https://doi.org/10.3329/dujase.v8i1.72994

Applying Sequential Correlation to Identify Order Dependent Activity Pairs

Md. Fahim Arefin, Maliha Tashfia Islam, Chowdhury Farhan Ahmed

Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh

**Email: farhan@du.ac.bd*

Received on 18 September 2023, Accepted for Publication on 25 January 2024

ABSTRACT

Mining sequential patterns has been one of the most useful fields in data mining. For example, one recent application of sequential pattern mining is analyzing logs of activity to predict and recognize activities. This is used for both daily life activities and virtual activities. However, only patterns are not always enough to represent the truly interesting information to the end user, especially when the support threshold is low and the number of frequent patterns is huge. A correlation measure is necessary to decipher the relationship between the logged activities. This data is generally collected as a sequence and there is no widely popular correlation measure for elements in sequential patterns. Therefore, we define Sequential Correlation, a novel correlation measure, for discovering important knowledge in sequential patterns and the corresponding full method, SCMine, to classify patterns based on the measure. We employed the measure to establish either a unidirectional or bidirectional association among the activities within a sequence and subsequently classified the sequences based on order dependency. Moreover, an efficient implementation approach for our measure is also discussed. Our performance study shows that, a significant number of activity patterns can be pruned when degree of order among the activities is important. So, it is also useful for classifying or pruning less significant activity patterns from a vast number of frequent sequential patterns.

Keywords: Sequential pattern, Sequential correlation, Data mining, Sequential dependency

1. Introduction

Data mining is a branch of science concerned with extracting information (potentially unknown or intriguing) from massive amounts of unstructured data or the repository [1]. These repositories include relational database, data warehouses, XML repository and such [2]. The main objective of this field of science is to pull out maximum beneficial information and store them in a structure that can be useful in the future [3].

There are several domains such as classification, clustering, regression, pattern mining, association rule mining[6] etc.[4]. Sequential pattern mining is a relatively new field of data mining but the classic data mining domain of frequent itemset mining is similar to sequential pattern mining. "The principal difference between the two lies in the fact that the order of items or data objects is inconsequential in the context of frequent itemset mining." In contrast, sequential pattern mining concerns data sequences in which items are sorted. Sequential pattern mining algorithms are frequently used to discover patterns that can then be used to build recommendation systems and text predictions, increase system usability, and make informed product selection decisions.

With the advent of computerized systems everywhere, terabytes of data are generated every day. Often it is the case that sequential patterns are not enough to depict the type of dependency or association that rests within the data items or objects. The biggest problems being:

- In instances where the minimum support threshold is set at a high level, the patterns extracted tend to reveal the most evident or intuitively expected 'knowledge.
- Conversely, when employing a low support threshold, a substantial volume of patterns typically arises, many of which prove to be redundant, lacking in informativeness, or constituted by arbitrary combinations of prevalent data objects.

Correlation analysis is a useful technique in these situations. Correlation analysis is the process of determining or quantifying the strength of a link between items, itemsets, or data objects.

In the field of activity recognition and monitoring, discovering order dependencies between different activities help in identifying daily routines and habits of individuals. Understanding these patterns can be crucial for applications in health monitoring, personalized services, and even in marketing.

The lack of widely used or accepted correlation measurements for sequential patterns is the primary driving force for our research. In the field of activity pattern recognition, by using our proposed measure, it can be indicated if two activities are strongly correlated in terms of order. In other words, we can specify if the activities have a dominant order by which they appear in a sequence. Let's consider an example with two common daily activities: "Morning Exercise" and "Breakfast."

Suppose in a dataset of daily activity sequences, we find that "Morning Exercise" is followed by "Breakfast" in 80% of the instances, while "Breakfast" is followed by "Morning Exercise" in only 20% of the cases. Both activities occur frequently in the dataset, indicating their importance in daily routines. At first glance, it might seem that these activities do not maintain a specific order. However, a deeper analysis reveals a strong sequential relationship.

When we set a lower support threshold, both sequences ("Morning Exercise" followed by "Breakfast" and vice versa) will appear in the analysis. But examining the relationship between these activities shows that there's a high likelihood (80%) that "Morning Exercise" will be followed by "Breakfast." This implies that "Breakfast" is typically consumed after "Morning Exercise," but it's less common for individuals to engage in "Morning Exercise" after having "Breakfast.

We must keep track of the sequence in which the activity patterns appeared during mining because we are dealing with sequential patterns. In real-life applications, this can be utilized to predict and recognize different activities, provide personalized fitness advice, identify behavioral patterns. If the sequence is not a critical factor, the activities can be executed in any format without concern for the order. Our objective is to find a correlation measure which measures the strength of order dependency between activities along with the direction of order between them, while considering performance advantages and the overhead for correlation analysis.

This article is an extended version of our previous work [5]. In that work, we had experimented on a small dataset on Daily Activity Log [20] to understand the applications of SCMine for Activity Log correlations. As our measure was able to isolate the inherent dependency between activities in that dataset, in this extended article, we provide real-life applications on activity log analysis, an in-depth discussion with examples of our proposed methodologies along with the necessary concepts, and a set of new extensive experimental discussions on other related datasets to understand the solutions' merits for activity log analysis.

With this work, we have addressed two important domains of data mining such as sequential pattern mining and correlation analysis. Our contributions are as follows:

- Use SCMine algorithm to indicate the strength of relationship among activities in a sequence based on order.
- Subjective evaluation of the SCMine algorithm on different activity log datasets.
- In depth testing on real life physical and virtual activity log datasets to solidify the importance of our proposed measure.

Section 2 contains the background study and existing works related to our domain. Section 3 consists of our proposed solutions to the problems. Section 4 gives a comparative analysis between our solutions and existing solutions, and conclusions are drawn in Section 5.

2. Background and Related Works

Multiple classifier systems for human activity detection were compared in [27]. Subsequently, recent advancements on Human activity recognition in artificial intelligence framework were discussed in [19]. The analysis of activity patterns could serve as a valuable, non-intrusive indicator for identifying signs of depression [16]. Similarly, the analysis of sedentary behavior and physical activity patterns in Chronic Obstructive Pulmonary Disease patients, revealed a strong correlation between high sedentary time and lower physical activity and exercise tolerance [17].

On the other hand, the importance of using sequence alignment in time-use diaries to efficiently analyze daily activity patterns within a set timeline was discussed in [15] and how sequential patterns can be used to analyze weblogs was discussed in [28]. Adedeji et al. discussed how analyzing weblog could help us extract user behavior pattern in [29] and Clustering-Based Pattern Mining (CBPM) uncovers relevant patterns by investigating correlations through clustering methodologies [10]. Inspired from these, we believe our sequential correlation measure can provide the necessary reinforcement that support-confidence framework needs. We intend to apply our measure to identify the sequential dependencies between different activities.

Sequence and Sequential Database: A sequence is a

collection or list of objects with a certain order[8]. A sequence database contains a list of transactions or sequences with ordered itemsets or events. A sequence S is written as <e_1 e_2 e_3 ... e_l >, where the event e_1 happens before event e_2, e_3 and so on. For example, <a(bc)d> is a sequence containing 3 events or itemsets. The brackets signify items which are contained within a single event. The sequence database is denoted as D.

Sequential pattern mining was first cited by Agrawal and Srikant in [12]. equential pattern mining aims to produce all subsequences whose frequency or count in the entire set is equal to or greater than min sup, given a collection of sequences made up of elements and a minimum support threshold, min sup.

PrefixSpan[13] is one of the leading and most popular algorithm for frequent sequence mining. It is an extension of Fp-growth algorithm[7] and uses some concepts from Freespan[14].

In our work, we will first use PrefixSpan to mine all the sequential patterns for a given threshold. Then we will use our SCMine algorithm to calculate the sequential correlation score for these patterns and group the order dependent patterns. This would help us uncover correlation between sequential activity patterns for both physical and virtual activities.

3. Proposed Approaches

In this section, we discuss our proposed strategies. In Section 3.1, we explain the terminologies required to explain our concept. Section 3.2 presents our algorithm. Section 3.3 contains a demonstration of the algorithm's use and Section 3.4 contains some possible applications of the algorithm in the field of activity dependency mining.

In this section, we delve into our proposed measure, Sequential Correlation, and provide an in-depth exploration of the entire process. In summary, throughout this chapter, we have accomplished the following:

1. We introduced our novel measure, Sequential Correlation, which plays a key role in assessing order dependency within sequential patterns.
2. We have presented a comprehensive algorithm called SCMine for pattern categorization. We included its pseudocode to facilitate a clear understanding of its functioning.
3. We have explained additional relevant terminology and concepts that are pertinent to the understanding and application of our measure and algorithm.
4. To enhance comprehension, we've offered a practical example that illustrates how Sequential Correlation and the SCMine algorithm can be applied in real-world scenarios.

Together, these components provide a holistic view of our contributions in the realm of sequential pattern analysis and order dependency assessment.

3.1 Terminologies

3.1.1 Itemset and Sequence: An itemset is denoted as (x_1, x_2, ..., x_k), where x_k is an item. A sequence is an ordered list of itemsets. A sequence S is denoted by s_1 s_2 .... s_l , where s_j is an itemset or an element of the sequence.

3.1.2 Order Dependent Sequences: A sequence in which the elements consistently adhere to a prevailing order of occurrence within the sequential database is formally termed an "order-dependent sequence." In practical terms, if we consider two items or itemsets, A and B, within a sequential dataset, the sequence AB will be classified as order-dependent if the event eA, encompassing A, frequently transpires before the event eB, which encompasses B. In other words, in most instances where both A and B are present, eA tends to precede eB, signifying an inherent order dependency between the two elements in the sequence.

3.1.3 Order Independent Sequences: An order-independent sequence is a sequence whose elements can appear in any order inside the sequential database. For instance, if there is no dominating order in the events eA, which includes A, and eB, which contains B, the sequence AB will be order independent. In other words, the frequency of eA < eB and eA > eB in the dataset is comparable.

3.1.4 Sequential Correlation: The key idea of our measurement, Sequential Correlation, is to calculate the ratio of an observed pattern to its inverse in terms of overall occurrence. Our metric, Sequential Correlation, calculates the ratio between an observed pattern and its inverse about all instances of both patterns combined.

Let F(A,B) be a function that calculates the frequency or overall number of sequences in the dataset where A appeared before B in the order of appearance. A and B in this case, might be either one item or a whole itemset. A series is shown by A, B. So the Sequential Correlation becomes:

$$SequentialCorrelation, SC(A,B) = \frac{F\,(A,B)\ -\ F\,(B,A)}{F\,(A,B)\ +\ F\,(B,A)}$$

Given that A and B are common items or itemsets, the Sequential Correlation score will always fall within [-1,1]. As depicted in Figure 1, Items with a Sequential Correlation, SC of 0 are independent, whereas those with a SC of 1 have a very high order dependency. We represented the strength of order dependency using a gradient of green in Figure 1. A value of -1 would mean the opposite order is the prevalent one. As our measure disregards null transactions, the score is also null-invariant.

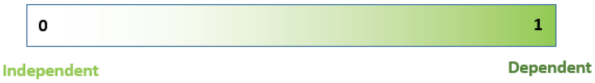


Fig.1. Normalized Sequential Correlation Range

Our measure cannot be equivalent to a certain support threshold because of the following reasons:

- Support represents the frequency of a sequence in terms of total data tuples or transactions. But total transactions have no effect on our score.
- Assuming the total number of transactions are constant, support has only one independent variable - the frequency of a pattern. Our measure has two such variables.
- Crossing a certain minimum support threshold does not guarantee that it will cross our correlation threshold.

3.1.5 Sequential Correlation Threshold

Sequential Correlation Threshold denotes the tolerance or benchmark level for order dependency. Setting a high SC.Threshold implies that only patterns displaying strong order dependency are deemed interesting, while a low SC.Threshold allows for patterns or sequences with more flexibility in the ordering of items to be considered

3.2 Proposed Algorithm

In this section we define our problem, explain the step by step procedure to calculate our measure and demonstrate an example for easy understanding.

3.2.1 Problem Statement

In the context of a sequential database of activities and a collection of frequent sequences determined by a user-specified support threshold, the task at hand involves identifying subsets of sequences characterized as order-dependent and order-independent. This classification is achieved through the application of the Sequential Correlation measure, with the SC.Threshold being a parameter defined by the user.

3.2.2 Step By Step Procedure

The SCMine algorithm's workflow requires a prior frequent pattern mining step. It operates under the assumption that an appropriate sequential pattern mining algorithm has been utilized to generate a list of frequent sequential patterns, each accompanied by its corresponding frequency count. Subsequently, SCMine proceeds through the following key steps:

A) Building Trie From Frequent Patterns:

In this stage, a trie data structure is constructed. Each frequent activity pattern, together with its associated frequency or count, is inserted into the trie. By the end of this step, the trie encapsulates the entire collection of frequent activity patterns.

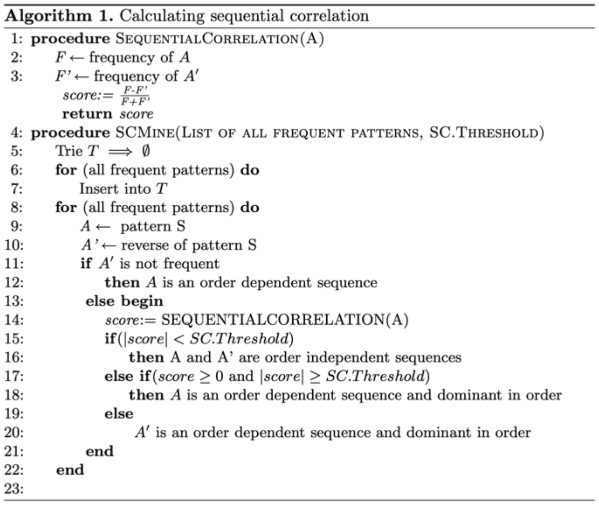
B) Calculating Sequential Correlation Score:

At this juncture, the algorithm computes the Sequential Correlation score for each frequent activity pattern using Equation (1). This calculation involves looking up the frequency of the reverse pattern from the trie as well.

C) Categorizing Patterns:

Following the computation of Sequential Correlation scores for each activity pattern, they can be categorized into either order-dependent or order-independent sequences, contingent upon the SC. Threshold value. If a activity pattern's score surpasses this threshold, it is classified as order-dependent.

These sequential steps within the SCMine algorithm work cohesively to analyze and categorize frequent sequential patterns based on their level of order dependency, making it a valuable tool for pattern analysis and understanding in sequential datasets.



```
Algorithm 1. Calculating sequential correlation
procedure SEQUENTIALCORRELATION(A)
    $F \leftarrow$ frequency of $A$
    $F' \leftarrow$ frequency of $A'$
     $score := \frac{F-F'}{F+F'}$
     return $score$
procedure SCMINE(LIST OF ALL FREQUENT PATTERNS, SC.THRESHOLD)
    Trie $T \implies \emptyset$
    for (all frequent patterns) do
        Insert into $T$
    for (all frequent patterns) do
        $A \leftarrow$ pattern S
        $A' \leftarrow$ reverse of pattern S
        if $A'$ is not frequent
            then $A$ is an order dependent sequence
         else begin
            $score :=$ SEQUENTIALCORRELATION(A)
            if($|score| < SC.Threshold$)
                then A and A' are order independent sequences
            else if($score \geq 0$ and $|score| \geq SC.Threshold$)
                then $A$ is an order dependent sequence and dominant in order
            else
                 $A'$ is an order dependent sequence and dominant in order
         end
    end
```

Algorithm 1: Calculating Sequential Correlation

The algorithm consists of two main procedures: `Sequential Correlation` and `SCMine`. The `Sequential Correlation` procedure takes a pattern `A` and calculates its sequential correlation by determining the frequency of `A` and its reverse `A'`. It computes a score by subtracting the frequency of `A'` from the frequency of `A` and then dividing the result by the sum of both frequencies. This score is then returned as the output of the procedure.

The second procedure, `SCMine`, operates on a list of all frequent patterns and a threshold value for sequential correlation. It initializes an empty Trie `T` and inserts all frequent patterns into this Trie. The procedure then iterates over all frequent patterns. If `A'` is not frequent, then `A` is considered an order dependent sequence. If `A'` is frequent, the procedure calculates the sequential correlation score for `A` and evaluates it against the threshold. If the absolute value of the score is less than the threshold, then `A` and `A'` are deemed order independent sequences. If the score is greater than or equal to zero and its absolute value is greater than or equal to the threshold, then `A` is an order dependent sequence and the dominant order. Conversely, if the score is less than zero and its absolute value is greater than or equal to the threshold, then `A'` is the order dependent sequence and the dominant order. The procedure ends after evaluating all patterns.

The algorithm, therefore, systematically evaluates and categorizes the frequent patterns based on their Sequential Correlation scores, ultimately leading to the partitioning of the dataset for further analysis.

3.3 Demonstration

We will use a minimized dataset containing 10 distinct activities for demonstration purposes:

Code	**Item**
C	Cleaning
D	Desk Work
E	Eating
K	Coffee Break
L	Listening to Music
M	Meditating
P	Play with Kid
R	Relaxing
S	Socializing
U	Unwinding

For the activities given above, the sequences are:

Sequences

L M S E R
D U S L E
L M S R E
L E S P M
L M S P E
L E C S M P
S E R L
L S E
S L E R P
S E M R K L

Single-item Sequences:

- L: 10, S: 10, E: 10, R: 5, M: 5, P: 4, D: 1, K: 1, C: 1, U: 1

If we apply a 20% minimum support threshold, we get 6 activities: L, S, E, R, M, P

Now, let's calculate sequences prefixed with "S":

Set of Suffixes with Prefix S in the Dataset:

[ER, LE, RE, PM, PE, MP, ERL, E, LERP, EMRKL]

From this set, we calculate the frequent activities in the projected database:

E: 8, R: 5, L: 4, P: 4, M: 3

Set of Suffixes with Prefix SE in the Dataset: [R, RL, RP, MRKL]

From this set, we calculate the frequent activities: R: 4, L: 2

Set of Suffixes with Prefix SER in the Dataset: [L, P, KL]
From this set, we calculate the frequent activities: L: 2

Set of Suffixes with Prefix SERL in the Dataset: This set is empty, indicating that this pattern will not get any longer.

This process is applied recursively to generate projected databases for all other frequent prefixes, including L, E, R, M, and P. This approach allows us to uncover frequent sequential patterns while considering the minimum support threshold.

Table 1: Sequential patterns with prefix S

Prefix	Projected (suffix) database	Sequential pattern
< S >	< ER > , < LE > , < RE > , < PM > , < PE > , < MP > , < ERL > , < E > , < LERP > , < EMRKL >	< S > , < SL > , < SE >, < SR > , < SM > , < SK > , < SP >
< SL >	< E > , < ERP >	< SLE >
< SE >	< R > , < RL > , < RP > , < MRKL >	< SEL > , < SER > , < SERL >
< SR >	< E > , < L > , < P > , < KL >	< SRL >
< SM >	< P > , < RKL >	-
< SK >	< L >	-
< SP >	< M > , < E >	-

We get 31 sequences with 2 activities, 21 sequences with 3 activities, and two sequences with 4 activities, all of these are highlighted in Table 1.

Let's compute Sequential Correlation for different item pairs:

For E-S (E followed by S): F(E-S) = 2, F(S-E) = 8.

Sequential Correlation, SC(E-S) = (2 - 8) / (2 + 8) = -0.6. Since the SC is negative, the dominant sequence is the reverse one (S followed by E). As the value is higher than 0.5, these two activities exhibit moderate dependency, with the order of dependency being S => E. Hence, S-E is an order-dependent sequence.

Similarly, for L-R (L followed by R): F(L-R) = 2, F(R-L) = 2.

Sequential Correlation, SC(L-R) = (2 - 2) / (2 + 2) = 0.

As the SC is 0, these two activities are completely independent, indicating no specific order. Therefore, L-R and R-L are order-independent sequences or patterns.

For M-R (M followed by R): F(M-R) = 2, F(R-M) = 0.

Sequential Correlation, SC(M-R) = (2 - 0) / (2 + 0) = 1.

With an SC of 1, these two activities exhibit strong dependency, with the order of dependency being M => R. M-R is an order-dependent pattern and the dominant sequence.

Based on the findings from Table 2, several noteworthy observations can be made. For instance, Listening to Music -Eating, Listening to Music -Meditating, and Socializing-Eating demonstrate high sequential correlation scores, indicating significant sequential dependency. These sequences are considered order-dependent.

Table 2: Sequential Correlations for 2 activity-sequences

Activity Sequence (A,B)	Fre (A,B)	Freq(B,A)	Total Freq.	SC(A-B)
Listening to Music - Socializing	6	4	10	0.2
Listening to Music - Eating	8	2	10	0.6
Listening to Music - Relaxing	2	2	4	0
Listening to Music - Meditating	4	0	4	1
Listening to Music - Play with Kid	4	0	4	1
Socializing - Eating	8	2	10	0.6
Socializing - Relaxing	5	0	5	1
Socializing - Meditating	3	2	5	0.2
Socializing - Play with Kid	4	0	4	1
Eating - Relaxing	4	0	4	1
Eating - Meditating	3	2	5	0.2
Eating - Play with Kid	3	0	3	1
Relaxing - Meditating	0	2	2	-1
Meditating - Play with Kid	2	0	2	1

Conversely, the relatively low sequential score observed for "Socializing" and "Meditating" implies that they lack a well-established order dependency. This suggests that the sequence "Socializing" followed by "Meditating" is an example of an order-independent sequence. The overall table thus mirrors a situation commonly encountered in real-life scenarios, where certain activity combinations exhibit a loose or non-deterministic relationship in terms of their order of occurrence.

Similarly, we can calculate Sequential Correlation for sequences with multi-item activity itemsets. Based on the Sequential Correlation, we can classify the sequences based on their Sequential Correlation score. If only the order-dependent patterns are deemed interesting to the end user, then the set of order-independent patterns can be pruned or removed.

As an illustration, by setting the SC.Threshold at 0.6, we can categorize two-length patterns into two distinct classes:

Order Independent Patterns: These patterns exhibit a SequentialCorrelation (SC) value less than 0.6.

Order Dependent Patterns: These patterns, on the other hand, demonstrate a SequentialCorrelation (SC) value greater than or equal to 0.6.

This classification helps discern the degree of order dependency within these patterns, providing valuable insights into their organization and significance.

Table 3: Order Dependent Patterns in Demo Dataset

Sequence(A,B)	SC(A,B)
Listening to Music - Eating	0.6
Listening to Music - Meditating	1
Listening to Music - Play with Kid	1
Socializing - Eating	0.6
Socializing - Relaxing	1
Socializing - Play with Kid	1
Eating - Relaxing	1
Eating - Play with Kid	1
Meditating - Relaxing	1
Meditating - Play with Kid	1

Table 4: Order Independent patterns in Demo Dataset

Sequence(A,B)	SC(A,B)
Listening to Music - Relaxing	0
Listening to Music - Socializing	0.2
Socializing - Meditating	0.2
Eating - Meditating	0.2

3.4 Application of Correlation between Activities

As our primary objective is to utilize this correlation to mine correlations from activity logs, in this section, we explain some possible applications of this correlation and how it can be useful for real life applications.

*3.4.1 Wellness Monitoring and Fitness Coachin*g

Discovering order dependencies helps in identifying daily routines and habits of individuals. Understanding these patterns can be crucial for applications in health monitoring, personalized services, and even in marketing. It can also reveal insights into a person's lifestyle, helping in areas like health advice, fitness coaching, and wellness monitoring.

3.4.2 Early Detection of Health Issues

Changes in the regular sequence of daily activities can indicate health issues. For example, a decrease in physical activity might suggest health decline. Similarly, irregularities or significant changes in daily routines can be indicators of mental health issues like depression or anxiety.

3.4.3. Personalization and Recommendation Systems

Context-aware Services: By understanding the typical order of activities, technology can offer more timely and relevant context aware services, such as suggesting a meal after a workout. This can also be used for automating home environments based on predictable sequences of activities (e.g., turning on the coffee machine after a morning workout).

3.4.4. Predictive Modeling

With established order dependencies, predictive models can forecast upcoming activities, enabling proactive decisions. Similarly, it can also help identify deviations from the usual sequence of activities can trigger alerts for unusual behavior or emergencies.

3.4.5. Time Management and Productivity

Identifying inefficient sequences of activities can lead to recommendations for better time management and help in balancing work-related and leisure activities.

3.4.6. Academic and Occupational Research

Understanding the sequence of daily activities contributes to research in areas like psychology, sociology, and human-computer interaction. It can also help analyze activity sequences in a work context can help in identifying stress patterns and improving workplace ergonomics.

4. Experimental Results

To contrast our strategy with the existing strategies, we employed a number of data sets from the UCI Machine Learning Repository [22]. We present the outcomes of four data sets because they are all comparable in terms of their findings. To create a string of characters, the datasets were discretized. The complexity to discretize the dataset was O(N) where N denotes the total number of elements in the dataset.

This section demonstrates how well our suggested approach, SCMine, performed overall across various real-life data sets related to Activity Mining. SCMine mines sequential patterns and categorizes them as order-dependent or order-independent. In the first half of this section, we show the connections between the data items discovered using sequential correlation in several datasets with various minimal support levels.

To underscore the significance of Sequential Correlation, we constructed a real-world dataset. This dataset was compiled by recording the purchase histories of 'tech products' from 30 users through a designated website [25]. For this purpose, we designed a website offering a selection of 16 items, and we asked users to sequentially indicate the products they had acquired over the course of the past four years. We meticulously logged each sequence of items, with each sequence serving as a representation of an individual customer's purchase history. It is worth noting that, for the sake of simplicity, we employed individual items as the fundamental building blocks of these sequences, foregoing the use of item sets.

After validating our theory in this dataset, we experimented with other large real-life datasets available publicly. According to the best of our knowledge, none of the currently available algorithms measures this correlation, so we compare it to the most widely used Sequential pattern mining algorithm – Prefix Span. We analyze the time overhead and memory overhead of running SCMine in comparison to Prefix Span at the end of the chapter.

4.1 Environment Setup for Experiment

We tested on a machine with a 1.70 GHz Intel Core i5 CPU and 6 GB of RAM running the Windows 10 operating system. The UCI Machine Learning Repository provided all the real-world datasets we utilized for this experiment. However, some of them needed to be preprocessed to meet our criteria for ordered sequence. We used real-life datasets only because they help us realize whether our Sequential Correlation is actually effective or not. We compare the efficiency of our algorithm SCMine with Prefix Span, both of which were Java implementations. The Prefix Span [2] algorithm implementation from spmf [23] was used directly for comparison, and the SCMine algorithm was built on top of it.

We don't consider the single-itemset sequences for performance measures because we need at least two itemsets to consider the correlation between them. To better highlight the benefits of SCMine and the value of Sequential Correlation, we additionally attempt to normalize the comparisons.

4.2 Performance Metrics

Various performance metrics have been taken into account to assess the effectiveness of SCMine. The performance metrics used for the comparison of our algorithm with Prefix Span are shortly described below:

A. Number of Order-dependent Patterns: Number of patterns varies with varying minimum support threshold value. With the increase of min_sup value, the number of patterns tends to decrease. Suitable min_sup threshold ranges are used in each database to observe the performance over the number of patterns more clearly.

B**. Runtime Analysis:** With varying minimum support threshold value, runtime will also vary. With the decrease of min_sup, more patterns will be generated, and it will take more time to run for generating more patterns. Suitable ranges for min_sup are applied for each dataset, and runtimes are shown in seconds.

C. Memory Consumption: To measure the performance over memory usage, we have used the maximum memory used by both Prefix Span and SCMine and then compared them to each other. While running any of the algorithms, we keep the value of maximum memory usage and keep on updating it if more than that value is used at any instance. Usually, memory consumption depends on the number of patterns generated.

4.3 Performance Analysis

In this section, we analyze the performance of our measure across different datasets in terms of the three metrics mentioned above. We used two datasets on "Daily Activity" to understand the results in the context of daily human activity and then we used two datasets on "Online Click Log Activity" to understand our measure's usefulness in virtual activity monitoring.

4.3.1 Number of Order-dependent Patterns

Only obvious patterns appear when the minimum support threshold is high, but there are too many patterns when it is low. With the help of our Sequential Correlation, we can divide the patterns into those with a high order dependency and those without. The proportion of patterns that are categorized as Order dependent patterns increase when the SC. Threshold is lowered. These are demonstrated by the outcomes displayed below from several real-life datasets.

Dataset 1: Body Sensor Data

Seven individuals with similar physical attributes collected accelerometer, device orientation, and GPS sensor data and labeled this data simultaneously. The data was collected using a smart-phone (sensor recording) and smart-watch (manually labeling), as outlined in reference [26]. As there were 47 unique activities, this data is helpful to visualize the dependency between different activities. When we generated a sequential database by processing this data, we had 1439 activities of 47 types. We created 74 sequences by compiling all activities done in sequence in a particular day by an individual. The average number of activities per sequence is 19.43. This indicates that on average, each subject performed around 19 activities per day. The average frequency of an activity across the entire dataset is approximately 30.60. This means that, on average, each unique activity occurs about 31 times in the dataset. For an easy visualization of the type of tasks we used, we have added a histogram to plot the top 20 tasks in terms of frequency. Some of the most interesting frequent sequences were grooming before socializing and relaxing before meal preparation. In almost all occurrences of these pairs, they followed this sequence, which meant they have sequential dependency.

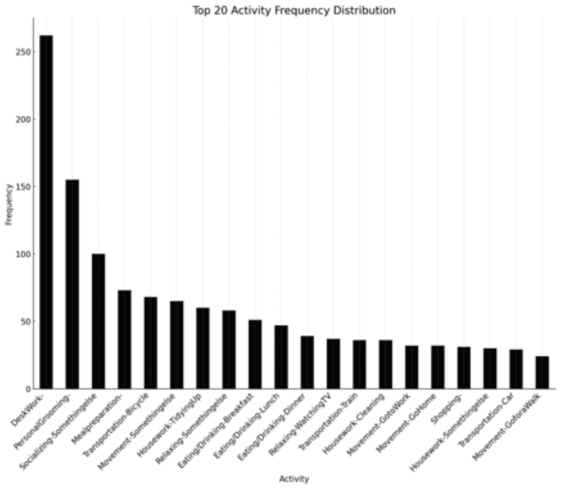


Fig. 2. Order Dependent Patterns in Body Sensor Dataset

Dataset 2: Activities of Daily Living (ADL)

This dataset is also comprised of the Activities of Daily Living performed by users daily in their own homes [20]. We used the daily activity logs of each day to generate one sequence.

This dataset was helpful in validating the usefulness of our correlation measure in real life, like the previous one. We could anticipate the relationship between the objects because the items were actual daily life events like showering, grooming, and sleeping. As anticipated, most sequences within this dataset exhibited order-dependent characteristics, with only a limited number displaying order-independent attributes. Some people, for example, shower before grooming, while others groom first. Almost no one watches TV shortly after waking up from a nap, but most people do so before bed.

In our experiments on the dataset, we systematically adjusted the minimum support thresholds and varied the SC. Threshold parameter to effectively illustrate the impact of SC. Threshold. As depicted in Figure 3, even when SC. Threshold was set to 0.5, nearly 90% of the sequences exhibited order-dependent characteristics.

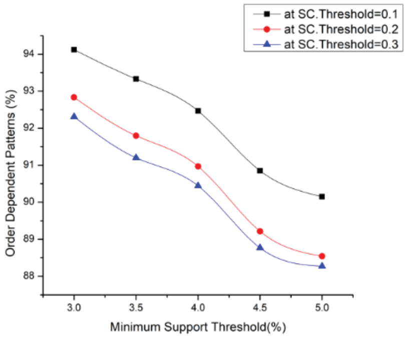


Fig. 3. Order Dependent Patterns in ADL Dataset

Dataset 3: MSNBC

To ensure the validity of our work in online activity, we applied our algorithm on weblog data. The data was extracted from the logs of msnbc.com and the news-related segments of msn.com, specifically on September 28, 1999, as documented in reference [22]. Each sequence within the dataset represents the sequence in which a user visited various categories on that particular day. While the original dataset encompassed 989,818 sequences, we opted to retain only the 31,790 sequences by removing the shortest ones. This dataset is relevant because this is a large dataset and allows us to analyze the browsing activity pattern.

With SC. Threshold established at 0.1, it was noted that over 40% of the patterns were categorized as order-dependent across various minimum support thresholds. However, this proportion declined to less than 30% when SC. Threshold was set at 0.3 or higher, as illustrated in Figure 4.

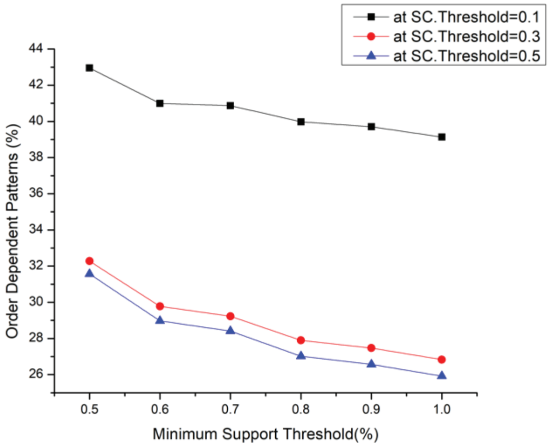


Fig. 4. Order Dependent Patterns in MSNBC Dataset

This dataset was used to analyze the usefulness of our Sequential Correlation in web log analysis and to measure the order dependency between different categories. At different minimum SC. Thresholds, we could see the majority of the patterns were order independent, which is normal for web browsing scenarios. The patterns which were order dependent were mostly closely related topics.

Similar findings were observed in the Fifa dataset which comprises clickstream data, as detailed in [9].

Dataset 4: UK Retail

This is a transactional dataset containing all the transactions occurring between December 1, 2010, and December 09, 2011, for a UK-based online retail store [21]. The company mainly sold unique all-occasion gifts. We pre-processed this dataset to create sequences where each sequence represented the purchase history of a customer, and each item represented a real item from the shop. We had 4069 distinct items

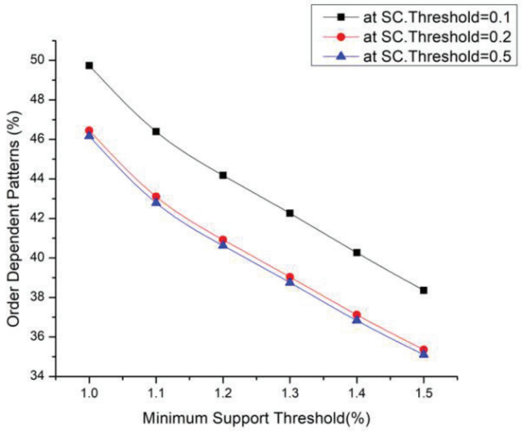


Fig. 5. Order Dependent Patterns in UKRetail Dataset

The advantage of using this dataset was to verify the universality of our dataset across different domains – as like activity log data, this dataset doesn’t have many order dependent pairs. We could observe what the correlations actually meant, and as expected in a retail scenario, few items had strong order dependency while most items didn't follow any particular order. At a 1% minimum support threshold, 102,927 frequent patterns were generated, where 101,359 patterns contained more than one item. When the SC. Threshold was set to 0.1; 50,419 of these patterns were found to be order-dependent, while the rest were order-independent.

4.3.2 Runtime Analysis

To enhance performance, we employed a trie-based approach for computing Sequential Correlation. In this method, the frequency of each shared sequential pattern was stored within a trie data structure. Notably, the computation of Sequential Correlation can be accomplished in constant time, denoted as O(1), while the time required for looking up the frequency of an N-length sequence is proportional to the length of the sequence itself, denoted as O(N).

Since we perform Sequential Correlation mining after extracting frequent patterns using Prefix Span, we can gauge the overhead of our algorithm by comparing our runtimes with those obtained when running only Prefix Span. We present the runtime escalation across various datasets and minimum support thresholds since both of these factors influence the runtime.

To measure the time complexity of SCMine, we experimented using all the aforementioned datasets, which yielded similar results, so we present the results found in MSNBC, as this is the largest datasets. These observations are reflected in Figure 6.

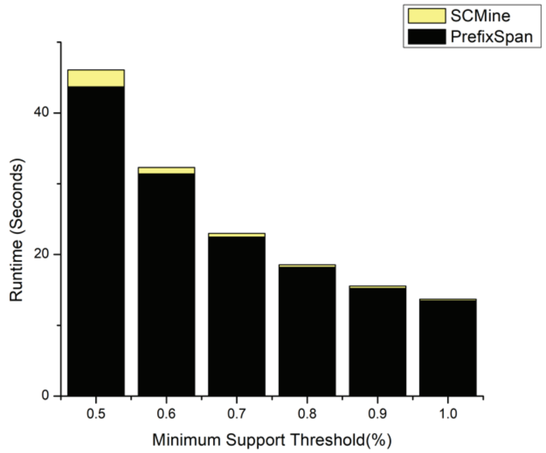


Fig. 6. Runtime comparison in MSNBC Dataset

4.3.3 Memory Analysis

As SCMine algorithm is applied after patterns are mined using Prefix Span, we measure the memory used when only Prefix Span is run and memory consumed when SCMine is run after Prefix Span to understand the memory overhead caused by SCMine. SCMine's memory overhead is mainly for the Trie structure, which stores the frequency of the common patterns. Consequently, as the count of common patterns expands, the corresponding memory demand also increases, leading to higher memory overhead as the minimum support threshold decreases. The extent of this memory overhead is contingent upon two key variables: the quantity of frequent patterns generated and the average length of those frequent patterns. Consequently, the specific memory overhead will fluctuate contingent on the dataset's inherent attributes and the chosen minimum support threshold.

Hence, we did a comprehensive comparison of results across datasets of varying characteristics, all assessed at different minimum support thresholds. Furthermore, we delve into an examination of the connection between the quantity of frequent patterns and the associated memory overhead across distinct datasets, further substantiating that our algorithm operates effectively with minimal memory overhead in all scenarios.

We ran our SCMine algorithm on the MSNBC dataset as it is a comparatively sparse dataset with only 4,118 frequent patterns at a 3% minimum support threshold. So we experimented at very low support thresholds because the memory overhead will not be significant unless a huge number of patterns are generated.

At a 0.6% minimum support threshold, PrefixSpan generates 1,70,819 frequent patterns and in this case, there is around 1.93% memory overhead caused by SCMine. When we increase the support threshold, the number of patterns decreases, and consequently the memory overhead decreases. The memory overhead is quite insignificant at a 0.9% minimum support threshold, as then the overhead is around 0.13%. Figure 7 represents this analysis.

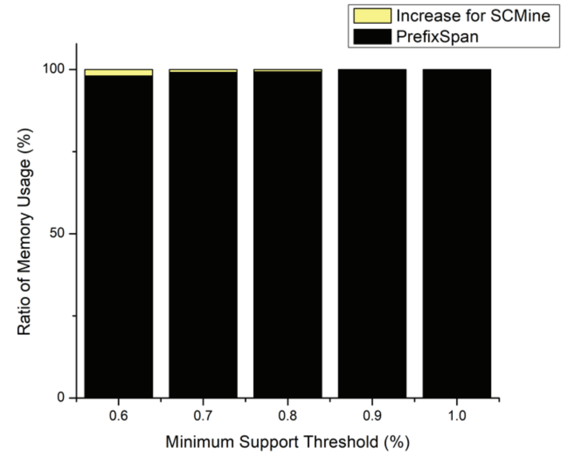


Fig. 7. Increase in Memory usage in MSNBC Dataset

Memory Usage Comparison across Different Datasets

As the ADL datasets are comparatively small, it is hard to extract any performance metric from them. Therefore, we compare the memory overhead in the datasets MSNBC, UKRetail, and Fifa to better understand how the nature of the dataset affects the memory overhead generated by SCMine, as depicted in Figure 8. We evaluate the correlation between the quantity of patterns produced and memory use across various datasets. Because longer patterns take up more space, their length causes this variation. The number of separate entries was an additional factor because the wider the tree, the fewer common prefixes there will be.

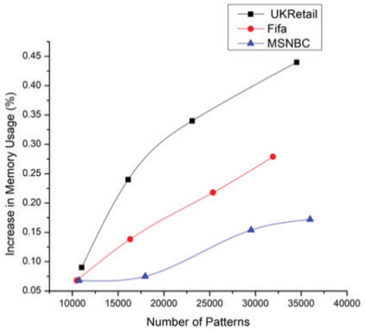


Fig.8. Memory Usage Comparison across Different Datasets

5. Conclusion

The question of how to discover truly useful patterns among the plethora of useless information has attracted substantial attention from researchers. We focused on combining the domains of Sequential pattern mining and correlation analysis for the field of Activity Log in our work. The research work as presented in this article is summarized in this section.

In this study, we have first tried to explore the preliminary literature that is required for a proper understanding of the field we had aimed to work on. With the help of numerous datasets and performance measures, we examined the effectiveness and workability of our approach for mining order dependency between activities. Below is a summary of our contributions:

- Our measure, Sequential Correlation, scrutinizes sequential patterns by evaluating their dependency on the order of activities.
- Large datasets from public online repositories related to physical and virtual activities were used to test performance, scalability, and efficiency.
- According to the performance analysis, our approach is deemed satisfactory as it effectively reduces the multitude of frequent patterns to a more manageable set of intriguing patterns, as determined by the selected SC. Threshold.
- Overhead in terms of time is negligible concerning the time consumed by the pattern mining step. Memory consumption remains comfortably within an acceptable threshold, primarily due to the utilization of a trie, a prefix-based data structure, in our approach.
- Our measure Sequential Correlation can be used for various real-life applications such as finding commercially useful information or studying personnel behavior, along with many other applications in the field of activity recognition etc.

Nevertheless, there is room for further refinement in pursuit of optimizing performance. Optimization techniques can be applied to reduce the time overhead as the number of patterns grows and Correlation among multi-activity combinations can be integrated into our score. The method holds potential for expanding its analysis to encompass the surrounding context of activities, including factors like the involvement of others or the specific setting. However, incorporating these aspects would necessitate the acquisition of additional metadata. Additionally, this methodology encourages the development of comprehensive theories addressing the overall determinants of daily behavior, rather than focusing solely on individual activities.

References

1. ICDM ’02: Proceedings of the 2002 IEEE International Conference on Data Mining, Washington, DC, USA, 2002. IEEE Computer Society.
2. ICDM ’04: Proceedings of the 4th IEEE International Conference on Data Mining, Brighton, UK, 2004. IEEE Computer Society.
3. M. H. Dunham, "Data Mining: Introductory and Advanced Topics," Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002.
4. U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Advances in knowledge discovery and data mining," pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.
5. Md. F. Arefin, M. T. Islam, C. F. Ahmed, (2018). "Mining Sequential Correlation with a New Measure." In: Perner, P. (eds) Advances in Data Mining. Applications and Theoretical Aspects. ICDM 2018. Lecture Notes in Computer Science, vol 10933. Springer, Cham.

6. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases." In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

7. J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation." SIGMOD Rec., 29(2):1–12, May 2000.

8. J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

9. V. Jaiswal and J. Agarwal, "The evolution of the association rules," pages 726–729, 2012.

10. Y. Djenouri, J. C.-W. Lin, K. Nørvåg, H. Ramampiaro, and P. S. Yu, "Exploring Decomposition for Solving Pattern Mining Problems." ACM Trans. Manage. Inf. Syst. 12, 2, Article 15 (June 2021), 36 pages.

11. T. Wu, Y. Chen, and J. Han, "Re-examination of interestingness measures in pattern mining: a unified framework." Data Min. Knowl. Discov., 21(3):371–397, 2010.

12. R. Agrawal and R. Srikant, "Mining sequential patterns." In Proceedings of the Eleventh International Conference on Data Engineering, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.

13. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns by prefix-projected growth." In Proceedings of the 17th International Conference on Data Engineering, pages 215–224, Washington, DC, USA, 2001. IEEE Computer Society.

14. J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu, "Freespan: Frequent pattern-projected sequential pattern mining." In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '00, pages 355–359, New York, NY, USA, 2000. ACM.

15. W. C. Wilson, “Activity Pattern Analysis by Means of Sequence-Alignment Methods“. Environment and Planning A: Economy and Space, 30(6), pp. 1017-1038, 1998.

16. S. F. Smagula et al., “Activity patterns related to depression symptoms in stressed dementia caregivers,” International Psychogeriatrics, vol. 35, no. 7, pp. 373–380, 2023.

17. S. W. M. Cheng, J. A. Alison, E. Stamatakis, S. M. Dennis & Z. J. McKeough, “Patterns and Correlates of Sedentary Behaviour Accumulation and Physical Activity in People with Chronic Obstructive Pulmonary Disease: A Cross-Sectional Study”, COPD: Journal of Chronic Obstructive Pulmonary Disease, 17:2, 156-164, DOI: 10.1080/15412555.2020.1740189, March 2020.

18. R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases." SIGMOD Rec., 22(2):207–216, June 1993.

19. N. Gupta, S. K. Gupta, R. K. Pathak, V. P., Jain, V. Rashidi, & J. S. Suri (2022). Human activity recognition in artificial intelligence framework: A narrative review. Artificial intelligence review, 55(6), 4755-4808.

20. F. J. Ordónez, P. De Toledo, and A. Sanchis, "Activity recognition using hybrid generative/discriminative models on home environments using binary sensors," Sensors 13.5 (2013): 5460-5477.

21. D. Chen, S. L. Sain, and K. Guo, "Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining," Journal of Database Marketing & Customer Strategy Management, 19(3):197–208, 2012.

22. M. Lichman, "UCI machine learning repository," 2013.

23. P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C. W. Wu, and V. S. Tseng, "SPMF: a Java open-source pattern mining library," J. Mach. Learn. Res. 15, 1 (January 2014), 3389–3393.

24. I. Jonassen, J. F. Collins, and D. G. Higgins, "Finding flexible patterns in unaligned protein sequences," Protein Science, 4(8):1587–1595, 1995.

25. Custom dataset collection website, 2017.

26. T. Sztyler, J. Carmona, J. Völker, H. Stuckenschmidt (2016). “Self-tracking Reloaded: Applying Process Mining to Personalized Health Care from Labeled Sensor Data.”, Transactions on Petri Nets and Other Models of Concurrency XI, Spring, vol 9930. 160-180, 2016

27. H. F. Nweke, Y. W. Teh, G. Mujtaba, M. A. Al-garadi, Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions, Information Fusion, Volume 46, 2019, Pages 147-170, ISSN 1566-2535,

28. M. A. Islam, M. R. Rafi, A. Azad, J. A. Ovi, Weighted frequent sequential pattern mining. Applied Intelligence, 52, 254–281 (2022).

29. F. Adedeji, A. Adekunle, A. Adebayo, & O. Alao, Development of a Process-Driven Model for Extracting User Behavioral Pattern from Web Access Log Files, 2022.