Analyzing the health data: an application of sequential data mining

Kadium Padmavathi
Department of CSE
SRM University, Andhra Pradesh
Guntur, India
padmavathi_kadium@srmap.edu.in

Sumalatha Saleti
Department of CSE
SRM University, Andhra Pradesh
Guntur, India
sumalatha.s@srmap.edu.in

Jayanth Sai Vinnakota
Department of CSE
SRM University, Andhra Pradesh
Guntur, India
jayanthsai_vinnakota@srmap.edu.in

***Abstract*—A branch of data mining called "pattern mining" looks for interesting patterns in data by using algorithms. These patterns can help with data interpretation, decision-making, and prediction, among other tasks. Sequence mining from big databases makes use of it. The majority of sequence mining algorithms are essentially derivatives of earlier algorithms. It is used in sequence mining from large databases. Most sequence mining algorithms are upgrades of previous algorithms. There are lots of applications in retail, healthcare, and security where sequential pattern mining is used. Data mining has been used in a variety of ways in the medical industry. During the previous research, the main technique was the analysis of a US health dataset covering the years 2013–2017. The dataset is organized into categories like illnesses, states, and fatalities. We can identify high-utility itemsets that provide information on the most common causes of death by looking at these categories and mortality rates. After obtaining the high-utility itemsets, we can run through a pattern-mining algorithm and get the sequence of causes that give the most number of deaths. An extremely popular sequential data mining algorithm called PrefixSpan is applied in the proposed methodology to extract sequential patterns.**

Categories and Subject Descriptors - [Health Database Application] Data Mining
General Terms - Algorithms, observations, and comparisons.
Keywords - High utility, mining algorithm, mortality rate, pattern mining; prediction; sequence;

I. INTRODUCTION

A. Sequential Pattern Mining

A set of itemsets organized in a particular order inside a sequence database is called a sequential pattern. This database consists of arranged items or occurrences, occasionally including a time reference but not always. Items with the same transaction-time value make up each itemset. Sequential patterns draw attention to correlations between transactions, as opposed to association rules, which concentrate on relationships within individual transactions [1]. Extracting particular sequences from a dataset under the condition that their frequency exceeds a minimum support threshold is known as sequential pattern mining. By identifying the most common behaviours in the dataset, this process seeks to provide insightful information about the domain. It is possible to eliminate pointless sequential patterns and keep attention on significant ones by establishing a minimum support threshold. Higher support typically indicates more people are interested in a sequential pattern. Applications for this mining technique can be found in many different fields. For example, computational biology helps investigate patterns of amino acid mutation, and in business, it helps analyze customer behaviours. Moreover, it makes it possible to extract patterns from dispersed weblogs across several servers for web usage mining [2]. A technique called sequential pattern mining is used to find important sequences in large databases. A sequence database recognises recurrent subsequences as patterns. Several industries are becoming more and more interested in extracting sequential patterns from their databases as the amount of data being collected and stored increases. This widely accepted and used mining technique is used in many different domains, including web-log analysis, customer purchase behaviour analysis, and medical record examination. For example, transaction records of consumers can be examined in the retail industry to identify sequential patterns [3]. Agrawal et al. (1993) developed the use of market basket analysis in association rule mining for frequent pattern mining, which was a groundbreaking example. For example, in a shopping history database, a frequent item set might consist of items like milk, tea, coffee, and sugar that are frequently purchased together for making tea or coffee. Sequential pattern mining and structural pattern mining are two classifications that fall under the umbrella of frequent pattern mining, depending on the kind of data being studied [1]. One popular method in sequential pattern mining is the support model, which seeks to locate the entire set of frequently occurring sequences in a set of sequences. Although a lot of work has gone into effectively identifying the patterns that the support model outlines, not as much has gone into carefully evaluating the usefulness of the patterns that are produced. The multiple alignment model is an alternate pattern definition used in sequential pattern mining. In order to identify the fundamental consensus patterns in the data, this model aims to arrange and condense sequences of sets. Clustering is a first step in the approximate algorithm ApproxMAP, which groups similar sequences together. After that, it uses multiple alignments to directly mine each cluster's underlying consensus patterns [4]. The best evaluation standards for a given problem are usually customized for the domain in question. This indicates that the requirements are based on the kinds of sequential patterns that are most pertinent to the use case in that field. As a result, the perfect benchmark should provide a variety of target patterns

along with evaluation criteria that are appropriate for each type of pattern [5]. This makes it easier for users to select the best mining technique for their application by letting them select the pertinent target patterns and evaluation criteria based on the domain requirements.

B. PrefixSpan Sequential Pattern Mining

The general idea behind it is to project only the corresponding postfix subsequences into projected databases and only look at the prefix subsequences. Sequential patterns are grown in each projected database by examining only local frequent patterns [8]. Level-by-level and bi-level database projections are two types of projections that have been studied by researchers to improve mining efficiency. To further cut costs and speed up processing, they have developed a pseudo-projection method based on main memory. If the projected sub-database and the pseudo-projection processing structure that goes with it can fit in the main memory, then this method enables greater processing speed [9]. PrefixSpan finds all frequently occurring sequential patterns in a sequence database subsequences that occur in more than the minimum number of minsup sequences in the database. Sequences, which are collections of itemsets, make up a sequence database. Unsorted groupings of distinct items are represented by itemsets. Take into consideration the four sequences in Table 1 below, for example. There are five itemsets in the first sequence, designated A. This means that item 1 was followed concurrently by items 1, 2, and 3, then items 1 and 3, then item 4, and finally items 3 and 6.

TABLE I
TRANSACTION DATABASE FOR PREFIXSPAN ALGORITHM

Transaction	Sequence
A	(1), (1 2 3), (1 3), (4), (3 6)
B	(1 4), (3), (2 3), (1 5)
C	(5 6), (1 2), (4 6), (3), (2)
D	(5), (7), (1 6), (3), (2), (3)

53 sequential patterns are found by the algorithm when PrefixSpan is run with a minimum support of 50% and a maximum pattern length of 100 items. The list is too long to be presented here in its entirety. ”(1,2),(6)” is an example of a pattern that was found; it has 50% support and appears in both the first and third sequences. This pattern has three items, making it three in length. ”(4), (3), (2)” is another example pattern that can be seen in the second and third sequences and has a 50% support rate. In the same way, this pattern has three pieces, which equals three lengths. Similarly, when we run the transactions as above table 1 we obtain the results as (2, 3), (1), (6), (2), (6), (2), (3) patterns by running on prefix span algorithm.

C. High Utility Sequential Pattern Mining

Researchers are becoming increasingly interested in mining High Utility Sequential Patterns (HUSP), which has become a focal point in data mining. The purpose of these HUSP mining algorithms is to search statistical sequence databases for sequential patterns of notable utility or significance. The time intervals between elements are important in real-world scenarios. However given these time intervals between elements, existing HUSP mining algorithms are unable to extract sequential patterns [6]. It can be difficult to manage the potentially large search space in HUSPM effectively, particularly in the absence of strong pruning techniques based on utility upper bounds. Fundamentally, the main goals of HUSPM are to minimize memory consumption and speed up execution times while maintaining HUSPs. Improving sequence data utility mining efficiency is still an open problem [7]. Developed by Zida et al. in 2012, the USpan algorithm is well known for its ability to find high-utility sequential patterns in a sequence database that is enhanced with utility data. An example of a database like this would be a customer transaction record, which would be a series of customer actions recorded, with each action represented by a set of items marked with the profit generated from the sale of those items.
High-utility sequential rule mining seeks to find patterns like A, B, and C, which suggest that many customers have bought items A, B, and C, respectively, and have made a sizable profit. Let us consider an example of Uspan for a better understanding. Considering 4 sequences of transactions with their sequence utility A, B, C, and D. In Table 2, we can see the transactions and their total utility by summing up the sequence’s individual utility.

TABLE II
CONSIDERING 4 SAMPLE TRANSACTIONS

Transaction	Sequence	Utility
A	{1[1],2[4]},{3[10]},{6[9]},{7[2]},{5[1]}	27
B	{1[1],4[12]},{3[20]},{2[4]},{5[1],7[2]}	40
C	{1[1]},{2[4]},{6[9]},{5[1]}	15
D	{1[3],2[4],3[5]},{6[3],7[1]}	16

After the transaction sequence is run on the Uspan algorithm we result in the sequences for the high profit rate. It’s calculated as Imagine a database containing customer transaction sequences, where each customer purchase is represented by a separate sequence. As an example, the first client, identified as ”s1,” purchased items 1 and 2, resulting in profits of $1 and $4, respectively. Then, this customer purchased item 3 for $10, item 6 for $9, item 7 for $2, and item 5 for $1.

TABLE III
HIGH UTILITY SEQUENTIAL PATTERN BY USPAN ALGORITHM

Patterns	Utility
1, 4), (3) (2)	37
(1, 4) (3) (7)	35
(1) (3) (7)	36)
(3)	35
(3) (7)	40
(4) (3) (2)	36
(4) (3) (2) (5)	37
(4) (3) (2) (7)	38
(4) (3) (2) (5, 7)	35

By adding up the maximum profit a pattern produces across all sequences in which it appears, one can calculate the utility or profit associated with a succession. As an example, let us look at rule (3)(7) that can be found in the sequences s1, and s2, for example. Sequence S1 yields a profit of 10 + 2 = $12 from this pattern. 20 + 2 = $22 is the value in sequence s2, and 5 + 1 = $6 is the value in sequence S4. This means that the rule's total database utility is equal to 12 + 22 + 6 = $40. A sequential pattern's utility or profit is calculated by adding up the maximum profit the pattern makes in all sequences in which it appears. Consider the rule (3)(7), for example, which can be found in the sequences s1, s2, and s4. This pattern yields a profit of 10 + 2 = $12 in sequence s1. Twenty plus two is twenty-two, and five plus one is six in sequence s4. Thus, 12 + 22 + 6 = $40 is the total utility of this rule in the database. From Table 3, we can obtain nine high-utility sequential patterns by running USPAN with a minimum utility of 35 and a maximum pattern length of 4 items.

II. RELATED WORK

The application of machine learning and sequential pattern mining methods to forecast medical outcomes, particularly the risk of in-hospital mortality in patients with acute coronary syndrome (ACS) [10]. From the French Hospital Discharge Database, sets of pertinent event sequences are extracted using sequential pattern mining. The care trajectories for specific patients are represented by these sequences. Researchers hope to obtain significant trends that can help predict patient outcomes by finding common sequences among patients. The researchers incorporate the extracted sequential patterns into predictive models to forecast the risk of in-hospital mortality. They use a text string distance metric in conjunction with machine learning algorithms, like Support Vector Machines (SVM), to gauge how similar patients' care patterns are to one another. Based on their care trajectory, this similarity measurement aids in estimating the probability that a patient will experience a specific outcome [10]. The best predictive model is the one that combines the SVM model with the edit-based distance metric. Receiver operating characteristic (ROC) curve scores, which gauge the model's capacity for discrimination, are used to evaluate the model's efficacy. The resulting ROC scores, which span from 0.71 to 0.99, show strong discriminating power across various permutations of similarity metrics and machine learning models. Overall, the study shows how useful sequential patterns are in forecasting medical events and raises the possibility of using these strategies as decision-support instruments to lower the number of ACS-related in-hospital deaths. Healthcare providers may be able to identify high-risk patients earlier and intervene more effectively to improve patient outcomes by utilizing patterns in patient care trajectories [11]. The difficulties encountered by conventional data mining algorithms when managing substantial amounts of intricate data that are gathered from shared resources. In particular, it is noted that compared to other pattern mining tasks like frequent itemset mining and association rule mining, sequential pattern mining (SPM), a fundamental task of data mining, is more difficult and complex. SPM also faces challenges when handling large amounts of data, including expensive memory, sluggish processing, and limited hard disc space [12]. Parallel or distributed computing techniques for sequential pattern mining have become significant answers with a range of uses to address these issues. The paper presents a comprehensive overview of parallel sequential pattern mining (PSPM) as it stands today. It offers insights into cutting-edge parallel SPM techniques and classifies conventional serial SPM approaches [12]. Apriori-based strategies, pattern growth-based tactics, hybrid algorithms, and partition-based algorithms are just a few of the parallel SPM approaches that are covered in the survey. Every strategy is covered in detail, with a discussion of its features, benefits, drawbacks, and summary. Furthermore, the paper delves into advanced topics related to PSPM, including hardware acceleration techniques, PSPM from uncertain data and stream data, and parallel quantitative, weighted, and utility sequential pattern mining. Additionally, it lists and evaluates a few well-known open-source PSPM software tools [13]. The paper concludes by summarizing PSPM's opportunities and challenges in the big data era and emphasizing the significance of addressing efficiency, scalability, and the increasing complexity of data mining tasks [13].

The results of mining the test data highlight the importance of using sequential pattern mining as a foundation for implementing additional data analysis tools in the future use of the system. Sequential pattern mining can be used to identify a variety of patterns that provide insights into various aspects of operations, including the behaviours of actors, the use of machinery, the spatial distribution of actors, the activities of forklifts, the attractiveness levels of actors, the social groups within manufacturing facilities, and opportunities for production process optimisation [14]. Every iteration of data prototyping yields new insights into the intended data and its possible uses. The discovery of important insights and the improvement of data collection tactics and analytical methods are made possible by this iterative process. It implies that extra value can be produced at a comparatively low cost during the development of data warehouses and intelligent manufacturing systems by using data mining techniques and continuous prototyping. Sequential pattern mining essentially lays the groundwork for future data-driven insights and manufacturing process optimisations by helping to comprehend and fully utilize the potential of the data gathered within the system. Through prototyping, organizations can iteratively refine data analysis approaches and improve the effectiveness and efficiency of their systems and processes, unlocking significant value [15].

The idea of "high utility pattern mining," which entails finding significant patterns based on variables like profit, frequency, and weight, is covered in the passage. High utility itemsets are one particular kind of pattern that has been thoroughly researched. These are sets of items whose utility exceeds a minimum threshold that the user defines. This methodology is especially useful in real-world settings like web services and retail marketing, where products have a

variety of attributes and finding significant patterns can help with decision-making [16]. High-utility itemset mining focuses on itemsets that may include rare items but have substantial utility in real-world scenarios, in contrast to frequent itemset mining, which identifies itemsets that occur frequently [16]. The use of high utility pattern mining in the medical domain is emphasized, whereby health datasets spanning several years and categories like illnesses, states, and fatalities can be analyzed. High utility itemsets can be generated to identify the main causes of death by looking at these categories in conjunction with mortality rates. In conclusion, high-utility pattern mining is a data science technique that seeks to find significant patterns according to measurable standards. It has proven useful in a number of domains, including medicine, where dataset analysis can reveal high utility itemsets pertaining to death rates and causes of death [17].

Data mining has been used in a variety of ways with regard to the medical industry. In this case, the main technique is the analysis of a US health dataset covering the years 2013–2017. The dataset is organized into categories like illnesses, states, and fatalities. These categories and mortality rates can be examined to obtain high-utility itemsets that identify the leading causes of death [17]. After concluding from previous research [17], the high utility itemsets are provided through which we will find the patterns i.e. sequential patterns for the most number of deaths for the causes in the US from 2013-2017. We will use the Prefixspan and Uspan algorithm for the comparison where Uspan will be taking the utility and uspan directly. The procedure for Uspan is longer as the calculations are extended through it. The comparative study will give us more conclusions regarding the sequences.

III. METHODOLOGY

The goal is to discover or extract useful knowledge from data for sequential pattern mining. Data can be analyzed in graphs, relational databases, time series, sequences etc. A sequence is an ordered list of symbols. Let us take an example as a sequence of items that can be purchased by customers over time like computer - monitor - router, list of words like I - go - back - home, list of locations like a-b-c-d etc. This is a popular determining task, introduced in 1994 by Agarwal and Srikant. A lot of algorithms have come into existence with the technology increasing like SPAM, SPADE, Prefixspan, GSP etc. Sequential algorithms have evolved by categories like mining high-utility sequential patterns, cost-efficient sequential patterns, progressive sequential pattern mining etc. We will be analyzing prefixspan and Uspan algorithms which are sequential mining algorithms. From the previous paper[17] after solving the application where we discussed the high utility mining over the health dataset of the US over a period from 2013-2017 a tenure of 5 years. This approach will lead to the high utility sets of the causes for the deaths in the US. Giving the itemsets of causes will let us know the itemsets which are causing a high number of deaths. This helps in recognising those causes which led to the most number of deaths during 2013-2017. We used HUI Miner on the dataset fixing different amounts of minimum utilities. We consider internal utility as deaths and external utility as the mortality rate of that cause in the period 2013-2017. Contemplating 10 causes by defining them as 1,2,3,4,5,6,7,8,9,10. The causes are considered as per the analysis. Causes are renamed as 1,2,3,4,5,6,7,8,9,10. Alzheimer's disease is renamed as 1, Cancer as 2, CLRD as 3, Diabetes as 4, Heart disease as 5, Influenza and pneumonia as 6, Kidney disease as 7, Stroke as 8, Suicide as 9 and Unintentional injuries as 10. After running on the HUIM algorithm we obtained high utility itemsets i.e. the causes which creates the most number of deaths. The high utility itemsets ranged from 1023 to 1, which can be observed going through the previous paper. We will be considering the top 5 i.e. 1 to 12 high utility itemsets and giving its itemsets to the prefix span and uspan algorithms which are changed and updated in the algorithm to read the file with less time. In Table 4, we can see the values and itemsets considered for the algorithm.

TABLE IV
ITEMSET DATABASE FROM HUIM

Minimum Utility	High Utility Itemsets count
1013847239	1
1003395205	4
1002395205	15
997943172	8
993943172	12

After considering the Minimum utility we will be obtaining the itemsets that will be taken for the further application of algorithms. They should be segregated as per the input and output files and run on it. From Table 5, we can see that itemsets which are obtained from HUIM from Table 4 are mentioned which gives the high rate of death.

TABLE V
HIGH UTILITY ITEMSETS FROM US DATASET

Minimum Utility	Set of itemsets
1013847239	(8, 3, 10, 2, 5)
1003395205	{(6, 4, 1, 8, 3, 10, 2, 5), (4, 1, 8, 3, 10, 2, 5), (1, 8, 3, 10, 2, 5), (8, 3, 10, 2, 5)}
1002395205	{(6, 4, 1, 8, 3, 10, 2, 5), (4, 1, 8, 3, 10, 2, 5), (4, 8, 3, 10, 2, 5), (1, 8, 3, 10, 2, 5), (8, 3, 10, 2, 5)}
997943172	{(6, 4, 1, 8, 3, 10, 2, 5), (6, 4, 8, 3, 10, 2, 5), (6, 1, 8, 3,10, 2, 5), (4, 1, 8, 3, 10, 2, 5), (4, 8, 3, 10, 2, 5), (1, 8, 3, 10, 2, 5), (8, 3, 10, 2, 5), (8, 3, 2, 5)}
993943172	{(9, 6, 4, 1, 8, 3, 10, 2, 5), (9, 4, 1, 8, 3,10, 2, 5), (6, 4, 1, 8, 3, 10, 2, 5), (6, 4, 8, 3, 10, 2, 5), (6, 1, 8, 3,10, 2, 5), (4, 1, 8, 3, 10, 2, 5), (4, 8, 3, 10, 2, 5), (1, 8, 3, 10, 2, 5), (8, 3, 10, 2, 5), (8, 3, 2, 5), (6, 8, 3, 10, 2, 5), (8, 10, 2, 5)}

Leading with this database we will be implementing them on the Uspan algorithm upright by considering the utility for the cause is calculated by internal and external. Internal utility is taken as minimum utility and external

as the mortality rate. The mortality rates are considered for the 2013-2017 US health dataset. 1,2,3,4,5,6,7,8,9,10 causes 1 has 5.25, 2 has 38.71, 3 has 9.24, 4 has 4.87, 5 has 43.63, 6 has 3.79, 7 has 2.91, 8 has 9.69, 9 has 2.37 and 10 has 8.37. Calculating the utility for the subset we need to multiply internal * external utility. Let us consider 1013847239 Minimum utility, calculating 1013847239*9.69 = 9824179745 where 8-cause 9.69 as the mortality rate. Similarly, 1013847239*9.24=9367948488, 8520542100, 39406234731, 44414725428. Similarly, we need to calculate the remaining. The total utility is calculated by summing up the utilities of individuals and gets as 111611975611. Calculating the remaining we get a long list of databases which should be sent as an input file to the Uspan algorithm and for Prefixspan we need to just give the set of itemsets which have considerable changes.

Uspan Algorithm

1. Begin with an empty pattern prefix.
2. Set the minimum utility threshold.
3. Initialize an empty set to store the discovered high utility sequential patterns.
4. Iterate over each item in the sequence database.
5. Extend the current pattern prefix with each item.
6. Check if the utility of the extended pattern prefix exceeds the minimum utility threshold.
7. If the utility is above the threshold, add the extended pattern to the set of discovered patterns.
8. Apply pruning techniques to reduce the search space and improve efficiency.
9. Prune patterns that cannot potentially become high utility patterns.
10. If an extension of the pattern prefix does not meet the minimum utility threshold or cannot be further extended, backtrack to the previous step and explore other possible extensions.
11. Repeat steps 2-10 until all possible extensions of the pattern prefix have been explored.
12. Terminate the algorithm when no further high utility sequential patterns can be found.

PrefixSpan Algorithm

1. Start with an empty prefix.
2. Scan the sequence database to find frequent items. Record the frequency of each item and their positions in the sequences.
3. For each frequent item found in step 2, extend the prefix with that item and recursively mine the database for the projected database, which contains all sequences that contain the prefix followed by the item.
4. For each projected database, check if the support (number of sequences containing the pattern) is greater than or equal to the minimum support threshold.
5. Output the patterns that meet the minimum support threshold.
6. Recursively call PrefixSpan on the projected database to continue pattern mining.
7. After mining all patterns with the current prefix, backtrack to the previous level of the prefix and try extending it with another frequent item.
8. Repeat steps 3-7 until all frequent sequential patterns are mined.

After running on the HUIM algorithm we will lead to the second path of the algorithm where we will run on Uspan and Prefixspan algorithms and get a comparative study which will be more efficient to run on.

IV. RESULTS AND DISCUSSIONS

We considered Minimum utilities as 1013847239, 1003395205, 1002395205, 997943172 and 993943172. We will be calculating the Time and memory for Uspan and Prefixspan. Running on both the algorithms we conclude with a sequence of 2 3 5 8 10, which is obtained by taking the minimum utility of 471384723 for Uspan as support and support as 1 for minsup 5 in prefixspan. We can see from Table 6, the time for Uspan is more than Prefixspan. We calculated the time as HUIM is run first as per the procedure.

TABLE VI
COUNT OF HIGH UTILITY SETS, TIME AND MEMORY WITH RESPECT TO MINIMUM UTILITY

Min Utility	Time Uspan	Time HUIM	Total Time	Time PrefixSpan	Time HUIM	Total Time
1013847239	21	14	35	8	14	22
1003395205	28	13	41	10	13	23
1002395205	33	11	44	13	11	24
997943172	41	8	49	16	8	24
993943172	43	8	51	21	8	29

We can see from table 7, the memory for Prefixspan is less than the memory for Uspan. The total memory is calculated including the memory for HUIM.

TABLE VII
COUNT OF HIGH UTILITY SETS, TIME AND MEMORY WITH RESPECT TO MINIMUM UTILITY

Memory PrefixSpan	Memory HUIM	Total Memory	Memory Uspan	Memory HUIM	Total Memory
3.4483	4.43073	7.8790	3.431	4.43073	7.8617
3.4404	3.9044	7.3449	3.452	3.9044	7.3564
3.44049	3.90444	7.34493	3.694	3.90444	7.5984
3.44003	3.4444	6.8844	3.722	3.4444	7.1664
3.43276	3.44441	6.8771	3.892	3.44441	7.3364

Getting to the graphical representation is below. From figure 1, we can see that the total time taken for Prefixspan is less than Uspan.

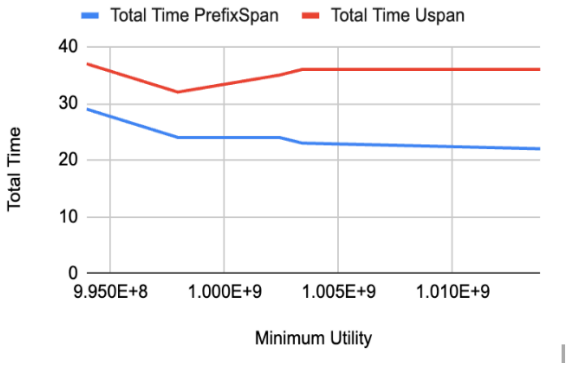

Fig. 1. Total Time - Prefixspan vs Uspan

In Figure 1, the X axis is taken as minimum utility and the Y axis is taken as Total Time. The graph shows us the comparison of time taken by Uspan and Prefixspan.

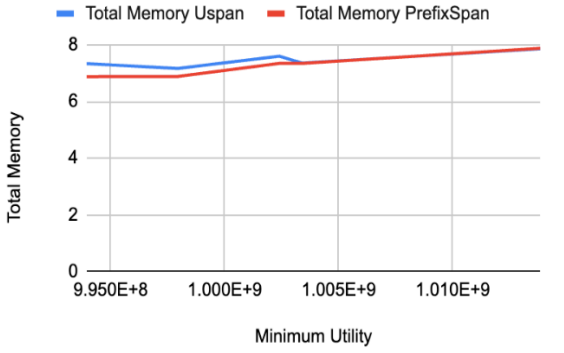

Fig. 2. Total Memory - Uspan vs Prefixspan

In Figure 2, we can see that the memory taken by Prefixspan is more than the Uspan. The X-axis is considered as minimum utility and the Y-axis is total memory. We can analyze that Uspan has efficient time and memory in our application. It is due to the data set taken. It differs based on the application.

V. CONCLUSIONS

In summary, pattern mining is a potent method for identifying significant and useful patterns in a variety of domains using massive datasets. Pattern mining helps businesses find important insights, make wise decisions, and streamline processes by extracting patterns like frequent itemsets, sequential patterns, or high utility itemsets. Pattern mining is essential for drawing insights from data, whether it is used in market basket analysis for retail, disease trends identification for healthcare, or user behaviour analysis for web services. Pattern mining algorithms efficiently navigate through the large search space of potential patterns, taking computational constraints into account and producing valuable results through the use of algorithms such as Apriori, FP-Growth, PrefixSpan, or USpan. All things considered, pattern mining remains a fundamental component of data analysis, enabling the retrieval of practical knowledge that spurs creativity and enhances judgment across a range of sectors. Generally speaking, USpan performs better than PrefixSpan in terms of time and memory usage, particularly for large-scale sequence databases. Its effective pruning methods and use of projection to lower the search space and memory requirements are to blame for this. However, the exact features of the dataset and the specifics of the implementation may affect the actual performance. In our case, Prefixspan showed better performance than Uspan where we concluded with a sequence of 2 3 5 8 10 which gives the number of deaths. It can be explained as Cancer as 2, CLRD as 3, Heart disease as 5, Stroke as 8 and Unintentional injuries as 10 occurring in a sequence creates the most number of deaths 480944524 due to these causes. This application can be used in the medical field for pre medication and diagnosis to reduce the mortality rate and for the better future. The pattern extracted can create a lot of change in the medical field. This application is based on a medium-sized data set. the population may vary and increase day by day but the technology created and evolving can make great changes to the world.

REFERENCES

[1] R.Agrawal and R.Srikant. Mining sequential patterns. Proceedings of the Eleventh International Conference on Data Engineering, 1995.
[2] R.Agrawal and R.Srikant. Fast algorithms for mining association rules in large databases. Proceedings of 20th International conference on Very Large Databases. 1994.
[3] Rakesh Agrawal Ramakrishna Srikant, “Mining Sequential Patterns”, 11th Int. Conf. on Data Engineering, IEEE Computer Society Press, Taiwan, 1995 pp. 3-14.
[4] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the International Conference on Very Large Databases (VLDB), 1994, pp. 487–499.
[5] J. Ayres, J. Flannick, J. Gehrke, T. Yiu, Sequential pattern mining using a bitmap representation, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2002, pp. 429–435.
[6] Truong Chi, T., P. F o u r n i e r-V i g e r. A Survey of High Utility Sequential Pattern Mining. – High-Utility Pattern Mining: Theory, Algorithms and Applications, Vol. 51, P. FournierViger, J. Lin, R. Nkambou, B. Vo, V. Tseng, EdsCham, Springer, 2019.
[7] W. Gan, J. C. W. Lin, H. C. Chao, and J. Zhan, “Data mining in distributed environment: a survey,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 7, no. 6, p. e1216, 2017.
[8] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf. Very Large Data Bases (VLDB 94), pages 487–499, Santiago, Chile, Sept. 19.
[9] R. Agrawal and R. Srikant. Mining sequential patterns. In Proc. 1995 Int. Conf. Data Engineering (ICDE 95), pages 3–14, Taipei, Taiwan, Mar. 1995.
[10] A. B. Jensen, “Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients,” Nature Communications, vol. 5, p. 4022, 2014.
[11] Jessica Pinaire . Etienne Chabert and Jerome Aze, “Sequential Pattern Mining to Predict Medical In-Hospital Mortality from Administrative Data: Application to Acute Coronary Syndrome”, Journal of Healthcare Engineering, Volume 2021.
[12] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In ACM SIGMOD Record, Vol. 22. ACM, 207–216.

[13] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Philip S. Yu. 2018. A Survey of Parallel Sequential Pattern Mining. ACM Trans. Knowl. Discov. Data. 0, 1, Article 00 (August 2018).

[14] A. Gerstenberg, H. Sjöman, T. Reime, P. Abrahamsson, and M. Steinert. "A Simultaneous, Multidisciplinary Development and Design Journey–Reflections on Prototyping." In Entertainment ComputingICEC 2015, pp. 409-416. Springer International Publishing, 2015.

[15] Heikki Sjöman, Martin Steinert, "Applying sequential pattern mining to portable RFID system data", Department of Engineering Design and Materials NTNU, 2016.

[16] R. Agrawal and R Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th Conf. Very Large Data Bases (VLDB), 1994.

[17] Kadium Padmavathi. Saleti Sumalatha and Tottempudi Sai Saran, "Analyzing the Health Data: An Application of High Utility Itemset Mining ", International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), 2023.