# EVALUATION ON DIFFERENT APPROACHES OF CYCLIC ASSOCIATION RULES

# Khushali Shah<sup>1</sup>, Mahesh Panchal<sup>2</sup>

Computer Science Department, Gujarat Technological university, Khushi\_4121989@yahoo.in, mkhpanchal@gmail.com

#### Abstract

Now a day's companies have large amount of data its exploration becomes complicated, especially if we emphasize the temporal aspect while considering association rules. Therefore, we are introducing the temporal association rules mining. In this paper, we focus on cyclic association rules, classified as a category of the temporal association rules. An association rule is cyclic if the rule has the minimum confidence and support at regular time intervals. Such a rule need not hold for the entire transactional Database, but only for a particular Periodic time interval. The cyclic association rules used to discover relations among items characterized by regular cyclic variation overt time. For finding cyclic association rules we are going through various different approaches and their comparative analysis. This Techniques will enable marketers to better Identify trends in sales and allow for better forecasting of Future demand. comparative study of different approaches are also provided.

Keywords— Data mining, cyclic association rule, association rules; frequent item set, constraints, partition, cycle length

#### **1. INTRODUCTION**

Different Organizations are collecting several kinds of data. This result in bulk of data might contain usual information, and since it is will take much more time for manual analysis, To short out this different algorithms for discovering potential useful information are developed. One of the major tasks in this area of data mining is association rule mining. Several efforts were dedicated to the coupling between the temporal aspect and the association rules by introducing temporal association rules .



Fig 1 different task of data mining

To solve the temporal association rules issue, five streams of approaches are identified:

• **Interval Association Rules:** involve discovering of Association rules during a time interval 1 related to each Item.

• Calendric Association Rules: Ramaswamy et al have introduced the calendric association rule. Then, Li et al generalized the already proposed model. The Calendric Association Rules offer a generic model of calendric association rules extraction based on calendar algebra expression. In fact, the latter is a collection of time intervals describing some real-life phenomenon. By integrating calendars and association rules, we can generate discovered patterns. For example, we buy cars equipped with air conditioning in summer: Cars => Air conditioning in summer. And we deduce that there are more accidents and victims on weekends: Accidents => Victims on Weekends;

• **Temporal Predicate Association Rules:** extend the association rule model by adding to the rules a conjunction of binary temporal predicates that specify the relationships between the time stamps of transactions. By finding associated items first and then looking for temporal relationships between them, it is possible to incorporate potentially valuable temporal semantics. The approach of temporal reasoning accommodates both point-based and interval-based models of time simultaneously. For example, we buy milk and butter in a given time-interval: Milk => Butter between 19 o'clock and 23 o'clock. And we buy bread and juice at a given time: Bread => Juice at nine o'clock;

• Sequential Association Rules: dedicated to the extraction of association rules from data represented as sequences. Indeed, a sequence contains sorted sets of items ordered by transaction time. For example, association rule mining does not take the time stamp into account, the rule can be Buy A => Buy B. If we take time stamp into account then we can get more accurate and useful rules likewise: Buy A implies Buy B within a week, or usually people Buy A every week;

• **Cyclic association rules :** aim to discover association rules from articles having regular cyclic variation over time.

In this paper, we focus on the last category: Cyclic Association Rules. The remainder of the paper is organized as follows. We scrutinize, in Section 2, the related work. We define the concepts of approach. We illustrate our approach through an algorithm in Section 3. Finally we conclude comparison of this approaches and sketching future research issues.

# 2. RELATED WORK

Objective of data mining is to mine meaningful knowledge from large databases using patterns like association rules. Cyclic rules are association rules that have minimum support and confidence at regular time intervals. In order to discover such rules, it is necessary to search for frequent patterns in a constrained of time, since they may occur repeatedly at specific regular time instants but on a small portion of the global time considered. A method to determine such rules is based on the application of any algorithm to obtain the set of traditional rules, and then to detect the cycles behind the rules An itemset has a cycle c, if the itemset occurs in every convex time interval of c. In this manner, an itemset is said to be a large cyclic itemset if its support in some cycles at least equal to the minimum support required by exploiting the relationship between Cycles and large itemsets. In order to deal with cyclic association rules, we enhance the transaction model by a time attribute that describes the time when the transaction was executed. Mining cyclic association rules (CAR) aims at discovering knowledge in the form of rules from data characterized by regular variation over time.

Additionally, if we generate the CAR from monthly sales data, we uncover the seasonal variation as certain rules are true, approximately, the same month each year. So general concept behind cyclic association rule is to extract Correlations between products which vary in a cyclic way.

The databases, from which are drawn the cyclic patterns, contain three data closely related to the market basket analysis: the first is the client identifier customerID, the second is the list of products and the third represents the date when this customer bought this product.

Our contribution focuses on cyclic association rules to discover associations which are cyclically repeated to reduce the number of generated cyclic rules, with an improvement of the runtime.

#### **3. THE FUNDAMENTAL CONCEPT**

• Cycle: A cycle c is a tuple (1, 0), such that 1 is the length cycle, being multiples of the unit of time; o is an offset designating the first time unit where the cycle appeared Thus, we conclude that 0<0 < 1.

For Example If we considered a length of cycle l = 2 and the corresponding offset is 1. So that, the cycle c = (l, 0) = (2, 1).

- Support: The support of an itemset X is the fraction of transactions that contain the itemset.
- *confidence* : The rule X->Y is the fraction of transactions containing X that also contain Y:
- Time Unit: Given a transactional database DB, each time unit Ui corresponds to the time scale on the database.
- Frequent Cyclic itemset: This concept refers to cyclic itemsets having supports exceeding the considered threshold. The itemset XY is considered as **Frequent Cyclic** denoted **FC** if the cyclic occurrences of the itemset XY are greater or equal to the given support threshold otherwise if sup (XY) >=*MinSup*.
- Partition and its length: A partition, noted Pi, is an adjacent subset of transactions of the original database. Its length, noted Ipil, is the number of partial transactions specific to a partition and it is set initially.

# 4. DISCOVERING CYCLIC ASSOCIATION

## RULES

Given a set of items and a set of transactions, where each transaction consists of a subset of the set of items, the problem of discovering association rules is defined as finding relationships between the occurrences of items within transactions.

An association rule of the form X->Y is a relationship between the two disjoint itemset and (an itemset is a set of items). An association rule is described in terms of support and confidence.

In order to deal with cyclic association rules, we enhance the transaction model by a time attribute that describes the time when the transaction was executed.

In this paper, we assume that a unit of time is given (e.g., by the user). We denote the  $i^{th}$  time unit i>=i 0; by  $t_i$ . That is  $t_i$ corresponds to the time interval [i\*t, (i+1)\*t]; where  $t_i$  t is B the unit of time. We denote the set of transactions executed in  $t_i$  ti by: [D<sub>i</sub>] we define the problem of discovering cyclic association rules as finding cyclic relationships between the presences of items within transactions. We define the problem of discovering cyclic association rules as finding cyclic relationships between the presences of items within transactions.

For example, if the unit of time is an hour and "coffee-> doughnuts" holds during the interval 7AM-8AM every day (i.e., every 24 hours), then "coffee-> doughnuts" has a cycle (24, 7). An association rule may have multiple cycles.

For example, if the unit of time is an hour and "coffee-> doughnuts" holds during the interval 7AM-8AM and 4PM-5PM every day (i.e., every 24 hours), then "coffee-> doughnuts" has two cycles  $C_1=((24;, 7))$  and  $C_{2=}(24; 16)$ : An association rule can be represented as a binary sequence where the ones correspond to the time units in which the rule holds and the zeros correspond to the time units in which the rule does not have the minimum confidence or support. For instance, if the binary sequence 001100010101 represents the X->Y then X->Y holds association rule in D[2],D[3],D[7],D[9],D[11]

## **5. DIFFERENT APPROACHES**

#### A. THE SEQUENTIAL APPROACH:

The sequential algorithm performs association Rule mining and cycle detection independently. The sequential algorithm be composed of two phases:

(i) The generation of association rules for each unit of time, once the rules in all the time units have been discovered, (ii) detection of cycles. In this procedure, each association rule is represented by a binary sequence with '1' corresponding to the time units in which the rule is generated and '0' else. In order to optimize the operation of this algorithm, the approach of interleaved algorithm was introduced.

For example, the binary sequence 001100010101 represents the association rule that holds in D2, D3, D7, D9, and D11. Given a binary sequence and the maximum cycle length of interest m, a procedure of detecting cycles consists of two steps.

In the first step, the sequence is scanned. Each time '0' is encountered at a position i, candidate cycles  $c = (j, i \mod j), 1$ <= j <=m are eliminated from the set of candidate cycles. If the maximum cycle length of interest is m, the set of candidate cycles contains all possible cycles c(1, 0), where 1 <= 1 <= m, 1<= o <= 1. Let m = 4, then the set of candidate cycles is {c(1, 0),  $c(2, 0), c(2, 1), \cdots, c(4, 0), \ldots c(4, 3)$ }. Given a sequence 001100010101, a digit '0' is found at position 0, thus cycles c(1, 0), c(2, 0), c(3, 0) and c(4, 0) are eliminated from the set of candidate cycles. This step is completed whenever the last bit of the sequence is scanned or the set of candidate cycles becomes empty, whichever is first. From the example, after the eleventh bit is scanned, the only remaining cycle is c(4, 3), representing a rule that holds in every fourth time unit starting from the time unit u3, followed by u7 and u11.

In the second step, large cycles are detected from the set of remaining cycles by eliminating cycles that are not large, that is, starting from the shortest cycle,

#### **B. THE INTERLEAVED APPROACH**

A more efficient approach to discover cyclic rules consists on inverting the process: first discover the cyclic patterns and then generate the rules. This algorithm is known as the *interleaved algorithm*.

It is apriori-based algorithms, it has two steps to follow: in the first step, it discovers the set of cyclic large itemsets (*cyclic patterns*) and on the second step, it discovers the cyclic rules. Since cyclic rules can be generated using the cycles and the support of itemsets, a simple extension of *gen\_rules* algorithm is enough The discovery of cyclic large itemsets is far more interesting, since it explores three new techniques: *cycle-pruning*, *cycle-skipping* and *cycle-elimination*.

Note that the time unit corresponds to a unitary time interval (intervals with a unitary time unit, like a day, a month or a week), and the cycle itself corresponds to a set of consecutive time units, which is itself a time interval with holes.

An itemset has a cycle c, if the itemset occurs in every convex time interval of c. In this manner, an itemset is said to be a large cyclic itemset if its support in some cycle is at least equal to the minimum support required.

Note that a cycle c is multiple of another cycle c' if c' is contained in c. additionally, consider a time segment D[i] as the portion of the dataset that occurs in the time interval ti.

Like apriori, the algorithm works in a candidate generation and test philosophy, generating the k-candidates based on the k-1 cyclic large itemsets. First, it considers as 1-candidates all single items over all possible time intervals, and determines 1-Cyclic large itemsets by counting their support and discarding the infrequent ones.

However this support counting can be enhanced by using cycleskipping and cycle elimination.

**Cycle-skipping** avoids counting the support for an itemset in time intervals that do not belong to the cycle of the itemset, since if a time interval  $t_i$  is not part of a cycle of an itemset, then there is no need to calculate its support in the corresponding time segment D[i]. On the other hand, if an itemset is not frequent in some time segment D[i], then it cannot have cycles defined over that time segment. This property allows for the reduction of the cycles that an itemset could have, and it is applied by **cycle elimination**.

After identifying 1-cyclic large itemsets, k-candidates are generated as in apriori, and using a third technique – *cycle*-*pruning*, which is based on the property that states

if an itemset has a cycle, then any of its subsets has the same cycle. In this manner, k candidates are just the itemsets generated by joining (k-1)-large itemsets over the cycles that are multiples of the cycles of any of its subsets. In this manner, the number of cycles of an itemset is less than or equal to the number of cycles of any of its subset.

#### The Interleaved Algorithm

Input: Database DB, MinSup, MinCon Ouput: set of cyclic association rules For each k; k > 1: 1. If k = 1; then all possible cycles are initially assumed to exist for each single itemset. Otherwise (if k > 1), cycle-pruning is applied to generate the potential cycles for k-itemsets using the cycles for (k - 1)-itemsets. 2. For each time unit  $t_i$ : 2.1 Cycle-skipping determines, from the set of candidates cycles for kitemsets, the set of k-itemsets for which support will be calculated in time segment D[i]. 2.2 If a *k*-itemset *X* chosen in Step 2.1 does not have the minimum support in time segment D[i]; then cycle-elimination is used to discard each cycle C, for which time unit  $t_i$  holds, from the set of potential

cycles of X. The evaluation of PCAR by its comparison with the SEOUENTIAL and the INTERLEVED algorithms as original

SEQUENTIAL and the INTERLEVED algorithms as original approaches dedicated to generate cyclic association rules.

# C. THE PCAR APPROACH

The PCAR algorithm is mainly premised on the mechanism of segmentation of the original database according to the usespecified number of partitions. Using an incremental side, the scan will be done step by step. Then, we scan the partitions one by one to extract all the frequent itemsets and generate the temporal cyclic association rules.

For Example If table contain 8 transaction and if we divide the table on 2 partitions. So, the length of the partitions is  $Ip_i l= Ip1l=Ip2I = 4$ .

P1 contains the first, the second, the third and the fourth transaction. And p2 is composed from the remainder of transactions otherwise the fifth, the sixth, the seventh and the eighth transaction.

After partitioning the database, we browse it partition by partition, each partition is scanned to generate the frequent cyclic itemsets. Obviously, the given partition i will be scanned only once, the next partition i + 1 will use the frequent cyclic item sets already generated in the partition i hence we emphasize the incremental side of our model. Based on frequent cyclic item sets, we extract the corresponding cyclic association rules as a final step.



#### Algorithm PCAR

Input: Database 'DB, Length Cycle, NbPartition, MinSup, MinCon! Ouput: set of cyclic association rules SetFrequentItemSet: = 0;for i = I to NbPartition do for j = I to IDBIINbPartition do j: given transaction For I = I to j do II I: Given itemset in a transaction j IlVerfify the itemset is cyclic and frequent if (Supp(I) > (MinSup \* Iii) and IsCyclic(I) then IsCyclic (I): Verify in the occurrence of I is Cyclic given the length Cycle if I E SetFrequentItemSet then Occurrence (I):= Occurrence (I) + j; Else SetFrequentItemS et SetFrequentItemSet U {I}; end if end if end for i := i + 1;end for i := i + 1: end for // 2. Generate set of cyclic rules SetRule: =0: For i = I to SetFrequentItemSet do R: = GenerateSetRule (I); II Generate the set of rules from an itemset I if conf(R) > MinCan! then II Add the rule to the set of cyclic ones SetRule: = setRule U R: End if

The drawbacks of this algorithm are that it generates rules which are not meeting the expert's expectations. To reduce the number of generated rules, we can use constraint-based association rules algorithms In which rules have to follow user defined constraints.

#### **D. THE CBCAR ALGORITHM**

The problem of mining constraint-based association rules aims to guide the search process and reduce the overall set of generated association rules. The enforced constraints may concern the content of rule, its premise or its consequence.

The constraint is a condition on the different values that can be taken by the itemsets. The integration of constraints in the rule is an important input for researchers to reduce the number of generated rules.

In the following section, we will present the main approaches of constraint-based association rules mining:

**Selected Items approach**: The main idea of this approach is to generate a set of selected items, denoted by S each itemset which satisfies the constraint B contains, necessarily, an item in S. S is generated by taking a combination of each element.

**The Direct approach:** The Direct algorithm does not go through the step of generation of the set of selected items, S. It performs the extraction of frequent itemsets directly satisfying the constraint B.

Algorithm: CBCAR: Constraint-Based Cyclic Association Rules

**Data**: DB, Minsupp, Minconf, FCT, dc, Q, PRM, CL, N, NB, L **Result**: R: all cyclic association rules extracted from cyclic stress DB.

```
Begin
For i=1 to NB do
For j=1 to |BD|/ NB do
For I=1 to j do
I: itemset in a transaction j
If (supp (I) > Minsupp * |i|) and Is Cyclic (I, L) then
If I C Fi then
Occurrence (I): = Occurrence (I) + j;
Else
Fi: = Fi U \{i\};
End if
End if
End for
i := i+1;
End for
i:=i+1;
End for
F1= find 1-frequent itemset in DB;
For (K=2; K≠Ø; K++) do
Ck = \{Generation-candidate (Ck-1); Ck \in PRM or Ck \in CL\}
If Ck is a multidimensional itemset && Is Cyclic (Ck, L) then
For each transaction T \in BD do
Ct = Subset (Ck. T)
For each candidate c \in Ct do
c.count = CalculSupport (BD, c)
If N > 1 // checking the Multidimensionality
If Q > 1 //verify quantitative
```

Fk { $c \in Ck$ , c.count > Minsupp} FCT (dc) Else  $Fk = \{c \in Ck-1, c.count > Minsupp\}$ Return Fk=UK Fk: For each subset s de Fi do If  $(s \in PRM)$  &&  $((Fi-s) \in CL)$  $r = s \rightarrow (Fi-s);$ If (confidence (R) > Minconf) && (Verify (metric (R)) R=R U r; **Return R:** End if End if End if End if End

#### E. THE IUPCAR ALGORITHM

The cyclic association rules have been introduced in order to discover rules from items characterized by their regular variation over time. In real life situations, temporal databases are often appended or updated. Rescanning the whole database every time is highly expensive while existing incremental mining techniques can efficiently solve such a problem.

The IUPCAR algorithm consists of three phases:- In the first phase, a scan of the initial database is done to class the founded itemsets on three classes namely the frequent cyclic itemsets, the frequent pseudo-cyclic itemsets and non frequent cyclic item sets.

- In the second phase, according to the second database, we categorize the itemsets into the three mentioned classes. Then, depending of the ancient class of the itemset with its ancient support and the new class with its new support in the increment database, an affectation of the suitable class is made according to a weighting model.

- In the final phase, given the founded frequent cyclic itemsets after the update operation, the corresponding cyclic association rules are generated.



**Table 1**: comaparative study on cyclic association rule method

Algorithm	Sequential	Interleaved	PCAR
Criteria			
Phases	3	2	3
Extended from Apriori	Yes	Yes	yes
Evaluation metrix	Support Confidence Cycle length	Support Confidence Cycle length	Support Confidence Cycle length No of partition
Runtime Performance	Less efficient	Better than sequential	The most efficient

# CONCLUSIONS

We have studied the problem of discovering cyclic association rules that display regular cyclic variation over time. Information about such variations will allow marketers to better identify trends in association rules and help better forecasting. By exploiting the relationship between cycles and large itemsets, we identified optimization techniques that allow us to minimize the amount of wasted work performed during the data mining process.

Other avenues for future work mainly address the following issues: (i) we can extend this concept in order to detect cyclic sequences

# REFERENCES

[1]Wafa Tebourski Wahiba Ben Abdessalem Karâa"Cyclic Association Rules Mining under Constraints"*International Journal of Computer Applications (0975 – 8887) Volume 49– No.20, July 2012* 

[2]Eya Ben Ahmed1 and Mohamed Salah Gouider"Towards an incremental maintenance of cyclic association rules"International Journal of Database Management Systems ( IJDMS ), Vol.2, No.4, November 2010

[3] E. Ben Ahmed and M.S. Gouider, (2010) "Towards a new mechanism of extracting cyclic

association rules based on partition aspect" Fourth International Conference on Research

Challenges in Information Science, RCIS, Nice, France.

[4] E.Ben Ahmed and M.S.Gouider, "Towards a new mechanism of extracting

cyclic association rules based on partition aspect", *IEEE International Conference on RCIS*, pp 69–78,2010.

[5] Eya Ben Ahmed, Ahlem Nabli, Faiez Gargouri "New Algorithm for Cyclic Mining in Temporal Databases" Progress in Computing Applications Volume 1 Number 1 March 2012
[6] Cláudia Antunes "Temporal Pattern Mining Using a Time Ontology" Instituto Superior Técnico / Technical University of Lisbon Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal

[7] B. Ozden, S. Ramaswamy and A. Silberschatz, (1998), Cyclic Association Rules, 14<sup>th</sup> International Conference on Data Engineering (ICDE'98),412, 16

[8] Sridhar Ramaswamy Sameer Mahajan Avi Silberschatz" On the Discovery of Interesting Patterns in Association Rules" Proceedings of the 24th VLDB Conference

New York, USA, 1998

[9] V. SRINIVASAN M.ARUNA "Mining Association Rules to Discover Calendar Based Temporal Classification" International Conference on Computing, Communication and Networking (ICCCN 2008)978-1-4244-3595-1/08/\$25.00 © 2008 IEEE

[18] Chiang, D., Wang, C., Chen, S., Chen, C. (2009). *The Cyclic Model Analysis on Sequential Patterns, IEEE Trans. on Knowl. and Data Eng.*, 21 (11) November 1617-1628,USA.

[19] Dong, G., Han, J., Lam, J., Pei, J., Wang, K., Zou, W. (2004). *Mining Constrained Gradients in Large Databases, IEEE Trans. on Knowl. and Data Eng.*, 16.

[20] Han, J., Gong, W., Yin, Y. (1998). Mining Segment-Wise Periodic Patterns in Time-Related Databases, KDD, 214-218. *and Data Eng.*, 21 (11) November 1617-1628,USA.[10] The

Cyclic Model Analysis on Sequential Patterns. Ding-An Chiang, Cheng-Tzu Wang, Shao-Ping Chen, and Chun-Chi Chen IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 21, NO. 11, NOVEMBER 2009

[11] Similarity-Profiled Temporal Association MiningJin
Soung Yoo, Member, IEEE, and Shashi Shekhar, Fellow, IEEE
IEEE TRANSACTIONS ON KNOWLEDGE AND DATA
ENGINEERING, VOL. 21, NO. 8, AUGUST 2009
[12] Parallel Algorithms for Discovery of Association Rules
Data Mining and Knowledge Discovery, 1, 343–373 (1997)
°c 1997 Kluwer Academic Publishers. Manufactured in The
Netherlands.

[13] Lu-Na Byon, Jeong-Hye Han, Fast algorithms for temporal association rules in a large database, Key engineering materials , 277-279,287-292, 15 Jan 2007.

[14] J. Ben Schafer, The Application of Data-Mining to Recommender Systems, Infomration System, 45-50, United States of America , 2009.

[15] Discover Relaxed Periodicity In Temporal Databases Tang Changjie Yu Zhonghua Zhang Tianqing Sichuan University[16] Yingjiu. L, P. Ning, X. Sean Wang, S. Jajodia,

Discovering Calendar-based Temporal Association Rules, In Proc. of the 8<sup>th</sup> Int'l Symposium on Temporal Representation and Reasoning, 111-118,2001.

[17] Cheng-Yue Chang, Ming-Syan Chen, Chang-Hung Lee, Mining General Temporal Association Rules for Items with Different Exhibition Periods, Proceedings of the 2002 IEEE International Conference on Data Mining, 59, 2002.