dissimilarity bound (maximum possible minimum distance) between the true support time sequence of temporal pattern and reference sequence monotonically increases with respect to the size of the temporal itemset.

Proof:

Here, we outline the generalized proof for the monotonicity property of maximum possible minimum bound dissimilarity value to true prevalence time sequence w.r.t dissimilarity measure introduced in section 4.1.

Let, $\overrightarrow{R} = (R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots\ldots, R_{r_m})$ and $\overrightarrow{T_I} = (T_{I_1}, T_{I_2}, T_{I_3}, \ldots\ldots\ldots\ldots, T_{I_m})$ be the reference and temporal pattern support time sequences of a size-k itemset, I then the maximum possible minimum dissimilarity value is given by Eq. (34)

$$D^{ulb}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) = \frac{1-\prod_{k=1}^{k=m} \begin{cases} exp^{-\left(\frac{R_{r_k} - T_{I_k}{}^{max}}{\sigma^g}\right)^2} & ; R_{r_k} > T_{I_k}{}^{max} \\ 1 & ; R_{r_k} \leq T_{I_k} \end{cases}}{1.3679} \tag{34}$$

Consider the size, (k + 1) item set, $I' = I \cup \{ i' \}$ where $I' \notin I$. The prevalence time sequence of this temporal pattern is denoted by $\overrightarrow{T_{I'}} = (T_{I'_1}, T_{I'_2}, T_{I'_3}, \ldots\ldots\ldots\ldots, T_{I'_m})$. From lemma-2, it is known that the prevalence value of a temporal pattern shows non-increasing behavior with the increase in pattern size i.e. $(T_{I'_t}) \leq (T_{I_t})$ for any t^{th} time slot. This holds true for all time slots in case of time stamped temporal database.

So, for any time slot 't', the prevalence value of superset temporal pattern is less than or equal to its subset temporal patterns. So, if $T_{I'_t} \leq T_{I_t}$, $T_{I_t} < R_{r_t}$ and $T_{I'_t} < R_{r_t}$ then, $R_{r_t} - T_{I_t} \leq R_{r_t} - T_{I'_t}$. This means that

$$\frac{1-\prod_{t=1}^{t=m}\begin{cases} exp^{-\left(\frac{T_{I_t}-R_{r_t}}{\sigma g}\right)^2}; R_{r_k} > {T_{I_k}}^{max} \\ 1 \qquad\qquad ; R_{r_k} \le T_{I_k} \end{cases}}{1.3679} \le \frac{1-\prod_{t=1}^{t=m}\begin{cases} exp^{-\left(\frac{T_{I'_t}-R_{r_t}}{\sigma g}\right)^2}; R_{r_t} > T_{I'_t} \\ 1 \qquad\qquad ; R_{r_t} \le T_{I'_t} \end{cases}}{1.3679} \tag{35}$$

i.e. $D^{ulb}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) \le D^{ulb}\left(\overrightarrow{T_{I'}}, \overrightarrow{R_r}\right)$.

On similar lines, it can also be proved that, $D^{ulb}\left(\overrightarrow{T_I^{max}}, \overrightarrow{R_r}\right) \le D^{ulb}\left(\overrightarrow{T_{I'_t}^{max}}, \overrightarrow{R_r}\right)$. i.e. the monotonicity of maximum possible minimum dissimilarity to maximum possible prevalence sequence also holds good.

Example:

For example, consider the case study discussed in section-6, $D^{ulb}(T_A, R) = 0.1863$, $D^{ulb}(T_B, R) = 0.0518$, $D^{ulb}(T_C, R) = 0$, $D^{ulb}(T_{AB}, R) = 0.3806$, $D^{ulb}(T_{AC}, R) = 0.1863$, $D^{ulb}(T_{BC}, R) = 0.1$, $D^{ulb}(T_{ABC}, R) = 0.3806$ It can be verified that $D^{ulb}(T_{AB}, R) \ge D^{ulb}(T_A, R)$ and $D^{ulb}(T_B, R)$, $D^{ulb}(T_{AC}, R) \ge D^{ulb}(T_A, R)$ and $D^{ulb}(T_C, R)$, $D^{ulb}(T_{BC}, R) \ge D^{ulb}(T_C, R)$ and $D^{ulb}(T_B, R)$. Also, $D^{ulb}(T_{ABC}, R) \ge D^{ulb}(T_{AB}, R)$,$D^{ulb}(T_{AC}, R)$ and $D^{ulb}(T_{BC}, R)$.

This proves monotonicity of proposed dissimilarity measure.

5.1.4 Temporal pattern pruning

We apply the pruning strategies like [11, 66–68, 78, 79, 86] but using proposed dissimilarity measure. Computational cost of pattern mining process is reduced by performing the pattern pruning process using minimum dissimilarity bound (lower bounding distance) and monotonicity of maximum possible minimum dissimilarity bound. Pattern pruning is divided into three strategies

Pruning using subset checkup The first strategy of pruning temporal patterns is through subset checkup. In this strategy, if the maximum possible minimum dissimilarity bound, $D^{ulb}\left(\overrightarrow{T_I^{max}}, \overrightarrow{R_r}\right)$ of any subset of a candidate temporal pattern is computed and if this dissimilarity value does not satisfy the threshold constraint then, the candidate temporal pattern is pruned by using the principle of monotonicity.

Pruning based on minimum dissimilarity bound The second strategy of pattern pruning is through computing minimum dissimilarity bound value of its maximum and minimum possible prevalence sequence. A candidate temporal pattern is pruned without the need for examining the true prevalence of temporal pattern, whenever its minimum dissimilarity bound exceeds the allowable dissimilarity limit.

Pruning using maximum possible minimum dissimilarity, $D^{ulb}\left(\overrightarrow{T_I^{max}}, \overrightarrow{R_r}\right)$ This strategy of pattern pruning is applied mainly to reduce the total number of next size candidate temporal patterns which are otherwise possible during pattern mining process. A candidate temporal pattern is pruned whenever the maximum possible minimum dissimilarity bound, $D^{ulb}\left(\overrightarrow{T_I^{max}}, \overrightarrow{R_r}\right)$ to

true prevalence sequence of temporal pattern exceeds the dissimilarity threshold value. In this case, the temporal pattern is not retained for generating higher size candidate temporal patterns.

5.2 Algorithm: MASTER- mining time profiled temporal associations

Input

Q : *Finite Set of all items*

TSDB : *Time stamped Temporal Database of Transactions*

R : *Reference Support Sequence*

$d_{Sim}^{(T,R)}$: *Dissimilarity measure*

δ : *User threshold*

δ^g : *Transformed threshold*

Output

Set of all valid temporal association patterns that are similar to reference w.r.t D^{true} *and* δ^g.

Variables

S : Itemset or Pattern Size

C_S : Finite set of Candidate Pattern or Itemset whose size is S

$Upper_S$: Finite set of upper bound support sequence of size-S association pattern

$Lower_S$: Finite set of lower bound support sequence of size-S association pattern

G_S : Support sequence set of Size-S pattern

G : Support sequences set of all possible subsets of patterns

J_S : Set of all association patterns of size-S, whose $D^{ulb} \leq \delta^g$

Z_S : Result set includes association patterns of size-S, whose True distance, $D^{true} \leq \delta^g$

Begin

1. S=1 and C_S = Q;
2. G_1 = generate_ true support sequences_ of_ temporal patterns(C_1, TSDB) ;
3. (Z_1 , J_1) = compute_ similar_ temporal patterns(C_1, G_1, TSDB, $d_{Sim}^{(T,R)}$, δ^g) ; // Singletons
4. S=2;
5. while (!empty(Z_{S-1}))

{

6. (C_S , $Upper_S$, $Lower_S$) = generate_candidate_pattern (J_{S-1} , G) ;
7. C_S = LB_BASED_PRUNE_CAND_PATTERN(C_s , $Upper_S$, $Lower_S$, $\vec{R}$, D^{ulb}, D^{llb}, δ^g);
8. G_S = generate-true-support-sequences-of-temporal patterns(C_S, TSDB) ;
9. (J_S , Z_S) = compute_ similar_ temporal patterns(C_S, G_S, R, D^{ulb}, *D,* δ^g);
10. G = {G} U {G_S}; S = S++;
11. }
12. return U(Z_1, Z_2, Z_3, Z_4, ……., Z_s);

End

Algorithm-2: Pruning_candidate_ temporal itemset_ based_ on_ minimum_ dissimilarity_ bound

LB_BASED_PRUNE_CAND_PATTERN (C_s , $Upper_S$, $Lower_S$, $\vec{R}$, D^{ulb}, D^{llb}, δ^g)

D^{ulb} : maximum-possible-minimum bound dissimilarity function
D^{llb} : minimum-possible-minimum bound dissimilarity function
{

For (each candidate temporal itemset , t ∈ C_s)
{
if ($D^{ulb}\left(\overrightarrow{T_g^{max}}, \overrightarrow{R_r}\right) + D^{llb}\left(\overrightarrow{T_g^{max}}, \overrightarrow{R_r}\right) > \delta^g$)
{
$C_s = C_s$ - t ;
}
}
return G_s ;
}

Algorithm-3 : Compute_ Similar_ Temporal_ Associations (C_S, G_S, R, D^{ulb}, D, δ^g)

D^{ulb} : maximum-possible-minimum bound dissimilarity function

D : dissimilarity function used to find the true distance

{

For (each candidate temporal itemset , t ∈ C_s)
{
if ($D^{ulb}\left(\overrightarrow{T_t^{max}}, \overrightarrow{R_r}\right) \leq \delta^g$)
{
$J_s = J_s$ U t ;
}
if ($D\left(\overrightarrow{T_l}, \vec{R}\right) \leq \delta^g$)
{
$Z_s = Z_s$ U t ;
}
}
return (J_s , Z_s) ;
}

Explanation:

Steps 1–3: Generate prevalence time sequences of singleton temporal items and then determine all similar time profiled singleton temporal items

Initially, all singleton items are candidate items and are of unit size. i.e. G_1 . We start with finding the true support sequences of singleton items. Then, each of these true support sequences ($\overrightarrow{G_1}$) are considered and the dissimilarity w.r.t reference sequence, $d_{Sim}^{(T,\ R)}$ is computed by applying the proposed dissimilarity measure, i.e. $D_{T,\ R}^{true}$ is

computed. All singleton temporal items that are found to be similar are retained and are candidate items for later stage. Also, singleton items that are dissimilar but whose upper-lower bounding distance $D^{ulb}\left(\overrightarrow{T}, \overrightarrow{R}\right)$ satisfies the dissimilarity threshold, δ^g are also retained and become the candidate temporal items for the next stage. It is to be noted that all items denoted by J_1 have their true distances satisfying the dissimilarity threshold and those denoted by Z_1 have their upper-lower distance value satisfying the threshold, δ^g. At the end of step-3, all retained items are stored in the result set.

Steps 4–8: Generate the candidate temporal itemset and their corresponding prevalence time sequence bounds. Perform pruning based on minimum dissimilarity distance bound

We move to next stage when the set J_{S-1} is not empty. This includes generating all candidate items of size (S > 1) denoted by G_S by considering the size, (S-1) itemset present in Z_{S-1}. In case, an item in G_{S-1} is not an element of Z_{S-1}, the candidate item is pruned by applying monotonicity property. On contrary, if the itemset in G_{S-1} belongs to $Z_{S-1,}$ then their prevalence time sequence bounds ($Upper_S$, $Lower_S$) are generated. From these prevalence time sequences, the minimum dissimilarity bound is computed. A candidate temporal itemset whose minimum dissimilarity bound exceeds the threshold limit shall be pruned by applying algorithm-2. Then, the support values of all candidate itemset are obtained for each time slot by performing the database scan and then their corresponding support time sequences are generated.

Step 9: Determine similar time profiled temporal associations by applying algorithm-3

The true distances between temporal itemsets and reference are computed and those temporal itemsets whose true distances satisfy the dissimilarity threshold condition are added to the result set R_k. For all itemset associations whose true dissimilarity exceeds threshold constraint but their corresponding maximum-minimum dissimilarity bound satisfies the dissimilarity threshold constraint, these itemsets are added to J_s for next stage candidate itemset generation. The size of the itemset denoted by S is incremented by 1 i.e. S = S + 1 and the procedure outlined in steps 6–10 is repeated until the set denoted by Z_s

5.2.1 Flow diagram of Temporal Association Mining Algorithm

Figures 7 and 8 represents the flow diagram to discover singleton and non-singleton time profiled temporal association patterns.

The dissimilarity measures proposed in our earlier works [11, 65, 68, 78–80] and any of the support estimation approaches in [21, 66, 67] may be used to obtain similar temporal association patterns. However, in this paper, we choose to use the proposed dissimilarity measure, ASTRA for retrieval of all similarity-profiled temporal associations.

5.3 Analytical analysis

The completeness and correctness of proposed dissimilarity measures for time stamped temporal databases and the computational analysis of the dissimilarity measures is discussed in the next subsection.

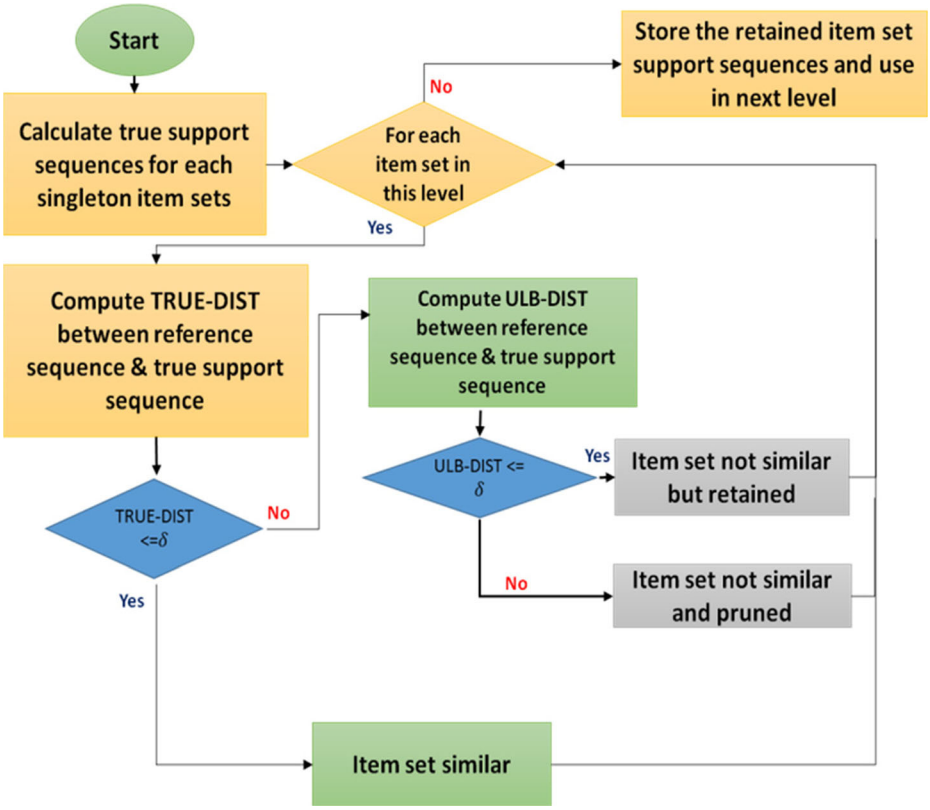

Fig. 7 Level-1 Temporal Patterns

5.3.1 Correctness and completeness

Correctness refers to dissimilarity measure whereas completeness refers to temporal pattern mining algorithm [86].

Correctness: Given an allowable dissimilarity limit (threshold) and a dissimilarity function, the distance value obtained between prevalence time sequences of all temporal patterns present in the result itemset and the reference must not exceed the allowable dissimilarity limit. This property is termed as "correctness" of the dissimilarity measure [85, 86].

Completeness: Given an allowable dissimilarity limit, the ability of the temporal pattern mining algorithm to output all valid temporal itemset (or temporal association patterns) whose prevalence time sequences vary similar to the given reference time sequence is called "completeness" of the algorithm [84–86].

Lemma-4:

Given a reference time sequence, $\overrightarrow{R_r}$ and an upper bound support time sequence of temporal pattern, $\overrightarrow{T_p}$. The upper-lower bounding distance, $D^{ulb}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)$ and the true dissimilarity, $D^{true}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)$ holds the inequality $D^{ulb}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)\leq D^{true}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)$.

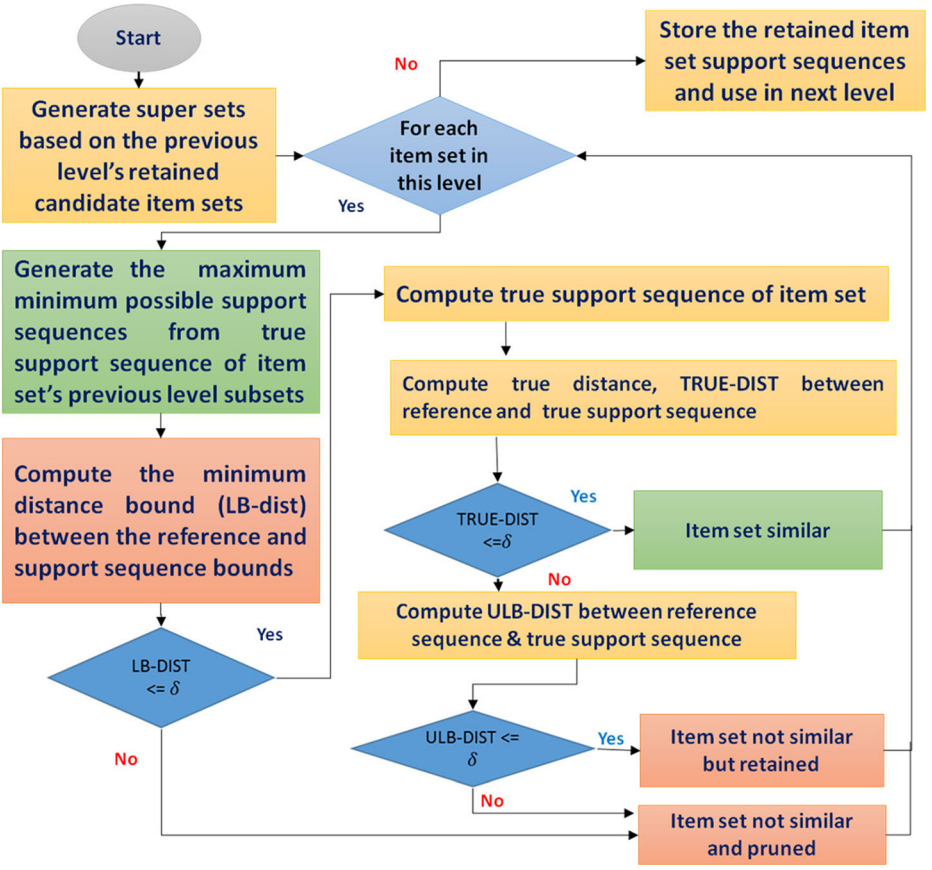

Fig. 8 Pattern Discovery for temporal patterns with size, S > 2

Proof:

The true distance obtained between any temporal pattern, $\overrightarrow{T_p}$ and the reference time sequence, $\overrightarrow{R_r}$ is given by Eq. (36),

$$D^{true}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)=\frac{1-\mathscr{M}_{R_r}^{T_p}}{1.3679}=\frac{1-\prod_{k=1}^{k=m}e^{-\frac{1}{m}\left(\frac{T_{p_k}-R_{r_k}}{\sigma g}\right)^2}}{1.3679} \tag{36}$$

The upper-lower bounding distance, $D^{ulb}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)$ considering true support time sequence of temporal pattern, $\overrightarrow{T_p}$ and the reference time sequence, $\overrightarrow{R_r}$ is given by Eq. (37),

$$D^{ulb}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)=1-\prod_{k=1}^{k=m}\frac{\begin{cases}exp^{-\left(\frac{T_{p_k}{}^{max}-R_{r_k}}{\sigma g}\right)^2}; R_{r_k}>T_{p_k}{}^{max}\\ 1 \quad\quad\quad\quad\quad\quad\quad\quad ; R_{r_k}\le T_{p_k}{}^{max}\end{cases}}{1.3679} \tag{37}$$

From Eqs. (36) and (37), it is visible that

$$D^{ulb}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right)\le D^{true}\left(\overrightarrow{T_p},\overrightarrow{R_r}\right) \tag{38}$$

if the proposed dissimilarity function is used as a similarity function.

Theorem-2. Algorithm MASTER is complete.

The MASTER algorithm uses the upper-lower bounding distance, $\boldsymbol{D^{ulb}}\left(\overrightarrow{\boldsymbol{T_p}}, \overrightarrow{\boldsymbol{R_r}}\right)$ and the minimum dissimilarity bound, $\boldsymbol{D^{LB}}\left(\overrightarrow{\boldsymbol{T_p}}, \overrightarrow{\boldsymbol{R_r}}\right)$ to prune all infeasible temporal patterns. The algorithm is said to be complete if and only if temporal pattern pruning performed using these dissimilarity bounds does not miss any valid time profiled temporal associations to a given reference time sequence. Let T_I be any size-k temporal pattern and T_J is subset of temporal pattern, T_I . Further, let δ be the allowable dissimilarity limit specified by the user in Euclidean space and δ^g is the allowable dissimilarity in Gaussian space.

From lemma-3 and lemma-4, the following inequality holds good, i.e. $D^{ulb}\left(\overrightarrow{T_J}, \overrightarrow{R_r}\right) \leq D^{ulb}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) \leq D^{true}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right)$. So, whenever $D^{ulb}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) > \delta^g$ holds good then, $D^{true}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) > \delta^g$ is also true. So, the temporal pattern T_I cannot be similar w.r.t reference. The function, generate_candidate_pattern in step-6 of MASTER algorithm, thus does not miss any valid similar temporal patterns.

Now consider the lower bounding distance. We have from lemma-1, the inequality, $D^{LB}\left(\overrightarrow{T_I{}^{max}}, \overrightarrow{T_I{}^{min}}, \overrightarrow{R_r}\right) \leq D^{true}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right)$. This means that if the lower bounding distance, $D^{LB}\left(\overrightarrow{T_I{}^{max}}, \overrightarrow{T_I{}^{min}}, \overrightarrow{R_r}\right) > \delta^g$, then $D^{true}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right)$ also exceeds δ^g. This proves that the step-7 of MASTER algorithm does not prune any valid temporal pattern. The second and eighth step of MASTER algorithm generates true support sequences of temporal patterns from the time stamped temporal database. Finally step-3 and step-9 computes all similar temporal patterns whose dissimilarities does not exceed,δ^g.

Theorem-3. The algorithm, MASTER is correct.

The algorithm is said to be correct as the step-3 and the step-9 of MASTER invokes the function, compute_similar_temporal_patterns and then discovers temporal patterns in the result set, only those whose dissimilarities does not exceed, δ^g.

5.3.2 Computational Analysis

In this section, the computational cost of MASTER to sequential approach [85] is discussed. Let $\mathbf{T}_{\boldsymbol{MASTER}}$ and $\mathbf{T}_{\boldsymbol{sequential}}$ respectively be the computational cost of MASTER and sequential approaches respectively.

Computational analysis for MASTER Let $\mathbf{T}_{\boldsymbol{generate\_size\_s\_candidate\_itemsets}}$ denote the computation cost for generating candidate itemset of size equal to s, $\boldsymbol{C_s}$ be the set consisting of all generated size-s candidates, $\mathbf{T}_{\boldsymbol{generate\_support\_bounds}}(\boldsymbol{C_s})$ is the computation cost for generating maximum and minimum prevalence bound sequence of size-s candidate itemset, $\mathbf{T}_{\boldsymbol{prune\_using\_lowerbound}}(\boldsymbol{C_s})$ is the computational cost for pruning candidate temporal patterns using lower bounding dissimilarity value, $\boldsymbol{C_s}'$ be the set of all filtered size-s candidate itemset, $\mathbf{T}_{\boldsymbol{gen\_supp\_seq\_true}}(\boldsymbol{C_s}')$ is the time taken to generate prevalence (or support) sequences of filtered candidate temporal patterns considering original temporal dataset, D^{TSDB}, $\boldsymbol{T}_{\boldsymbol{compute\_true\_patterns}}(\boldsymbol{C_s}')$ is the computational cost for finding similar temporal patterns of size 's' all those which satisfy the dissimilarity constraint and 'S' denotes size of candidate

temporal pattern which is generated in MASTER algorithm. Then the computational cost of MASTER is given by Eq. (39)

$$T_{MASTER} = \sum_{s=1}^{s=m} [T_{generate\_k\_candidate\_itemsets} + T_{generate\_support\_bounds}(C_s) + T_{prune\_using\_lowerbound}(C_s)] + T_{gen\_supp\_seqs\_true}\left(C_s', D^{TSDB}\right) + T_{compute\_true\_patterns}\left(C_s'\right) \quad (39)$$

Computational analysis for sequential Let, 't' be the number of time slots in time stamped temporal dataset with D^i consisting transactions present in the ith timeslot, 'n' be the size of the largest candidate temporal pattern (or temporal itemset) generated at every time slot, C_s'' refers to size 's' candidate temporal patterns generated at each time slot, $T_{prune\_using\_lowerbound'}(C_s'')$ is the computational cost for separating candidate temporal patterns using variable minimum support threshold values, $T_{gen\_partial\_\text{supp}\_seqs}(C_s''', D^i)$ is the computational cost required for scanning the current data subset, D^i and to generate partial prevalence sequences until the current ith time slot of the filtered candidate set, C_s'''.

The sequential approach for obtaining similar temporal patterns repeats the procedure for generating candidate temporal patterns, performs pattern pruning and filters feasible and infeasible temporal patterns, and generates partial prevalence sequences of temporal patterns with increasing candidate pattern size (s). $T_{compute\_true\_patterns}$ is the computational cost for finding all the valid similarity profiled temporal patterns from the available true prevalence computations. Then, the overall computational cost of sequential method is given by Eq. (40)

$$T_{sequential} = \sum_{i=1,t}\left\{\sum_{s=1}^{s=n}\left(T_{generate\_k\_candidate\_itemsets} + T_{prune\_using\_lowerbound'}\left(C_s''\right) + T_{gen\_partial\_supp\_seqs}\left(C_s'', D^i\right)\right)\right\} + T_{compute\_true\_patterns} \quad (40)$$

Comparison of computational cost for MASTER and sequential approaches Consider both sequential and MASTER approaches. In sequential approach, 'n' is the size of the largest size of the candidate temporal patterns (or temporal itemset) generated at every time slot. Similarly, 'm' is the largest size of candidate temporal patterns generated in MASTER. The total number of candidate temporal patterns generated in sequential approach is equal to $\sum_{i=1,t}\sum_{s=1,n} C_s''$ and this value is greater than $\sum_{s=1,m} C_s$. Thus, $T_{gen\_partial\_\text{supp}\_seqs} \gg T_{gen\_\text{supp}}$

Table 2 Temporal support sequences

Temporal Support sequences
$\overrightarrow{T_A} = (0.6, 0.4)$
$\overrightarrow{T_B} = (0.3, 0.7)$
$\overrightarrow{T_C} = (0.8, 0.8)$
$\overrightarrow{T_{AB}} = (0.3, 0.3)$
$\overrightarrow{T_{AC}} = (0.4, 0.4)$
$\overrightarrow{T_{BC}} = (0.3, 0.5)$
$\overrightarrow{T_{ABC}} = (0.3, 0.3)$

Table 3 Dissimilarity values

Temporal Pattern	Euclidean	Proposed	Similar
$\overrightarrow{T_A}, \mathrm{R}$	0.2	0.3251	✗
$\overrightarrow{T_B}, \mathrm{R}$	0.1	0.1	✓
$\overrightarrow{T_C}, R$	0.3162	0.5631	✗
$\overrightarrow{T_{AB}}, R$	0.2236	0.3806	✗
$\overrightarrow{T_{AC}}, R$	0.1414	0.1863	✗
$\overrightarrow{T_{BC}}, R$	0.238	0.1	✓
$\overrightarrow{T_{ABC}}, R$	0.3366	0.8306	✗

$_{seqs\_true}$. This is due to the substantial number of candidate patterns generated in sequential approach. Hence, we can deduce that computational cost of sequential is very much higher than the MASTER, i.e. $\mathrm{T}_{sequential} \gg \mathrm{T}_{G-Spamine}$ even if the MASTER has a computational cost associated represented by $\mathrm{T}_{generate\_support\_bounds}(C_s)$.

$$T_{MASTER}/T_{sequential} \approx \frac{\sum_{s=1}^{s=m}\begin{bmatrix} T_{generate\_s\_candidate\_itemsets} + T_{generate\_support\_bounds}(C_s)+ \\ T_{prune\_using\_lowerbound}(C_s) + T_{gen\_supp\_seq\_true}\left(C_s{}', D^{TSDB}\right) \\ +T_{compute\_true\_patterns}\left(C_s{}'\right) \end{bmatrix}}{\left\{\sum_{s=1}^{s=n}\left\{t^*\begin{pmatrix} T_{generate\_k\_candidate\_itemsets} + T_{prune\_using\_lowerbound'}\left(C_s{}''\right)+ \\ T_{gen\_partial\_supp\_seq\_true}\left(C_s{}'', D^i\right) \end{pmatrix}\right\} + T_{compute\_true\_patterns}\right\}} \tag{41}$$

6 Working example

To understand the working of algorithm, we choose to consider the time stamped transaction database in Fig. 1. The time stamped transaction database is defined over two time slots and consists of ten (10) transactions in each time slot. The total transactions are twenty (20). The database is defined over 3 transaction items A, B, C which form the finite set of items. Supports of each singleton item are computed and represented as a table in Fig. 2.

Table 2 gives the support sequences (true support) of all temporal itemset associations for each time slot. It can be seen that supports of temporal patterns are multi-dimensional. Since the number of time slots are two, in the present example, the support sequences of temporal patterns are 2-dimensional vector sequences. This makes the conventional frequent itemset mining and temporal association mining algorithm fail.

For the present example, we choose the reference sequence, $\overrightarrow{R}$ = (0.4, 0.6) and a dissimilarity threshold equal to Δ=0.1. Table 3 gives the dissimilarity values computed by applying Euclidean distance function and the proposed dissimilarity measure of each temporal pattern w.r.t reference. In naïve approach, we must find true supports of each pattern combination and then verify for similarity. This requires 7 true computations. Table 3 depicts similar and dissimilar temporal pattern.

To find the dissimilarity values of temporal pattern to the reference by applying the proposed dissimilarity measure between temporal pattern and reference, we require the value for deviation and the transformed threshold value. These values can be computed using the expression for deviation and threshold given by Eqs. (13) and (15) respectively. All patterns whose dissimilarity value exceeds $\Delta^g = 0.1$ are dissimilar w.r.t proposed dissimilarity measure. It can be verified that both these distance functions determine same set of similar temporal patterns.

$$\sigma^g = \frac{\Delta}{\sqrt{ln_e\left(\frac{1}{abs\left(1-1.3679^*\Delta\right)}\right)}} = \frac{0.1}{\sqrt{ln_e\left(\frac{1}{abs\left(1-1.3679^*\Delta\right)}\right)}} = 0.2607$$

$$\Delta^g = \frac{1-e\left(\frac{\Delta}{\sigma^g}\right)^2}{1.3679} = \frac{1-e\left(\frac{0.1}{0.2607}\right)^2}{1.3679} = 0.1$$

We now apply the temporal pattern mining algorithm for this example database. Initially, we start with computing support sequences for singleton items. The true distance of these singleton items w.r.t reference are as follows: $D^{true}\left(\overrightarrow{T_A}, \overrightarrow{R_r}\right) = 0.3251$; $D^{true}\left(\overrightarrow{T_B}, \overrightarrow{R_r}\right) = 0.1$; $D^{true}\left(\overrightarrow{T_C}, \overrightarrow{R_r}\right) = 0.5631$. This gives us that temporal pattern, T_B is the only similar temporal pattern. Since, temporal pattern T_A and T_C are not similar, we must compute their maximum-minimum bounding distances to judge whether to retain these pattern or to prune them. The maximum-minimum bounding distance for these temporal pattern are $D^{ULB}\left(\overrightarrow{T_A}, \overrightarrow{R_r}\right) = 0.1863$; $D^{ULB}\left(\overrightarrow{T_C}, \overrightarrow{R_r}\right) = 0$. The distance of T_A exceeds the threshold, $\Delta^g = 0.1$ and hence the temporal pattern T_A must be pruned and cannot be retained. However, temporal pattern, T_C is retained as its distance, D^{ULB}does not exceed the threshold, Δ^g. Since, T_A is not retained all the superset patterns associated with this pattern, i.e. T_{AB}, T_{AC}, T_{ABC} are directly pruned without computing their true support sequences and true distances. Since, temporal patterns T_B and T_C are retained, we compute the lower bounding distance of pattern, T_{BC} by approximating its support time sequence bounds and the distance,

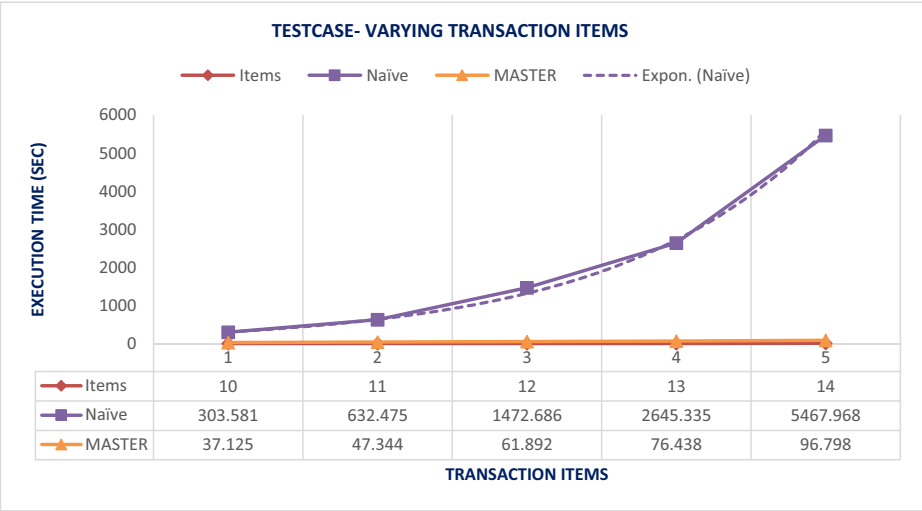

Fig. 9 Execution times of naïve and MASTER – varying items

$D^{LB}\left(\overrightarrow{T_{BC}}, \overrightarrow{R_r}\right) = 0.05 < 0.1$. Since, $D^{LB}\left(\overrightarrow{T_{BC}}, \overrightarrow{R_r}\right)$ did not exceed the threshold, its true support is to be computed. The true distance of this pattern is 0.1 and hence it is considered as a similar pattern to reference. For this example, using the proposed approach we required only 4 true support computations as against to naïve approach that requires 7 true support computations. This is the advantage of the proposed algorithm which uses the monotonicity of dissimilarity measure to prune the invalid temporal associations much early in the mining process.

7 Results and discussions

Experiments are performed on an Intel core -i5 3470 3.24 GHz CPU with 4 GB of memory running on windows operating system using the temporal pattern mining tool implemented in Java shown in the Figure 26. The experiments are conducted by considering five different test cases i) varying transaction items ii) varying time slots iii) varying transactions per time slot iv) varying threshold and v) true support computations performed. This is done to study the scalability of the proposed approach w.r.t naïve, sequential [84–86], Spamine [84–86] approaches. The results obtained proved that the MASTER approach using the proposed dissimilarity measure has shown comparatively better results to other three approaches that use the Euclidean distance measure. The improvement in our approach is essentially because of the proposed approach for support estimation of temporal associations and the proposed dissimilarity measure. The experiment results obtained are discussed below for various test cases.

7.1 Varying number of transaction items

The execution time of proposed approach is compared to naïve [86], sequential [84–86], Spamine [84–87] and G-Spamine [11] approaches. In our approach, we use the proposed dissimilarity measure. Other three approaches naïve, sequential and Spamine uses the

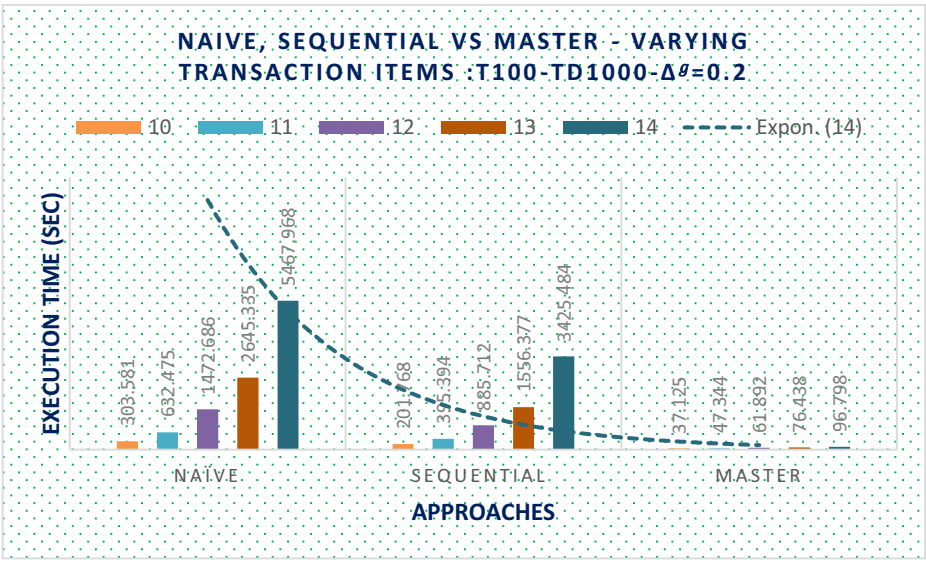

Fig. 10 Naïve, Sequential and MASTER– varying items

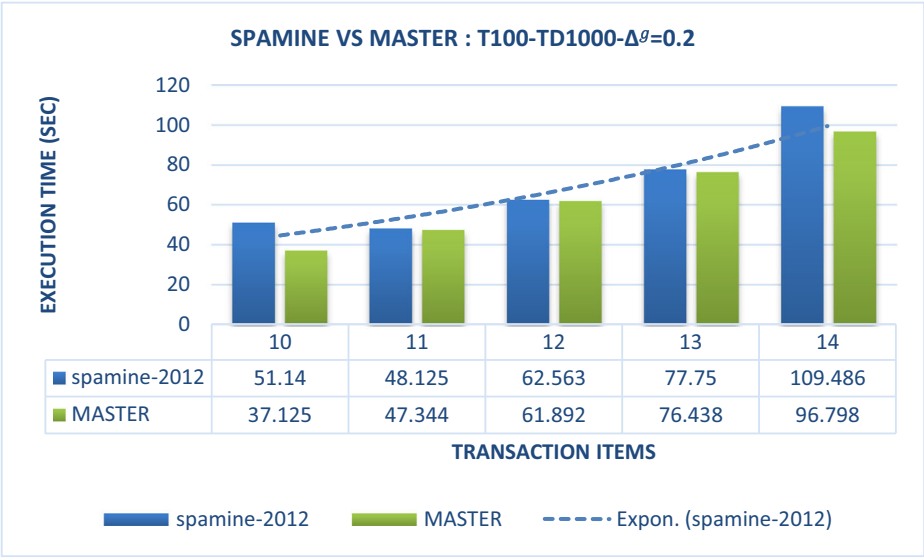

Fig. 11 Spamine (Euclidean) vs MASTER for varying transaction items

Euclidean distance measure for finding similarity between temporal associations and reference. The threshold limit chosen is equal to 0.2. The reference sequence is considered over 100 time slots and each time slot consists of 1000 transactions per time slot. The number of items is varied from 10 to 14. Figure 9 depicts the graph plotted for execution times of naïve and proposed approach. It is evident that the proposed algorithm is tractable whereas the naïve approach shows exponential behavior and has chances of becoming intractable.

Figure 10 depicts the comparison of execution times of naïve, sequential and proposed approaches over a time stamped temporal database consisting of 100-time slots with 1000

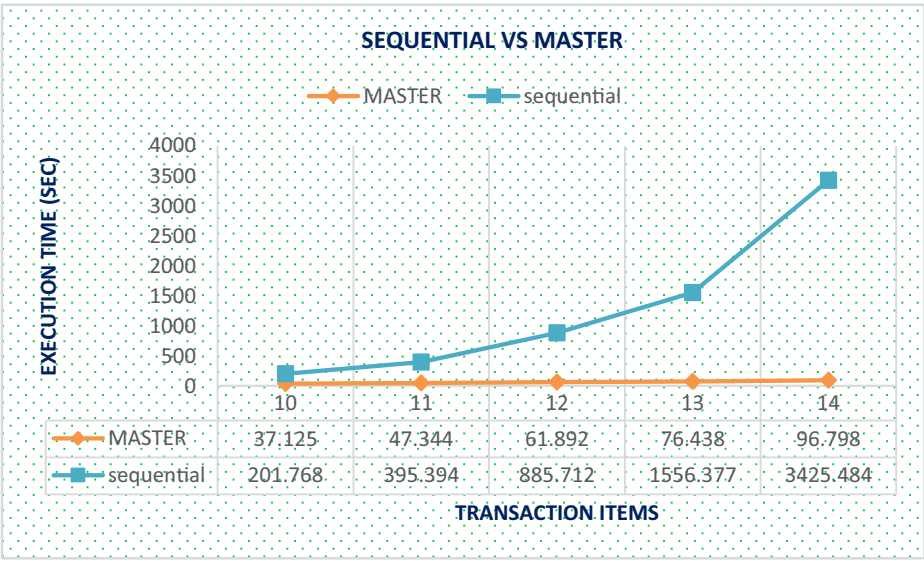

Fig. 12 Sequential (Euclidean) vs MASTER for varying transaction items

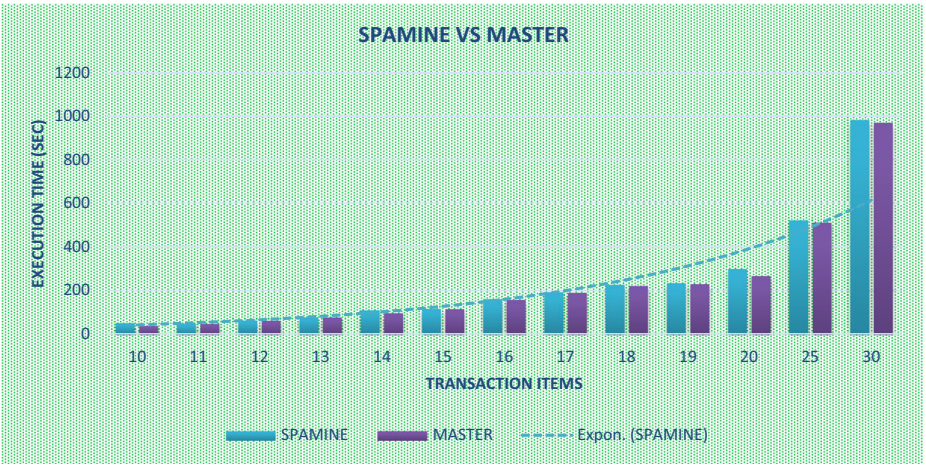

Fig. 13 Spamine vs MASTER

transactions per time slot for a threshold equal to 0.2. The number of transaction items are varied from 10 items to 14 items and execution times are recorded. Similarly, the execution times for spamine and proposed approach are plotted as depicted using graph in Fig. 11. The comparison of sequential and proposed approaches is depicted using graph in Fig. 12 for 10, 11 12, 13 and 14 transaction items.

Figure 13 depicts the comparison of execution times of Spamine and proposed approach over a time stamped temporal database consisting of 100-time slots with1000 transactions per time slot for a threshold equal to 0.2. The number of transaction items considered are 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25 and 30 items and execution time that are recorded are plotted as bar graph depicted in Fig. 13.

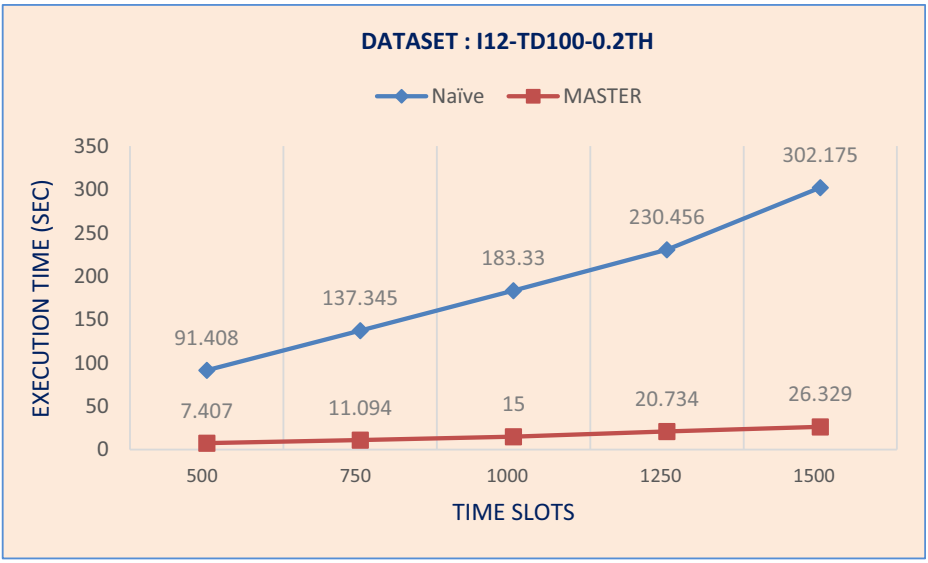

Fig. 14 Naïve vs MASTER for varying time slots

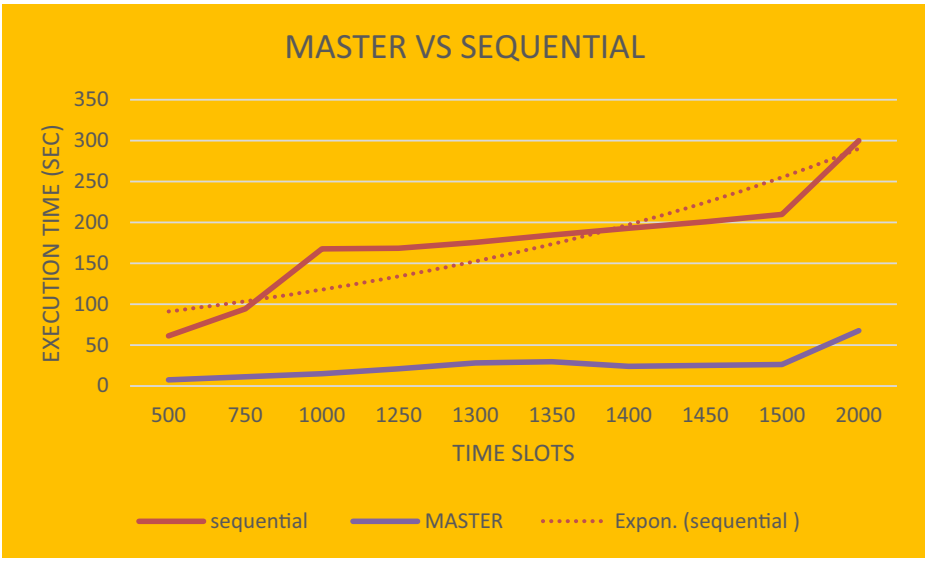

Fig. 15 Sequential (Euclidean) vs MASTER for varying time slots

7.2 Varying total number of time slots

In the second experiment, we consider varying number of time slots for a constant number of transactions per time slot equal to 100, threshold equal to 0.2 and number of transaction items equal to 12. The minimum number of transactions is 50 k and maximum number of transactions is 200 k. Fig. 14 compares the execution time of naïve and proposed approach. Figure 15 compares the execution time of sequential and proposed approach. The graph depicted in Figure 16 compares naïve, sequential, spamine to the proposed approach.

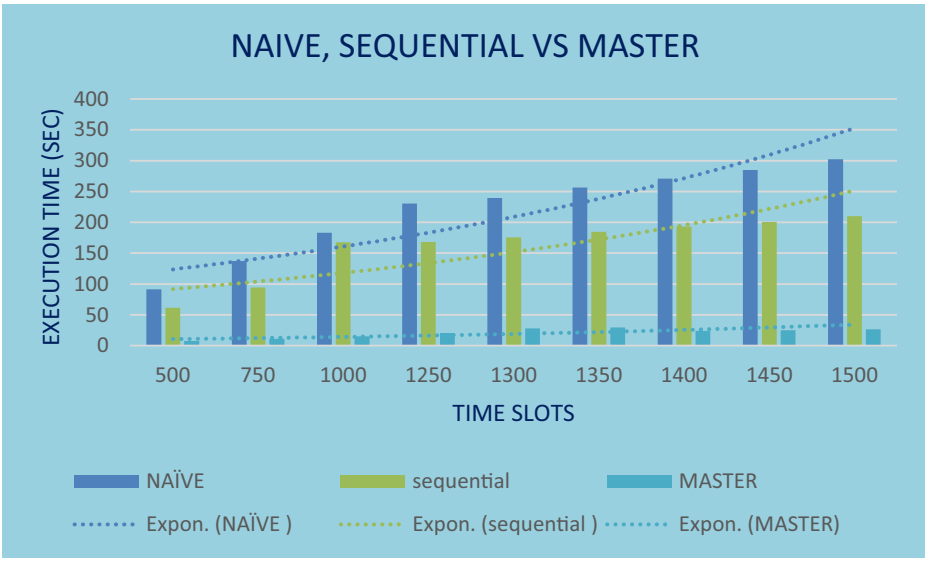

Fig. 16 Comparison of all approaches

7.3 Varying total number of transactions per time slot

In the third experiment, the total number of transactions per time slot is varied with the fixed number of transaction items, number of time slots and threshold. For experimentation, the number of transaction items is 10, time slots are set to 100 and the threshold value chosen is 0.2.

Figure 17 depicts the execution times of naïve, sequential and z-spamine approaches for varying transactions per time slot equal to 500, 750, 1000, 1250 and1500. The dotted line graph indicates the linear and exponential trends of naïve and sequential approaches. The total number of transactions are 1, 00, 000 defined over 100 time slots. It can be verified that the proposed approach performs better to sequential and naïve approaches. Both sequential and naïve uses Euclidean distance measure. The reduction in time taken for the MASTER using proposed dissimilarity measure to output the result set is due to the reduced number of true support computations that are performed.

Figure 18 depicts the execution times of Spamine and MASTER approaches for varying transactions per time slot equal to 500, 750, 1000, 1250 and1500. The proposed approach performs better to Spamine even as the number of transactions per time slots increase substantially. Figure 19 depicts the execution times of Spamine, G-Spamine [11] using fuzzy dissimilarity function in which the membership function is summation based and the proposed approach MASTER (G-spamine algorithm but with the fuzzy dissimilarity measure whose membership function is product based for varying transactions) per time slot equal to 500, 750, 1000, 1250, 1300, 1350, 1400, 1450, 1500 and 2000. The proposed approach i.e. MASTER performs better to Spamine and G-Spamine even as the number of transactions per time slot increase substantially. It is clearly visible that with increase in number of transactions per time slot, the proposed approach has an improvement in execution time. For fewer number of transactions per time slot, the present approach performs at least same as the other two approaches.

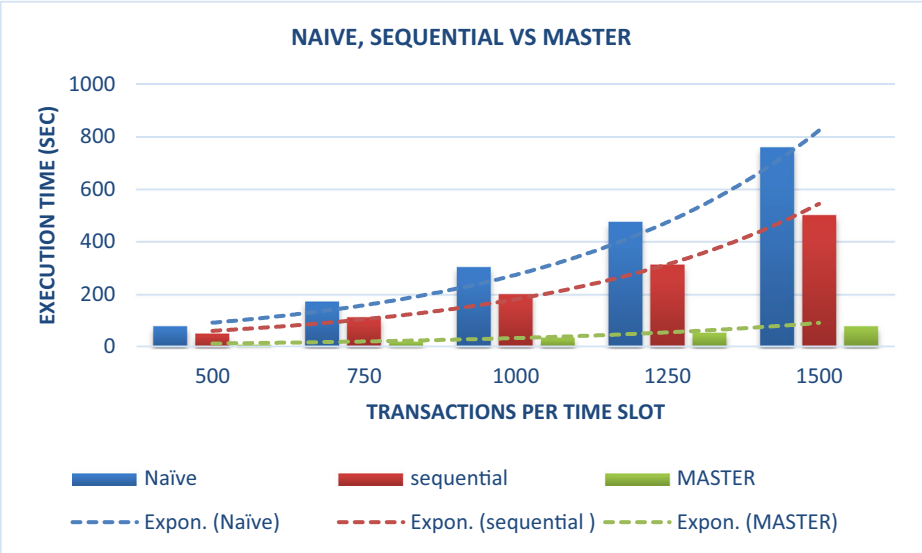

Fig. 17 Naïve, Sequential and MASTER – Execution times

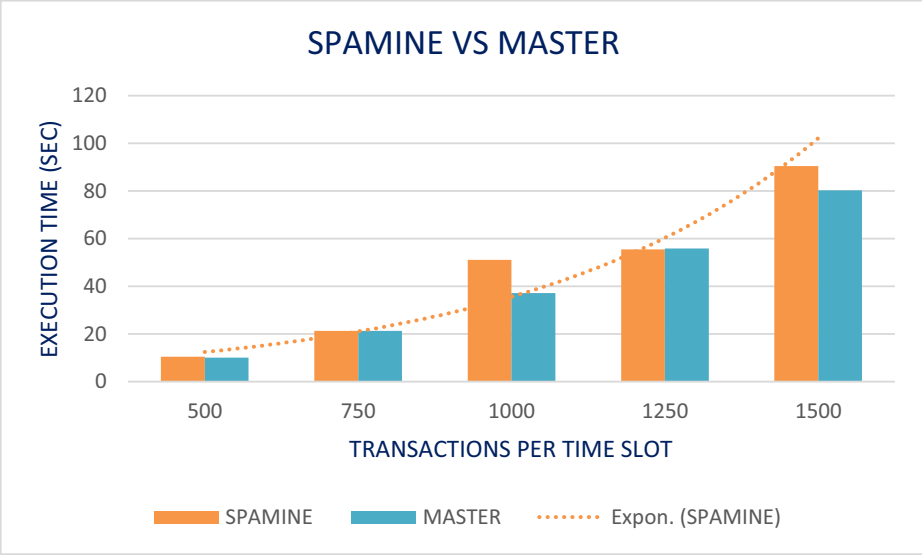

Fig. 18 Spamine and MASTER – Execution times

Figure 20 depicts the execution times of naïve, sequential and MASTER approaches for varying transactions per time slot equal to 500, 750, 1000, 1250, 1300, 1350, 1400, 1450, 1500 and 2000. The total number of transactions are 1,00,000 defined over 100-time slots and the number of transaction items are 12. The proposed approach performs better to sequential and naïve approaches and can be depicted from Fig. 20. The graph plotted in Fig. 21 considering sequential and proposed approach demonstrates the advantage of proposed approach over sequential approach. The time taken by proposed algorithm is very much less compared to sequential algorithm.

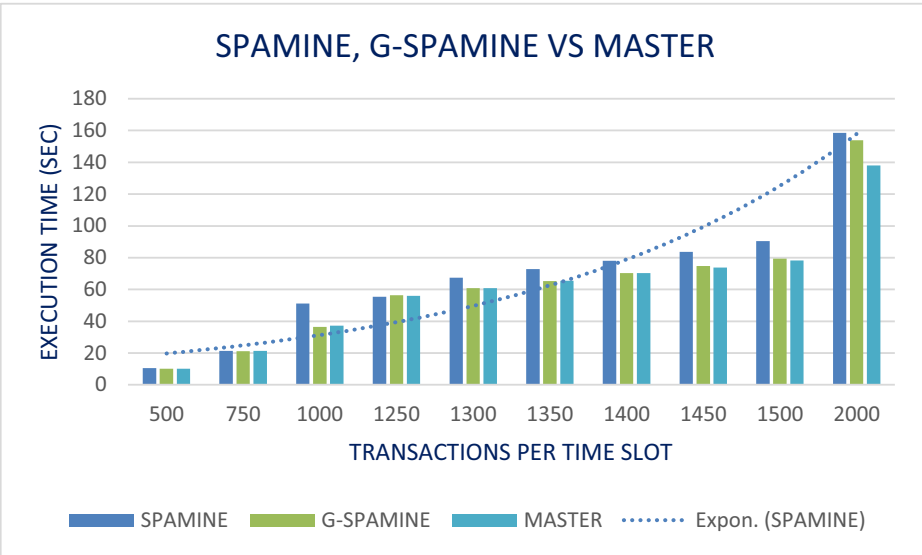

Fig. 19 Spamine, G-Spamine and MASTER – Execution times

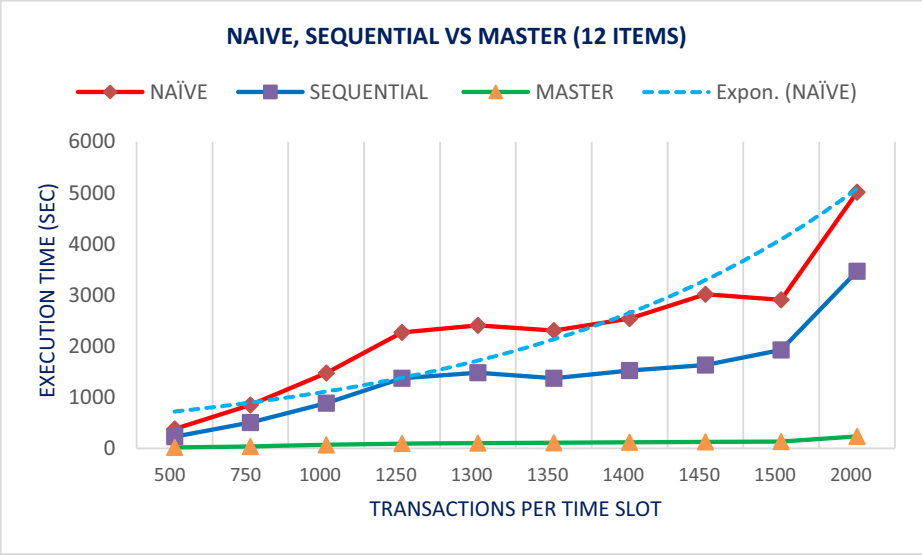

Fig. 20 Execution times of naïve, sequential and proposed

7.4 Varying threshold values

The execution times of naïve, sequential approaches w.r.t proposed approach are plotted as a graph in Figs. 22 and 23 for varying thresholds equal to 0.12, 0.14, 0.16, 0.18, 0.2, 0.22, 0.24 and 0.26. The number of transaction items considered are equal to 10, 100 time slots and 1000 transactions per time slot. Finally, the graph depicted in Fig. 24 shows the execution time of all approaches.

7.5 Effect on true support computations

This section outlines some of the results obtained by applying the proposed prevalence estimation approach in section 2. Figure 25 depicts true support computations required

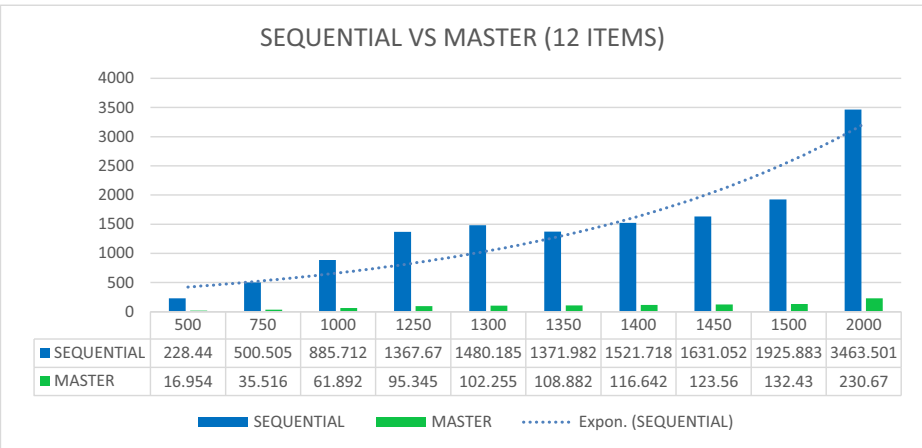

Fig. 21 Sequential (Euclidean) vs MASTER

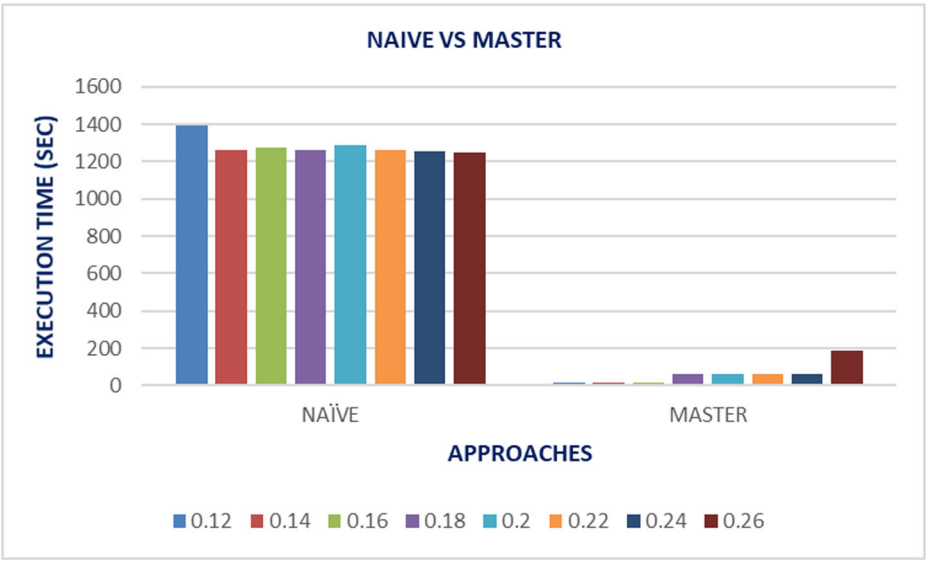

Fig. 22 Naïve vs MASTER – varying thresholds

for a temporal dataset TD1000-T100-I20 that is random generated. TD indicates number of transactions per time slot, T is number of time slots, I is the total number of items in finite itemset. The temporal database generated from IBM data generator [85] comprises of one lakh transactions. The total number of possible temporal association patterns possible is 2^{20} which are 1 billion temporal patterns. For example, a database generated over 10 items has 1024 different possible pattern combinations.

Fig. 23 Sequential vs MASTER – varying thresholds

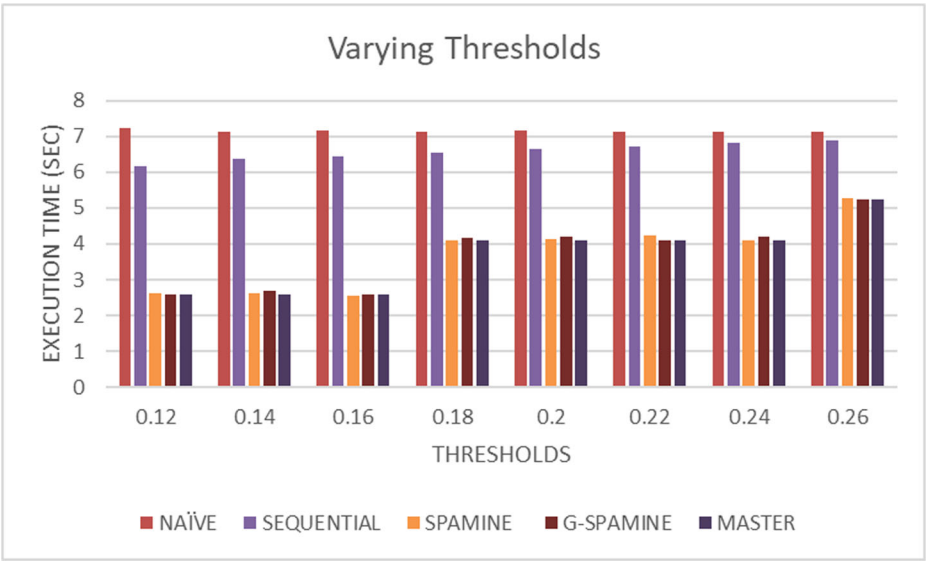

Fig. 24 Varying thresholds – all approaches

7.6 GUI of Tool devoloped for time profiled association mining

To visualize number of temporal associations for which true supports are computed, number of retained temporal associations and the temporal trends we have devoloped a pattern mining tool for similarity based time profiled temporal association mining.

Figure 26 shows the layout of the visual mining tool which has a provision to generate synthetic time stamped transcation database for a given set of specifications such as (i) number of time slots (ii) number of transactions per time slot (iii) number of items (iv) threshold (v) reference sequence. A provision is also made available to choose any available and existing dataset.

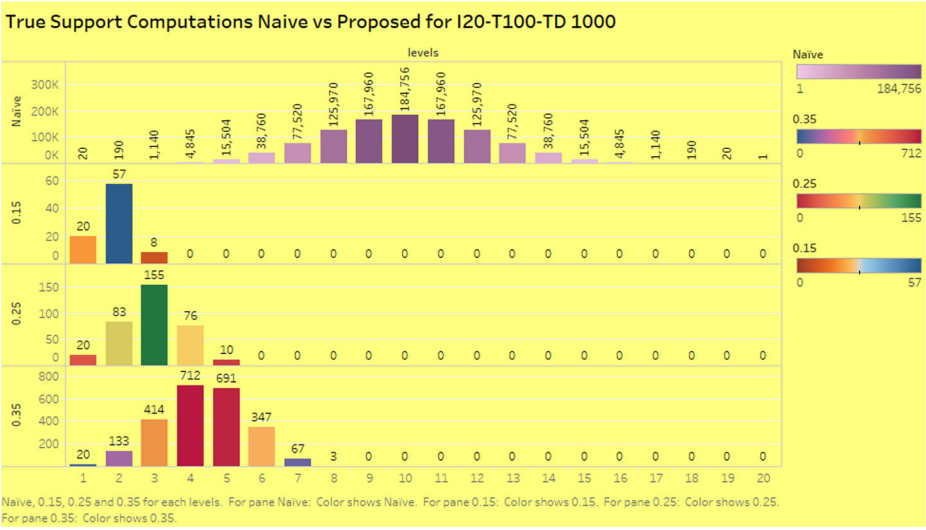

Fig. 25 True Support Computations – Naïve vs MASTER

8 Conclusions

One of the two important challenges that need to be addressed in time profiled association mining are the approaches that can gauge the support bounds of temporal associations and the dissimilarity measure that also has monotonicity property. Our prevalence bound computation approach is designed to be feasible with the proposed dissimilarity measure to hold monotonicity with respect to lower bounding distance. In this paper, we have also proposed a novel dissimilarity measure for mining time profiled temporal associations. The dissimilarity measure holds the monotonicity property with respect to the maximum-minimum distance bound and true support time sequence of temporal association. The proposed dissimilarity measure considers the standard deviation and performs the similarity computation in the transformed space and is comparatively less sensitive to high dimensionality when compared to Euclidean distance. MASTER addressed in the present research extends G-Spamine by proposing a novel similarity measure. The completeness and correctness of the proposed algorithm is also discussed analytically. The computational performance of the proposed algorithm is also compared to sequential approach analytically. Several experiments are performed by considering various test cases such as effect of varying thresholds, effect of varying time slots, transactions and transactions per time slot by using synthetic datasets generated using the IBM synthetic data generator. Experimental results prove the computational efficiency of the proposed algorithm is better when compared to naïve, sequential, spamine and G-spamine approaches. In future, this research can be extended by coming out with new approaches for support estimation and pattern pruning techniques. The proposed measure may also be applied for all real world applications which requires similarity value computation.

Acknowledgements Vangipuram Radhakrishna is heartfully thankful to his mentor and esteemed professor P.V. Kumar for his guidance and motivation throughout this research. We are also thankful to Aravind Cheruvu, graduated student from Department of Information Technology, VNR VJIET for his participation in this research work.

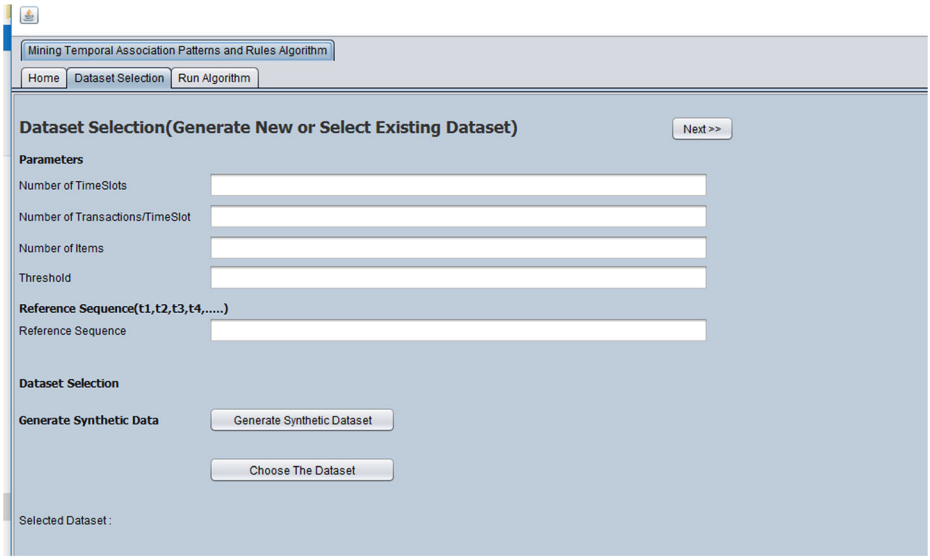

Fig. 26 Visualization – Data Selection Screen

References

1. Agrawal R and Shafer JC (1996) Parallel Mining of Association Rules: Design, Implementation, and Experience. IEEE Trans. Knowledge and Data Eng, pp. 487–499
2. Agrawal R, Srikant R (1994) Fast Algorithms for Mining Association Rules in Large Databases. In: Bocca JB, Jarke M, Zaniolo C (eds) Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94). Morgan Kaufmann Publishers Inc., San Francisco, pp 487–499
3. Agrawal R, Srikant R (1995) Mining Sequential Patterns. In: Proc. IEEE Int'l. Conference on Database Engineering, 3–14
4. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. SIGMOD Rec 22(2):207–216. https://doi.org/10.1145/170036.170072
5. Agrawal R, Imielinski T, Swami A (1994) Database mining: A performance perspective. IEEE TOKDE 5(5):914–925
6. Ale JM, Rossi GH (2000) An approach to discovering temporal association rules. In: Carroll J, Damiani E, Haddad H, Oppenheim D (eds) Proceedings of the 2000 ACM symposium on Applied computing - Volume 1 (SAC '00), vol 1. ACM, New York, pp 294–300. https://doi.org/10.1145/335603.335770
7. Aljawarneh S, Aldwairi M Muneer Bani Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model". J Comput Sci, ISSN 1877-7503. https://doi.org/10.1016/j.jocs.2017.03.006
8. Aljawarneh S, Radhakrishna V, Kumar PV, Janaki V (2016) A similarity measure for temporal pattern discovery in time series data generated by IoT. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–4
9. Aljawarneh SA, Elkobaisi MR, Maatuk AM (2016) A new agent approach for recognizing research trends in wearable systems. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2016.12.003
10. Aljawarneh SA, Moftah RA, Maatuk AM (2016) Investigations of automatic methods for detecting the polymorphic worms signatures. Futur Gener Comput Syst 60:67–77, ISSN 0167-739X. https://doi.org/10.1016/j.future.2016.01.020
11. Aljawarneh SA, Radhakrishna V, Kumar PV, Janaki V (2017) G-SPAMINE: an approach to discover temporal association patterns and trends in internet of things. Futur Gener Comput Syst 74:430–443. https://doi.org/10.1016/j.future.2017.01.013
12. Bettini C, Wang X, Jajodia S (1996) Testing complex temporal relationships involving multiple granularities and its application to data mining. Proceedings of the Fifteenth ACM SIGACTSIGMOD-SIGART Symposium on principles of database systems, series PODS '96, Montreal, Quebec, Canada. ACM, New York. Proc. of the ACM PODS'96: 68–78. https://doi.org/10.1145/237661.237680
13. Bettini C, Wang XS, Jajodia S (1998) Mining temporal relationships with multiple granularities in time sequences. Data Engineering Bulletin 21(1):32–38. http://131.107.65.22/pub/debull/98mar/98MAR-CD.pdf#page=34
14. Bettini C, Wang XS, Jajodia S, Lin JL (1998) Discovering frequent event patterns with multiple granularities in time sequences. IEEE Trans Knowl Data Eng 10(2):222–237. https://doi.org/10.1109/69.683754
15. Borgelt C (2005) Keeping things simple: finding frequent item sets by recursive elimination. In: Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations (OSDM '05). ACM, New York, 66–70. https://doi.org/10.1145/1133905.1133914
16. Calders T, Paredaens J (2003) Axiomatization of frequent itemsets. Theor Comput Sci 290(1):669–693, ISSN 0304-3975. https://doi.org/10.1016/S0304-3975(02)00081-6
17. Chanda AK, Ahmed CF, Samiullah Md., Leung CK (2017) A new framework for mining weighted periodic patterns in time series databases. Expert Systems with Applications (79) :207-224. https://doi.org/10.1016/j.eswa.2017.02.028
18. Chen X, Petr I (2000) Discovering temporal association rules: algorithms, language, and system. In: Proc. 2000 Int'l Conf Data Eng
19. Chen YC, Peng WC, Lee SY (2015) Mining Temporal Patterns in Time Interval-Based Data. IEEE Trans Knowl Data Eng 27(12):3318-3331. https://doi.org/10.1109/TKDE.2015.2454515
20. Chen C-H, Lan G-C, Hong T-P, Lin S-B (2016) Mining fuzzy temporal association rules by item lifespans. Appl Soft Comput 41:265–274. https://doi.org/10.1016/j.asoc.2016.01.008
21. Cheruvu A, Radhakrishna V (2016) Estimating temporal pattern bounds using negative support computations. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–4. https://doi.org/10.1109/ICEMIS.2016.7745352
22. Cheung D, Han J, Ng V, Wong CY (1996) Maintenance of discovered association rules in large databases: an incremental updating technique. In: Proc. 1996 Int'l Conf. Data Eng., pp. 106–114
23. Cohen E et al (2001) Finding Interesting Associations without Support Pruning. IEEE Trans Knowl Data Eng 13(1):64–78

24. Dong G, Li J (1999) Efficient mining of emerging patterns: discovering trends and differences. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '99). ACM, New York, NY, USA, 43-52. https://doi.org/10.1145/312129.312191
25. Gharib TF, Nassar H, Taha M, Abraham A (2010) An efficient algorithm for incremental mining of temporal association rules. Data Knowl Eng 69(8):800–815
26. Grahne G, Zhu J (2005) Fast algorithms for frequent itemset mining using FP-trees. IEEE Trans Knowl Data Eng 17(10):1347–1362. https://doi.org/10.1109/TKDE.2005.166
27. Guil F, Marín R (2012) A tree structure for event-based sequence mining. Knowl-Based Syst 35:186–200. https://doi.org/10.1016/j.knosys.2012.04.027
28. Guil F, Bailón A, Álvarez JA, Marín R (2013) Mining generalized temporal patterns based on fuzzy counting. Expert Syst Appl 40(4):1296–1304, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2012.08.061
29. Gunupudi RK, Nimmala M, Gugulothu N, Gali SR (2017, ISSN 0167-739X) CLAPP: A self constructing feature clustering approach for anomaly detection. Futur Gener Comput Syst. https://doi.org/10.1016/j.future.2016.12.040
30. Han J, Yongjian F (1995) Discovery of Multiple-Level Association Rules from Large Databases. In: Dayal U, Gray PMD, Nishio S (eds) Proceedings of the 21th International Conference on Very Large Data Bases (VLDB '95). Morgan Kaufmann Publishers Inc., San Francisco, pp 420–431
31. Han J, Dong G, Yin Y (1999) Efficient mining of partial periodic patterns in time series database. Proc. 15th Int'l Conf. Data Eng., pp. 106–115
32. Han J, Pei J, Yin Y, Mao R (2004) Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Min Knowl Disc 8:53–87
33. Imran A, Aljawarneh SA, Sakib K Web Data Amalgamation for Security Engineering: Digital Forensic Investigation of Open Source Cloud. J Univer Comput Sci 22(4):494–520
34. Jiang JY, Liou RJ, Lee SJ (2011) A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification. IEEE Trans Knowl Data Eng 23(3):335–349. https://doi.org/10.1109/TKDE.2010.122
35. Kumar GR, Mangathayaru N, Narasimha G (2015) An improved k-means clustering algorithm for intrusion detection using gaussian function. In: Proceedings of the The International Conference on Engineering & MIS 2015 (ICEMIS '15). https://doi.org/10.1145/2832987.2833082
36. Kumar GR, Mangathayaru N, Narsimha G (2016) Design of novel fuzzy distribution function for dimensionality reduction and intrusion detection. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–6
37. Kumar GR, Mangathayaru N, Narsimha G An Approach for Intrusion Detection Using Novel Gaussian Based Kernel Function. J Univers Comput Sci 22(4):589–604
38. Last M, Klein Y, Kandel A (2001) Knowledge discovery in time series databases. IEEE Trans Syst, Man, Cybern, B (Cybern) 31(1):160–169. https://doi.org/10.1109/3477.907576
39. Lee W-J, Lee S-J (2004) Discovery of fuzzy temporal association rules. IEEE Trans Syst Man Cybern B (Cyber) 34(6):2330–2342. https://doi.org/10.1109/TSMCB.2004.835352
40. Lee CH, Lin CR, Chen MS (2001) Sliding-window filtering: an efficient algorithm for incremental mining. In: Proceedings of the Tenth International Conference on Information and Knowledge Management. ACM, New York, pp 263-270. https://doi.org/10.1145/502585.502630
41. Lee C-H, Chen M-S, Lin C-R (2003) Progressive partition miner: an efficient algorithm for mining general temporal association rules. IEEE Trans Knowl Data Eng 15(4):1004–1017. https://doi.org/10.1109/TKDE.2003.1209015
42. Lee WJ, Jiang JY, Lee SJ (2004) An efficient algorithm to discover calendar-based temporal association rules. 2004 I.E. International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583) 4: 3122–3127. https://doi.org/10.1109/ICSMC.2004.1400819
43. Lee YJ, Lee JW, Chai DJ, Hwang BH, Ryu KH (2009) Mining temporal interval relational rules from temporal data. J Syst Softw 82(1):155–167. https://doi.org/10.1016/j.jss.2008.07.037
44. Li Y, Ning P, Wang XS, Jajodia S (2001) Discovering Calendar Based Temporal Association Rules. Proceedings Eighth International Symposium on Temporal Representation and Reasoning. pp 111–118. https://doi.org/10.1109/TIME.2001.930706
45. Li Y, Ning P, Wang XS, Jajodia S (2001) Discovering calendar-based temporal association rules. Proceedings Eighth International Symposium on Temporal Representation and Reasoning. TIME 2001, Cividale del Friuli, pp. 111–118. https://doi.org/10.1109/TIME.2001.930706
46. Li Y, Ning P, Sean Wang X, Jajodia S (2003) Discovering calendar-based temporal association rules. Data Knowl Eng 44(2):193–218, ISSN 0169-023X. https://doi.org/10.1016/S0169-023X(02)00135-0
47. Lin MY, Lee SY (2002) Fast Discovery of Sequential Patterns by Memory Indexing. In: Kambayashi Y, Winiwarter W, Arikawa M (eds) Data Warehousing and Knowledge Discovery. DaWaK 2002. Lecture Notes in Computer Science, vol 2454. Springer, Berlin, Knowledge Discovery. DaWaK 2002. Lecture Notesin Computer Science, vol 2454. Springer, Berlin, https://doi.org/10.1007/3-540-46145-0_15

48. Lin M-Y, Hsueh S-C, Chang C-W (2008) Fast discovery of sequential patterns in large databases using effective time-indexing. Inf Sci 178(22):4228–4245. https://doi.org/10.1016/j.ins.2008.07.012
49. Lin YS, Jiang JY, Lee SJ (2014) A Similarity Measure for Text Classification and Clustering. IEEE Trans Knowl Data Eng 26(7):1575–1590. https://doi.org/10.1109/TKDE.2013.19 http://ieeexplore.ieee.org/document/6420834/
50. Lind DA, Marchal WG, Wathen SA (2004) Statistical techniques in business and economics, 12e: Chapter 7: Continuous Probability Distributions. The McGraw-Hill Companies, New York
51. Liu B, Hsu W, Ma Y (1999) Mining association rules with multiple minimum supports. In: Proc. Int'l Conf. Knowledge Discovery and Data Mining
52. Mannila H, Toivonen H, Verkamo I (1995) Discovering Frequent Episodes in Sequences. KDD'95. AAAI, Menlo Park, pp 210–215
53. Ozden B, Ramaswamy S, Silberschatz A (1998) Cyclic association rules. In: Proceedings of the Fourteenth International Conference on Data Engineering (ICDE). IEEE Computer Society, Washington, DC, pp 412–421. http://dl.acm.org/citation.cfm?id=645483.656222. Accessed 10 Nov 2017
54. Park JS, Chen MS, Yu PS (1997) Mining association rules with adjustable accuracy. In: Proc. ACM Sixth Int'l Conf. Information and Knowledge Management, pp. 151–160
55. Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering Frequent Closed Itemsets for Association Rules. In: Beeri C, Buneman P (eds) Proceedings of the 7th International Conference on Database Theory (ICDT '99). Springer-Verlag, London, 19:398–416. http://dl.acm.org/citation.cfm?id=645503.656256. Accessed 10 Nov 2017
56. Pei J, Han J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu M (2001) PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. In: Proceedings of the 17th International Conference on Data Engineering. IEEE Computer Society, Washington, DC, pp 215–224
57. Radhakrishna V, Kumar PV, Janaki V (2015) A survey on temporal databases and data mining. In: Proceedings of the The International Conference on Engineering & MIS 2015 (ICEMIS '15). ACM, New York, Article 52, 6 pages. https://doi.org/10.1145/2832987.2833064
58. Radhakrishna V, Kumar PV, Janaki V (2015) A Novel Approach for Mining Similarity Profiled Temporal Association Patterns Using Venn Diagrams. In : Proceedings of the The International Conference on Engineering & MIS 2015 (ICEMIS '15). ACM, New York, NY, USA, Article 58, 9 pages. http://dx.doi.org/10.1145/2832987.2833071
59. Radhakrishna V, Kumar PV, Janaki V (2016) A computationally optimal approach for extracting similar temporal patterns. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–6. https://doi.org/10.1109/ICEMIS.2016.7745344
60. Radhakrishna V, Kumar PV, Janaki V (2016) An Approach for Mining Similar Temporal Association Patterns in Single Database Scan. In: Satapathy S, Das S (eds) Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2. Smart Innovation, Systems and Technologies, vol 51. Springer, Cham. https://doi.org/10.1007/978-3-319-30927-9_60
61. Radhakrishna V, Kumar PV, Janaki V (2016) Mining of outlier temporal patterns. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–6. https://doi.org/10.1109/ICEMIS.2016.7745343
62. Radhakrishna V, Kumar PV, Janaki V, Aljawarneh S (2016) A similarity measure for outlier detection in timestamped temporal databases. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–5. https://doi.org/10.1109/ICEMIS.2016.7745347
63. Radhakrishna V, Kumar PV, Janaki V (2016) Looking into the possibility of novel dissimilarity measure to discover similarity profiled temporal association patterns in IoT. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–5
64. Radhakrishna V, Kumar PV, Janaki V, Aljawarneh S (2016) A computationally efficient approach for temporal pattern mining in IoT. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–4. https://doi.org/10.1109/ICEMIS.2016.7745354
65. Radhakrishna V, Aljawarneh SA, Kumar PV, Choo K-KR (2016) A novel fuzzy gaussian-based dissimilarity measure for discovering similarity temporal association patterns. Soft Comput. https://doi.org/10.1007/s00500-016-2445-y
66. Radhakrishna V, Kumar PV, Janaki V (2016) A computationally efficient approach for mining similar temporal patterns. In: Proceedings of the 22nd International Conference on Soft Computing (MENDEL 2016) held in Brno, Czech Republic, Vol 576, Advances in Intelligent Systems and Computing. https://link.springer.com/chapter/10.1007/978-3-319-58088-3_19
67. Radhakrishna V, Kumar PV, Janaki V (2016) Estimating prevalence bounds of patterns to discover similar temporal association patterns. In: Proceedings of the 22nd International Conference on Soft Computing (MENDEL 2016) held in Brno, Czech Republic, Vol 576, Advances in Intelligent Systems and Computing. https://link.springer.com/chapter/10.1007/978-3-319-58088-3_20

68. Radhakrishna V, Aljawarneh SA, Kumar PV, Janaki V (2017) A novel fuzzy similarity measure and prevalence estimation approach for similarity profiled temporal association pattern mining. Futur Gener Comput Syst. https://doi.org/10.1016/j.future.2017.03.016 ISSN 0167-739X
69. Radhakrishna V, Kumar PurluV, Janaki V (2017) Multimed Tools Appl. 10.1007/s11042-017-5185-9
70. Radhakrishna V, Kumar PV, Janaki V A Novel Similar Temporal System Call Pattern Mining for Efficient Intrusion Detection. J Univers Comput Sci 22(4):475–493
71. Ramaswamy S, Mahajan S, Silberschatz A (1998) On the discovery of interesting patterns in association rules. In: Proc. Int'l Conf. Very Large Databases (VLDB). Morgan Kaufmann Publishers Inc., San Francisco, 12:368–379. http://dl.acm.org/citation.cfm?id=645924.671170. Accessed 10 Nov 2017
72. Sohrabi MK, Barforoush AA (2012) Efficient colossal pattern mining in high dimensional datasets. Knowl-Based Syst 33:41–52. https://doi.org/10.1016/j.knosys.2012.03.003
73. Srikant R, Agrawal R (1995) Mining Generalized Association Rules. In: Dayal U, Gray PMD, Nishio S (eds) Proceedings of the 21th International Conference on Very Large Data Bases (VLDB '95). Morgan Kaufmann Publishers Inc., San Francisco, pp 407–419
74. Srikant R, Agrawal R (1996) Mining quantitative association rules in large relational tables. In: Widom J (ed) Proceedings of the 1996 ACM SIGMOD international conference on Management of data (SIGMOD '96). ACM, New York, 25(12):1–12. https://doi.org/10.1145/233269.233311
75. Srikant R, Agrawal R (1996) Mining sequential patterns: Generalizations and performance improvements. In: Apers P, Bouzeghoub M, Gardarin G (eds) Advances in Database Technology — EDBT '96. EDBT 1996. Lecture Notes in Computer Science, vol 1057. Springer, Berlin
76. Tansel UA, Imberman SP (2007) Discovery of Association Rules in Temporal Databases. Information Technology, 2007. ITNG '07. Fourth International Conference on, Las Vegas, NV, 2007, pp. 371-376. https://doi.org/10.1109/ITNG.2007.78
77. Tung AKH, Han J, Lakshmanan LVS, Ng RT (2001) Constraint-based clustering in large databases. In: Proc. 2001 Int'l Conf. Database Theory
78. Vangipuram R, Kumar PV, Janaki V Design and analysis of similarity measure for discovering similarity profiled temporal association patterns. IADIS International Journal on Computer Science and Information Systems 12(1):45–60
79. Vangipuram R, Kumar PV, Janaki V, Cheruvu AA dissimilarity measure for mining similar temporal association patterns. IADIS International Journal on Computer Science and Information Systems 12 142(1):126
80. Vangipuram R, Kumar PV, Janaki V Normal Distribution Based Similarity Profiled Temporal Association Pattern Mining (N-SPAMINE). Database Systems Journal 7(3):22–33
81. Villafane R, Hua KA, Tran D, Maulik B (1999) Mining Interval Time Series. Data Warehousing and Knowledge Discovery:318–330. https://doi.org/10.1007/3-540-48298-9_34
82. Winarko E, Roddick JF (2007) An algorithm for discovering richer relative temporal association rules from interval-based data. Data Knowl Eng 63(1):76–90, ISSN 0169-023X. https://doi.org/10.1016/j.datak.2006.10.009
83. Yang C, Fayyad U, Bradley PS (2001) Efficient discovery of error-tolerant frequent itemsets in high dimensions. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '01). ACM, New York, 194–203. 10.1145/502512.502539
84. Yoo JS (2012) Temporal data mining: similarity-profiled association pattern. Data mining: foundations and intelligent paradigms 23:29–47
85. Yoo JS, Shekhar S (2008) Mining temporal association patterns under a similarity constraint. In: Proceedings of the 20th international conference on Scientific and Statistical Database Management. Springer-Verlag, Berlin, Heidelberg, 17:401–417 https://doi.org/10.1007/978-3-540-69497-7_26
86. Yoo JS, Shekhar S (2009) Similarity-Profiled Temporal Association Mining. IEEE Trans Knowl Data Eng 21(8):1147–1161. https://doi.org/10.1109/TKDE.2008.185
87. Yoo JS, Zhang P, Shekhar S (2005) Mining Time-Profiled Associations: An Extended Abstract. In: Ho TB, Cheung D, Liu H (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2005. Lecture Notes in Computer Science, vol 3518. Springer, Berlin
88. Zaki MJ (2000) Scalable algorithms for association mining. IEEE Trans Knowl Data Eng 12(3):372–390. https://doi.org/10.1109/69.846291
89. Zaki MJ, Gouda K (2003) Fast vertical mining using diffsets. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03). ACM, New York, 10:326–335. https://doi.org/10.1145/956750.956788
90. Zhu F, Yan X, Han J, Yu PS, Cheng H (2007) Mining Colossal Frequent Patterns by Core Pattern Fusion. IEEE 23rd International Conference on Data Engineering, Istanbul, pp 706–715. https://doi.org/10.1109/ICDE.2007.367916
91. Zhuang DEH, Li GCL, Wong AKC (2014) Discovery of Temporal Associations in Multivariate Time Series. IEEE Trans Knowl Data Eng 26(12):2969–2982. https://doi.org/10.1109/TKDE.2014.2310219

Vangipuram Radhakrishna is presently associated with Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, INDIA. Prior to this, he served in various positions as a Lecturer, at Kakatiya Institute of Technology and Science, Warangal, as an Assistant Professor at Balaji Institute of Technology and Science, and Associate Professor at Balaji Institute of Technology and Science. He has 14 years of teaching experience and two years industry experience. He is a Professional Member of IEEE (MemberID-91,086,459), IEEE Technical Council on Data Engineering, IEEE Computer Society Technical Committee on Software Engineering and Association of Computing Machinery. He is awarded the prestigious ACM SENIOR MEMBER award in 2016 (Senior Member ACM, Member.ID- 6967456). His research interests include Data mining, Temporal databases, Temporal Data Mining, Network security, Software Engineering, Machine Learning and Algorithm Design. He is a certified SQL associate from Cambridge intercontinental university. He received several best paper awards at International Conferences within and abroad and has been an active researcher under guidance and footsteps of professors P.V. Kumar and V. Janaki who have always been the major inspiration for all his past, present and future achievements. His passion for research is also inspired from his beloved father and Professor Dr. V. Narasimha Charyulu. His Scopus Author ID is 56,118,344,300 and Thomson Reuters Researcher ID is I-5990-2014. He has more than 50 publications in peer-reviewed and refereed international conferences and international journals.

Shadi Aljawarneh is an associate professor, Software Engineering, at the Jordan University of Science and Technology, Jordan. He holds a BSc degree in Computer Science from Jordan Yarmouk University, a MSc degree in Information Technology from Western Sydney University and a PhD in Software Engineering from Northumbria University-England. He worked as an associate professor in faculty of IT in Isra University, Jordan since 2008. His research is centered in software engineering, web and network security, e-learning, bioinformatics, Cloud Computing and ICT fields. Aljawarneh has presented at and been on the organizing committees for a number of international conferences and is a board member of the International Community for ACM, Jordan ACM Chapter, ACS, and IEEE. A number of his papers have been selected as “Best Papers” in conferences and journals.

Puligadda Veereswara Kumar is Professor of Computer Science and Engineering at Osmania University. He is awarded Ph.D. in Computer Science and Engineering in the area of temporal databases from Osmania University. He is currently working as the Head and Professor in the department of Computer Science and Engineering, Acharya Institute of Technology, Bangalore, India. He has more than 30 years of Teaching and R&D experience. A number of research scholars are working under his esteemed guidance towards their Ph.D. He has to his credit nearly 50 research papers in various fields of Engineering, in various national and peer reviewed International Journals. He has published and also presented several research papers at National and International conferences. He served as a Chairman, Board of studies, Computer Science and Engineering at Osmania University College of Engineering and has organized and conducted various staff development programs and workshops. He is a Life Member of MISTE. His interested areas include Temporal databases and Temporal data mining, Bio Informatics, Data mining and Artificial Intelligence. Several research scholars from various reputed universities are pursuing research in his guidance and are also awarded successfully.

Vinjamuri Janaki received Ph.D. degree from J.N.T. University Hyderabad, India and M.Tech degree from R.E.C Warangal, Andhra Pradesh, India. She is currently working as Head and Professor of CSE, Vaagdevi College of Engineering, Warangal, India. Her research interest includes Network security, Data mining, Mobile Adhoc Networks and Artificial Intelligence. She has been involved in the organization as a chief member for various conferences and workshops. She published more than 50 research papers in National and International journals and conferences. She is presently supervising more than 10 scholars towards their research.