Multimed Tools Appl (2018) 77:17643–17692
DOI 10.1007/s11042-017-5185-9

SRIHASS - a similarity measure for discovery of hidden time profiled temporal associations

Vangipuram Radhakrishna[1] · **Puligadda Veereswara Kumar**[2] · **Vinjamuri Janaki**[3]

Received: 1 June 2017 / Revised: 8 August 2017 / Accepted: 30 August 2017 /
Published online: 21 September 2017

Abstract Mining and visualization of time profiled temporal associations is an important research problem that is not addressed in a wider perspective and is understudied. Visual analysis of time profiled temporal associations helps to better understand hidden seasonal, emerging, and diminishing temporal trends. The pioneering work by Yoo and Shashi Sekhar termed as SPAMINE applied the Euclidean distance measure. Following their research, subsequent studies were only restricted to the use of Euclidean distance. However, with an increase in the number of time slots, the dimensionality of a prevalence time sequence of temporal association, also increases, and this high dimensionality makes the Euclidean distance not suitable for the higher dimensions. Some of our previous studies, proposed Gaussian based dissimilarity measures and prevalence estimation approaches to discover time profiled temporal associations. To the best of our knowledge, there is no research that has addressed a similarity measure which is based on the standard score and normal probability to find the similarity between temporal patterns in z-space and retains monotonicity. Our research is pioneering work in this direction. This research has three contributions. First, we introduce a novel similarity (or dissimilarity) measure, SRIHASS to find the similarity between temporal associations. The basic idea behind the design of dissimilarity measure is to transform support values of temporal associations onto z-space and then obtain probability sequences of temporal associations using a normal distribution chart. The dissimilarity measure uses these probability sequences to estimate the similarity between patterns in z-space. The second contribution is the prevalence bound estimation approach. Finally, we give the algorithm for time profiled associating mining called Z-SPAMINE that is primarily inspired from SPAMINE. Experiment results prove that our approach, Z-SPAMINE is computationally more efficient and scalable

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

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

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

[3] Department Computer Science and Engineering, Vaagdevi College of Engineering, Warangal, India

compared to existing approaches such as Naïve, Sequential and SPAMINE that applies the Euclidean distance.

Keywords Temporal · Prevalence · Time stamp · Support bounds · Seasonal patterns · Association · Distance function

1 Introduction

Time profiled temporal association mining is one of the challenging areas of research interest in temporal data mining. The problem of mining interesting seasonal or temporal trends, emerging patterns hidden in time stamped temporal data throws several challenges for researchers to address approaches for improving computational efficiency. One of the drawbacks is the dearth of similarity measures that can address high dimensionality challenges of time stamped temporal data. This area of research is comparatively understudied when compared to its counterparts such as temporal clustering, classification, search & retrieval [8]. The pioneering work that addressed mining temporal associations in time stamped temporal data is SPAMINE by Jin Soung Yoo and Shashi Shekhar [59–61]. Yoo and Shashi Shekhar uses the Euclidean distance measure and propose approaches for support estimation of temporal associations. No effort has been made to devise new dissimilarity measure and this was coined as the future work of authors [60]. It is known that the Euclidean distance measure is sensitive to high dimensional data [34] and hence it is not suitable for efficient mining of temporal associations from time stamped temporal data. Studies and research addressed in [59–61] is extended in our previous research by devising novel fuzzy Gaussian-based dissimilarity measures [8, 44–46]. However, the design of dissimilarity measures proposed in all the previous studies [8, 23, 34, 44–47, 49, 50, 56, 59–61] is not based on the standard score and normal probability. To the best of our knowledge this research proposes a novel approach for mining time profiled associations applying the concept of standard score and probability distribution. In this paper, a dissimilarity measure based on the concept of normal distribution is introduced. i.e. the design of dissimilarity measure is now extended to suit the possibility of mining time profiled temporal association patterns through computing standard scores and normal probability. The basic idea is to transform support value and support sequences into equivalent z-score value and z-score sequences. For these z-score sequences, the probability value is computed using a normal distribution chart. Finally, temporal patterns are expressed as sequences of z-score probability values. This paper extends our previous research studies with a novel contribution by proposing a new dissimilarity measure for retrieving all possible and valid time profiled temporal association patterns from the given input time stamped temporal database.

This paper is outlined as follows: Section-1 explores some of the important and significant studies related to frequent pattern mining, association rule mining, temporal association rule mining, closest related works to time profiled temporal association mining. The scope for present research, basic terminology and notations are also outlined in this section. The proposed prevalence estimation approach and z-score based dissimilarity measure are discussed in section-2 and section-3 respectively. Section-4 outlines the algorithm design and Section-5 gives the time profiled association mining algorithm, Z-Spamine. Experiment results and discussions are discussed in Section-6 by considering various test cases that study algorithm scalability and

performance. Section-7 gives the normal distribution chart used to design the dissimilarity measure in this research. Section-8 concludes this paper.

1.1 Related works

1.1.1 Mining associations in transaction databases

Some important algorithms that address discovery of frequent items in a transaction database are discussed in this subsection. Discovery of association rules in transaction database is initially addressed by introducing two algorithms called AIS and SETM [3]. Both AIS and SETM algorithms generate candidate itemsets on the fly in a given pass as and when data is being read. The idea of AIS and SETM is to verify for itemsets that are found to be large in the previous pass and then extend itemsets in the current pass that contain itemsets from previous pass. In otherwords, the supports are computed for these items in the current pass. The drawback of these two algorithms is that all unnecessary combinations are also considered for generation and counts too many candidate items that are actually not large. This drawback of AIS and SETM algorithms has been well addressed and overcome in the later research by agarwal and srikant [2] which is considered as the significant milestone in the field of data mining. The pioneering work is the apriori [2] and aprioriTid [2] algorithms addressed for mining frequent itemsets and hidden association rules in a transaction database. 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.

Zaki (in year, 2000) proposed scalable algorithm for mining association rules that consider structural properties of frequent items called Eclat algorithm [62]. Eclat algorithm uses lattice traversal technique for finding frequent itemsets and aims at minimizing I/O costs. This work was later extended by Zaki (in year, 2001) by coming with a vertical mining based approach called Dclat algorithm [63] for frequent itemset mining. Dclat algorithm uses novel vertical data representation called Diffset. Mining frequent patterns in time series databases and transaction databases have been extensively studied in data mining and most of the earlier studies use candidate set generation and test such as apriori which is computationally expensive.

Han [21] proposed a compressed tree based approach for finding frequent patterns called FP-tree approach. The advantages of FP-tree approach are i) it generates a highly compact FP-tree that is substantially smaller than original transaction database ii) it avoids costly candidate generation and test process by concatenating the frequent-1 itemsets present in the conditional FP-trees iii) the partitioning based divide and conquer approach reduces the size of the conditional patterns. The FP-tree approach of frequent pattern mining [21] is extended in [11] which applies recursive elimination principle. The advantage of this approach is simple tree structure. Following works of [2] several works on association rule mining have been addressed that includes generalized association rule mining [52], multiple-levl association rule mining [19], quantitative association rule mining [53], high-dimensional association rule mining [58], constraint based and multiple minimum support based ARM [36, 55], incremental association rule mining [16, 30], parallel association rule mining [1, 38]. An interesting work (in year, 2001) carried by cohen [17] address correlation based

association rule mining that is useful in many data mining applications such as clustering web data, finding similar web documents, colloborative filtering and other data mining related applications. Association rules may be extended to web data amalgation [22] w.r.t cloud and concept of similarity measure may be extended to find outliers.

1.1.2 Temporal association rule mining

The fundamental objective of traditional association rule mining (ARM) is to retreive the set of all rules that satisfy certain constraints such as support, confidence of an itemset and the interestingness. Association rule mining for non-temporal databases is extended to temporal databases by introducing the concept of time and the rules so obtained are termed as temporal association rules [4]. The conventional support in ARM does not consider the lifespan of an itemset because of which all transactions are considered irrespective of the lifespan of itemset. This limitation is overcome by considering lifespan of an itemset in which it is valid and introducing the concept of temporal support and confidence. The apriori algorithm for non-temporal databases [2] is extended for temporal databases in [4]. This is followed by several approaches for mining association rules from subset database that consider the time aspect [10, 12, 13, 18, 20, 32, 57]. Although, algorithms such as FP-tree and constraint based approaches exist unfortunately, all these algorithms do not help to discover interesting rules from publication databases. An approach called "progressive partioned miner" [31] is addressed to discover temporal association rules from publication databases and causal relationalship between itemsets that are actually infrequent.

Another type of temporal association rules called cyclic association rules [37] are proposed by Ozden, Ramaswamy, and Silberschatz. Cyclic association rules are association rules that satisfy periodicity. i.e. if an association rule satisfies at a given time point or time instant, then this rule also holds good for all other cycles at that particular instant. Similarly, if a rule does not hold true for a time instance, then for all cycles it also does not hold good. On the otherhand, most of the real life patterns are actually not perfect and the objective is to find all imperfect patterns. Another limitation is that they are not addressed for multiple time granularities but have only been addressed to a single time point. Hence, these cannot at least address a query of the form, "second holiday of every year". Given a time stamped transaction dataset, the problem of mining association rules in calendar schema is addressed in [32, 33].

Most of these studies did not address time profiled temporal association mining that has various applications in stock market exchange, analyzing sales trends in market-basket, climate measurement (such as temperature, moisture, precipitation etc) to mention a few of them. Although, studies [28, 29, 33, 51, 64] have considered transaction data that is implicitly related to time, all these studies 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". Studies [59–61] addressed the problem of "similarity-based temporal association mining" but they were restricted to the use of Euclidean distance measure for mining time profiled associations. Mining temporal patterns from interval databases is addressed in [14] that proposed Gaussian based similarity measure. Summary, detailed information and implementation of various data mining algorithms and respective synthetic and real time data sets 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, clustering and classification are available as open source (http://www.philippe-fournier-viger.com/spmf).

1.2 Time profiled temporal association mining

Similarity profiled temporal association pattern mining is one of the topics of wide research interest in the context of temporal data mining. The pioneering work to address the solution in this direction is by the authors, Jin Soung Yoo and Shashi Shekhar [59–61]. All these studies use Euclidean distance measure. From the extensive literature survey performed and to the best of our knowledge, there are no significant findings recorded in the literature in the direction of proposing new measures to address the above said problem. This fact has motivated us to come up with new similarity measures, so that these measures can be used to retrieve all valid similar temporal patterns w.r.t any chosen reference pattern. Some of our earlier studies [5, 15, 39–43, 48] proposed new similarity (or dissimilarity) measures that extend the basic Gaussian function and the approaches to estimate the supports of temporal association patterns.

In [39], we come up with a dissimilarity measure for mining temporal association patterns, all those patterns whose prevalence variations are same as prevalence variations of reference pattern. The drawback [39], is that it is not addressed as to what deviation must be chosen for applying the dissimilarity measure. This drawback was later overcome and addressed by proposing the expression for computing deviation [8, 44–47, 56] and for choosing proper threshold value corresponding to the deviation. Approaches for estimating supports of temporal association patterns are discussed in [15, 48–50]. These similarity measures designed may also be applied to different applications related to [6, 7, 9]. Application of similarity measures for dimensionality reduction is discussed in [25–27].

1.3 Research scope

The present research is inspired from [23, 34, 59–61]. Past research [49, 50, 59–61] that addressed mining time profiled temporal associations considered the widely known Euclidean distance measure. It is a well-known fact that the Euclidean distance falls prey to high dimensionality and hence does not suit for time profiled association mining, which have the support sequences that are implicitly highly dimensional. Some of our previous works [5, 8, 42–47, 56] propose the fuzzy Gaussian based distance measures for finding similarity between temporal trends and patterns. However, all these distance measures that are proposed does not consider transforming support time sequences to a different time space. The following are several findings that lead to the following research

a. There is a scope for research to find the similarity of temporal associations to a given query time sequence in transforming space. This scope for research has motivated the present work. The idea is to find the standard score of support time sequences of time stamped temporal patterns and then transform these sequences to z-score probability sequences.
b. There is also scope for coming out with a dissimilarity measure that can find the similarity between the temporal pattern in z-space. Our research thus addresses a novel dissimilarity measure to find similarity between time stamped temporal patterns in such transforming space.
c. There is a scope for proposing a method for estimating prevalence time sequence limits of temporal associations that can address computational complexity and is compatible with proposed similarity/distance measure

1.4 Problem statement

Given a time stamped transaction database and subset constraints that include i) reference time sequence ii) allowable dissimilarity threshold and iii) a dissimilarity function that maps the prevalence time sequence of temporal itemset and reference time sequence onto a transformed space in which the similarity between two given temporal associations can be accurately estimated. The problem of time profiled temporal association mining is to discover set of all valid time profiled associations that satisfy all those subset constraints through estimating prevalence time sequence of temporal associations by performing a minimum number of true support computations to achieve improved computational efficiency.

1.5 Basic terminology

In this section, we introduce the basic terms and notations followed in this paper.

1.5.1 Time stamped transaction database

It is defined as the transaction data that is defined over a finite number of 'n' disjoint time slots where each time slot is a point in time.

1.5.2 Temporal item or itemset

An item or itemset present in a given transaction (or transactions) of a time stamped temporal database.

1.5.3 Positive temporal item (or positive itemset)

A positive temporal item or itemset denotes the existence of itemset in the time stamped transaction database. A positive itemset is also called as the positive temporal pattern. Unless specified, the default itemset is considered as a positive temporal item or itemset. A positive itemset say 'I' is denoted using T_I where T denotes temporal nature of itemset.

1.5.4 Negative temporal item (or negative itemset)

A negative temporal item or itemset denotes the non-existence of itemset in the time stamped transaction database. A negative itemset is also called as the negative temporal pattern. A negative temporal itemset is denoted using $\bar{T}_I$ where T denotes temporal nature of itemset.

1.5.5 Prevalence (or support)

Let 't_i' denote i^{th} time slot and the transaction database defined at t_i be represented using the notation, D_i. Prevalence of an itemset, I at timeslot, t_i is denoted using T_{I_i} and is equal to the fraction of transactions that contain the itemset in D_i. It is also called as the support of itemset. The prevalence value is always between 0 and 1. For example, if a particular time slot, t_3 contains 100 transactions and a given item is present in 40 transactions, then the prevalence value of itemset at the time slot, t_3 is equal to 40/100 = 0.4.

1.5.6 Positive prevalence (or support)

Let t_i be the i^{th} time slot and the transaction database defined at t_i be denoted as D_i. The positive prevalence of a temporal item or itemset in a given time slot, t_i is defined as the fraction of transactions that contain the temporal item or itemset in the time slot, t_i. It is also called as support of itemset. The positive prevalence value always lies between 0 and 1. For example, if a particular time slot say, t_1 contains 250 transactions and a given item is present in 25 transactions, then the prevalence value at the time slot considered is 25/250 = 0.1.

1.5.7 Negative prevalence (or support)

Let 't_i' denote i^{th} time slot and the transaction database defined at t_i be represented using the notation, D_i. The negative prevalence of an itemset, I at timeslot, t_i is denoted using $\overline{T}_{I_i}$ and is equal to the fraction of transactions that do not contain the itemset in the database, D_i. Negative prevalence value defines the probability of non-existence of an item or itemset. For example, if a particular time slot, t_3 contains 100 transactions and a given item is present in 40 transactions, then the negative prevalence value of itemset at the time slot, t_3 is equal to 60/100 = 0.6.

1.5.8 Positive prevalence time sequence $\left(\overrightarrow{T_I}\right)$

Let I be any itemset, then the prevalence time sequence is the sequence of prevalence values of a time stamped temporal item or itemset defined over 'n' disjoint time slots and is denoted using $\overrightarrow{T_I}$. The prevalence time sequence corresponding to positive itemset is defined as positive prevalence time sequence. A prevalence time sequence is also called positive prevalence time sequence.

1.5.9 Negative prevalence time sequence $\left(\overrightarrow{\bar{T}_I}\right)$

Let I be any itemset, then the negative prevalence time sequence is the sequence of prevalence values of a time stamped temporal item or itemset defined over 'n'disjoint time slots for a negative temporal itemset and is denoted using $\overrightarrow{\bar{T}_I}$.

2 Prevalence bounds estimation

One of the important challenges that are to be addressed when discovering all the valid similar temporal association patterns from a time stamped temporal database of disjoint transactions is the number of true support computations that are required to be performed. For example, for 'N' items, there exists 2^N item set combinations and hence the complexity is $O(2^N)$. This means that the total number of true support computations required is 2^N in the worst case. Hence, if we can come up with approaches for estimating prevalence bounds of association patterns (or item sets) by devising a suitable procedure then, it shall help to reduce the total number of true

support scans that must be carried out. Sections 2.2 and 2.3 give the approaches for estimating prevalence values of temporal associations.

Let 'N' be the number of items in the finite itemset, 'I' and pattern size be represented using notation 'S'. Suppose that P and Q are any two items chosen from 'I'. Then, notations T_P and T_Q each denote corresponding positive temporal pattern for P and Q respectively. Similarly, notations $\bar{T}_P$ and $\bar{T}_Q$ each denote corresponding negative temporal pattern. For estimating prevalence bounds of patterns, the complete set of possible patterns are mainly divided into three categories, each representing, temporal patterns of sizes, S = 1, S = 2 and S > 2. Necessary expressions which are required to estimate support bounds of temporal association patterns of size (i) |S| = 2 and (ii) |S| > 2 are discussed in subsections below.

2.1 Prevalence bounds for single time slot

Let, T_P and T_Q are any singleton temporal patterns and T_{PQ} is temporal association pattern generated from temporal patterns, T_P and T_Q. The support bounds for temporal association pattern, T_{PQ} are computed using expressions defined in Eq. (1)

$$T_{PQ} = \begin{cases} T_{p_1} - \min\left(T_{p_1}, 1 - T_{Q_1}\right) \\ T_{p_1} - \max\left(T_{p_1} - T_{Q_1}, 0\right) \end{cases} \tag{1}$$

The expression to compute the minimum prevalence bound is given by Eq. (2)

$$T_{PQ}^{min} = T_{p_1} - \min\left(T_{p_1}, 1 - T_{Q_1}\right) \tag{2}$$

The expression to compute the maximum prevalence bound is given by Eq. (3)

$$T_{PQ}^{max} = T_{p_1} - \max\left(T_{p_1} - T_{Q_1}, 0\right) \tag{3}$$

Throughout this discussion, the notation T_{I_k} represents support value of temporal item or itemset at k^{th} time slot.

2.2 Prevalence bounds for level-2 temporal pattern (size, S = 2)

Let, T_P and T_Q are any two temporal patterns defined over 'n' time slots. For 'n' time slots, we denote respective pattern support time sequences as $\overrightarrow{T_P} = (T_{P_1}, T_{P_2}, T_{P_3}, \ldots\ldots\ldots\ldots, T_{P_n})$ and $\overrightarrow{T_Q} = (T_{Q_1}, T_{Q_2}, T_{Q_3}, \ldots\ldots\ldots\ldots, T_{Q_n})$. The bounds for all temporal association patterns (i.e of the form T_{PQ}) at level-2, are obtained applying Eq. (4),

$$\overrightarrow{T_{PQ}} = \begin{cases} \left(\left(T_{P_1} - \min\left(T_{P_1}, 1 - T_{Q_1}\right)\right), \left(T_{P_2} - \min\left(T_{P_2}, 1 - T_{Q_2}\right)\right), \ldots, \left(T_{P_n} - \min\left(T_{P_n}, 1 - T_{Q_n}\right)\right) \right) \\ \left(\left(T_{P_1} - \max\left(T_{P_1} - T_{Q_1}, 0\right)\right), \left(T_{P_2} - \max\left(T_{P_2} - T_{Q_2}, 0\right)\right), \ldots, \left(T_{P_n} - \max\left(T_{P_n} - T_{Q_n}, 0\right)\right) \right) \end{cases} \tag{4}$$

For a temporal pattern of the form, T_{PQ}, the minimum prevalence time sequence is denoted by $\overrightarrow{T_{PQ}^{min}}$ and is given by Eq. (5),

$$\overrightarrow{T_{PQ}^{min}} = \left(T_{PQ_1}^{min}, T_{PQ_2}^{min}, T_{PQ_3}^{min}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, T_{PQn}^{min}\right) \tag{5}$$

where

$$T^{min}_{PQ_1} = T_{P_1} - \min\left(T_{P_1}, 1 - T_{Q_1}\right)$$
$$T^{min}_{PQ_2} = T_{P_2} - \min\left(T_{P_2}, 1 - T_{Q_2}\right)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$T^{min}_{PQ_n} = T_{P_n} - \min\left(T_{P_n}, 1 - T_{Q_n}\right)$$

In similar lines, its maximum prevalence sequence bound is given by Eq. (6),

$$\overrightarrow{T^{max}_{PQ}} = \left(T^{max}_{PQ_1}, T^{max}_{PQ_2}, T^{max}_{PQ_3}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, T^{max}_{PQn}\right) \tag{6}$$

where

$$T^{max}_{PQ_1} = \left(T_{P_1} - \max\left(T_{P_1} - T_{Q_1}, 0\right)\right)$$
$$T^{max}_{PQ_2} = \left(T_{P_2} - \max\left(T_{P_2} - T_{Q_2}, 0\right)\right)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$T^{max}_{PQ_n} = \left(T_{P_n} - \max\left(T_{P_n} - T_{Q_n}, 0\right)\right)$$

2.3 Pattern support bound for 'n' time slots and S > 2

Let, P and Q are any two items of sizes equal to (S-1) and 1 respectively, in a time stamped transaction database of 'n' number of disjoint time slots. Then, notations T_P and T_Q each denote temporal itemset (or pattern) of size, equal to (S-1) and 1 respectively. The respective support time sequence of T_P and T_Q 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 $\overrightarrow{T_Q} = (T_{Q_1}, T_{Q_2}, T_{Q_3}, \ldots\ldots\ldots\ldots, T_{Q_n})$.

A temporal association pattern T_{PQ} is generated from itemset association PQ (or may be viewed as generated from temporal patterns, T_P and T_Q). The size of temporal association pattern, T_{PQ} is equal to |S| (or S) while the size of patterns T_P and T_Q is equal to (|S|-1) and 1 respectively. 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 which are obtained from their superset itemset association, PQ of size equal to |S|. It should be noted that, itemset represented by Ss(PQ) and S(PQ) together form the itemset association 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 whose 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 possible subset combinations possible at level (l-1) and level-1 using which superset itemset combination PQ at level 'l' can be generated.

For example, consider the itemset ABC of size equal to 3, then the possible size-2 itemset are AB, AC and BC while the size-1 itemset are C, B and A. Itemset ABC may be obtained by considering any of these three possible combinations. It can be easily verified that {A, B}∩ {C} = ∅ where Ss(ABC) ≡ AB and S(ABC)≡ C. Similarly, {A, C}∩ {B} = ∅ and {B, C}∩ {A} = ∅. To find the support bounds for temporal itemset ABC, i.e. T_{ABC} we consider all these possible subset combinations.

In general, notations, $T_{\mathrm{Ss}(PQ)_t}$ and $T_{\mathrm{S}(PQ)_t}$ are used to represent the 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 above. Subsection 2.4.4 demonstrates the computation of support bounds of temporal itemset association, ABC.

2.3.1 Minimum support time sequence

The minimum 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 (7) represents the support time sequence of temporal association pattern, T_{PQ} obtained from the k[th] possible subset denoted by Ss(PQ) of size equal to |S|-1 and singleton pattern S(PQ).The minimum support time sequence of temporal association pattern, T_{PQ} obtained over 'n' time slots by considering k[th] subset (i.e $Ss^k(PQ)$) is given by (7)

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

where

$$\begin{aligned} T^k_{PQ_1} &= \left(T_{Ss^k(PQ)_1} - minimum\left\{T_{Ss^k(PQ)_1}, 1 - T_{S(PQ)_1}\right\}\right) \\ T^k_{PQ_2} &= \left(T_{Ss^k(PQ)_2} - minimum\left\{T_{Ss^k(PQ)_2}, 1 - T_{S(PQ)_2}\right\}\right) \\ &\ldots\ldots\ldots\ldots \\ T^k_{PQ_n} &= \left(T_{Ss^k(PQ)_n} - minimum\left\{T_{Ss^k(PQ)_n}, 1 - T_{S(PQ)_n}\right\}\right) \end{aligned}$$

From all possible support time sequences obtained by applying Eq. (7) through considering each subset itemset association denoted by $Ss^k(PQ)$ and $S(PQ)$, the minimum support time sequence is obtained by considering maximum support value at every time slot over all possible subsets of itemset association, PQ. The minimum support time sequence, $\overrightarrow{T_{PQ}{}^{min}}$ of temporal association pattern, T_{PQ} is given by Eq. (8)

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

where

$$\begin{aligned} T^{min}_{PQ_1} &= \text{maximum}\left\{T^1_{PQ_1}, T^2_{PQ_1}, \ldots., T^k_{PQ_1}\right\} \\ T^{min}_{PQ_2} &= \text{maximum}\left\{T^1_{PQ_2}, T^2_{PQ_2}, \ldots., T^k_{PQ_2}\right\} \\ &\ldots.. \\ T_{PQ_n} &= \text{maximum}\left\{T^1_{PQ_n}, T^2_{PQn}, \ldots., T^k_{PQ_n}\right\} \end{aligned}$$

2.3.2 Maximum support time sequence

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(PQ) ∩S(PQ)=∅. Equation (9) represents the support time sequence of temporal association pattern T_{PQ} obtained by considering the kth possible subset denoted by $Ss^k(PQ)$ of size equal to |S|-1 and the singleton pattern S(PQ)

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

where

$$T^k_{PQ_1} = \left(T_{Ss^k(PQ)_1} - \max\left(\left\{T_{Ss^k(PQ)_1} - T_{S(PQ)_1}\right\}, 0\right)\right)$$
$$T^k_{PQ_2} = \left(T_{Ss^k(PQ)_2} - \max\left(\left\{T_{Ss^k(PQ)_2} - T_{S(PQ)_2}\right\}, 0\right)\right)$$
$$\ldots\ldots\ldots\ldots$$
$$T^k_{PQ_n} = \left(T_{Ss^k(PQ)_n} - \max\left(\left\{T_{Ss^k(PQ)_2} - T_{S(PQ)_n}\right\}, 0\right)\right)$$

In all the expressions above $T_{Ss^k(PQ)_t}$ and $T_{S(PQ)t}$ refers to support of kth itemset combination denoted by Ss(PQ) and singleton pattern at tth time slot respectively. From all possible support time sequences obtained by applying Eq. (9) through considering each subset itemset associations denoted by $Ss^k(PQ)$ and $S(PQ)$, the maximum support time sequence is obtained by considering minimum support value at every time slot over all possible subsets of itemset association, PQ. The maximum support time sequence, $\overrightarrow{T_{PQ}{}^{max}}$ is given by Eq. (10)

$$\overrightarrow{T_{PQ}{}^{max}} = \left(T^{max}_{PQ_1}, T^{max}_{PQ_2}, T^{max}_{PQ_3}, \ldots\ldots\ldots\ldots\ldots, T^{max}_{PQ_n}\right) \tag{10}$$

where

$$T^{max}_{PQ_1} = \text{minimum}\left\{T^1_{PQ_1}, T^2_{PQ_1}, \ldots., T^k_{PQ_1}\right\}$$
$$T^{max}_{PQ_2} = \text{minimum}\left\{T^1_{PQ_2}, T^2_{PQ_2}, \ldots., T^k_{PQ_2}\right\}$$
$$\ldots\ldots$$
$$T^{max}_{PQ_n} = \text{minimum}\left\{T^1_{PQ_n}, T^2_{PQn}, \ldots, T^k_{PQ_n}\right\}$$

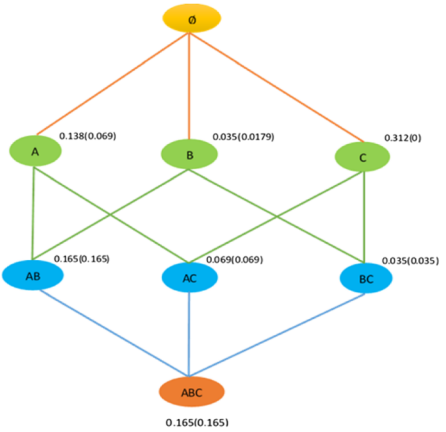
In sub-expressions of Eq. (9) the representation $T^k_{PQ_t}$ denotes the support value obtained by considering the kth possible subset itemset combination at tth time slot and $T^{max}_{PQ_t}$ denotes the maximum possible support value of temporal association pattern T_{PQ} at tth time slot.

2.4 Case study

This section explains the approach for estimating prevalence time sequence bounds of temporal association patterns by applying the method discussed in sections 2.1 to 2.3. For this, the time stamped transaction database generated using IBM data generator [8, 44] as shown in Fig. 1a is considered. The database is defined over two-time slots (denoted by T2). It 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 temporal dataset is denoted as TD10-D20-I3-L2-T2.

D_1		D_2	
Time Slot–1 : (t_1)		Time Slot–2 : (t_2)	
Time	Transaction Items	Time	Transactions
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}

a



b

Fig. 1 **a** Example dataset. **b** monotonicity property of proposed dissimilarity measure w.r.t $D_Z^{max-min}$ showing distance values in the form: D_Z^{true} $(D_Z^{max-min})$

The database is defined over three items A, B and C which form the finite set of items. Figure 1b shows the lattice diagram depicting the distance computations using proposed similarity function. Table 1 shows prevalence values of level-1 (singleton) positive and negative 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 of itemset size equal to one at level-1. T_{A_1}, T_{B_1} are positive supports of patterns at time slot t_1 and T_{A_2}, T_{B_2} are positive supports of patterns at time slot t_2. Similarly, $\bar{T_{A_1}}$, $\bar{T_{B_1}}$ are negative supports at time slot t_1 and $\bar{T_{A_2}}$, $\bar{T_{B_2}}$ are negative supports at time slot t_2.

In subsections 2.4.1 to 2.4.4, the proposed approach for estimating support bounds of temporal association patterns is explained by considering itemset associations AB, AC, BC and ABC.

2.4.1 Prevalence bound of temporal itemset, AB i.e. T_{AB}

Consider the temporal itemset, T_{AB}. The computation of prevalence sequence bounds of temporal patterns can be obtained by applying Eqs. (5) and (6). Figure 2a shows the maximum possible support sequence bound and minimum possible support sequence bound for the temporal itemset, T_{AB}.

Maximum support time sequence of T_{AB}, $(\overrightarrow{T_{AB}^{max}})$ The temporal support sequence of temporal itemset, T_{AB} is denoted by $\overrightarrow{T_{AB}^{max}}$ and can be computed using $\overrightarrow{T_{AB}^{max}} =$

Table 1 Support values of singleton temporal items

Item (I)	Positive Temporal Pattern (T1) (Level-1)	Prevalence at t1 T_{t_1}	Prevalence at t2 T_{t_2}	Negative Temporal Pattern $\overline{(T_1)}$ (Level-1)	Prevalence at t1 $\overline{(T_{t_1})}$	Prevalence at t2 $\overline{(T_{t_2})}$
A	$\mathbf{T_A}$	0.6	0.4	$\overline{T_A}$	0.4	0.6
B	$\mathbf{T_B}$	0.3	0.7	$\overline{T_B}$	0.7	0.3
C	$\mathbf{T_C}$	0.8	0.8	$\overline{T_C}$	0.2	0.2

$\left(T^{max}_{AB_1}, T^{max}_{AB_2} \right)$ where $T^{max}_{AB_1} = T_{A_1} - \max(T_{A_1} - T_{B_1}, 0)$ and $T^{max}_{AB_2} = T_{A_2} - \max(T_{A_2} - T_{B_2}, 0)$. In expressions for $T^{max}_{AB_1}$ and $T^{max}_{AB_2}$ the notation T_{A_1} and T_{B_1} represent positive supports of temporal pattern at time slot, t_1 and T_{A_2}, T_{B_2} are positive supports of temporal patterns at time slot, t_2. In the present example, we have T_{A_1}=0.6, T_{A_2} =0.4, T_{B_1}=0.3, T_{B_2} =0.7. So, $T^{max}_{AB_1} = T_{A_1} - \max(T_{A_1} - T_{B_1}, 0) = 0.6 - \text{maximum } (0.6–0.3, 0) = 0.6\text{- maximum } (0.3, 0) = 0.6–0.3 = 0.3$. Similarly, $T^{max}_{AB_2} = T_{A_2} - \max(T_{A_2} - T_{B_2}, 0) = 0.4 - \text{maximum } (0.4–0.7, 0) = 0.4 - \text{maximum } (-0.3, 0) = 0.4–0 = 0.4$. Hence, $\overrightarrow{T^{max}_{AB}} = (0.3, 0.4)$.

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

From Fig. 2a, it can be verified that the true support sequence of temporal itemset, T_{AB} lies between the maximum possible support sequence ($\overrightarrow{T^{max}_{AB}}$) and minimum possible support

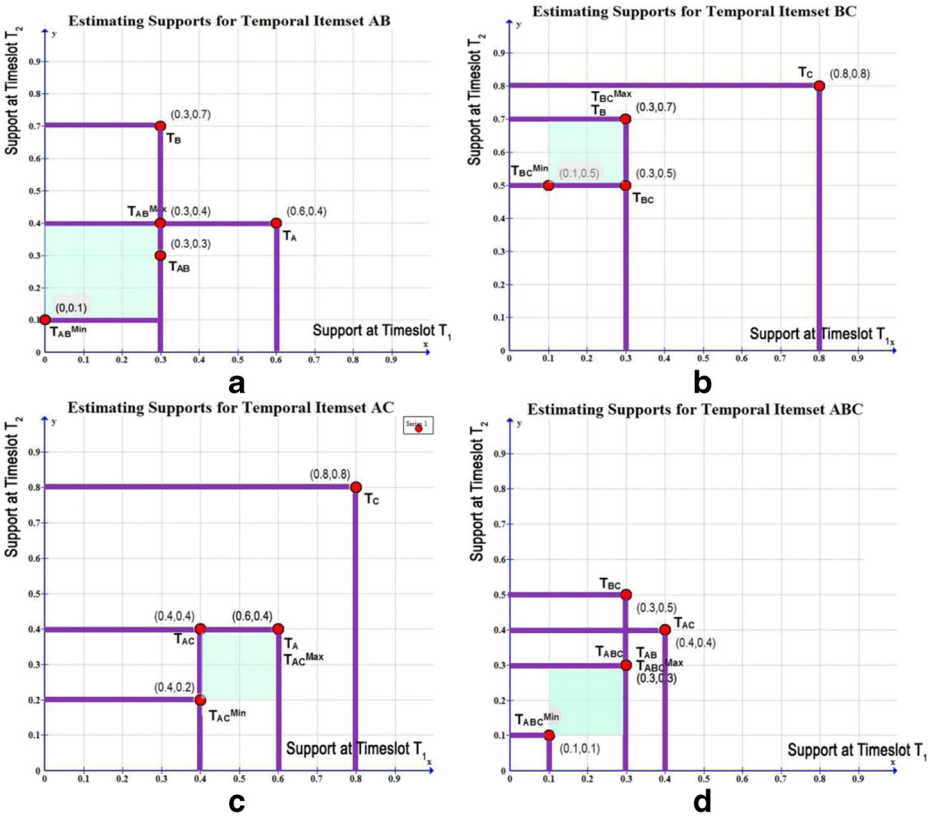


Fig. 2 Support bounds for temporal associations

sequence ($\overrightarrow{T_{AB}^{min}}$) as represented by the shaded region. The shaded region in Fig. 2a is used to represent the fact that the true support of temporal pattern, T_{AB} can only belong to this region.

2.4.2 Prevalence bound of temporal pattern, AC i.e. T_{AC}

The computation of prevalence sequence bounds of temporal pattern, T_{AC} can be obtained by applying Eqs. (5) and (6). Figure 2c shows the maximum possible support sequence bound and minimum possible support sequence bound for the temporal itemset, T_{AC}.

Maximum support time sequence of T_{AC},($\overrightarrow{T_{AC}^{max}}$) The maximum support time sequence of temporal itemset, T_{AC} is denoted by $\overrightarrow{T_{AC}^{max}}$ and can be computed using $\overrightarrow{T_{AC}^{max}} = \left(T_{AC_1}^{max}, T_{AC_2}^{max} \right)$ where $T_{AC_1}^{max} = T_{A_1} - \max(T_{A_1} - T_{C_1}, 0)$ and $T_{AC_2}^{max} = T_{A_2} - \max(T_{A_2} - T_{C_2}, 0)$. In the expressions for $T_{AC_1}^{max}$ and $T_{AC_2}^{max}$ the notations T_{A_1}, T_{C_1} represent positive supports of temporal patterns at time slot, t_1 and T_{A_2}, T_{C_2} are positive supports of temporal patterns at time slot, t_2 . In the present example, we have T_{A_1}=0.6, T_{A_2} =0.4, T_{C_1}=0.8, T_{C_2} =0.8. So, $T_{AC_1}^{max} = T_{A_1} - \max(T_{A_1} - T_{C_1}, 0) =$ 0.6 – maximum (0.6–0.8, 0) = 0.6- maximum (−0.2, 0) = 0.6–0.0 = 0.6. Similarly, $T_{AC_2}^{max} = T_{A_2} -$ $\max(T_{A_2} - T_{C_2}, 0)$ = 0.4 – maximum (0.4–0.8, 0) = 0.4 – maximum (−0.4, 0) = 0.4–0 = 0.4. Hence, $\overrightarrow{T_{AC}^{max}} = (0.6, 0.4)$

Minimum support time sequence of T_{AC}, ($\overrightarrow{T_{AC}^{min}}$) The minimum support time sequence of temporal itemset, T_{AC} is denoted by $\overrightarrow{T_{AC}^{min}}$ and can be computed using $\overrightarrow{T_{AC}^{min}} = \left(T_{AC_1}^{min}, T_{AC_2}^{min} \right)$ where $T_{AC_1}^{min} = T_{A_1} - \min(T_{A_1}, 1 - T_{C_1})$ and $T_{AC_2}^{min} = T_{A_2} - \min(T_{A_2}, 1 - T_{C_2})$. In the present case, T_{A_1}=0.6, T_{A_2} =0.4, T_{C_1}=0.8, T_{C_2} =0.8. So, $T_{AC_1}^{min} = T_{A_1} - \min(T_{A_1}, 1 - T_{C_1})$ = 0.6− min(0.6, 0.2) = 0.4. Similarly, $T_{AC_2}^{min} = T_{A_2} - \min(T_{A_2}, 1 - T_{C_2})$ = 0.4− min(0.4, 0.2) = 0.2. Hence, $\overrightarrow{T_{AC}^{min}}$ =(0.4, 0.2)

2.4.3 Prevalence bound of temporal pattern, T_{BC}

The computation of prevalence time sequence bounds of temporal patterns can be obtained by applying Eqs. (5) and (6). Figure 2b shows the maximum possible support time sequence and minimum possible support time sequence for the temporal itemset, T_{BC}

Maximum support time sequence of T_{BC},($\overrightarrow{T_{BC}^{max}}$) The temporal support sequence of temporal itemset, T_{BC} is denoted by $\overrightarrow{T_{BC}^{max}}$ and may be computed using $\overrightarrow{T_{BC}^{max}} = \left(T_{BC_1}^{max}, T_{BC_2}^{max} \right)$ where $T_{BC_1}^{max} = T_{B_1} - \max(T_{B_1} - T_{C_1}, 0)$ and $T_{BC_2}^{max} = T_{B_2} - \max(T_{B_2} - T_{C_2}, 0)$. In the expressions for $T_{BC_1}^{max}$ and $T_{BC_2}^{max}$ the notations T_{B_1}, T_{C_1} represent positive supports of temporal patterns at time slot, t_1 and T_{B_2}, T_{C_2} are positive supports of temporal patterns at time slot, t_2. From the given dataset, we have T_{B_1}=0.3, T_{B_2} =0.7, T_{C_1}=0.8, T_{C_2} =0.8. So, $T_{BC_1}^{max} = T_{B_1} - \max(T_{B_1} - T_{C_1}, 0) = 0.3$ – maximum (0.3–0.8, 0) = 0.3- maximum (−0.5, 0) = 0.3–0.0 = 0.3. Similarly, $T_{BC_2}^{max} = T_{B_2} - \max(T_{B_2} - T_{C_2}, 0) = 0.7$ – maximum (0.7–0.8, 0) = 0.7 – maximum (−0.1, 0) = 0.7. Hence, $\overrightarrow{T_{BC}^{max}}$ =(0.3, 0.7)

Minimum support time sequence bound of T_{BC}, ($\overrightarrow{T_{BC}^{min}}$) The minimum temporal support sequence of temporal itemset, T_{BC} is denoted by $\overrightarrow{T_{BC}^{min}}$ and can be computed using $\overrightarrow{T_{BC}^{min}} = \left(T_{BC_1}^{min}, T_{BC_2}^{min} \right)$ where $T_{BC_1}^{min} = T_{B_1} - \min(T_{B_1}, 1-T_{C_1})$ and $T_{BC_2}^{min} = T_{B_2} - \min(T_{B_2}, 1-T_{C_2})$. In the presentcase, $T_{B_1}=0.3, T_{B_2}=0.7, T_{C_1}=0.8, T_{C_2}=0.8$. So, $T_{BC_1}^{min} = T_{B_1} - \min(T_{B_1}, 1-T_{C_1}) = 0.3 - \min(0.3, 0.2) = 0.1$. Similarly, $T_{BC_2}^{min} = T_{B_2} - \min(T_{B_2}, 1-T_{C_2}) = 0.7 - \min(0.7, 0.2) = 0.5$. Hence, $\overrightarrow{T_{BC}^{min}} = (0.1, 0.5)$

2.4.4 Prevalence bound of temporal pattern, T_{ABC}

The computation of prevalence sequence bounds of temporal pattern, T_{ABC} can be obtained by applying Eqs. (7) to (10). Figure 2d shows the maximum possible support sequence bound and minimum possible support sequence bound for the temporal itemset, T_{ABC}.

Maximum support time sequence of T_{ABC},($\overrightarrow{T_{ABC}^{max}}$) 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 following three cases

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

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

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

$$\begin{aligned}\overrightarrow{T_{ABC}^2} &= \left(T_{ABC_1}^2, T_{ABC_2}^2 \right) \\ &= (T_{AC_1} - \max(T_{AC_1} - T_{B_1}, 0), T_{AC_2} - \max(T_{AC_2} - T_{B_2}, 0)) \\ &= (0.4 - \max(0.4 - 0.3, 0), 0.4 - \max(0.4 - 0.7, 0)) = (0.3, 0.4)\end{aligned}$$

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

$$\begin{aligned}\overrightarrow{T_{ABC}^3} &= \left(T_{ABC_1}^3, T_{ABC_2}^3 \right) \\ &= (T_{BC_1} - \max(T_{BC_1} - T_{A_1}, 0), T_{BC_2} - \max(T_{BC_2} - T_{A_2}, 0)) \\ &= (0.3 - \max(0.3 - 0.6, 0), 0.5 - \max(0.5 - 0.4, 0)) = (0.3, 0.4)\end{aligned}$$

$$\overrightarrow{T_{ABC}^{max}} = \left(T_{ABC_1}^{max}, T_{ABC_2}^{max} \right) = (\min(0.3, 0.3.0.3), \min(0.3, 0.4, 0.4)) = (0.3, 0.3)$$

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

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 following three cases

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

$$\begin{aligned}\overrightarrow{T^1_{ABC}} &= \left(T^1_{ABC_1}, T^1_{ABC_2}\right)\\ &= (T_{AB_1} - \min(T_{AB_1}, 1 - T_{C_1}), T_{AB_2} - \min(T_{AB_2}, 1 - T_{C_2}))\\ &= (0.3 - \min(0.3, 0.2), 0.3 - \min(0.3, 0.2))\\ &= (0.1, 0.1)\end{aligned}$$

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

$$\begin{aligned}\overrightarrow{T^2_{ABC}} &= \left(T^2_{ABC_1}, T^2_{ABC_2}\right)\\ &= (T_{AC_1} - \min(T_{AC_1}, 1 - T_{B_1}), T_{AC_2} - \min(T_{AC_2}, 1 - T_{B_2}))\\ &= (0.4 - \min(0.4, 0.7), 0.4 - \min(0.4, 0.3))\\ &= (0.0, 0.1)\end{aligned}$$

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

$$\begin{aligned}\overrightarrow{T^3_{ABC}} &= \left(T^3_{ABC_1}, T^3_{ABC_2}\right)\\ &= (T_{BC_1} - \min(T_{BC_1}, 1 - T_{A_1}), T_{BC_2} - \min(T_{BC_2}, 1 - T_{A_2}))\\ &= (0.3 - \min(0.3, 0.4), 0.5 - \min(0.5, 0.6))\\ &= (0.0, 0.0)\end{aligned}$$

$$\overrightarrow{T_{ABC}{}^{min}} = \left(T^{min}_{ABC_1}, T^{min}_{ABC_2}\right) = (max(0.1, 0, 0), max(0.1, 0.1, 0)) = (0.1, 0.1)$$

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, T_{ABC} are $\overrightarrow{T_{ABC}{}^{min}} = (0.1, 0.1)$ and $\overrightarrow{T_{ABC}{}^{max}} = (0.3, 0.3)$ while the true support sequence of temporal pattern, T_{ABC} is (0.3, 0.3).

3 SRIHASS – the proposed Z-score based dissimilarity measure

Let T_p and R_r be the temporal and reference pattern and their respective prevalence values at k^{th} time slot are denoted by T_{p_k}, R_{r_k}. The corresponding prevalence time sequences over 'm' time slots are represented using $\overrightarrow{T_P} = (T_{p_1}, T_{p_2}, T_{p_3}, \ldots\ldots\ldots, T_{p_m})$ and $\overrightarrow{R_r} = (R_{r_1}, R_{r_2}, R_{r_3}, \ldots\ldots\ldots, R_{r_m})$. The dissimilarity measure discussed in this section is motivated from the basic

Gaussian membership function [23] and is extended using [8, 44–47, 56]. Discovering time profiled (or similarity profiled) temporal patterns using proposed approach requires transforming support time sequences of temporal patterns (association pattern) into their equivalent standard score (z-score) values. The idea is to obtain z-score sequences and their corresponding normal probability values using standard normal distribution table.

3.1 Z-score of temporal pattern

The z-score of a temporal pattern at a given time slot is defined as the standard score obtained by considering the support value of a temporal and reference pattern for a chosen deviation (σ^z). The deviation value is a function of threshold value specified in euclidean space denoted using notation, Δ. Formally, the z-score value of a temporal pattern, T_p w.r.t reference, R_r at k^{th} time slot is denoted using $Z(T_{p_k})$ and is computed using Eq. (11),

$$Z(T_{p_k}) = \frac{(T_{p_k} - R_{r_k})}{\sigma^z} \tag{11}$$

where

$$\sigma^z = \frac{\Delta}{\sqrt{ln\left(\frac{1}{abs(1-0.2212^{*}\Delta)}\right)}} \tag{12}$$

3.2 Z-score and probability sequence of temporal pattern

Z-score sequence is the standard score sequence obtained by representing z-score values of a temporal pattern for all 'm' time slots as an m-tuple. Z-score sequence of a temporal pattern over 'm' time slots is denoted by $\overrightarrow{Z(T_p)}$ and is formally represented using Eq. (13)

$$\overrightarrow{Z(T_p)} = \left(Z(T_{p_1}), Z(T_{p_2}), Z(T_{p_3}), \ldots\ldots, Z(T_{p_m})\right) \tag{13}$$

The normal probability of z-score of a temporal pattern, T_p at k^{th} time slot is denoted using the notation, $P(Z(T_{p_k}))$ and the corresponding probability sequence, $\overrightarrow{P(Z(T_p))}$ over 'm' time slots is represented using Eq. (14)

$$\overrightarrow{P(Z(T_p))} = \left(P(Z(T_{p_1})), P(Z(T_{p_2})), P(Z(T_{p_3})), \ldots\ldots\ldots, P(Z(T_{p_m}))\right) \tag{14}$$

3.3 Z-score based dissmilarity measure

Consider the probability sequence, $\overrightarrow{P(Z(T_p))}$ represented by Eq. (14). The membership value of a temporal pattern, T_p at k^{th} time slot w.r.t reference is given by Eq. (15)

$$\mathcal{M}_{R_{r_k}}^{T_{p_k}} = e^{-\left(\frac{P\left(Z\left(T_{p_k}\right)\right)}{\sigma^z}\right)^2} \tag{15}$$

Extending Eq. (15) for 'm' time slots, the normalized similarity of temporal pattern w.r.t reference is given by Eq. (16),

$$\mathcal{M}^{avg}_{T_p,R_r} = \frac{\sum_{k=1}^{k=m} \mathcal{M}^{T_{pk}}_{R_{rk}}}{m} \tag{16}$$

The true dissimilarity between temporal pattern and reference is denoted by D^{true}_Z and is defined using Eq. (17)

$$D^{true}_Z = \frac{1-\mathcal{M}^{avg}_{T_p,R_r}}{0.2212} = \frac{1-\frac{\sum_{k=1}^{k=m} e^{-\left(\frac{P\left(Z\left(T_{p_k}\right)\right)}{\sigma^z}\right)^2}}{m}}{0.2212} \tag{17}$$

Statement: Given $'\Delta'$, T_P, and T_q, two temporal patterns T_P, T_q are considered to be similar, if the computed dissimilarity value denoted by $D^{true}_{T_P,T_q}$ does not exceed, Δ^z. i.e. $D^{true}_{T_P,T_q} \leq \Delta^z$.

3.4 Threshold in Z-space

Let $'\Delta'$ be the threshold specified in Euclidean space which represents the allowable dissimilarity limit between temporal pattern and reference pattern, then, the z-score of Δ is obtained using, $z_\Delta = \frac{\Delta}{\sigma^z}$. The probability of z_Δ that is obtained using normal distribution chart is denoted by $P(z_\Delta)$. The expression for threshold in (normalized space) transformed space is given by Eq. (18)

$$\Delta^z = \frac{1-e^{-\left(\frac{P(z_\Delta)}{\sigma^z}\right)^2}}{0.2212} \tag{18}$$

The value for σ^z used in the Eq. (18) is obtained by applying Eq. (12).

3.5 Deviation

The derivation of expression for deviation is straight forward. We can derive the expression by equating the dissimilarity expression for single time slot using proposed measure and dissmilarity value provided by the user in Euclidean space as depicted in Eq. (19)

$$\frac{1-e^{-\left(\frac{P\left(Z\left(T_{p_k}\right)\right)}{\sigma^z}\right)^2}}{0.2212} = \Delta \tag{19}$$

This results in Eq. (20)

$$e^{-\left(\frac{P\left(Z\left(T_{p_k}\right)\right)}{\sigma^z}\right)^2} = 1-0.2212^*\Delta \tag{20}$$

Solving Eq. (20), we get expression for deviation given by Eq. (21) as specified in Eq. (12)

$$\sigma^z = \frac{\Delta}{\sqrt{ln\left(\frac{1}{abs\left(1-0.2212^*\Delta\right)}\right)}} \tag{21}$$

3.6 Analysis

In this section, we analyze possible values for the similarity measure for different cases.

3.6.1 Best case

In the best case, the dissimilarity between temporal patterns is zero. i.e. the similarity between temporal pattern and the given reference pattern is unity.

For the best case, for k[th] time slot,

$$Z(T_{P_k}) = \frac{\left(T_{p_k} - R_{r_k}\right)}{\sigma^z} = 0$$

This is because $T_{p_k} \cong R_{r_k}$. This gives P($Z(T_{P_k})$) = 0. Hence, $\mathcal{M}_{R_{r_k}}^{T_{p_k}} = e^{-\left(\frac{\mathrm{P}\left(Z\left(T_{P_k}\right)\right)}{\sigma^z}\right)^2} = 1$

Extending this to 'm' number of time slots, we get the average value for similarity as

$$\mathscr{M}_{T_p,R_r}^{avg} = \frac{\sum_{k=1}^{k=m} \mathscr{M}_{R_{r_k}}^{T_{p_k}}}{m} = \frac{1+1+1\ldots\ldots m}{m} = 1$$

Hence the dissimilarity value for the best case situation is

$$D_Z^{true} = \frac{1 - \mathscr{M}_{T_p,R_r}^{avg}}{0.2212} = \frac{1-1}{0.2212} = 0 \tag{22}$$

In other words, similarity between temporal pattern and the reference is, $sim = 1 - D_Z^{true}$ = 1–0 = 1.

3.6.2 Worst case

In the worst case, the dissimilarity between temporal patterns is one (or unity). i.e. the similarity between temporal pattern and the given reference pattern is zero. Similar to the best case, we have in the worst case,

$$\mathscr{M}_{T_p,R_r}^{avg} = \frac{\sum_{k=1}^{k=m} \mathscr{M}_{R_{r_k}}^{T_{p_k}}}{m} = \frac{0.7788 + \ldots m \text{ times}}{m} = 0.7788$$

Hence, the dissimilarity value for the worst case situation is

$$D_Z^{true} = \frac{1 - \mathscr{M}_{T_p,R_r}^{avg}}{0.2212} = \frac{1-0.7788}{0.2212} = 1 \tag{23}$$

In other words, similarity between temporal pattern and the reference is, $sim = 1 - D_Z^{true}$ = 1–1 = 0.

3.7 Distance bound computations

3.7.1 Max-min distance bound

Let, $\overrightarrow{T_p^{max}}$ = ($T_{p_1}^{max}$, $T_{p_2}^{max}$, $T_{p_3}^{max}$,……., $T_{p_m}^{max}$) be the maximum possible support time sequence of temporal pattern, T_p. To obtain maximum possible minimum dissimilarity value, z-score value at each time slot is to be computed by considering $\overrightarrow{T_p^{max}}$ and $\overrightarrow{R_r}$. The Z-score of a temporal pattern at k[th] time slot may be obtained by using Eq. (24)

$$Z\left(T_{p_k}^{max}\right) = \begin{cases} \dfrac{\left(R_{r_k} - T_{p_k}^{max}\right)}{\sigma^z} ; R_{r_k} > T_{P_k}^{max} \\ 0 \qquad\qquad ; R_{r_k} \le T_{P_k}^{max} \end{cases} \tag{24}$$

All these z-score values obtained by applying Eq. (24) for 'm' time slots are represented as the z-score sequence denoted by Eq. (25),

$$\overrightarrow{Z\left(\overrightarrow{T_p^{max}}\right)} = \left(Z\left(T_{p_1}^{max}\right), Z\left(T_{p_2}^{max}\right), Z\left(T_{p_3}^{max}\right)\ldots\ldots\ldots, Z\left(T_{p_m}^{max}\right)\right) \tag{25}$$

The probability sequence, $\overrightarrow{P\left(Z\left(\overrightarrow{T_p^{max}}\right)\right)}$ obtained by computing probability value using normal distribution chart for each $Z(T_{P_k}{}^{max})$ is represented using Eq. (26)

$$\overrightarrow{P\left(Z\left(\overrightarrow{T_p^{max}}\right)\right)} = \left(P\left(Z\left(T_{p_1}^{max}\right)\right), P\left(Z\left(T_{p_2}^{max}\right)\right), \ldots., P\left(Z\left(T_{p_k}^{max}\right)\right)\right) \tag{26}$$

The membership degree between temporal pattern, $\overrightarrow{T_p^{max}}$ and the reference at k[th] time slot is represented using $\mathcal{M}_R^{T_{p_k}U}$ and is given by Eq. (27)

$$\mathcal{M}_R^{T_{p_k}U} = \begin{cases} e^{-\left(\frac{P\left(Z\left(T_{p_k}^{max}\right)\right)}{\sigma^z}\right)^2} ; P\left(Z\left(T_{p_k}^{max}\right)\right) \neq 0 \\ 1 \qquad\qquad\qquad\qquad ; P\left(Z\left(T_{p_k}^{max}\right)\right) = 0 \end{cases} \tag{27}$$

Equation (28) gives the normalized similarity of temporal pattern w.r.t reference considering all, 'm' dis-joint time slots,

$$\mathcal{M}_{T_p}^{avgU} = \frac{\sum_{k=1}^{k=m} \mathcal{M}_R^{T_{p_k}U}}{m} \tag{28}$$

The dissimilarity bound denoted by $D_Z^{max-min}$ is computed using Eq. (29),

$$D_Z^{\max-\min} = \frac{1 - \frac{\sum_{k=1}^{k=m} \begin{cases} e^{-\left(\frac{P\left(Z\left(T_{p_k}^{max}\right)\right)}{\sigma^z}\right)^2} ; P\left(Z\left(T_{p_k}^{max}\right)\right) \neq 0 \\ 1 \qquad\qquad\qquad\qquad ; P\left(Z\left(T_{p_k}^{max}\right)\right) = 0 \end{cases}}{m}}{0.2212} \tag{29}$$

3.7.2 Min-min distance bound

Similar to the computation of max-min distance bound, the distance bound, $D_Z^{min-min}$ is given by Eq. (30)

$$D_Z^{\min-\min} = \frac{1-\frac{\sum_{k=1}^{k=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{p_k}^{min}\right)\right)}{\sigma^2}\right)^2} & ;P\left(Z\left(T_{p_k}^{min}\right)\right)\neq 0 \\ 1 & ;P\left(Z\left(T_{p_k}^{min}\right)\right)=0 \end{cases}}{m}}{0.2212} \tag{30}$$

3.7.3 Minimum distance bound

The dissimilarity degree between temporal pattern (T_p) and reference (R_r) obtained by summing two dissimilarity bounds, $D_Z^{max-min}$ and $D_Z^{min-min}$ is termed as the minimum bound dissimilarity and is denoted by D_Z^{min}. Equation (31) gives the expression for D_Z^{min},

$$D_Z^{min} = D_Z^{max-min} + D_Z^{min-min} \tag{31}$$

4 Algorithm design

In the naive approach for temporal association pattern mining, we must find true supports of all patterns to judge if a temporal pattern is similar or not. This makes the computational complexity class, NP. If these resulting number of true support scans can be reduced, the computational efficiency shall be improved. The proposed Z-Spamine algorithm (which uses proposed dissimilarity measure that is based on standard score named as **SRIHAAS**) is designed to reduce the overall computation cost. The improvement in the computation cost is addressed by

- Proposing approach for estimating temporal pattern prevalence time sequence bounds without examining the temporal dataset before computing true supports for early pruning (section-2)
- Reducing temporal pattern search space through defining temporal dissimilarity measure that hold monotonicity. (section-3)

The following subsections outlines the algorithm design.

4.1 Cover of prevalence time sequences

One of the severe data sensitive operations when discovering time profiled association patterns is generating true prevalence time sequences of temporal patterns (or temporal itemset). This is because in the worst case, this can generate prevalence sequences of all possible temporal

pattern combinations. Approaches for estimating temporal pattern support values are proposed in the work of Calders [54], Jin Soung Yoo [59–61] and Vangipuram [49, 50]. The prevalence estimation approach addressed in section-2 is used in the proposed Z-Spamine algorithm. The dissimilarity measure used in Z-Spamine algorithm is introduced in section-3.

4.2 Minimum dissimilarity bound

The minimum bound dissimilarity of a temporal pattern to a given reference is equal to the sum of the maximum and minimum bounding dissimilarities of temporal pattern. 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 for the reference exceeds user threshold) much ahead in the pattern mining process. This objective is achieved through computing the minimum dissimilarity bound value in Z-spamine pattern mining algorithm. 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.

4.2.1 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 $\overrightarrow{R^U} = (r_1, r_2, \ldots\ldots\ldots, r_w)$ and $\overrightarrow{T_I^L} = (T_{I_1}^{max}, T_{I_2}^{max}, \ldots\ldots\ldots\ldots\ldots\ldots, T_w^{max})$ be the subsequences of $\overrightarrow{R}$ and $\overrightarrow{T_I^{max}}$ respectively, where $r_t > T_{I_t}^{max}$; $1 \le t \le w$. The maximum possible minimum dissimilarity value between temporal patterns, $\overrightarrow{R}$ and $\overrightarrow{T_I^{max}}$, $D_Z^{max-min}\left(\overrightarrow{T_I^{max}}, \overrightarrow{R}\right)$ 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 distance between temporal itemset and the reference sequence. For example, when the proposed dissimilarity function introduced in section-3 is used then,

$$D_Z^{\max-\min}\left(\overrightarrow{T_I^{max}}, \overrightarrow{R}\right) = D\left(\overrightarrow{T_I^L}, \overrightarrow{R^U}\right) = \frac{1-\frac{\sum_{t=1}^{t=m} \mathcal{M}_R^{T_{I_t}^{max}}}{m}}{0.2212} \tag{32}$$

where, $\mathcal{M}_R^{T_{I_t}^{max}} = e^{-\left(\frac{P\left(Z\left(T_{I_t}^{max}\right)\right)}{\sigma^2}\right)^2}$; if $P\left(Z\left(T_{I_t}^{max}\right)\right) \neq 0$ and is equal to 1; if $P\left(Z\left(T_{I_t}^{max}\right)\right) = 0$.

Similarly, the maximum possible minimum dissimilarity between true support sequence of temporal itemset association and the reference is

$$D_Z^{true}\left(\overrightarrow{T_I},\overrightarrow{R_r}\right)=D\left(\overrightarrow{T_I^L},\overrightarrow{R^U}\right)= \tag{33}$$

$$\frac{1-\mathcal{M}_{T_I,R_r}^{avg}}{0.2212}=\frac{1-\frac{\sum_{k=1}^{k=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{I_k}\right)\right)}{\sigma^2}\right)^2} & ;P(Z(T_{I_k}))\neq 0 \\ 1 & ;P(Z(T_{I_k}))=0 \end{cases}}{m}}{0.2212}$$

4.2.2 Definition-2

Given a reference support time sequence, $\overrightarrow{R}$ = $(R_{r_1},R_{r_2},R_{r_3},\ldots\ldots\ldots\ldots,R_{r_m})$, the maximum possible prevalence sequence of an item set, $\overrightarrow{T_I^{max}}$ = ($T_{I_1}^{max}$, $T_{I_2}^{max}$, $T_{I_3}^{max}$, ……,$T_{I_m}^{max}$) and the minimum possible prevalence sequence of an item set, $\overrightarrow{T_I^{min}}$ = ($T_{I_1}^{min}$, $T_{I_2}^{min}$, $T_{I_3}^{min}$, ……,$T_{I_m}^{min}$). The minimum dissimilarity bound, $D_Z^{min}\left(\overrightarrow{T_I^{max}},\overrightarrow{T_I^{min}},\overrightarrow{R_r}\right)$ is equal to the sum of maximum possible minimum dissimilarity bound, $D_Z^{max-min}\left(\overrightarrow{T_I^{max}},\overrightarrow{R_r}\right)$ and minimum possible minimum dissimilarity bound, $D_Z^{min-min}\left(\overrightarrow{T_I^{min}},\overrightarrow{R_r}\right)$. This is formally denoted as

$$D_Z^{min}\left(\overrightarrow{T_I^{max}},\overrightarrow{T_I^{min}},\overrightarrow{R_r}\right)=D_Z^{max-min}\left(\overrightarrow{T_I^{max}},\overrightarrow{R_r}\right)+D_Z^{min-min}\left(\overrightarrow{T_I^{min}},\overrightarrow{R_r}\right) \tag{34}$$

Explanation: The distance is computed considering 'm' time slots and applying proposed dissimilarity measure. If true distance is to be computed then we consider support values of temporal and reference pattern for all time slots. Alternatively, if lower bounding distance is to be computed then, for computing upper-lower bound distance, only those support values which satisfy $R_{r_m} > T_{I_m}^{max}$ at 'm^{th}' time slot are considered. Similarly, for lower-lower bound computation only those pattern support values which satisfy $R_{r_m} < T_{I_m}^{min}$ are considered. This makes the above definition hold well.

4.2.3 Lemma-1

Given the maximum possible prevalence sequence, $\overrightarrow{T_I^{max}}$ = ($T_{I_1}^{max}$, $T_{I_2}^{max}$, $T_{I_3}^{max}$, ……,$T_{I_m}^{max}$), minimum possible prevalence sequence, $\overrightarrow{T_I^{min}}$ = ($T_{I_1}^{min}$, $T_{I_2}^{min}$, $T_{I_3}^{min}$, ……,$T_{I_m}^{min}$), 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 the true distance holds the inequality, $D_Z^{min}\left(\overrightarrow{T_I^{max}},\overrightarrow{T_I^{min}},\overrightarrow{R_r}\right)$ $\leq D_Z^{true}\left(\overrightarrow{T_I},\overrightarrow{R_r}\right)$,if the proposed measure of section-3 is used as a similarity function.

Proof: According to definition of lower-bounding distance using proposed dissimilarity measure, it is known that $D_Z^{min}\left(\overrightarrow{T_I^{max}}, \overrightarrow{T_I^{min}}, \overrightarrow{R_r}\right) = D_Z^{max-min}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) + D_Z^{min-min}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right)$

i.e.

$$D_Z^{min}\left(\overrightarrow{T_I^{max}}, \overrightarrow{T_I^{min}}, \overrightarrow{R_r}\right) =$$

$$\frac{1-\frac{\sum_{k=1}^{k=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{I_k}^{max}\right)\right)}{\sigma^2}\right)^2} & ; R_{r_k} > T_{I_k}^{max} \\ 1 & ; else \end{cases}}{m}}{0.2212} + \frac{1-\frac{\sum_{k=1}^{k=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{I_k}^{min}\right)\right)}{\sigma^2}\right)^2} & ; R_{r_k} < T_{I_k}^{min} \\ 1 & ; else \end{cases}}{m}}{0.2212}$$

$$\leq \frac{1-\frac{\sum_{t=1}^{t=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{I_t}^{max}\right)\right)}{\sigma^2}\right)^2} & ; P\left(Z\left(T_{I_t}^{max}\right)\right) \neq 0 \\ 1 & ; P\left(Z\left(T_{I_t}^{max}\right)\right) = 0 \end{cases}}{m}}{0.2212} + \frac{1-\frac{\sum_{t=1}^{t=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{I_t}^{min}\right)\right)}{\sigma^2}\right)^2} & ; P\left(Z\left(T_{I_t}^{max}\right)\right) \neq 0 \\ 1 & ; P\left(Z\left(T_{I_t}^{max}\right)\right) = 0 \end{cases}}{m}}{0.2212}$$

$$\equiv \frac{1-\frac{\sum_{t=1}^{t=m} e^{-\left(\frac{P\left(Z\left(T_{I_t}\right)\right)}{\sigma^2}\right)^2}}{m}}{0.2212} = D_Z^{true}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) \tag{35}$$

4.3 Non-decreasing property of maximum-minimum dissimilarity bound

This section explores pruning scheme which is used to reduce the temporal itemset (or pattern) search space. Monotonicity property of the support (or prevalence) measure is the most popular technique which is used to reduce the search space of itemset [2]. The prevalence values of all possible superset temporal itemset (or patterns) of a given itemset cannot be greater than item set's prevalence values. Hence, according to the monotonicity property of support measure [2], if a temporal itemset does not satisfy support threshold, then all its superset temporal itemset can also be pruned. If we can come up with an interest measure (also called as dissimilarity measure) which has a property that is like monotonicity then, the search space can be reduced, thus achieving improved computational efficiency. The supporting argument or proof for this is discussed in the following subsection 4.3.1.

4.3.1 Lemma-2

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

Proof: The prevalence time sequence of a temporal pattern is obtained by considering 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 time sequences for all 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).

4.3.2 Lemma-3

The maximum possible minimum (upper-lower) dissimilarity bound 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, the generalized proof for monotonicity of maximum possible minimum bound dissimilarity value to true prevalence time sequences using proposed dissimilarity measure in the section-3 is outlined. 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

$$D_Z^{\max-\min}\left(\overrightarrow{T_I}, \overrightarrow{R_r}\right) = \frac{1-\frac{\sum_{k=1}^{k=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{I_k}^{max}\right)\right)}{\sigma^2}\right)^2}; & P\left(Z\left(T_{I_k}^{\max}\right)\right)\neq 0 \\ 1 ; & P\left(Z\left(T_{I_k}^{\max}\right)\right)=0 \end{cases}}{m}}{0.2212} \quad (36)$$

Consider the size, (k + 1) item set, $\dot{I} = I \cup \{\dot{i}\}$ where $\dot{I} \notin I$. The prevalence time sequence of this temporal pattern is denoted by $\overrightarrow{T_{\dot{I}}} = (T_{\dot{I}_1}, T_{\dot{I}_2}, T_{\dot{I}_3}, \ldots\ldots\ldots\ldots, T_{\dot{I}_m})$. 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. $(T_{\dot{I}_t}) \leq (T_{I_t})$ 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 $T_{\dot{I}_t} \leq T_{I_t}$, $T_{I_t} < R_{r_t}$ and $T_{\dot{I}_t} < R_{r_t}$ then, $R_{r_t} - T_{I_t} \leq R_{r_t} - T_{\dot{I}_t}$. This means that

$$\frac{1-\frac{\sum_{t=1}^{t=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{I_t}\right)\right)}{\sigma^2}\right)^2}; & P(Z(T_{I_t}))\neq 0 \\ 1 ; & else \end{cases}}{m}}{0.2212} \leq \frac{1-\frac{\sum_{t=1}^{t=m}\begin{cases} e^{-\left(\frac{P\left(Z\left(T_{\dot{I}_t}\right)\right)}{\sigma^2}\right)^2}; & P\left(Z\left(T_{\dot{I}_t}\right)\right)\neq 0 \\ 1 ; & else \end{cases}}{m}}{0.2212} \quad (37)$$

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

4.4 Temporal pattern pruning

We follow pruning strategies discussed in [8, 44, 59–61] but apply the pruning strategy using the 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. The pattern pruning scheme is explained below.

4.4.1 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, $D_Z^{max-min}\left(\overrightarrow{T_I^{max}},\overrightarrow{R_r}\right)$ 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.

4.4.2 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, D_Z^{min} exceeds the allowable dissimilarity limit.

4.4.3 Pruning using maximum possible minimum dissimilarity $D_Z^{max-min}\left(\overrightarrow{T_I^{max}},\overrightarrow{R_r}\right)$

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, $D_Z^{max-min}\left(\overrightarrow{T_I^{max}},\overrightarrow{R_r}\right)$ to true prevalence sequence of a temporal pattern exceeds the dissimilarity threshold value. In this case, the temporal pattern is not retained for generating higher size candidate temporal patterns.

5 Z-SPAMINE

In this section, we outline the algorithm for mining time profiled temporal associations, Z-Spamine. Z-Spamine algorithm uses the proposed dissimilarity measure, SRIHASS as the similarity function to find similarity between temporal pattern and reference. Z-Spamine is

extended by considering spamine proposed by Yoo and Sashi Sekhar [59–61] that uses Euclidean distance measure as the similarity function. Our approach requires transforming the threshold value to a new space, say z-space.

5.1 Algorithm

The Z-Spamine approach is outlined as an algorithm in this subsection.

Algorithm: Z-Spamine

Input: A time stamped temporal database, reference temporal trend, value of dissimilarity threshold

Output: Time profiled temporal associations

Similarity function: Srihass

For Level-1 item sets ($L = 1$)

1. From the time stamped temporal dataset, obtain the prevalence time sequence for each singleton temporal item set defined in I
2. **for each singleton temporal item set**
3. Compute True Distance, $D_Z^{true}(\overrightarrow{T_I}, \overrightarrow{R_r})$ of temporal itemset, T_I to reference, R_r .
4. **If** ($D_Z^{true}(\overrightarrow{T_I}, \overrightarrow{R_r}) \leq \Delta^z$) then
5. Item set is **similar**
6. **else**
7. Compute $D_Z^{max-min}(\overrightarrow{T_I}, \overrightarrow{R_r})$ for each Item set
8. **If** ($\boldsymbol{D_Z^{max-min}(\overrightarrow{T_I}, \overrightarrow{R_r}) \leq \Delta^z}$) then
9. Item set is **retained**
10. **else**
11. Item set is **pruned**
12. **End**
13. Using all the **similar** and **retained** item sets generate the item set combination sequences.

For Other Level item sets ($L \neq 1$)

14. **For each item set**
15. **Bound Estimation**
16. Generate ($\overrightarrow{T_I^{max}}$) and ($\overrightarrow{T_I^{min}}$) of next level itemset using the true prevalence of previous level subset prevalence sequences.
17. Calculate $\boldsymbol{D_Z^{max-min}(\overrightarrow{T_I}, \overrightarrow{R_r})}$ and $\boldsymbol{D_Z^{min-min}(\overrightarrow{T_I}, \overrightarrow{R_r})}$ for each itemset using $\overrightarrow{T_P^{max}}$ and $\overrightarrow{T_P^{min}}$ along with reference sequence.
18. Compute $\boldsymbol{D_Z^{min}(\overrightarrow{T_I}, \overrightarrow{R_r}) = D_Z^{max-min}(\overrightarrow{T_I}, \overrightarrow{R_r}) + D_Z^{min-min}(\overrightarrow{T_I}, \overrightarrow{R_r})}$
19. **If** ($\boldsymbol{D_Z^{min}(\overrightarrow{T_I}, \overrightarrow{R_r}) \leq \Delta^z}$) then
20. Item set is similar (Bounded Approach)
21. Compute True Distance value $\boldsymbol{D_Z^{true}(\overrightarrow{T_I}, \overrightarrow{R_r})}$ for each Item set.
22. **If** ($\boldsymbol{D_Z^{true}(\overrightarrow{T_I}, \overrightarrow{R_r}) \leq \Delta^z}$) then
23. Item set is **similar**
24. **else**
25. Compute $\boldsymbol{D_Z^{max-min}(\overrightarrow{T_I}, \overrightarrow{R_r})}$ for each Item set using Z-Score probability tuple $P(\boldsymbol{Z})$
26. **If** ($\boldsymbol{D_Z^{max-min}(\overrightarrow{T_I}, \overrightarrow{R_r}) \leq \Delta^z}$) then
27. Item set is **retained**
28. **else**
29. Item set is **pruned**
30. **else**
31. Item set is **not Similar and pruned**
32. **End**
33. Continue iterations for next levels until all item sets are pruned or level equals the number of item sets in I.

Explanation • Steps 1–12: Finding patterns at first level

The computation process starts with obtaining true supports of first level temporal itemsets from the input dataset. These supports obtained at every time slot are tranformed to standard scores and probability scores using deviation given by Eq. (12). Compute D_Z^{true} between reference and candidate pattern. If this satisfies the similarity condition, then the pattern is similar. Otherwise compute $D_Z^{max-min}$. If this distance satisfies similarity condition, then retain the pattern else kill the pattern. Record all retained and similar patterns at first level.

- **Step 14–32: Finding patterns at level, L > 1**

For each next level itemset, estimate the support bounds of itemsets at each time slot by considering the previous stage itemset true supports. Obtain D_Z^{min} using the computed support bounds. If D_Z^{min} satisfies the similarity constraint, then find the true support of itemset. Now, compute $D_Z^{max-min}$ and D_Z^{min}. The pattern is similar if D_Z^{min} satisfies similarity condition. If D_Z^{min} exceeds the similarity condition, $D_Z^{max-min}$ is computed. If this value satisfies similarity constraint, then itemset is retained. Other wise, it is pruned. Record all retained and similar itemsets at each level.

- **Step 33:**

From the retained itemsets of previous levels, generate the combinations of itemsets for next level and repeat the steps from 14 to 32. Prune all itemset combinations that are supersets of subset itemset combinations that are not similar and not retained at the previous level. Continue the process in steps 14–32.

5.2 Working example

This section explains the working of Z-spamine algorithm using proposed dissimilarity measure. Consider the dataset shown in Fig. 1a and Lattice diagram depicted in Fig. 1b. The true distance and maximum-minimum dissimilarity values are shown in the lattice diagram. The true distance values of temporal itemset are as follows: $D_Z^{true}\left(\overrightarrow{T_A},\overrightarrow{R_r}\right) = 0.138$, $D_Z^{true}\left(\overrightarrow{T_B},\overrightarrow{R_r}\right) = 0.035$, $D_Z^{true}\left(\overrightarrow{T_C},\overrightarrow{R_r}\right) = 0.312$, $D_Z^{true}\left(\overrightarrow{T_{AB}},\overrightarrow{R_r}\right) = 0.165$, $D_Z^{true}\left(\overrightarrow{T_{AC}},\overrightarrow{R_r}\right) = 0.069$, $D_Z^{true}\left(\overrightarrow{T_{BC}},\overrightarrow{R_r}\right) = 0.035$, $D_Z^{true}\left(\overrightarrow{T_{ABC}},\overrightarrow{R_r}\right) = 0.165$. It is easy to verify that true distance does not satisfy monotonicity. For example, $D_Z^{true}\left(\overrightarrow{T_{AC}},\overrightarrow{R_r}\right) < D_Z^{true}\left(\overrightarrow{T_A},\overrightarrow{R_r}\right)$, $D_Z^{true}\left(\overrightarrow{T_C},\overrightarrow{R_r}\right)$. Similarly, $D_Z^{true}\left(\overrightarrow{T_{BC}},\overrightarrow{R_r}\right) < D_Z^{true}\left(\overrightarrow{T_C},\overrightarrow{R_r}\right)$. i.e. the condition that the dissimilarity of superset temporal patterns must be greater than the distance of its subset temporal patterns fails in these cases. However, this property holds good w.r.t $D_Z^{max-min}$ of temporal pattern to the reference. For example, the maximum-minimum dissimilarity values for itemset associations AB, AC, BC, ABC are as follows:

$$D_Z^{max-min}\left(\overrightarrow{T_{ABC}},\overrightarrow{R_r}\right) = 0.165, D_Z^{max-min}\left(\overrightarrow{T_{AB}},\overrightarrow{R_r}\right) = 0.165, D_Z^{max-min}\left(\overrightarrow{T_{AC}},\overrightarrow{R_r}\right) = 0.0692\,.$$

i.e. the distance values of all superset temporal patterns are greater than their subset temporal patterns. This also holds good for all superset temporal patterns w.r.t subset temporal patterns. It is to be noted that the values of max-min dissimilarity bound as indicated within parentheses while the true distance values are shown outside the parentheses in Fig. 1b.

6 Results and discussions

This section outlines results obtained by applying the prevalence estimation approach of section 2 and dissimilarity measure of section 3 in z-spamine algorithm. Test cases considered for the experiment are i) comparison of true support computations performed by naive using Euclidean and Z-Spamine using proposed dissimilarity measure ii) effect of threshold value iii) effect of the total number of transaction items, iv) effect of total number of time slots, v) effect of number of transactions per time slot. Subsections 6.1 to 6.5 outlines the results obtained from experiments performed by considering various test cases for scalability that includes

i) comparison of true support computations
ii) Effect of varying threshold value
iii) Effect of varying number of transaction items
iv) Effect of varying number of time slots
v) Effect of varying number of transactions per time slot

6.1 Comparison of true support computations

This section outlines some of the results obtained by applying the proposed prevalence estimation approach in section 2.

Figure 3 depicts true support computations required for a randomly generated temporal database denoted as TD1000-T100-I20. 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 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. Figure 4 shows total number of true support computations required using proposed support bound estimation procedure for different thresholds 0.15, 0.25 and 0.35 applying dissimilarity measure in section 3 and Z-Spamine algorithm.

Figure 5 depicts the graph comparing the true support computations carried using naïve and proposed approaches for a threshold, $\delta = 0.35$.

Figure 6 depicts the graph comparing retained association patterns to consider for similarity when adopting the naïve and proposed approaches for thresholds, δ equal to 0.15, 0.25 and 0.35.

6.2 Varying threshold value

In the second experiment, we considered varying threshold values by maintaining constant values for transaction items, time slots and number of transactions per time slot. The

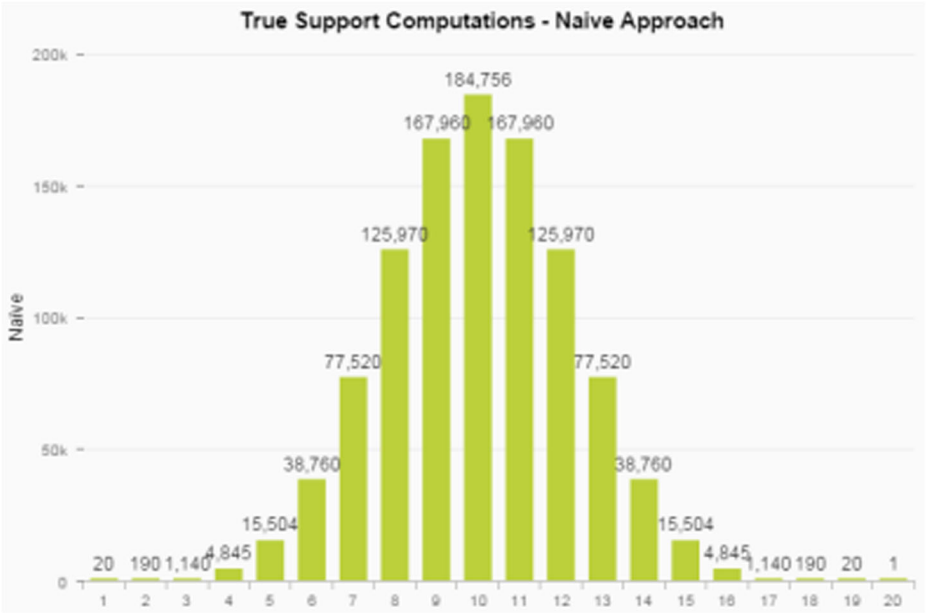


Fig. 3 True support computations – naïve approach

experiment is conducted by considering a time stamped temporal dataset generated for 10 transaction items, 100 time slots and 1000 transactions per time slot. Threshold values are varied from 0.12 to 0.26 in steps of 0.02 and execution times for naïve, sequential, spamine and Z-Spamine approaches are plotted as a bar graph shown in the Fig. 7. It can be verified

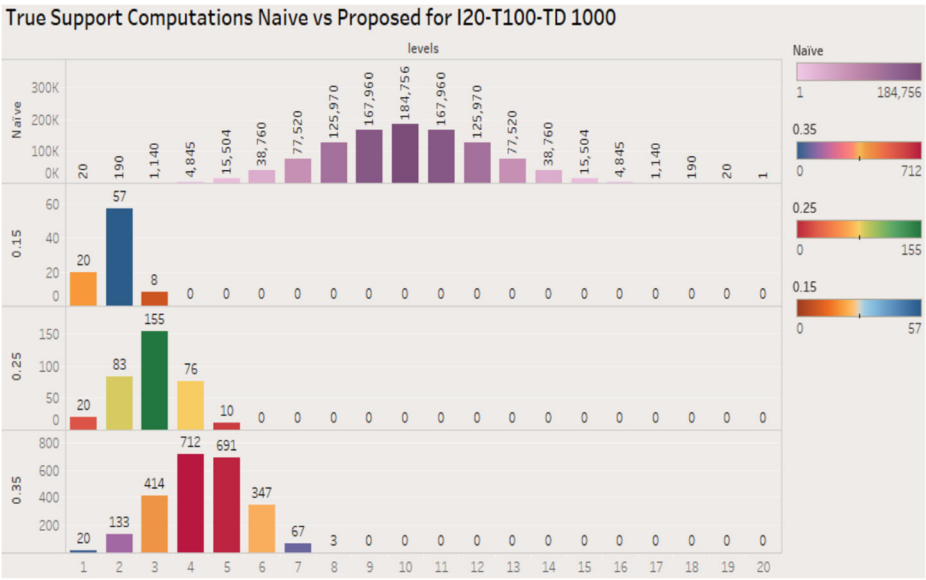


Fig. 4 True support computations – Z-Spamine

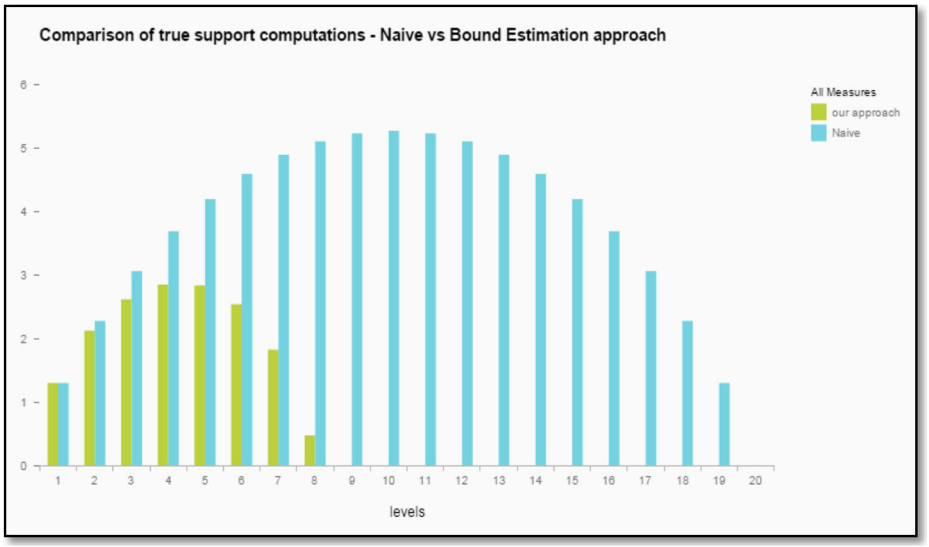


Fig. 5 Naïve vs. proposed - I20-T100-TD1000 and $\delta = 0.35$

from the graph that Z-Spamine performs better to naïve and sequential approaches and has comparatively similar or better execution times w.r.t Spamine approach.

Figure 8 shows the execution times of sequential and Z-Spamine approaches for various thresholds varied from 0.12 to 0.26 in steps of 0.02. The number of transaction items considered are 12. The total number of time slots is 100 and transactions in each time slot are 1000. From the graph depicted in Fig. 8, it can be verified that the sequential approach is very sensitive to variations in the threshold value.

Figure 9 shows the effect of varying thresholds on naïve, sequential, spamine and z-spamine approaches on a time stamped temporal synthetic dataset generated consisting of 12

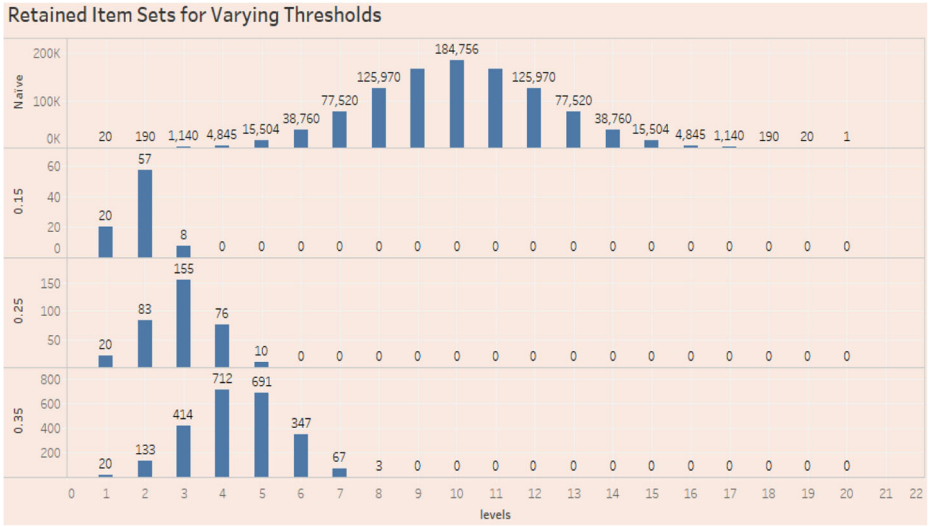


Fig. 6 Retained patterns - naïve vs. proposed by level for, $\delta = 0.25$

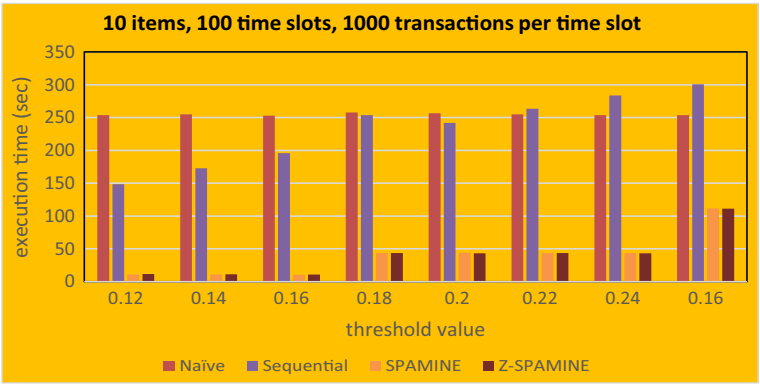


Fig. 7 Varying thresholds – 10 Items

items, 100 time slots, and 1000 transactions per time slot. The execution time of naïve and sequential approaches are very high compared to spamine and z-spamine. Hence, we considered to plot the graph considering log value of execution time. Z-Spamine has better performance when compared to naïve and sequential approaches and has almost similar performance to spamine. The effect of varying thresholds on execution times of spamine and Z-Spamine algorithms can be viewed from the graph represented using Fig. 10 as this graph only compares spamine to Z-Spamine.

The performance of our algorithm w.r.t spamine is clearer from the bar graph in Fig. 10. It shows that the execution time of Z-Spamine is better to spamine.

6.3 Effect of varying total number of transaction per time slot

In the third experiment, the total number of transactions per time slot are 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 11 depicts the execution times of naïve, sequential and z-spamine approaches for varying transactions per time slot equal to 500, 750, 1000, 1250, 1300, 1350, 1400, 1450, 1500 and 2000. The dotted line graph indicates the exponential trend line of naïve approach.

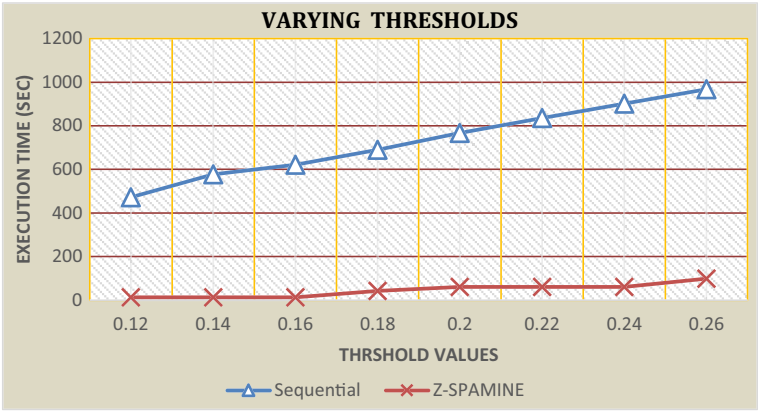


Fig. 8 Varying thresholds – 12 Items

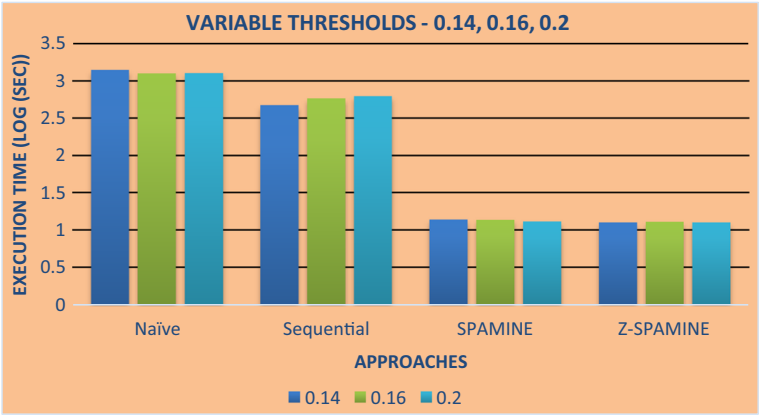


Fig. 9 Effect of varying thresholds on four approaches

The total number of transactions are 1,00,000 defined over 100 time slots. It can be verified that the proposed approach (Z-Spamine) performs better to sequential and naïve approaches. Both sequential and naïve uses Euclidean distance measure. The reduce in time taken for Z-Spamine to output the result set is due to the reduced number of true support computations that are carried by Z-Spamine.

The comparison of execution times of Naive and Z-Spamine approaches for varying transactions per time slot equal to 500, 750, 1000, 1250, 1500 and 2000 is shown in Fig. 12. The dotted line graph indicates the linear trend line for naïve approach. The graph shows that Z-Spamine performs substantially better to naïve approach.

The performance of Sequential [61] and Z-Spamine approaches for varying transactions per time slot equal to 500, 1000, 1500 and 2000 are depicted in Fig. 13. The dotted line graph indicates the exponential trend for sequential approach and is denoted by the exponential function expression, $y = 25.07e^{0.9391x}$. The trend denoted by dotted exponential line shows that the execution time of sequential approach is sensitive to the total number of transactions per time slot and increases in an exponential manner. On the other hand, execution time of Z-Spamine is comparatively very much better to sequential approach and less sensitive to number of transactions per time slot.

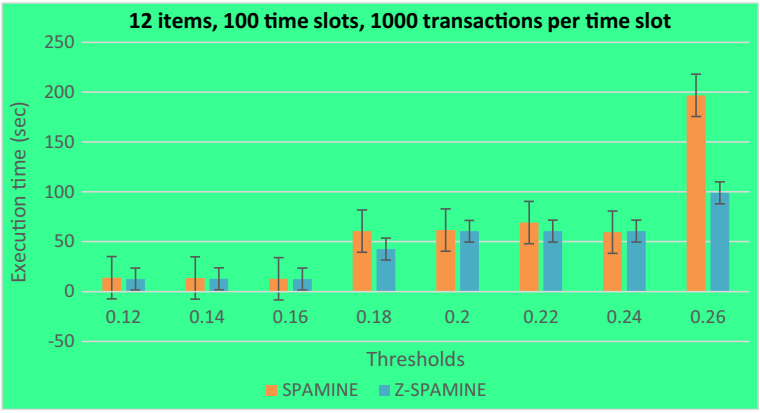


Fig. 10 Spamine vs Z-Spamine – 12 Items

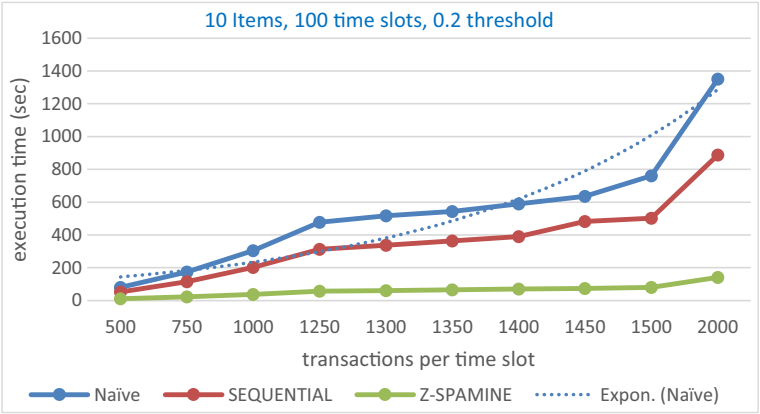


Fig. 11 Comparison of naïve, sequential and Z-Spamine

The performance of Spamine using Euclidean distance measure and proposed Z-Spamine approach using support bound estimation is recorded in the Fig. 14. Two dotted lines in the graph show the exponential trend line plotted w.r.t Spamine (top dotted line) and 2- point moving average (bottom dotted line). The execution times plotted in the graph of Fig. 14 proves that the execution time of Spamine has comparatively more exponential behavior than Z-Spamine approach. This is because the execution time of spamine meets the exponential trend line which is not true for Z-spamine as visible clearly for 2000 time slots. So, it can be deduced that, as the number of time slots increases, Z-Spamine is comparatively more scalable to Spamine approach.

Figure 15 represents execution times of naïve, sequential and z-spamine approaches for varying transactions per time slot equal to 500, 750, 1000, 1250, 1300, 1350, 1400, 1450, 1500 and 2000 for a constant number of transaction items equal to 12. The two dotted line graphs indicate the exponential trend line of naïve and sequential approaches. The total number of transactions are 1,00,000 defined over 100 time slots. From the above graph, Z-Spamine has comparatively better performance to sequential and naïve approaches. Also, the

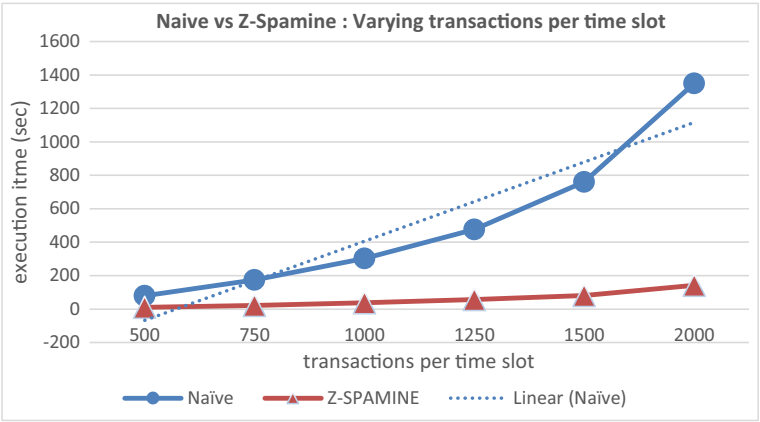


Fig. 12 Execution times - naïve and Z-Spamine

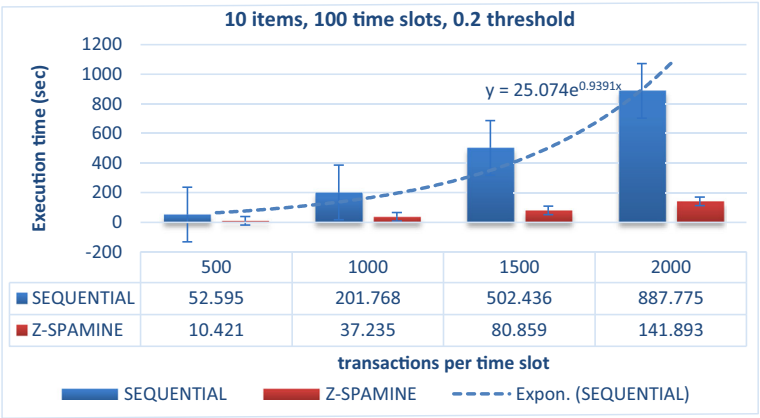


Fig. 13 Sequential vs Z-Spamine for varying transactions per time slot

two-dotted exponential trend in Fig. 15 proves that the execution time of Z-Spamine is less sensitive to the total number of transactions per time slot.

The performance of Z-Spamine and Spamine approaches is recorded in Fig. 16 by varying transactions per time slot. The total number of transactions per time slot chosen are 500, 1000, 1500 and 2000 for a time-stamped temporal database defined over 100 time slots and the total number of transaction items equal to 12. Two trend lines are plotted corresponding to Spamine showing linear and exponential behaviors. The bar graph in Fig. 16 shows that the performance of Z-Spamine is almost same or better to Spamine. In general, Z-Spamine performance is comparatively better than Spamine.

6.4 Varying total number of time slots

In the fourth experiment, we consider varying the 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 10. The minimum number of transactions is 50 k and maximum number of transactions is 200 k. Figure 17a compares the execution time of naïve, sequential and z-

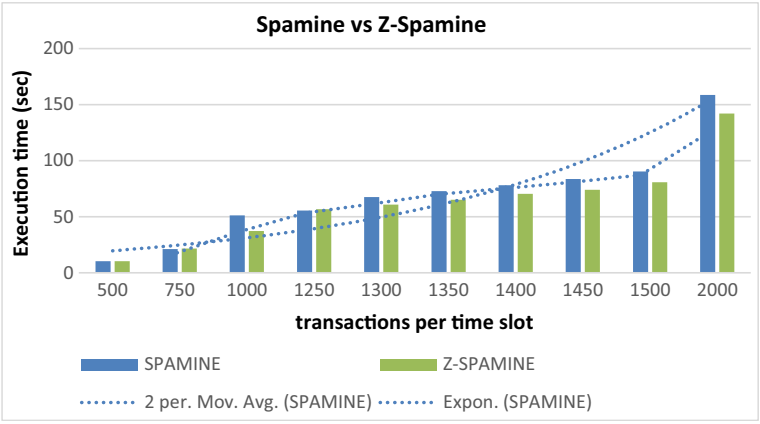


Fig. 14 Spamine vs Z-Spamine for varying transactions per time slot

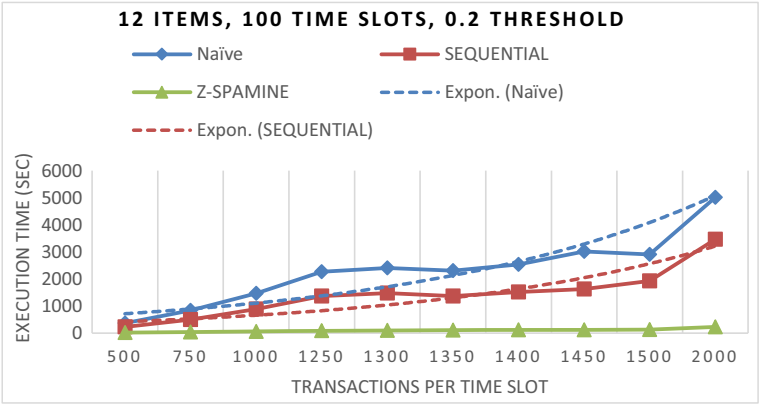


Fig. 15 Effect of varying transactions per time slot – 12 Items

spamine approaches. The proposed approach performs better to naïve and sequential approaches. Execution times of Spamine and Z-Spamine are compared in Fig. 17b. The performance comparison of naïve, sequential, Spamine using Euclidean distance and Z-Spamine using proposed dissimilarity measure is depicted in Fig. 17c. The execution time of Z-Spamine is better compared to all other approaches. It can be concluded that execution time of z-spamine is less sensitive to change in time slots when compared to other approaches.

The graph shown in Fig. 18 depicts the execution time comparison of spamine and z-spamine approaches for 12 items, 100 transactions per time slot and 0.2 threshold. The performance of z-spamine is better to Spamine approach that applies the Euclidean distance.

6.5 Varying total number of transaction items

In the fifth experiment, the execution time and performance of naïve, sequential, spamine and Z-Spamine approaches are studied and recorded by varying the total number of transaction items. Figure 19 shows the line graph plotted depicting, the execution times of naïve, sequential and z-spamine approaches. The number of transaction items considered are 10,

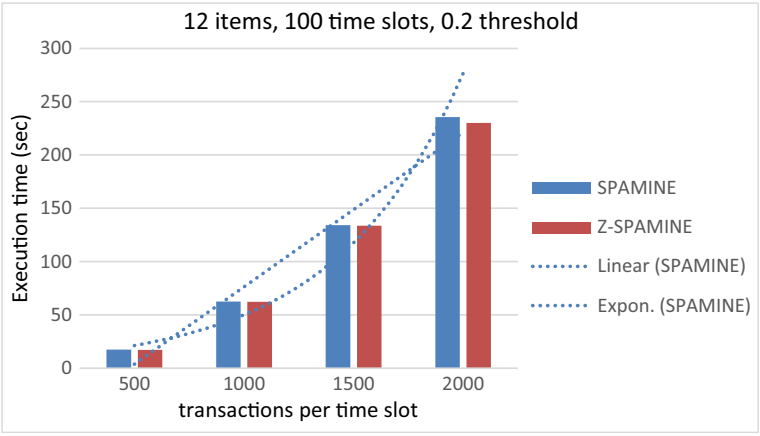


Fig. 16 Effect of varying transactions per time slot on Spamine and Z-Spamine - 12 Items

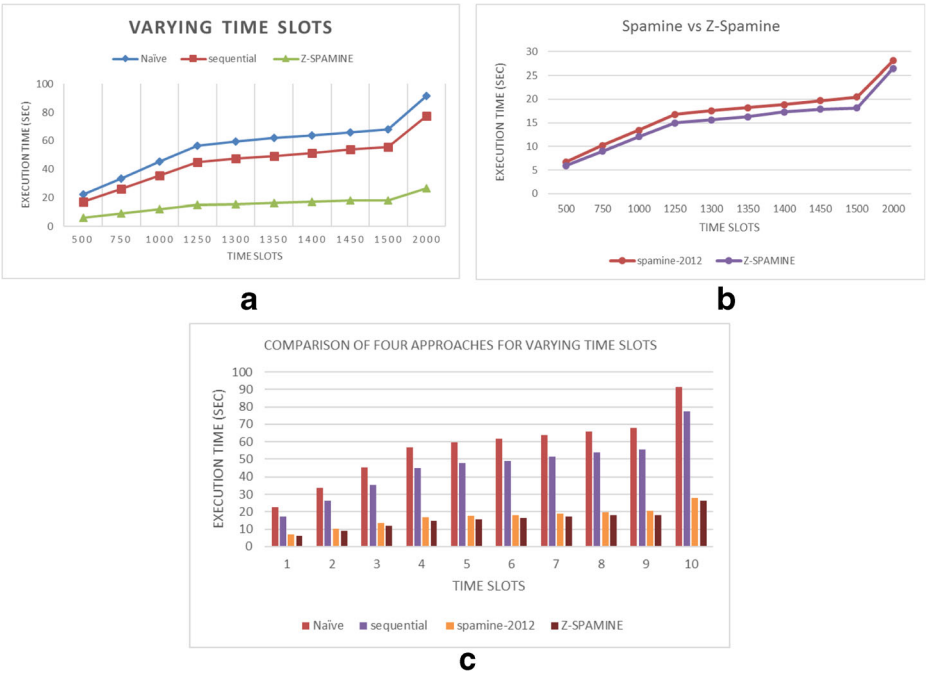


Fig. 17 **a** Comparison of naïve, sequential, Z-Spamine. **b** Comparison of Spamine, Z-Spamine for varying time slots for varying time slots. **c** Comparison of all approaches

11, 12, 13 and 14. The time stamped temporal database considered for this experiment has 100 time slots and each time slot has 1000 transactions which are constant. The threshold chosen is 0.2. The graph proves that Z-spamine is less sensitive to the increase in number of transaction items whereas the other two approaches i.e. naïve and sequential are more sensitive to number of transaction items and shows exponential increase in time.

The comparison of execution times of Spamine and Z-Spamine for variable transaction items (10, 11, 12, 13 and 14) is shown in Fig. 20 using a line graph. Similarly, Fig. 21 depicts execution times of Spamine and Z-Spamine for varying number of transaction items equal to

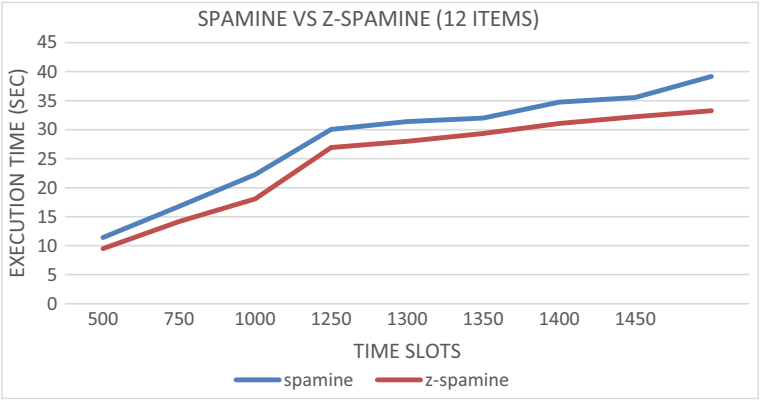


Fig. 18 Z-Spamine vs Spamine

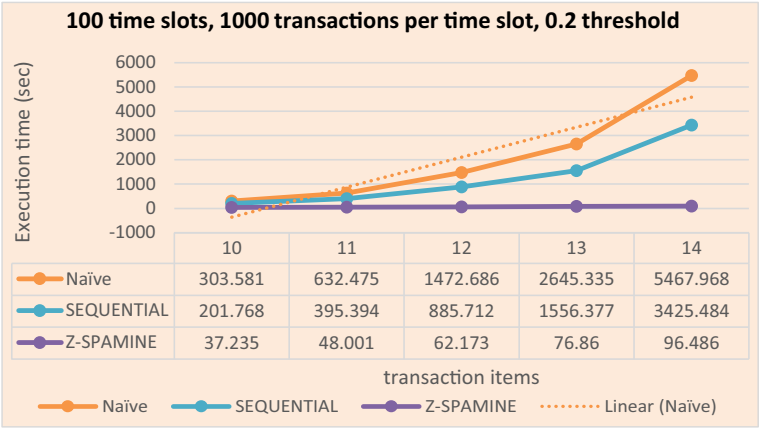


Fig. 19 Effect of varying transaction items

10,11,12,13,14,15,16,17,18,19 and 20. Both these graphs depict that Z-Spamine has nearly same or lesser execution time when compared to Spamine that applies the Euclidean distance. It is to be noted that Z-Spamine uses the proposed dissimilarity measure and support estimation technique.

The bar chart in Fig. 22 shows the execution time of two approaches (spamine and Z-Spamine) for number of transaction items equal to 10, 15, 20 and 25. In general, we can conclude that Z-Spamine has almost similar execution time or less execution time compared to Spamine approach. It is also seen that Z-Spamine is less sensitive to increase in transaction items compared to Spamine.

6.6 Visualization

To visualize the number of temporal associations for which true supports are computed, number of retained temporal associations and temporal trends, we have devoloped a visual tool for time profiled temporal association mining. Figure 23 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

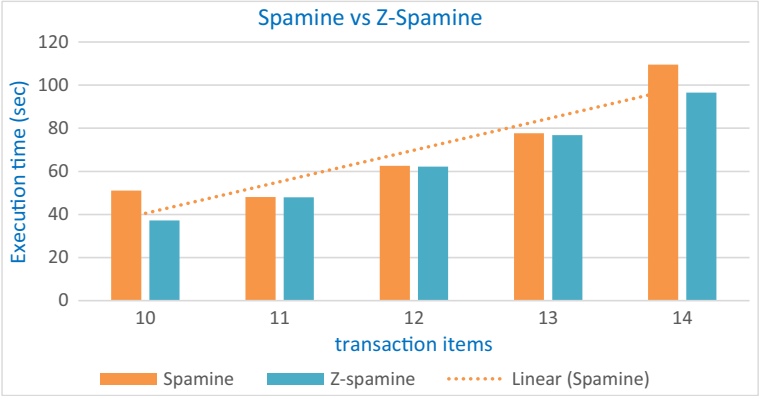


Fig. 20 Effect of varying transaction items

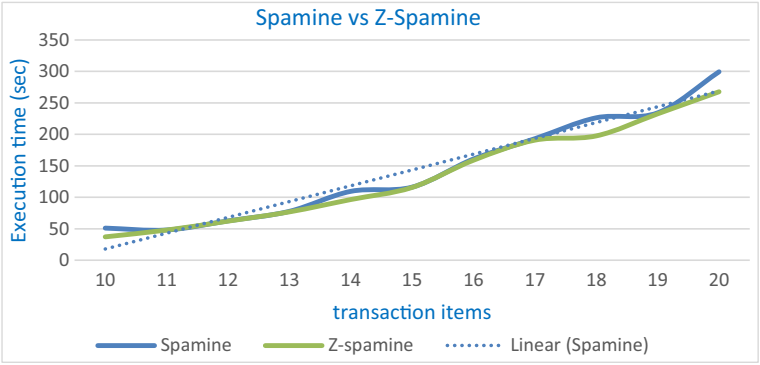


Fig. 21 Effect of varying transaction items

slot iii) number of items iv) threshold v) reference sequence. A provision is also made available to choose any available and existing time stamped temporal dataset.

Figure 24 shows various algorithms that may be chosen for visualizing and understanding temporal trends of patterns to a given user reference sequence pattern.

Figures 25 and 26 depicts the number of true support computations and number of retained itemsets over a time stamped temporal data having 100 time slots, 1000 transactions/timeslot, 14 items for a chosen threshold equal to 0.2.

Figures 27 and 28 depicts the number of true support computations and retained itemsets required on a time stamped temporal data consisting of 100 time slots, 500 transactions/timeslot, 12 items for a chosen threshold equal to 0.25 for naïve approach. It can be seen that both the number of true support computations and retained itemsets are equal for all levels for naïve approach and the graphs are bell-shape curve.

Figures 29 and 30 depicts the number of true support computations and retained itemsets required on a time stamped temporal data consisting of 100 time slots, 500 transactions/timeslot, 12 items for a chosen threshold equal to 0.25 with proposed z-spamine approach. It can be seen that, both the number of true support computations and retained itemsets are

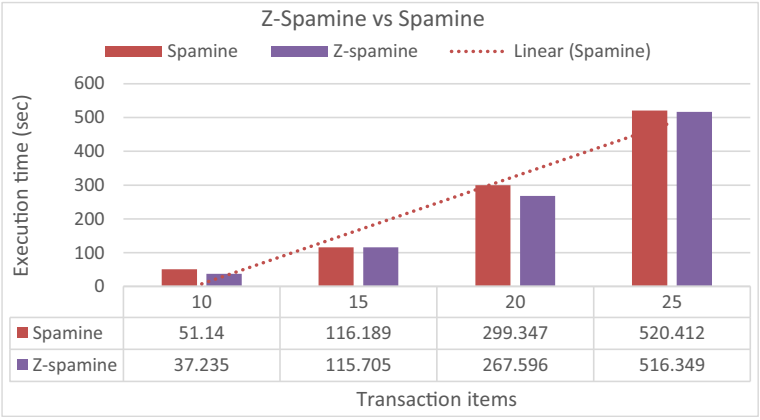


Fig. 22 Effect of varying transaction items

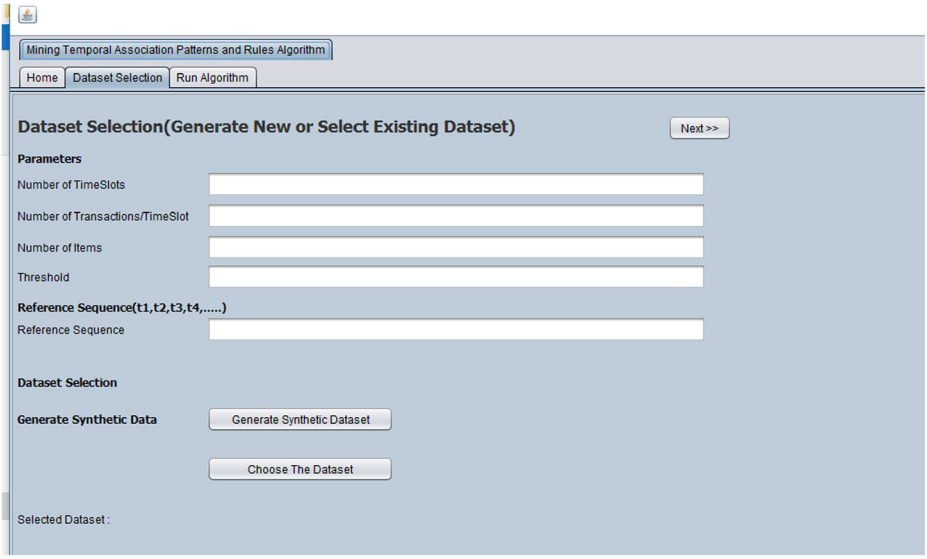


Fig. 23 Visualization – data selection screen

reduced compared to the naïve approach. It is also visually clear that the true supports and retained itemsets after level-4 for z-spamine is almost zero, whereas for naïve approach it keeps increasing till level-6 and then starts decreasing till level-12. Similarly, the seasonal pattern trends may also be visualized by selecting the trend visualization option. This provides the seasonal temporal trends that are hidden in the dataset w.r.t reference.

The following section gives details of the probability distribution chart used for design of similarity measure.

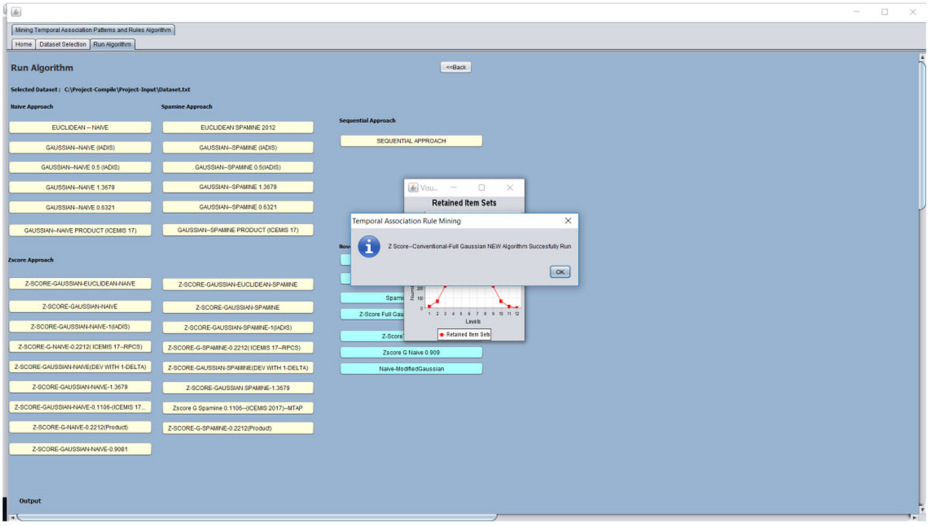


Fig. 24 Visualization – algorithm selection screen

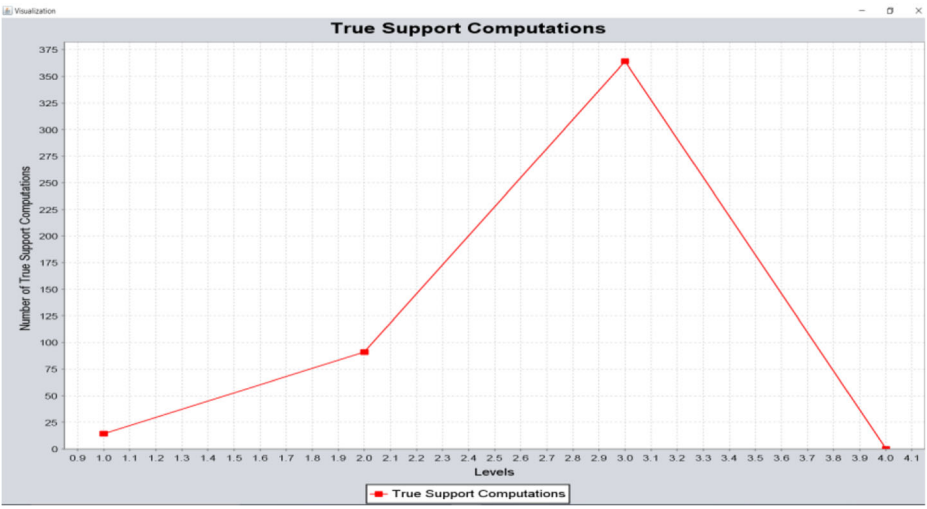


Fig. 25 Visualization – true support computations

7 Normal distribution chart

Figure 31 shows the normal curve standard deviation graph and Fig. 32 shows the normal distribution probability chart using which the probailities for z-score value are obtained. To compute z-score value between temporal pattern and reference, we apply Eqs. (11) and (12). The probability value for this z-score is obtained using the distribution chart (http://www.westbrookstevens.com/continuous.htm), [35] in Fig. 32.

The probability sequence of temporal associations obtained from these z-score sequences are used to compute the dissimilarity between patterns. For example, the minimum value of z-score considered is 0 and the maximum possible value considered is 3.09 as shown in the distribution chart in Fig. 32. The normal probability for z = 0 is equal to 0 and for z = 3.09 it is

Fig. 26 Visualization – retained itemsets

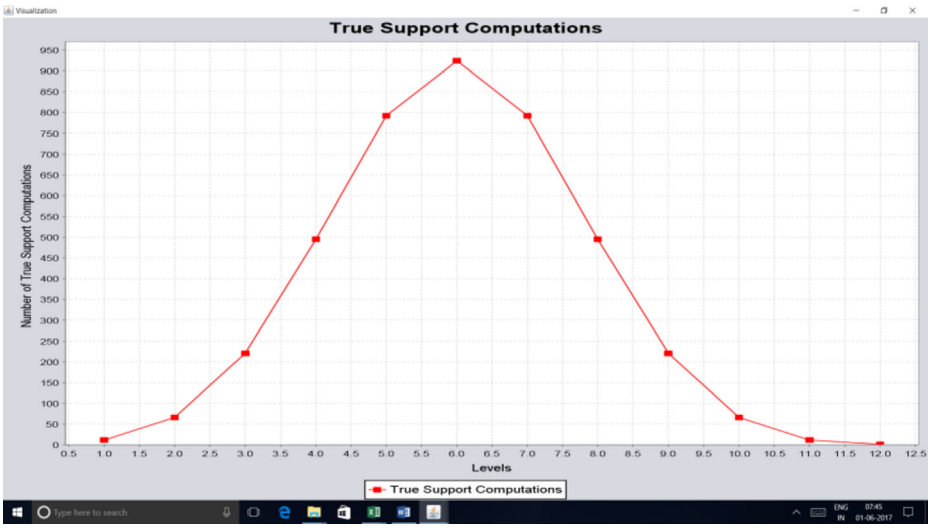


Fig. 27 visualization – retained itemsets

equal to 0.5. The normal probability value, hence lies between 0 and 0.5. In this research, the similarity measure is designed by considering the probability distribution chart of Fig. 32. The following is the procedure to use the z-chart

i) Find the z-value between temporal pattern support and reference support using standard normal value formula.
ii) After finding the z-value, use the distribution chart to find the probability value.
iii) Locate the z-value number from the left side of the chart by going vertically down. Once the first part of the number is found then locate the second part of the number by moving horizontally across the probability chart.

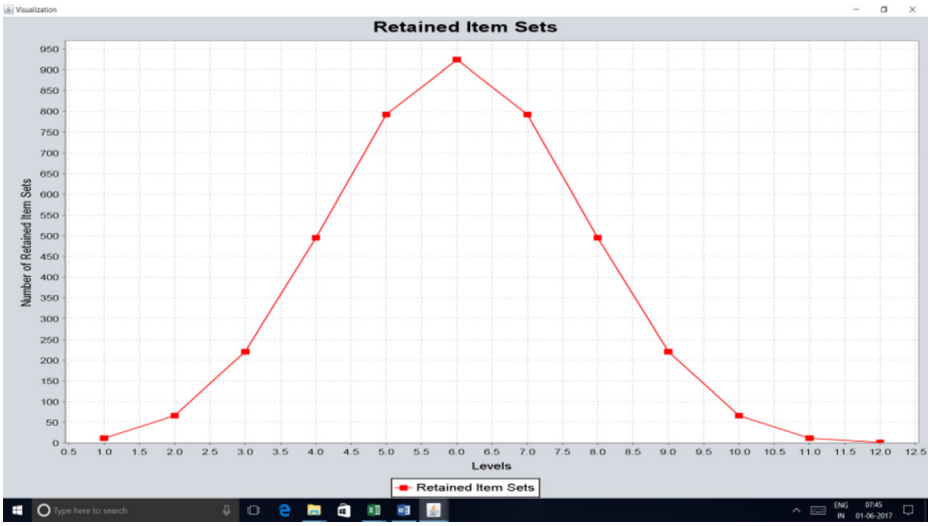


Fig. 28 visualization – retained itemsets

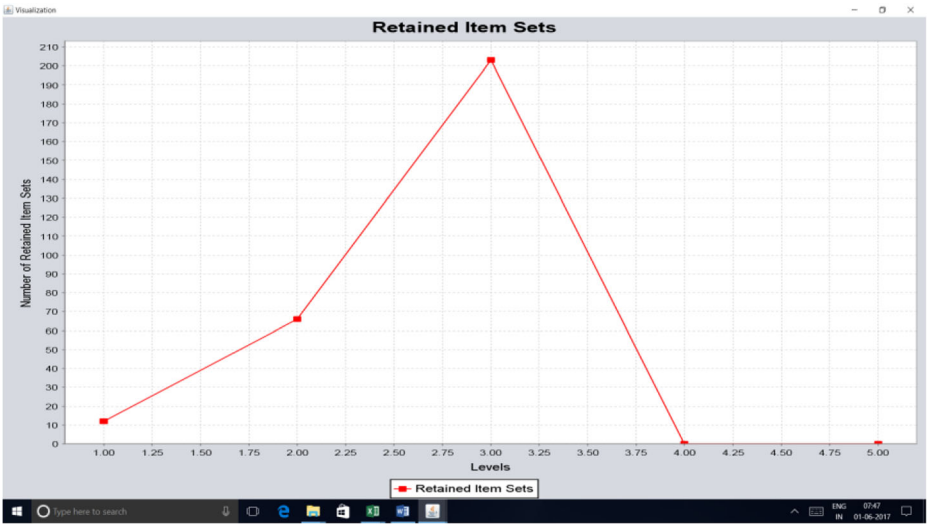


Fig. 29 Visualization – retained itemsets

iv) The value obtained by the defining cell is the probability of z-score.

8 Conclusions

Similarity-based temporal associating mining was initially coined by Yoo and Sashi Sekhar. Yoo and Sashi Sekhar applied the Euclidean distance and support estimation approaches to reduce computational complexity. Following the initial research of Yoo, subsequent works were restricted to the use of Euclidean distance. To the best of our knowledge, there are no known research studies, that have proposed novel dissimilarity measures, that apply standard

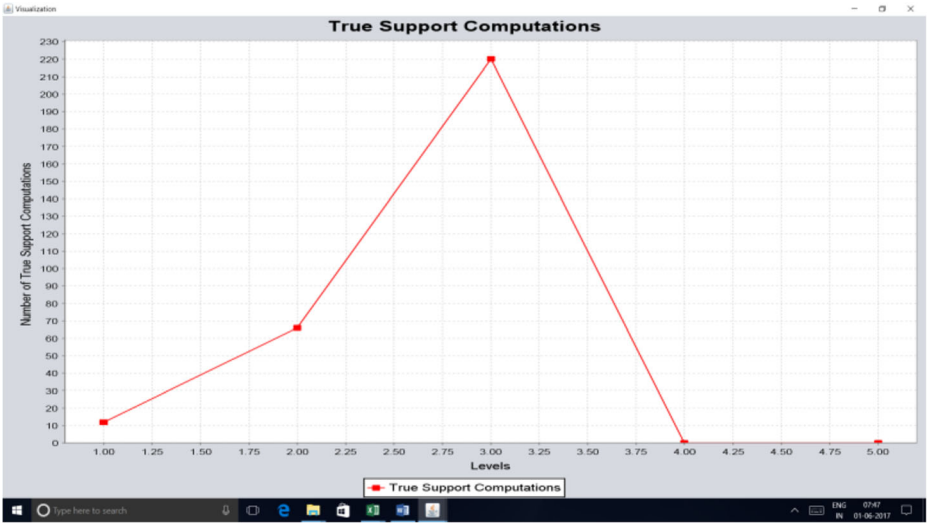


Fig. 30 Visualization – retained itemsets

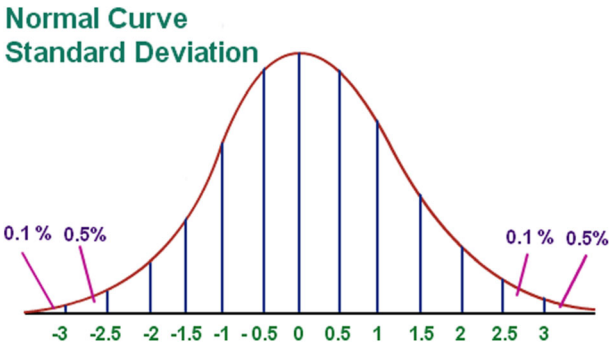


Fig. 31 Normal curve standard deviation

score and probability distribution for time profiled temporal association mining and retain monotonicity property. Our research is pioneering work in this direction. In this research, we apply standard score and probability distribution concept to discover similarity-based time profiled temporal associations. This research introduces a novel dissimilarity measure, SRIHASS that is based on standard score computation for discovering time profiled temporal associations.

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.0000	0.0040	0.0080	0.0120	0.0160	0.0199	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.0910	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.1480	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.1700	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.1950	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.2190	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2517	0.2549
0.7	0.2580	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.2910	0.2939	0.2967	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.3340	0.3365	0.3389
1.0	0.3413	0.3438	0.3461	0.3485	0.3508	0.3531	0.3554	0.3577	0.3599	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.3770	0.3790	0.3810	0.3830
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.3980	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.4370	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.4750	0.4756	0.4761	0.4767
2.0	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.4830	0.4834	0.4838	0.4842	0.4846	0.4850	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.4890
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.4920	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.4940	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.4960	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.4970	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.4980	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3.0	0.4987	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.4990	0.4990

Fig. 32 Normal distribution probability chart

Our similarity measure facilitates to understand hidden seasonal and emerging temporal trends among temporal patterns. We also proposed an approach for estimation of prevalence values of association patterns which helps to know the limits of support values and fits proposed measure. Estimating support limits helps to perform early pruning of invalid temporal associations by applying proposed dissimilarity measure. It is also analytically proved that the proposed measure holds closure property which can be used to prune temporal patterns. Several experiments have been carried under various test cases and the computational performance of Z-Spamine using proposed similarity measure is compared to the naïve, sequential and spamine approaches that apply the Euclidean distance. Experiment results and analysis prove that our approach is computationally more efficient and is substantially scalable compared to other approaches.

Acknowledgments Vangipuram Radhakrishna is heartfully thankful to professor Dr. P.V. Kumar and Dr. V. Janaki for their continuous support and constructive suggestions, through out this research. I am personally thankful to my mentor and professor P.V. Kumar for whatever I am today. I thank my father and professor Dr. Vangipuram Narasimha Charyulu for creating passion towards research. I am thankful to Dr. D.N. Rao, honorable president, and Dr. C.D. Naidu, Principal, VNR VJIET for his constant inspiration and motivation throughout my stay at VNR VJIET. Authors also thank Mr. Aravind Cheruvu, who successfully graduated from Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology for his sincere effort and involvement in shaping this research for the past few years. Finally, I thank my loving son, Srihass without whose cooperation this research shall not have been possible.

References

1. Agrawal R, Shafer JC (1996) Parallel mining of association rules. IEEE Trans Knowl Data Eng 8(6):962–969 https://doi.org/10.1109/69.553164
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, 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
4. 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
5. 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 https://doi.org/10.1109/ICEMIS.2016.7745355
6. Aljawarneh SA, Elkobaisi MR, Maatuk AM (2016) A new agent approach for recognizing research trends in wearable systems. Computers & Electrical Engineering, Available online 16 December 2016, https://doi.org/10.1016/j.compeleceng.2016.12.003
7. 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
8. 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 ISSN 0167-739X, https://doi.org/10.1016/j.future.2017.01.013
9. Aljawarneh S, Aldwairi M, Yassein MB (2017) 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
10. 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
11. Christian Borgelt (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, pp 66–70. https://doi.org/10.1145/1133905.1133914
12. Chen X, Petrounias I (2000) Discovering temporal association rules: algorithms, language and system. In: Proceedings of 16th International Conference on Data Engineering (Cat. No.00CB37073), pp 306–306. https://doi.org/10.1109/ICDE.2000.839423

13. Chen X, Petrounias I (1999) Mining temporal features in association rules. In: Żytkow JM, Rauch J (eds) Principles of data mining and knowledge discovery. PKDD 1999. Lecture Notes in Computer Science, vol 1704. Springer, Berlin, Heidelberg
14. 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
15. 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
16. Cheung D, Han J, Ng V, Wong CY (1996) Maintenance of discovered association rules in large databases: an incremental updating technique. Proc. 1996 Int'l Conf. Data Eng, pp 106–114. https://doi.org/10.1109/ICDE.1996.492094
17. Cohen E, Datar M, Fujiwara S, Gionis A, Indyk P, Motwani R, Ullman JD, Cheng Y (2001) Finding Interesting Associations without Support Pruning. IEEE Trans on Knowl and Data Eng 13(1):64–78 https://doi.org/10.1109/69.908981
18. 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, pp 43–52. https://doi.org/10.1145/312129.312191
19. Han J, Fu Y (1995) Discovery of multiple-level association rules from large databases. In: Dayal U, PMD G, 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
20. Han J, Dong G, Yin Y (1999) Efficient mining of partial periodic patterns in time series database. In: Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337), Sydney, pp 106–115. https://doi.org/10.1109/ICDE.1999.754913
21. 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(1):53–87. Kluwer Academic Publishers. https://doi.org/10.1023/B:DAMI.0000005258.31418.83
22. Imran A, Aljawarneh SA, Sakib K Web Data Amalgamation for Security Engineering: Digital Forensic Investigation of Open Source Cloud. J Univers Comput Sci 22(4):494–520 https://doi.org/10.3217/jucs-022-04-0494
23. 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
24. 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). ACM, New York, pp 69:1–69:7. https://doi.org/10.1145/2832987.2833082
25. Kumar GR, Mangathayaru N, Narasimha G (2016a) An approach for intrusion detection using novel Gaussian based Kernel function. J Univers Comput Sci 22(4):589–604. https://doi.org/10.3217/jucs-022-04-0589
26. Kumar GR, Mangathayaru N, Narsimha G (2016b) Design of novel fuzzy distribution function for dimensionality reduction and intrusion detection. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, pp 1–6 https://doi.org/10.1109/ICEMIS.2016.7745346
27. Kumar GR, Mangathayaru N, Gugulothu N, Suresh Reddy G (2016) CLAPP: A self constructing feature clustering approach for anomaly detection, Future Generation Computer Systems, Available online 4 January 2017, ISSN 0167-739X, https://doi.org/10.1016/j.future.2016.12.040
28. Last M, Klein Y, Kandel A (2001) Knowledge discovery in time series databases. IEEE Trans Syst Man Cybern Part B Cybern 31(1):160–169 https://doi.org/10.1109/3477.907576
29. Lee W-J, Lee S-J (2004) Discovery of fuzzy temporal association rules. IEEE Trans Syst Man Cybern Part B Cybern 34(6):2330–2342 https://doi.org/10.1109/TSMCB.2004.835352
30. Lee C-H, Lin C-R, Chen M-S (2001) Sliding-window filtering: an efficient algorithm for incremental mining. In: Paques H, Liu L, Grossman D (eds) Proceedings of the tenth international conference on Information and knowledge management (CIKM '01). ACM, New York, pp 263–270 https://doi.org/10.1145/502585.502630
31. 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
32. Li Y, Ning P, Wang XS, Jajodia S (2001) Discovering calendar-based temporal association rules. In: 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
33. Li Y, Ning P, Wang XS, Jajodia S (2003) Discovering calendar-based temporal association rules, data & knowledge engineering, Volume 44, Issue 2, Pages 193-218, ISSN 0169-023X, https://doi.org/10.1016/S0169-023X(02)00135-0

34. 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
35. 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
36. Liu B, Hsu W, Ma Y (1999) Mining association rules with multiple minimum supports. In: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining (KDD '99). ACM, New York, pp 337–341. https://doi.org/10.1145/312129.312274
37. Ozden B, Ramaswamy S, Silberschatz A (1998) Cyclic association rules. In: Proceedings 14th International Conference on Data Engineering, pp 412–421. https://doi.org/10.1109/ICDE.1998.655804
38. Park JS, Yu PS, Chen M-S (1997) Mining association rules with adjustable accuracy. In: Proceedings of the sixth international conference on Information and knowledge management (CIKM '97). ACM, New York, 151–160. https://doi.org/10.1145/266714.266886
39. 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
40. 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
41. 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
42. 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 https://doi.org/10.1109/ICEMIS.2016.7745353
43. 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
44. 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, future generation computer systems, Available online 14 March 2017, ISSN 0167-739X, https://doi.org/10.1016/j.future.2017.03.016
45. Radhakrishna V, Kumar PV, Janaki V (2017) 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 http://www.iadisportal.org/ijcsis/papers/2017200104.pdf
46. Radhakrishna V, Kumar PV, Janaki V, Cheruvu A (2017) A dissimilarity measure for mining similar temporal association patterns. IADIS International Journal on Computer Science and Information Systems 12(1):126–142 http://www.iadisportal.org/ijcsis/papers/2017200109.pdf
47. Radhakrishna V, Kumar PV, Janaki V (2017) Normal distribution based similarity profiled temporal association pattern mining (N-SPAMINE). Database Systems Journal 7(3):22–33
48. Radhakrishna V, Kumar PV, Janaki V (2017) A Novel Similar Temporal System Call Pattern Mining for Efficient Intrusion Detection. J Univers Comput Sci 22(4):475–493 https://doi.org/10.3217/jucs-022-04-0475
49. Radhakrishna V, Kumar PV, Janaki V (2017) A computationally efficient approach for mining similar temporal patterns. In: Matoušek R (ed) Recent advances in soft computing. ICSC-MENDEL 2016. Advances in intelligent systems and computing, vol 576. Springer, Cham
50. Radhakrishna V, Kumar PV, Janaki V, Rajasekhar N (2017) Estimating prevalence bounds of temporal association patterns to discover temporally similar patterns. In: Matoušek R (ed) Recent advances in soft computing. ICSC-MENDEL 2016. Advances in intelligent systems and computing, vol 576. Springer, Cham. https://doi.org/10.1007/978-3-319-58088-3_20
51. Ramaswamy S, Mahajan S, Silberschatz A (1998) On the discovery of interesting patterns in association rules. In: Gupta A, Shmueli O, Widom J (eds) Proceedings of the 24rd International Conference on Very Large Data Bases (VLDB '98). Morgan Kaufmann Publishers Inc., San Francisco, pp 368–379
52. Srikant R, Agrawal R (1995) Mining generalized association rules. In: Proceedings of the 21th international conference on very large data bases (VLDB '95). Morgan Kaufmann Publishers Inc., San Francisco, pp 407–419
53. Srikant R, Agrawal R (1996) Mining quantitative association rules in large relational tables. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data (SIGMOD '96). ACM, New York, pp 1–12. https://doi.org/10.1145/233269.233311
54. Srikant R, Agrawal R (1997) Mining generalized association rules. Futur Gener Comput Syst 13(2):161–180. https://doi.org/10.1016/S0167-739X(97)00019-8

55. Tung AKH, Ng RT, Lakshmanan LVS, Han J (2001) Constraint-based clustering in large databases. In: Proceedings of the 8th international conference on database theory (ICDT '01). Springer, Verlag, 405–419
56. Radhakrishna V, Aljawarneh SA, Kumar PV, Choo KKR (2016) A novel fuzzy gaussian-based dissimilarity measure for discovering similarity temporal association patterns. Soft Comput: 1–17. https://doi.org/10.1007/s00500-016-2445-y
57. Villafane R, Hua KA, Tran D, Maulik B (1999) Mining interval time series. In: Mohania M, Tjoa AM (eds) DataWarehousing and Knowledge Discovery. DaWaK 1999. Lecture Notes in Computer Science, vol 1676. Springer, Berlin https://doi.org/10.1007/3-540-48298-9_34
58. 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. https://doi.org/10.1145/502512.502539
59. Yoo JS (2012) Temporal data mining: similarity-profiled association pattern. In: Holmes DE, Jain LC (eds) Data mining: foundations and intelligent paradigms. Intelligent systems reference library, vol 23. Springer, Berlin https://doi.org/10.1007/978-3-642-23166-7_3
60. Yoo JS, Shekhar S (2008) Mining Temporal Association Patterns under a Similarity Constraint. In: Ludäscher B, Mamoulis N (eds) Scientific and Statistical Database Management. SSDBM 2008. Lecture Notes in Computer Science, vol 5069. Springer, Berlin https://doi.org/10.1007/978-3-540-69497-7_26
61. 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
62. Zaki MJ (2000) Scalable algorithms for association mining. IEEE Trans Knowl Data Eng 12(3):372–390 https://doi.org/10.1109/69.846291
63. 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, p 326–335. https://doi.org/10.1145/956750.956788
64. 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 Computer Society Technical Committee on Data Engineering, IEEE Computer Society Technical Council on Software Engineering. He is awarded the prestigious ACM Senior Member award in 2016, (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 professor 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 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.

Puligadda Veereswara Kumar served as a distinguished Professor of Computer Science and Engineering at Osmania University. He earned his Ph.D. in Computer Science and Engineering in the area of temporal databases from Osmania University. He is currently 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. Several research scholars are working under his esteemed guidance. 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 several National and International conferences. He served as a Chairman, Board of studies, CSE 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. Research scholars from various reputed universities are pursuing research in his guidance and are also awarded successfully.

Vinjamuri Janaki received Ph.D from J.N.T. University Hyderabad, India and M.Tech from National Institute of Technology, Warangal, India. She is currently Head and Professor for the Department Computer Science and Engineering, Vaagdevi College of Engineering, Warangal, India. Her research interest includes Network security,

Data mining, Mobile Adhoc Networks and Artificial Intelligence. She has also 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.