International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

A Kernel Based Apriori Algorithm for Sequential Pattern Mining

Ravi Kumar V & Dr. Purna Chander Rao

[1]Associate Professor, Dept of IT, TKRCET & Research Scholar, JNTUH, India.
[2]Professor, SVIET, India

Abstract: *Several researchers focused on the sequential pattern mining problem and many algorithms weredeveloped to mine sequential patterns. In this paper, we propose an efficient algorithm K-Apriori which makes use of a new pruning methodcalled Hamming Distance that allows the early detection of sequential patterns during themining process. This approach is an extension to the most popular Apriori algorithm. This K-Apriori algorithm forms the kernels based on the membership of each items and then grouped them into one kernel those having same membership values. Since, the items are grouped; there is no need to scan all the items for every mining process. This approach reduces the time taken for mining greatly. The extensive simulations reveal the performance of proposed approach under various test cases.*

Keywords:*Sequential Patten mining, Apriori, M-Apriori, Runtime, Hamming Distance, Kernel*

I.INTRODUCTION

In recent days, the sequential pattern mining (SPM) is becoming more research issue due to the wide variety of storage applications over the internet. Mining [1] the useful information from internet is becoming more critical due to the large size of data storages. Sequential pattern mining [2], [3] has numerous applications, including the discovery of motifs in DNA sequences, the analysis of web log and customer shopping sequences, the study of XML query access patterns, and the investigation of scientific or medicalprocesses.Sequential pattern mining istrying to find the relationships between occurrences of sequential events for looking forany specific order of the occurrences. In the other words, sequential pattern mining is aiming at finding the frequently occurred sequences to describe the data or predict future data or mining periodical patterns. To gain a better understanding of sequential pattern mining problem, let's start by looking at an example. From a shopping store database, we can find frequent sequential purchasing patterns, for example "70% customers who bought the TV typically bought the DVD player and then bought the memory card with certain time gap." It is conceivable that achieving this pattern has great impact to better advertisement and better management of shopping store.Sequential pattern mining methods often use the support, which is the criterion to evaluate frequency but this parameter is not efficient to discover some patterns. Various techniques are proposed to obtain frequent itemset through sequential pattern mining, however, every method having various drawbacks. In earlier, Agrawal et al. proposed the Apriori algorithm [2] to generate-and-test candidates for mining the sequential patterns from a static database. However, there are two limitations of this algorithm: one is complex candidate itemset generation process which consumeslarge memory and enormous execution time and second problem is excessive database scans for candidate generation. In [4], an enhanced Apriori (M-Apriori) algorithm was proposed to overcome the problems of Apriori algorithm by deleting the itemset which are not frequent. However, the deletions of non-frequent items will effects on the accuracy.

In this paper, a new SPM algorithm named as K-Apriori was proposed to overcome the above mentioned drawbacks with conventional algorithms with the base of Apriori algorithm. This approach, all the items will be stored in the dataset. No single item will be deleted, because, if the item with minimum count was deleted, the accuracy will be reduced and the system is not able to recognize in future. So, in this approach, the items those having similar characteristics are grouped into kernels. During the mining, for a given sequence, only the kernel with matched membership will be scanned and the remaining kernels will not be scanned. This feature of proposed approach reduces the execution time greatly. Simulation results are carried out over various dataset with various sizes reveals the efficiency of K-Apriori algorithm.

The rest of the paper is organized as follows: section II illustrates the details about the earlier approaches proposed over SPM. Section III illustrates the details of the sequence pattern alignment. Section IV gives the complete details about the proposed approach. Simulation results were shown in the section V. runtime was evaluated under this section to measure the performance of proposed approach. a comparative analysis is also carried out with conventional approaches.Finally the conclusions are given in section VI.

II. RELATED WORK

Various approaches were proposed in earlier to achieve the reduced computational complexity, reduced computational time and increased accuracy. Pei et al. designed the PrefixSpan [5] algorithm to efficiently mine the sequential patterns based on the projection mechanism. A sequence database is recursively projected into several smaller sets of projected database to speed up the computations for mining sequential patterns. Zaki et al. proposed a SPADE [6] algorithm to fast mine the sequential patterns. The SPADE algorithm utilizes the combinational properties based on the efficient lattice search techniques with join operations. Based on SPADE algorithm, the sequential patterns can be derived with three database scans. Many algorithms have been proposed to mine the sequential patterns, but most of them are performed to handle the static database. When the sequences are changed whether sequence insertion [7] or deletion [8] in the original database, the discoveredsequential patterns may become invalid or new sequentialpatterns may arise. An intuitive way to update the sequentialpatterns is to re-process the updated database in batch mode, which is inefficient in real-world applications.To handle the dynamic database with sequence insertion,Lin et al. proposed a FASTUP algorithm [9] to incrementallymaintain and update the discovered sequential patterns withsequence insertion. The original database is still, however,required to be rescanned if the discovered sequential pattern islarge in the added sequences but small in the original databasebased on the FASTUP concept. Hong et al. then extended thepre-large concept of association-rule mining to respectivelymaintain and update the discovered sequential patternswith sequence insertion [10] and sequence deletion [11] in alevel-wise way, which requires more computations of multiple database rescans. Lin first designed a fast updated sequentialpattern (FUSP)-tree and developed the algorithms for efficiently handling an incremental database with sequence insertion [12]. The FUSP tree is built in advance beforethe sequences are inserted into the original database. Twoparts with four cases are then divided based on the FASTUPconcept to maintain and update the FUSP tree for later miningprocess. Lin et al. also proposed the maintenance algorithm for sequence deletion [13]. The original database is, however,required to be rescanned if it is necessary to maintain asequence which is small in the original database but large inthe inserted sequences with sequence insertion or a sequenceis small both in the original database and in the deletedsequences with sequence deletion. Some more approaches [17-22] are proposed in earlier, however, all approaches having their own drawbacks.

III. SEQUENTIAL PATTERN ALIGNMENT

Since, the sequences can be represented in the binary format, which can be buffered in the large dataset. The patterns will be varied with few bits of transitions thus there exists a correlative nature among them. For any two sequences, the correlation will represent the strength of relation between them. Due to this correlative nature, there is a possibility of misclassification of sequences which tends to reduction in the accuracy. Also, there are multiple sequences will be exist in the large dataset, the correlation will be high. Generally, the correlative nature of binary sequences can be measured through the metric Hamming Distance (HD) [14]. The mathematical formulation for hamming distance is represented as,

$$d_{HD}(s_1, s_2) = \sum_{s_{1i} \neq s_{2i}} 1\,, \quad i = 1, \dots \dots m \qquad (1)$$

Where, d_{HD} measures the number of varying positions in two strings s_1 and s_2 of equal length m.

For Example, s_1: “ABCDDCBD”, s_2: “ABDDCCBD”, $d_{HD} = 2$.

This evaluation is carried out for equal length sequence. In the case of non-equal length sequence, the hamming distance is evaluated through the number of transitions.

For example, s_1: “ABCDDCBD, The number of transitions (t_1)=6.

s_2: “ABCBD”, The number of transitions (t_2)=4.

$d_{HD}(s_1, s_2) = t_1 - t_2 = 6 - 4 = 2.$

In the process of pattern matching the hamming distance provides the correlation strength. However, applying these approaches for measuring distances, the bit pattern plays and important role. It is observed that more the pattern transition exist the probability of pattern matching error get higher. The pattern matching errors could be minimized via updated sequencing of binary patterns. To develop such approach in recent past [15] a 1-D Local binary pattern (LBP) representation of signal is presented. A LBP is locally transformed binary pattern used for the optimization of pseudo random nature of a binary pattern. They are more dominantly used as a filtration approach in image processing, signal processing, speech processing etc.[15].A LBP sequence is generated as uniform patterns if they have at most two circularly bitwise transitions from 0 to 1 or vice versa, and non-uniform patterns if otherwise. For example, “11110000” is a uniform pattern as $U = 2$, whereas “01010111” is a non-uniform pattern as $U = 6$. With the realignment of such pattern the error robustness is achieved, the LBP is hence robust to alignment error [16]. With this approach the LBP coding is applied over sequence pattern, and the patterns are categorized into uniform and non-uniform patterns to reduce alignment and matching error. The spectral feature of the pattern distribution using Histogram bin is computed and the patterns are realigned to achieve lower transition matching error. For this local binary patterns the sequence are stored as a data set, and mining is then carried out over such binary pattern.

IV. PROPOSED K-APRIORI

The earlier proposed Apriori and the M-Aprioriare succeeded in achieving better accuracy as well as reduced computation time. But, both the approaches have individual drawbacks as mentioned above. To overcome the drawbacks of conventional Apriori approaches, in this paper, a new Apriori algorithm is proposed by considering the correlative nature of items present in the database. For this, purpose, the proposed approach initially forms some kernels based on the individual hamming distances of Items. Let's define some definitions such as $T = \{T_1, T_2, \dots, T_m\}$ be the transaction set, $I = \{I_1, I_2, \dots, I_n\}$ be the set of items in each and every transaction and k-item set $= \{i_1, i_2, \dots, i_k\}$ is also a itemset such that it is a subset to item set I, $k \subseteq I$. Suppose $D = \{D_1, D_2, \dots, D_n\}$ be the individual hamming distances, i.e., the number of transitions of individual item. Based on the obtained Hamming distances, totally n kernels will be formed. The items with same hamming distance will be formed as a kernel.

Table 1. The Transactions

Transactions	Item sets
T1	Rice, oil, Ginger
T2	Rice, wheat, sugar
T3	Milk, dal, ginger
T4	Dal, sugar, oil
T5	Wheat, rice, potato
T6	Rice, dal, sugar, milk, potato
T7	Rice, dal, sugar, milk, oil

Table.2. TheHamming distances

Items	Hamming Distance (D)
Rice	3
Oil	2
Wheat	4
Sugar	4
Milk	3
potato	5
Dal	2
ginger	5

Based on the obtained hamming distances, here totally three clusters will be formed. Kernel with membership of 5, Kernel with membership of 4, Kernel with membership of 3and Kernel with membership of 2. Then the items with similar HD will be grouped into one kernel as,

Table 3.Cluster itemset

Kernel	Membership	Items	Transaction IDs
Kernel1 (K1)	5	Potato	T5, T6
		Ginger	T1, T3
Kernel2 (K2)	4	Wheat	T2, T5
		sugar	T2, T4, T6, T7
Kernel3 (K3)	3	Milk	T3, T6, T7
		Rice	T1, T2, T6, T7
Kernel4 (K4)	2	Oil	T1, T4, T7
		Dal	T3, T6, T7

By forming a kernels and allotting them to the respective items, all the items are placed in any one of the cluster. No any item was missed or deleted. This process increases the accuracy. In the case of modified Apriori [4], the item with least frequency was being deleted. If the query is from that least frequent item, then the accuracy will be reduced due to deletion of that item form the dataset. The complete kernelling of the entire dataset is shown in the figure. 3.

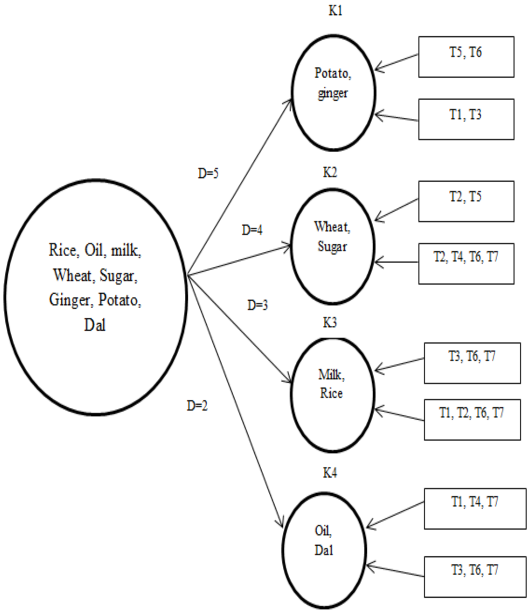

Figure.1. Kernelling of total dataset along with transaction IDs

The final output of the kernel tree is shown in the figure.1, from figure.1, the minimum membership is two and the maximum membership is five. Now, find thefrequent patterns from the kernel tree. The items of the database and their memberships areshown in table.3 for each item.First and foremost, we need to prioritize all the itemsets according to their memberships and then see each item one by one from bottom to top. Now, to mine the given query, the proposed approach evaluates the hamming distance and then compares with the predefined memberships. If any one of the membership was matched with the membership of query, it processes into that kernel. Further, the query compares with the hamming distances of individual items of that particular kernel. If the evaluated hamming distance for any item with the query item is zero, then the item is said to be matched. The pattern matching criterion for the given pattern or sequence is shown in the following algorithm.

Step 1: given input sequence, for example Sugar.

Step 2: evaluate individual hamming distance of input sequence, i.e., the number of transitions, t=4.

Step 3: perform a matching process by evaluating the difference between the HD of input pattern with the predefined membership values. The membership of input sequence is said to be matched if the difference was zero for any one of the membership values. Thus, the given sequence belongs to a kernel with membership 4.

Step 4: Now, the input sequence was compared with the items present in the kernel with membership using the formula given in Equation.(1).

Step 5: The obtained hamming distances are sorted from minimum to maximum.

Step 6: The item with minimum hamming distance is revealed as the matched item for a given input sequence.

The computational time of the proposed approach will be reduced very much due to the process carried out at step 3. Because, there is no need to scan the entire items. Since the hamming distance evaluation is also a simple process, the computational complexity of proposed approach is also less compared to conventional approaches.

V. SIMULATION RESULTS

In this section, the performance evaluation proposed algorithm is carried out on various data sets with various kinds of sizes and data distributions, and also a comparative analysis is carried out between the proposed approach and the conventional Apriori and M-Apriori. All experiments were conducted on a 4GHz Intel Core2 Duo processor PC with 500GB mainmemory running Microsoft Windows XP.The performance was analyzed by measuring the time taken to mine every item for a given dataset. The runtime for various datasets with various sizes is represented in the figure.3. The simulation was carried out over a synthetic dataset. This synthetic dataset carries the details of super market purchases of 10000 customers for one year over 3000 distinct item categories.

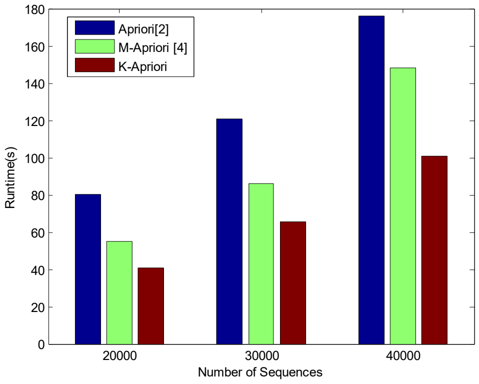

Figure.3. Runtime for varying data size

From the above figure, it can be observed that the runtime for the proposed approach is less compared with conventional approaches. For every case of dataset, the run time is observed as low. Because, the proposed approach mines the given pattern only by comparing it with the kernel of same membership. Then there is no need to scan the remaining kernels.
The runtime also varies with number of transaction. As the number of transaction increases, the runtime also increase. The runtime variations for varying transaction is represented in the figure.4

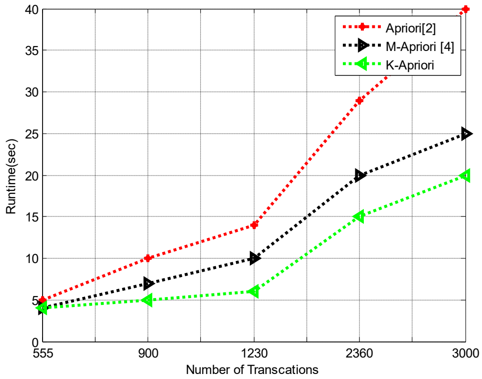

Figure.4 runtime for varying number of transactions

Figure.4 describes the details of the time taken for runtime with varying number of transaction. The number of transactions is varied from 500 to 2500 and the respective time consumption results for both the proposed and the conventional approaches are shown in figure.4. Since the proposed approach considers the hamming distance as a main characteristic of item, the time taken by proposed K-Apriori must be less when compare to Apriori and M-Apriori. In the above figure, the time taken is increasing with number of transaction, but the increment is less for K-Apriori compared to Apriori and M-Apriori.
Here two more cases were carried out to show the efficiency of proposed approach. K-Apriori is completely based on the kernels. So, the varying number of kernels also plays an important role in the mining process. As the number of categories of items increases, the number of kernels also increases.

Because, the kernels are formed based on the individual hamming diatnsces of every item which is nothing but named as membership. The runtime variations for the varying items and also for varying kernels are represented in the figure.5 and figure.6 respectively.

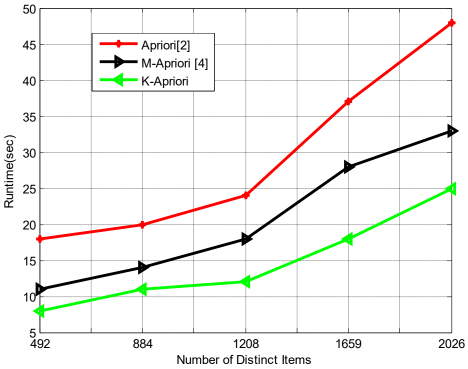

Figure.5 runtime for varying number of distinct items

The obtained runtime details of the proposed and conventional approaches were illustrated in the figure.5. As the number of distinct items increases, the runtime will increases. In the proposed approach also, this is normal. But compared to the conventional approaches the increment is less due to the kernelling of items.

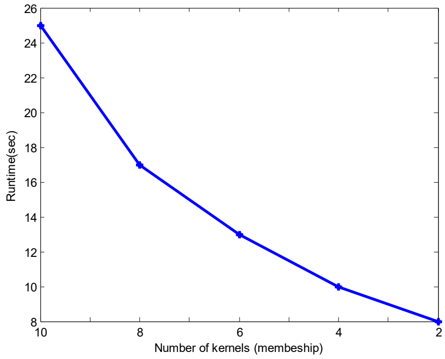

Figure.6 runtime for varying number of kernels

All the above experiments confirm that the proposed algorithm K-Apriori is efficient and takes lessrun time comparing to conventional approaches. Because K-Apriori considered the internal features of patterns such number of transitions and grouped as kernels those having same features, whereas the conventional Apriori and the M-Apriori scans the entire items to mine the given query sequence.

VI. CONCLUSION

In this paper, a new approach was proposed for sequential pattern mining based on the earlier Apriori algorithm. This approach successfully reduces the memory space and also reduces the time required for execution even for large datasets. The enhancement of the proposed approach can be observed when there is an increment in the itemsets and also in the number of transactions. The time consumed to generate candidate sequence in our proposed Apriori is less than the time consumed in the conventional Apriori. The results also reveal the efficiency of proposed approach in the view of time consumption

VII. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, ``Database mining: A performance perspective,'' *IEEE Trans. Knowl. Data Eng.*, vol. 5, no. 6, pp. 914_925, Dec. 1993

[2] R. Agrawal and R. Srikant, ``Mining sequential patterns,'' in *Proc. Int. Conf. Data Eng.*, 1995, pp. 3-14.

[3] C. H. Mooney and J. F. Roddick, ``Sequential pattern mining: Approaches and algorithms,'' *ACMComput. Surveys*, vol. 45, no. 2, pp. 1-39, Feb. 2013.

[4] MohammedAl-Maolegi, “An improved Apriori Algorithm for Association rules”, International journal on natural language computing (IJNLC), Vol.3, No.1, February 2014.

[5] J. Pei *et al.*, ``Mining sequential patterns by pattern-growth: The Prefix Span approach,'' *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1424-1440, Nov. 2004.

[6] M. J. Zaki, ``SPADE: An efficient algorithm for mining frequent sequences,'' *Mach. Learn.*, vol. 42, nos. 1-2, pp. 31-60, Jan. 2001.

[7] D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong, ``Maintenance of discovered association rules in large databases: An incremental updating technique,'' in *Proc. 25th Int. Conf. Data Eng.*, Mar. 1996, pp. 106-114.

[8] D.W.-L. Cheung, S. D. Lee, and B. Kao, ``A general incremental technique for maintaining discovered association rules,'' in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Apr. 1997, pp. 185-194.

[9] M.-Y. Lin and S.-Y. Lee, ``Incremental update on sequential patterns in large databases,'' in *Proc. IEEE Int. Conf. Tools Artif. Intell.*, Nov. 1998, pp. 24-31.

[10] T.-P. Hong, C.-Y. Wang, and S.-S. Tseng, ``an incremental mining algorithm for

maintaining sequential patterns using pre-large sequences," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7051-7058, Jun. 2011.

[11] C.-Y. Wang, T.-P. Hong, and S.-S. Tseng, ``Maintenance of sequential patterns for record deletion," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2001, pp. 536-541.

[12] C.-W. Lin, T.-P. Hong, W.-H. Lu, and W.-Y. Lin, ``an incremental FUSP-tree maintenance algorithm," in *Proc. 8th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2008, pp. 445_449.

[13] C.-W. Lin, T.-P. Hong, and W.-H. Lu, ``An efficient FUSP-tree update algorithm for deleted data in customer sequences," in *Proc. Int. Conf. Innovative Comput., Inf. Control*, Dec. 2009, pp. 1491-1494.

[14] Bioinformatics Day 6: Classification in Next Generation Sequencing Data Analysis", 2013.

[15] NavinChatlani, John J. Soraghan, "Local Binary Patterns for 1-D Signal Processing", EUSIPCO-2010.

[16] JianfengRen, "Noise-Resistant Local Binary Pattern With an Embedded Error-Correction Mechanism", IEEE Transactions On Image Processing, 2013.

[17] WeeKeong,YewKwongAmitabha Das, "Rapid Association Rule Mining," in *Information and KnowledgeManagement*, Atlanta, Georgia, 2001, pp. 474-481.

[18] Jian Pei, Jiawei Han, "Mining Frequent patterns without candidate generation," in *SIGMOD '00 Proceedingsof the 2000 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 2000,pp. 1-12.

[19] PanchalMayur, LadumorDhara, KapadiyaJahnvi, Desai Piyusha, Patel Tushar S., "An Analytical Study ofVarious Frequent Itemset Mining Algorithms," *Research Journal of Computer and Information TechnologySciences*, p. 4, 2013.

[20] Chistopher.T, PhD SaravananSuba, "A Study on Milestones of Association Rule Mining ," *InternationalJournal of Computer Applications*, p. 7, June 2012.

[21] Borgelt C., "Efficient Implementations of Apriori and Eclat," in *1st IEEE ICDM Workshop on Frequent ItemSet*, 2003, p. 9.

[22] SrinivasanParthasarathy, and Wei Li Mohammed JaveedZaki, "A Localized Algorithm for ParallelAssociation Mining," in *In 9th ACM Symp. Parallel Algorithms & Architectures.* , 1997.