IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

A Novel Methodology for Mining Frequent Itemsets from Temporal Dataset

B. Sowndarya[1*], T. Meyyappan [2], S.M. Thamarai[3]

[1,2]Department of Computer Science, Alagappa University, Karaikudi, Tamil Nadu, India
[3]Alagappa Government Arts college, Karaikudi, Tamil Nadu, India

**Corresponding Author: sowndaryabalu@gmail.com Tel:7418894502*

Available online at: www.ijcseonline.org

Accepted: 23/Jul/2018, Published: 31/Jul/2018

Abstract- Traditional data mining techniques predict frequent itemsets without considering the temporal data. Due to this, efficiency of the frequent itemsets mining is not upto the mark on the temporal data. A new extended apriori algorithm proposed in this research work handles the time interval while identifying the frequent itemsets. The main objective of this research work is to identify patternset in periodic intervals from the temporal data sets. Datasets from UCI data repository is subjected to this proposed method. Experimental results are tabulated and plotted. The results show improvement over the traditional apriori algorithm.

Keywords: Data mining, Apriori Algorithm, Frequent Itemsets, Temporal Data.

I. INRODUCTION

Data Mining is a process of discovering and knowledge from large amounts of data. One of the most important applications in data mining is analysis of transactional data. Data mining is the process of extracting interesting information or patterns from large information repositories such as: relational database, data warehouses, XML repository, etc. In that, Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to large and larger item sets as long as those item sets appear sufficiently often in the database. Apriori algorithm is a classical algorithm in data mining. It is used for mining frequent itemsets and relevant association rules. Association rules mainly used to support and confidence of the itemset in data repository. For example, a supermarket transactional database, we may have *{*sugar, milk*}* bought together with support of 20%.It means sugar and milk together of all transactions contain of 20%. This association rules are straightly forward to generate frequent itemsets in the repository datasets. Apriori is used for relation to the dataset from the algorithm and prediction subset, then to generate the rules. It is not considering for the time-stamping in apriori. But in my case this time-stamping is one of the major factors of the temporal dataset. For example, transactional database of a supermarket.

Basic concept of frequent items in temporal dataset: In this section to describe the data selection and rule generation which can be used for Filtering process. A data generally from UCI data repository. The data from the dataset which can be used to Filtering the data. By this step the data can be reduced which means filter the frequent items in the data. Usually frequent data can be selected from dataset by data selection method.

The rest of the paper is as follows. In Section II, we discuss some related works in comparison with our work. In Section III, the proposed algorithm is presented in details. The notion of Time-Stamp is described and the essence of a density threshold is illustrated with an example. In Section IV, we present the experimental evaluation of our algorithms using datasets and give analysis on experimental results. In Section V, we conclude the paper with some discussions.

II. RELATED WORKS

The objective of the existing algorithm is to find itemset X in a contiguous subset of database, where the support of X is above the minimum support and the size of the time interval is optimal. Also by introducing density threshold, validity of the rules are ensured (i.e., excluding time intervals which are not densed). Considering time hierarchy, our approach toward discovering frequent itemsets is to first partition the database into many small segments. We use cubes (hypercubes for time hierarchies more than three) to show these segments. Candidates that have support more than the minimum support in at least one TC are considered to be frequent. Neighboring TCs of the same itemsets are merged if they are frequent. In the

following, we present and explain our algorithm to mine frequent itemsets[1].

A relationship between data items in Association Rule (AR). Association rule mining (ARM) is a class of computing techniques used to discover interesting relationships among a set of items by finding itemsets that frequently appear together in transactions [2]. The time attribute is to discover Temporal Association Rules (TARs) in another important utilization. In contrast to the traditional association rule, the temporal association rule adds a time constraint (it can be a point in time or a time range) to the rule to indicate when it holds. The problem is finding of the complete set of class association rules that satisfies their minimum support and their minimum confidence thresholds from a dataset [3]. Given pair of confidence and support thresholds, the problem of mining association rules is to identify all association rules that have confidence and support greater than the corresponding minimum support threshold (denoted as min supp) and minimum confidence threshold(denoted as min conf).

Association rule mining algorithms [2] work in two steps: 1) generate all frequent itemsets that satisfy min_supp and 2) generate all association rules that satisfy min_conf using the frequent itemsets. This problem can be reduced to the problem of finding all frequent itemsets for the same support threshold [4]. The discovery of association relationship among the data in a huge database has been known to be useful in selective marketing, decision analysis, and business management [10, 11]. Association rule mining algorithms [1] work in generate all frequent itemsets that satisfy *min supp*; (2) generate all association rules that satisfy *min conf* using the frequent itemsets. The existing model of the constraint based association rule mining is not able to efficiently handle the time-variant database due to two fundamental problems, i.e., (1) lack of consideration of the *exhibition period* of each individual transaction; (2) lack of an intelligent support counting basis for each item. However, since different transactions have different exhibition periods in a time-variant database, only considering the occurrence count of each item might not lead to interesting mining results [5].

The database storing temporal information can be named as temporal database. Temporal data reflects the development process of things which is of great benefit to uncovering the essence of things evolution. Temporal association rule mining is to discover the valuable relationship among the items in the temporal database. Till now, the research and practise of temporal association rule mining have been popular, and many literatures have reported (H Mannila *et al*, 1997; Jef Wijsen *et al*, 1997; Sridhar Ramaswamy *et al*, 1998; Anthony K. H. Tung *et al*, 1999; Juan M.Ale *et al*, 2000; Sherri K. Harms *et al*, 2002; Sherri K.Harms *et al*, 2004). Here, we propose a new algorithm: T-Apriori based on time constraint and try to look for a new application of temporal association rule mining from a different viewpoint. We develop the concept of sequence of ecological events and apply T-Apriori algorithm to the analysis of ecological phenomena generating and vanishing process. The remainder of the paper is organized as follows. The concepts of association rule mining and temporal association rule mining. The temporal association rule mining methodology and T-Apriori algorithm [6].

The problem of mining association rules was first explored by [1].Many variants of mining association rules are studied to explore more mining capabilities, such as incremental updating[2,3],mining of generalized and multi-level rules[4],mining associations among correlated or infrequent items[5],and temporal association rule discovery[6,7].While these are important results toward enabling the integration of association mining and fast searching algorithms, their mining methods, however cannot be effectively applied to the transaction database where the exhibition periods of the items are different from one to another[7].

Temporal data mining become a core technical data processing in dealing with changeable data recently. Unlike conventional data mining, temporal data mining has its exclusive characteristics. Temporal data mining can be classified into temporal schema and similarity. Temporal schema mining focuses on time-series schema mining, temporal causal relationship mining, and association rules mining. The paper focuses on an algorithm of association rules mining for temporal data [8].

Association rules was first proposed by [9].In has two part, finding frequent itemsets and generating association rules. The major and time consuming part of the algorithm is discovering frequent itemsets and generating association rules is straightforward. In our literature review, we consider association rules the same frequent itemset mining. Many studies have been done to extend association rules in different ways. Classification association rule [10, 11], context-based association rule [13, 14], negative association rule [12], fuzzy association rule [15], generalized association rule [16, 17] are some of the research areas in this field. There are several kinds of meaningful patterns, when time attribute is considered. We aim to review the most relevant researches to our study.

III. MINING FREQUENT ITEMSETS WITH TIME-STAMP

3.1 Proposed Algorithm for mining frequent itemsets

The objective of the proposed algorithm is to find the itemset in a subset of database, where the support of itemset of the dataset is considering their starting date to end date for the time-stamp interval. Also introducing the density of threshold, then the validity of rules are

generated,(i.e. excluding time-stamp which are not densed). To filter the data in particular date filter or preprocess the date of subset instances. And then date attribute to removed from the filtered data. Finally the output in the input of Apriori Algorithm and then the preprocessed data to generate the rules.

In another way of rule generation in my case, using CFS with filtering dataset to removed the attributes from the dataset of ARFF instances. To filter the data in particular time-stamp for the subset of dataset. In that attribute to filter the starting date to end date for the time-stamp interval. To filter the data in particular date filter or preprocess the date of subset instances. And then date attribute to remove from the filtered data. Finally the output in the input of Apriori Algorithm and then the preprocessed data to generate the rules.

Algorithm 3.1: Algorithm for mining frequent itemset with time-stamp

Input:

Dataset in the format of ARFF or CSV or DB Table.

Output:

Generate association rule.

Steps:

1. Read the dataset.
2. for each itemset do
3. Find the timestamp attribute
4. if timestamp is between the starting and ending date then
5. Generate the new dataset
6. Add the itemset into the new dataset
7. end if
8. end for
9. Pass the generated new dataset as an input to the Apriori Algorithm
10. It generates the association rule.

Algorithm3 .2: Algorithm for preprocessing dataset of mining frequent itemset with time-stamp.

Input:

Dataset in the format of ARFF or CSV or DB Table.

Output:

Generate association rule.

Steps:

1. Read the dataset.
2. Filtered dataset=call FSA (dataset)
3. Read the filtered dataset
4. for each itemset do
5. Find the timestamp attribute
6. if timestamp is between the starting and ending date then
7. Include itemset into the new dataset
8. end if
9. end for
10. Pass the new dataset as an input to the Apriori Algorithm
11. Gets the generated association rules.

Algorithm 3.3: Feature Selection Algorithm

Input:

Feature format of FSA (F1, F2, ……., F_k, F_c)

Output:

S_{best}

Steps:

1. begin
2. for i = 1 to k do begin
   r = calculate correcoeff (F_i, F_c);
   end;
3. let $\rho = 0$
4. for i = 1 to k do begin
   t = calculate signi(r, ρ) for F_i ;
   if t > CV
   $S_{best} = S_{list}$;
   end;
5. return S_{best};
6. end

IV. COMPUTATIONAL STUDY

In this section, the proposed algorithm is experimented with chosen data sets. The performance of the algorithm is evaluated with weather data set.

4.1 Nominal Dataset

A weather dataset is chosen by rather than real dataset that controlled the experiment can be validating the efficacy of our approach. A temporal dataset and can be handled by our algorithm. The program takes the starting date and ending date with weather data as inputs and randomly distributes data between the starting and ending dates. Dataset features such as outlook, temperature, humidity, windy, play, and date.

The Apriori algorithm is used to identify the frequent itemset on weather nominal dataset. One of the itemset is selected and generates the new itemset in the dataset.

4.2 Experiment on Result

The dataset is a weather database containing transactions made up by 15 items from 1980-01-01 to 1980-12-01. Then an itemset is chosen as a target solution. The main purpose to behind using such a large dataset for analysis, in our proposed algorithm is capable of handling any frequent itemset for mining problem.

The proposed algorithm is coded by Java Programming language and is run on genuine computer and we expected results on target solution. In our proposed algorithm using java programming and is run in Eclipse environment. And the design purpose we used to JGuiD.

4.2.1: Without Filtering Dataset To Generate Association Rules Result:

The nominal dataset was without preprocessing and their using filtering process for the timestamp range. And then the filtering process to reduce the original dataset using timestamp for date attribute to particularly mention their starting and ending date. Then it filter the data and then added the new dataset. The new dataset is the input of Apriori Algorithm. Finally, the algorithm to generate the association rules.

Apriori
=======
Minimum support: 0.35 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 13

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10

Size of set of large itemsets L(2): 22

Size of set of large itemsets L(3): 9

Best rules found:

1. temperature=mild 3 ==> humidity=high 3 <conf:(1)> lift:(1.75) lev:(0.18) [1] conv:(1.29)
2. humidity=normal 3 ==> temperature=cool 3 <conf:(1)> lift:(2.33) lev:(0.24) [1] conv:(1.71)
3. temperature=cool 3 ==> humidity=normal 3 <conf:(1)> lift:(2.33) lev:(0.24) [1] conv:(1.71)
4. outlook=overcast 2 ==> play=yes 2 <conf:(1)> lift:(1.75) lev:(0.12) [0] conv:(0.86)
5. outlook=rainy humidity=high 2 ==> temperature=mild 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)
6. outlook=rainy temperature=mild 2 ==> humidity=high 2 <conf:(1)> lift:(1.75) lev:(0.12) [0] conv:(0.86)
7. outlook=rainy humidity=normal 2 ==> temperature=cool 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)
8. outlook=rainy temperature=cool 2 ==> humidity=normal 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)
9. windy=TRUE play=no 2 ==> outlook=rainy 2 <conf:(1)> lift:(1.75) lev:(0.12) [0] conv:(0.86)
10. outlook=rainy play=no 2 ==> windy=TRUE 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)

4.2.2: With Filtering Dataset To Generate Association Rules Result:

The nominal dataset was with preprocessing and their using correlation based feature selection to filtering dataset. And then the filtering dataset using timestamp for date attribute to particularly mention their starting and ending date. Then it filter the data and then added a new dataset. The new dataset is the input of Apriori Algorithm. Finally, the algorithm to generate the association rules.

Apriori
=======
Minimum support: 0.35 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 13

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8

Size of set of large itemsets L(2): 13

Size of set of large itemsets L(3): 5

Best rules found:

1. temperature=mild 3 ==> humidity=high 3 <conf:(1)> lift:(1.75) lev:(0.18) [1] conv:(1.29)
2. humidity=normal 3 ==> temperature=cool 3 <conf:(1)> lift:(2.33) lev:(0.24) [1] conv:(1.71)
3. temperature=cool 3 ==> humidity=normal 3 <conf:(1)> lift:(2.33) lev:(0.24) [1] conv:(1.71)
4. temperature=mild windy=FALSE 2 ==> humidity=high 2 <conf:(1)> lift:(1.75) lev:(0.12) [0] conv:(0.86)
5. humidity=high play=no 2 ==> temperature=mild 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)
6. temperature=mild play=no 2 ==> humidity=high 2 <conf:(1)> lift:(1.75) lev:(0.12) [0] conv:(0.86)
7. humidity=normal windy=TRUE 2 ==> temperature=cool 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)
8. temperature=cool windy=TRUE 2 ==> humidity=normal 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)
9. humidity=normal play=yes 2 ==> temperature=cool 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)
10. temperature=cool play=yes 2 ==> humidity=normal 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)

Comparison between the performance of the algorithm with and without filtering the dataset to generate the association rules. The proposed method generates the best rules to find the targeted solution. CFS based filtering proves to be the best way to generate association rules.

Table 4.1 Different Approach Dataset with Result

Sl. No	Dataset	Approach	No of Itemsets	Result (Generated Rules)
1.	Weather Nominal Dataset	Nominal Dataset without Filtering process	15	Best rules found are: 1.temperature=mild3==>humidity=high 3 <conf:(1)> lift:(1.75)lev:(0.18)[1]conv:(1.29) 2.humidity=normal3==>temperature=cool3 <conf:(1)>lift:(2.33)lev:(0.24) [1] conv:(1.71) 3.temperature=cool 3 ==> humidity=normal 3<conf:(1)>lift:(2.33)lev:(0.24)[1]conv:(1.71) 4.outlook=overcast 2 ==> play=yes 2 <conf:(1)>lift:(1.75)lev:(0.12) [0] conv:(0.86) 5.outlook=rainyhumidity=high 2 ==> temperature=mild2<conf:(1)>lift:(2.33) lev:(0.16) [1] conv:(1.14) 6.outlook=rainy temperature=mild 2 ==> humidity=high2<conf:(1)>lift:(1.75) lev:(0.12) [0] conv:(0.86) 7.outlook=rainy humidity=normal 2 ==> temperature=cool2<conf:(1)>lift:(2.33) lev:(0.16) [1] conv:(1.14) 8.outlook=rainy temperature=cool 2 ==> humidity=normal2<conf:(1)>lift:(2.33) lev:(0.16) [1] conv:(1.14) 9.windy=TRUE play=no 2 ==> outlook=rainy2<conf:(1)>lift:(1.75) lev:(0.12) [0] conv:(0.86) 10.outlook=rainy play=no 2 ==> windy=TRUE2<conf:(1)>lift:(2.33) lev:(0.16) [1] conv:(1.14)
2.	Weather Nominal Dataset	Nominal Dataset with filtering process	15	Best rules found are: 1. temperature=mild 3 ==> humidity=high 3 <conf:(1)> lift:(1.75) lev:(0.18) [1] conv:(1.29) 2.humidity=normal 3 ==> temperature=cool 3<conf:(1)> lift:(2.33) lev:(0.24) [1] conv:(1.71) 3.temperature=cool 3 ==> humidity=normal 3<conf:(1)> lift:(2.33) lev:(0.24) [1] conv:(1.71) 4.temperature=mild windy=FALSE 2 ==> humidity=high2 <conf:(1)> lift:(1.75) lev:(0.12) [0] conv:(0.86) 5.humidity=high play=no 2 ==> temperature=mild 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14) 6.temperature=mild play=no 2 ==> humidity=high 2 <conf:(1)> lift:(1.75) lev:(0.12) [0] conv:(0.86) 7.humidity=normal windy=TRUE 2 ==> temperature=cool 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14) 8.temperature=cool windy=TRUE 2 ==> humidity=normal 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14) 9.humidity=normal play=yes 2 ==> temperature=cool 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14) 10.temperature=cool play=yes 2 ==> humidity=normal 2 <conf:(1)> lift:(2.33) lev:(0.16) [1] conv:(1.14)

V. CONCLUSION

In this paper, we proposed an algorithm for mining frequent itemsets from the temporal dataset. It extracts the frequent itemsets those are in between the timestamp interval. In our algorithm comparison between the performance of the algorithm with and without filtering the dataset to generate the association rules. The proposed method generates the rules to find the solution. CFS based filtering proves to be the best way to generate association rules. We enhanced the algorithm using the preprocessing technique namely Correlation-based Feature Selection. The proposed method yields better results compared to existing methods. Mining outcome helps in making make better decisions.

As a future research direction, the proposed algorithm can be augmented with information Gain based filter method.

REERENCES

[1]. Mazaher Ghorbani and Masound Abessi,"A New Methdology for Mining Frequent itemsets on Temporal Data,"IEEE Transactions on Engineering Management.,vol:64,issue:4,Nov.2017.

[2]. Y. Xiao, Y. Tian, and Q. Zhao, "Optimizing frequent time-window selection for association rules mining in a temporal database using a variable neighbourhood search," *Comput. Oper.Res.*, vol. 52, pp. 241–250, 2014.

[3]. D. Nguyen, B. Vo, and B. Le, "CCAR: An efficient method for mining class association rules with itemset constraints," *Eng. Appl. Artif. Intell.*,vol. 37, pp. 115–124, 2015.

[4]. C.-H. Lee, M.-S. Chen, and C.-R. Lin, "Progressive partition miner: An efficient algorithm for mining general temporal association rules," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 1004–1017, Jul./Aug. 2003.

[5]. C.-H. Lee, J. C. Ou, and M.-S. Chen, "Progressive weighted miner: An efficient method for time-constraint mining," in *Advances in Knowledge Discovery and Data Mining*. New York, NY, USA: Springer, 2003,pp. 449–460.

[6]. Temporal Association Rule Mining Based On T-Apriori Algorithm And Its Typical Application.

[7]. Efficient Algorithm for Mining Temporal Association Rule, vol 7, no 4, April 2007.

[8]. Temporal Association Rules in Mining Method, This paper to survey content.

[9]. R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993.

[10]. B. Liu,W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proc. 4th Int. Conf. Knowl. Discovery Data Mining*, 1998, pp. 80–86.

[11]. D. Nguyen, B. Vo, and B. Le, "CCAR: An efficient method for mining class association rules with itemset constraints," *Eng. Appl. Artif. Intell.*,vol. 37, pp. 115–124, 2015.

[12]. X. Wu, C. Zhang, and S. Zhang, "Efficient mining of both positive and negative association rules," *ACM Trans. Inf. Syst.*, vol. 22, no. 3,pp. 381–405, 2004.

[13]. Y.-L. Chen, K. Tang, R.-J. Shen, and Y.-H. Hu, "Market basket analysis in a multiple store environment," *Dec. Support Syst.*, vol. 40, no. 2, pp. 339–354, 2005.

[14]. M. Shaheen, M. Shahbaz, and A. Guergachi, "Context based positive and negative spatio temporal association rule mining," *Knowl.-Based Syst.*,vol. 37, pp. 261–273, 2013.

[15]. Y.-L. Chen and C.-H. Weng, "Mining fuzzy association rules from questionnaire data," *Knowl.-Based Syst.*, vol. 22, no. 1, pp. 46–56, 2009.

[16]. J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in *Proc. 21st Int. Conf. Very Large Data Bases*, 1995,pp. 420–431.

[17]. F. Benites and E. Sapozhnikova, "Hierarchical interestingness measures for association rules with generalization on both antecedent and consequent sides," *Pattern Recognit. Lett.*, vol. 65, pp. 197–203, 2015

[18]. Krutika. K .Jain , Anjali . B. Raut "Review paper on finding Association rule using Apriori Algorithm in Data mining for finding frequent pattern", vol 3, issue 1, 2015.

Authors Profile

Ms. B. Sowndarya received the UG degree in G.T.N Arts College, Dindigul from Madurai Kamaraj University in 2015, and the PG degree in NPR Arts and Science College, Natham, Dindigul from Madurai Kamaraj University in 2017. Currently working toword the M.phil degree in Alagappa University, Karaikudi.

Dr. T. Meyyappan M.Sc, M.Tech., M.B.A., M.Phil, Ph.D. currently, Professor, Department of Computer Science, Alagapp University, Karaikudi, TamilNadu. He has organized conferences, workshops at national and international levels. He has published 90 numbers of research papers in National, International journals and conferences. He has developed Software packages for Examination, Admission Processing and official Website of Alagappa University. As a Co-Investigator, he has completed Rs.1 crore project on smart and secure environment funded by NTRO, New Delhi. As principal Investigator, he has completed Rs. 4 lakhs project on Privacy Preserving Data Mining funded by U.G.C. New Delhi. He has been honoured with Best Citizens of India Award 2012 research areas include Operational Research, Digital Image Processing, Fault Tolerant computing, Network security and Data Mining.

Dr. SM. Thamarai currently, guest lecturer, Alagappa Government Arts College, Karaikudi, received her Diploma in Electronics and Coomunication Engineering, Department of Technical Education, Tamilnadu in 1989 and her B.C.A. M.Sc. (University First Rank holder and Gold medalist), M.Phil. (First Rank holder) degrees in Computer Science(1998-2005) from Alagappa University. She has published 27 research papers in International, National Journals and conferences. She received her Ph.D. degree in Computer Science in 2014. Her current research interests include Operational Research and Fault Tolerant Computing.

