Analysis and Implementation some of Data Mining Algorithms by Collecting Algorithm based on Simple Association Rules

Nadia Moqbel Hassan
Assistant Lecturer
University of Al-Mustansiriya
College of Engineering
Computers & Software Engineering Department

ABSTRACT

Association analysis is utilized to detect the learning and set up tenets from a huge dataset. The minimum support value in the association investigation is a discriminating element to influence the execution of this detection. Association rule mining represent to a data mining method and its objective is to discover intriguing association or correlation relationships among a huge set of data elements. In this paper new algorithm has been proposed which to collecting the (Sample Association Rules) taken from (Basic Apriori Algorithm) with the (Multiple Minimum Support utilizing Maximum Constraints Algorithms). The algorithm is executed, and is compared with its other algorithms, using a new proposed comparison algorithm. Comparisons have been on various groups of data. Consequences of applying the proposed algorithm indicate speedier implementation than different algorithms. At the end, both of execution and results shows: Effortlessness, exactness, and velocity to new algorithm, as well as reliability of the another algorithms.

General Terms

Data Mining, Simple Association Rules.

Keywords

Rule Mining, A priori Algorithm, Knowledge discovery, Minimum Support.

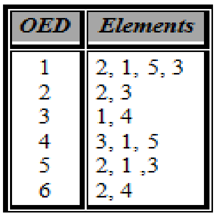
1. INTRODUCTION

Association analysis, too named market-basket analysis, portrays the co-event among data elements in a big size of provided data set. In the association analysis, the minimum support value specifically influences the value of the mining out rules in the association analysis. A bigger minimum support value is viewed, the less essential information is removed, bad habit verso, a little minimum support inclines toward, and a lot unnecessary rules are indicated. Data mining guarantees to make life less demanding for business chiefs and analysts. Data mining can be seen as the investigation and examination of big amounts of data keeping in mind the end goal to find important styles and rules. Data mining is the operation of finding fascinating styles from huge sizes of data. As a learning disclosure process, it regularly includes data purification, data coordination, data choice, data conversion, style detection, style assessment, and learning presentation. A data repository is a depot for long stockpiling of data from numerous sources, composed in order to encourage administration choice making. The data are put away under a consolidated planner. In data mining, the data is stocked electronically and the search is automated or at least increased by computer. [1,2,3,4]. Association rules are a standout amongst the most explored ranges of data mining and have as of late gotten much consideration from the database society. Association rule mining tries to find associations among operands encoded in a database. An association rule picks the shape $X\rightarrow Y$ where X (the precursor) and Y (the resulting) is sets of predicates. Association rule mining involves the notions of support and certainty to Specify rules that are especially intriguing or startling [5,6,2].

Association analysis has been extensively utilized as a part of numerous application areas. One of the best known is the business field where the detect of procurement styles or associations between elements is extremely valuable for choice making and for successful promoting. In the most recent years the application ranges have expanded fundamentally. A few samples of modern applications are discovering styles in biological databases, extraction of learning from software engineering measurements or getting clients profiles for web framework personalization [7,8,9,10].

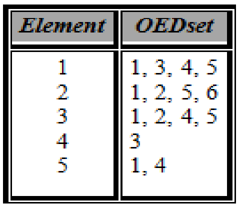
Case in point (1): Let E = {e1, e2, . . . , e m} be an arrangement of elements and let D be a database having set of operands where every operand O is a subset of E. An association principle is an association relationship of the type: $\rightarrow B$, where $A\subset E$, $Y\subset E$, and $A\cap B=\emptyset$. The support of rule $A\rightarrow B$ is known, as the percentage of operands containing both A and B in D. The confidence of $A\rightarrow B$ is known as the percentage of operands containing A that too contain B in D. The mission of association rules mining is to locate all powerful association rules that fulfill minimum support threshold (min-supp) and a minimum confidence threshold (min-conf). Mining association rules comprises of two stages, Table (1) and Table (2) explains the two stages.

Table 1.
Horizontal Representation



OED	Elements
1	2, 1, 5, 3
2	2, 3
3	1, 4
4	3, 1, 5
5	2, 1 ,3
6	2, 4

Table 2.
Vertical Representation



Element	OEDset
1	1, 3, 4, 5
2	1, 2, 5, 6
3	1, 2, 4, 5
4	3
5	1, 4

In the first stage, all repeat elementsets that fulfill the min-supp are found. In the second stage, powerful association rules are produced from the repeat elementsets found in the first stage. Most research considers just the first stage on the grounds that once repeat elementsets are discovered; mining association rules is trifling [11]. Rules blast that outcomes from creating colossal number of repeat elementsets particularly in heavy datasets is an issue of concern. Rule interestingness is an idea that is utilized to channel the pointless and excess rules as in [12,13].

Case in point (2): Consider the rule {Milk, Food} → {Juice}. Since the support count for {Milk, Food, Juice} is 2 and the aggregate number of operands is 5, the rule's support is 2/5 = 0.4. The rules confidence is gotten by separator the support count for {Milk, Food, Juice} by the support count for {Milk, Food}. Since there are 3 operands that contain milk and sustenance, the confidence for this standard is 2/3 = 0.67.

2. P LITERATURE SURVEY

Knowledge discovery and data mining as known by Fayyad et al. (1996) is "the operation of recognizing legitimate, novel, valuable, and justifiable examples in data". Data mining has risen especially in circumstances where using so as to analyzing the data manually or straightforward inquiries is either unimaginable or very complicated (Cant'u-Paz & Kamath, 2001). Data mining is a multi-disciplinary field that joins learning from numerous controls, basically from machine learning, artificial intelligence, statistics, signal and image processing, mathematical optimization, and pattern recognition (ibid.). Information detection and data mining comprise of three primary strides to change over a gathering of crude data to valuable information. These three stages are data pre-processing, learning extraction, and data post-processing (Freitas, 2003). The found learning ought to be precise, conceivable, important and fascinating for the end client keeping in mind the end goal to consider the data mining procedure as effective (Cant'u-Paz & Kamath, 2001). This segment gives a review of data mining pre-processing, data mining assignments, and the routine systems for data mining [22]. Some exploration work like [13,15] talk about the rule era issue, they proposed that mining Simple Association Rules (SAR) that have a solitary component as its subsequent will be more efficient. In paper [16], the work focuses on association rules usage mining. Furthermore improve diverse assessment of pave the unpredictability of managing the information from users, clients, and business analyst. They have utilized Apriori and enhanced FP tree to discover association rule. Apriori the traditional mining algorithm is an approach to discover certain potential, general learning from the gigantic ones. Apriori algorithm [17] is the mining of repeat elementset and association rule learning [18] over operand databases. It filters the repeated elementsets by scanning the database until those element shows up repeat in database. This is utilized to discover the association rule [19]. The FP-Tree Algorithm is another approach to discover repeat styles [20] without using candidate generations [20], accordingly enhancing execution. It utilizes a divide-and-conquer plan. The central portion of this strategy is the utilization rehash example tree (FP tree), which keeps the piece set association information. The objective is to discover association rule with minimum support at least s and minimum confidence at least c and minimizing the information uncover about the private database. They have proposed FDM [21] algorithm to safely mine the association rule on horizontally partitioned database [9].

3. MINING ASSOCIATION RULES ALGORITHMS (DILEMMA DEFINITION)

Association rule mining discovers fascinating association or correlation relationships among an extensive arrangement of data elements [23,25]. The association rules are viewed as fascinating on the off chance that they fulfill both a minimum support threshold and a minimum confidence threshold [26]. A more formal definition is the accompanying. Let E={ $e_1,e_2,\ldots\ \ldots\ ..i_m$} be a set of elements. Let D, the errand applicable data, be a set of database operands where every operand O is a set of elements such that O⊆E. Every operand is connected with an identifier, called OED. Let A be a set of elements. A operand O is said to contain A if and only if A⊆O. An association rule is implication of the form A→B, where A⊂E, B⊂E and A∩B=∅ The rule A→B holds in the operand set D with support s, where s is the percentage of operands in D that contain A∪B (i.e., both A and B). This is taken to be the probability, P (A∪B) the rule A→B has certainty c in the operand set D if c is the percentage of operands in D containing A that likewise contain B. This is taken to be the restrictive likelihood, P(B/A). That is,

support (A→B) =P (A∪B)…………………………. (1)

confidence (A→B) =P (B/A)……………………... (2)

The meaning of a rehash style depends on the following contemplations. A set of elements is alluded to as an elementset (style). An elementset that contains k elements is a k elementset. For instance the set {name, semester} is a 2-elementset. The event recurrence of an elementset is the quantity of operands that contain the elementset. This is likewise referred to, clearly, as the recurrence, support count, count of elementset. An elementset fulfills minimum support if the event recurrence of the elementset is more prominent than or equivalent to the result of minimum support and the aggregate number of operands in D. The quantity of operands needed for the elementset to fulfill minimum support is in this way alluded to as the minimum support count. If an elementset fulfills minimum support, then it is a repeat elementset (repeat style). The most widely recognized way to deal with discovering association rules is to separate the problem into two portions [24]:

1. Find all repeat elementsets: By definition, each of these elementsets will happen at any rate as repeat as pre-determined minimum support count [25].
2. Produce solid association rules from the repeat elementsets: By definition, these rules must fulfill minimum support and minimum confidence [25].

Extra interestingness measures can be implemented, if wanted. The second step is the simpler of the two. The general performance of mining association rules is specified by the first step. As indicated in [27], the execution, for big databases, is most affected by the combinatorial blast of the repeat of conceivable rehash elementsets that must be considered furthermore by the quantity of database scans that must be performed [24]. Since the answer for the second sub issue is direct, real research endeavors have been spent on the first sub issue like [28, 29]. To represent Algorithms of Mining Association Rules:

Algorithm 1:
Apriori Algorithm

Apriori Algorithm is utilized to discover relationship between diverse sets of data. Every group of data has a number of elements and is known as an operand. The yield of Apriori is sets of rules that let us know how regularly elements are contained in sets of data [30]. The associations that Apriori finds are called Association rules. An association rule has two portions. The Antecedent is a subset of elements found in sets of data. The Consequent is a element that is found in blend with the precursor. Two terms depict the importance of the association rule. The Confidence is a percentage of data sets that contain the forerunner. The Support is a percentage of the data sets with the antecedent that likewise contain the resulting. Figure (1) gives the Apriori algorithm work process.

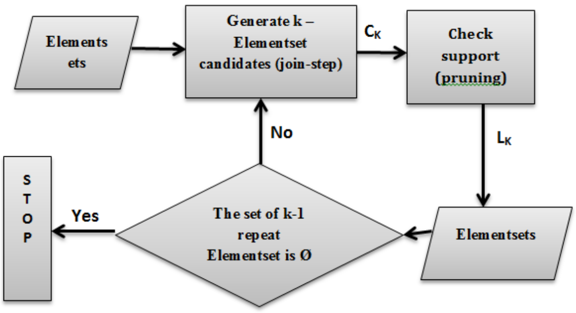

Elementsets
Generate k – Elementset candidates (join-step)
C_K
Check support (pruning)
L_K
No
Elementsets
The set of k-1 repeat Elementset is Ø
Yes
STOP

Fig 1: Apriori Algorithm work process

To enhance the productivity of the level insightful generation of repeated elementsets, a property called Apriori property is utilized to lessen the search space. This property expresses that all nonempty subset of a repeat elementset must likewise be repeated. A two-step procedure is utilized to discover L_k-1 from L_k:

1. The join step: To discover L_k, a group of $_k$ elementsets is produced by joining L_k-1 with itself. This group of candidate elementsets is meant C_k.
2. The prune step: Ck is a superset of L_k, that is, its members might possibly be repeat, and however all the repeat k-elementsets are incorporated in C_k.

A scan of the database is done to determine the count of every candidate C_k, the individuals who fulfill the min-supp is added to Lk. To diminish the quantity of competitors in C_k, the Apriori property is utilized. A case of Apriori algorithm is found in [9,10].

Algorithm 2:
Mining Association Rules with Multiple Minimum Supports using Maximum Constraints:
At the point when the minimum support value of an elementset is defined as the lowest minimum supports of the elements in it, the elementset may be big, yet elements included in it might be little. For this situation, it is uncertainty capable whether this elementset is worth considering. For the illustration depicted above, if the support of element B is 30%, littler than its minimum support 40%, then the 2-elementset {A, B} ought not be worth considering. It is along these lines sensible in some feeling that the event frequency of a fascinating elementset must be bigger than the maximum of the minimum supports of the elements contained in it. Wang et al. [8] proposed a mining methodology, which permitted the minimum support value of an elementset to be any job of the minimum support values of elements contained in the elementset. Despite the fact that their methodology is adaptable in allocating the minimum supports to elementsets, its time unpredictability is high because of its generality. In this paper, it has been proposed a straightforward and productive algorithm taking into account the Apriori way to deal with create the big elementsets under the maximum constraints. Note that if the mining issue is not under the greatest requirement, then Wang et al's methodology is a decent choice [8,10].

Algorithm 3:
Simple Association Rules
Simple association rules unlike functional dependencies, association rules are not transitive and don't create. Case in point, given that A⇒B and A⟹C, one can't presume that A⟹BC. The accompanying evidence was given in [14]. Given:

$A,B,C\subset A\cap B=\emptyset, B\cap C=\emptyset, A\cap C=\emptyset$, where:

1. conf(A⟹BC)=conf(A⟹B)×conf(AB⟹C)=conf(A⟹C) ×conf (AC⟹B).

2. supp (A⟹BC)=supp(AB⟹C=supp(AC⟹B).

Confirmation: According to the meaning of conf, there:

$$conf(A \Longrightarrow BC) = \frac{supp\ (ABC)}{supp\ (A)} = \frac{(\mathrm{supp\ (AB)\times conf\ (AB \Longrightarrow C)}}{\mathrm{supp\ (A)}})$$

conf (A⟹B)×conf(AB⟹C)

By trading B and C is acquired:

conf (A⟹BC)=conf(A⟹C)×conf(AC⟹B)

As indicated by the meaning of supp, there are

supp (A⟹BC)=supp(ABC)=supp(AB⟹C)=supp(AC⟹B).

An illustration for an example of SAR is found in [14].

4. IDEA OF ALGORITHMT

The thought of algorithm relies on upon the algorithms illustrated in section III, new algorithm is suggested that implies simple association rules however with indicating diverse min-supp to every single element. The algorithm is called mining simple association rules with numerous minimum supports utilizing maximum constraints.

5. PROPOSED ALGORITHM

The finding of the blend between the two algorithms proposed in [14,28]. The detailed algorithm of generating mining simple association rules with multiple minimum supports. The algorithm is showed below.

1. Create candidate one elementset (C_1).

2. Input a min-supp for every element in (C_1).

3. for every element in (C_1).

4. In the event that element supp-count>=its min-supp then output (L_1): element //generate repeat 1-elementset.

5. Perform join as Apiori.

6. for each elementset.

7. m1=max of min-supp of every element in elementset.

8. for every element in elementset.

9. If element supp-count > = mE then output C_k elementset //produce candidate elementsets.

10. for each elementsets in C_k.

11. If elementset supp-count > =mE then output L_k elementset // produce repeat elementsets.

12. for every 1_k in L_k // 1^k is a k-repeat elementsets of L_k.

13. SB= {(k-1) - elementsets f^{k-1} /$f^{k-1} \subset f^k$}.

14. for every $f^{k-1} \in$ SB.

15. conf=supp-count (f^k)/supp-count (f^{k-1}).

16. If conf ≥minconf then output r: $1^{k-1} \Longrightarrow (f^k - f^{k-1})$ with conf, supp; // output ruleset.

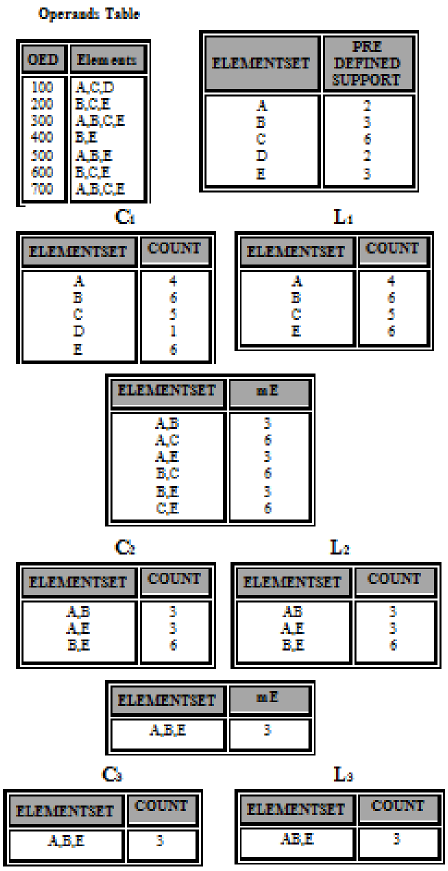
17.end for;

18.end for;

The algorithm steps which can be clarified are:

1. A min-supp is indicated for every element, then check if every elements supp-count is >= it's predefined min-supp and creates the repeat 1-elementset in.
2. Candidate elementsets are produced just if every element's supp-count in this elementset >= to the maximum predefined min-supp determined for every of those elements (mE).
3. Repeat elementsets are produced if elementset supp-count is >= to mE.
4. In the wake of discovering the repeat k-elementsets, the elementsets subsets at level k-1 just are discovered then produce simple rules and check if the rule's conf is >= min-conf.

In the example explain in Figure (2), the elementsets are produced by steps said above; if the min-conf determined is 75%, seven rules are created in Apriori; one of them is not a simple rule which is A⇒BE. In this example, this rule is not produced as it could be gotten from the rules (A⇒B; AB⇒E; A⇒E; AE⇒B).



Operands Table

OED	Elements
100	A,C,D
200	B,C,E
300	A,B,C,E
400	B,E
500	A,B,E
600	B,C,E
700	A,B,C,E

ELEMENTSET	PRE DEFINED SUPPORT
A	2
B	3
C	6
D	2
E	3

C1

ELEMENTSET	COUNT
A	4
B	6
C	5
D	1
E	6

L1

ELEMENTSET	COUNT
A	4
B	6
C	5
E	6

ELEMENTSET	mE
A,B	3
A,C	6
A,E	3
B,C	6
B,E	3
C,E	6

C2

ELEMENTSET	COUNT
A,B	3
A,E	3
B,E	6

L2

ELEMENTSET	COUNT
AB	3
A,E	3
B,E	6

ELEMENTSET	mE
A,B,E	3

C3

ELEMENTSET	COUNT
A,B,E	3

L3

ELEMENTSET	COUNT
AB,E	3

Fig 2: Simple association rules withmultiple minimum supports Example

6. COMPARISON PROCEDURES BETWEEN SINGLE AND MULTIPLE SUPPORTS ALGORITHMS

The comparison between any mining association rules algorithms is either made on number of rules produced by every algorithm or on their processing time. To look at the processing time, same algorithm parameters ought to be utilized which are the min-supp and min-conf. But utilizing the same min-supp causes disarray when comparing an algorithm that takes one min-supp value and another that takes multiple min-supp criteria. On the off chance that the yield of the single and multiple supports algorithms is the same; it implies that both had equivalent parameters.

The sub program show in the graph of (Fig. 3) are utilized to indicate a min-supp to every element to unite the output of single and multiple supports algorithm. This will make comparing the processing times is uniting so as to take into account a dependable angle the output.

In figure (3) these strides are taken to choose what min-sup ought to be indicated to every single element when comparing Simple Association Rules with mining Simple Association Rules with multiple Minimum Supports utilizing Maximum Constraints. The comparison methods are:

1. Take the rules created from SAR algorithm with a particular min-sup i.e. 1%.
2. For every rule, determine which elementsets are contained in this rule.
3. Get the sup-count of every elementset.
4. Compute the values of mE of every elementset.
5. For every element in the elementset, specify a min-sup equivalent to mE
6. If any element is specified >1 min-sup, pick the littlest value.
7. If some elements were not determined any min-sup, this implies that they didn't show up in the rules produced. They ought to be indicated a min-sup more noteworthy than their sup-count to be avoided from repeat 1-elementset.

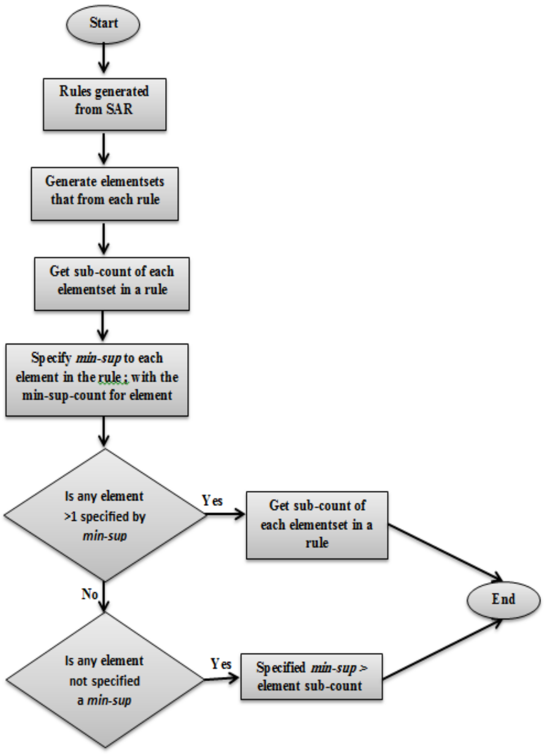

Start
Rules generated from SAR
Generate elementsets that from each rule
Get sub-count of each elementset in a rule
Specify min-sup to each element in the rule ; with the min-sup-count for element
Is any element >1 specified by min-sup
Yes
Get sub-count of each elementset in a rule
End
No
Is any element not specified a min-sup
Yes
Specified min-sup > element sub-count

Fig 3: Comparison sub program

For example: In the wake of mining a set of operands utilizing Simple Association Rules (SAR) algorithm, number of rules are created. If the first rule was: AB⇒C

The elementsets in this rule and their supp-count are:

A, B Supp = 589 // A, C Supp = 725// B, C Supp = 1623//

A, B, C Supp =589.

The mE should not exceed any of the support counts specified above, If their specified min-supp is equal to their least number = 589, and guarantee that those four elementsets are generated and so this rule will be generated.

min-supp (A) = 589//min-supp (B) = 589//min-supp(C) = 589.

If the second rule was: DB⇒E. The elementsets in this rule and their support counts are:

D, B Supp = 485//D, E Supp = 559//B, E Supp = 1513//

D, B, E Supp = 485. For those elementsets to be created the min-supp of those elements ought not surpass 485 which is the slightest number among their support count so,

min-supp (D) = 485//min-supp (B)= 485//min-supp (E) = 485.

Be that as it may, the min-supp (B) was determined before to be equivalent to 589. For this situation the littlest amount is picked which are 485 to make sure that all elementsets that contain this component is produced. Repeat the methods (procedures) mentioned above for each rule to discover the min-supp that ought to be determined to every element. The elements which don't show up in the rules ought to be indicated a min-supp > its supp-count to be prohibited from repeat 1-elementset.

7. PERFORMANCE EVALUATION AND SIMULATION RESULTS

The comparison methods(sub programs) clarified in segment IV are applied when comparing Apriori with maximum constraints and when comparing mining Simple Association Rules with multiple Minimum Supports utilizing Maximum Constraints.

A. Evaluated for time:

The four algorithms Listed in the paper are tested and assessed for time and accuracy as showed in part B. Follow the methods (sub programs or procedures) Listed in section IV, indicate a min-supp for every element to test the multiple supports algorithms and after that produce rules at steady min-conf. The processing time of every algorithm when implemented on AdventureWorksDW as shown in Table (3) and the comparison diagram as show in Figure (4).

Table 3. Processing Times of Apriori, SAR, Maximum Constraints and Simple Association Rules with multiple Minimum Supports utilizing Maximum Constraints for AdventureWorksDW

Apriori Equivalents supp-count(%)	*Apriori Time (sec)*	*SAR Time (sec)*	*Maximum Constraints Time(sec)*	*Proposed Algorithm Time (sec)*
0.2	7	0.5	5	0.2
0.175	8	0.6	6	0.3
0.150	9	0.7	7	0.4
0.125	10	0.8	8	0.5
0.1	14	0.9	11	0.6
0.075	15	1	13	0.7

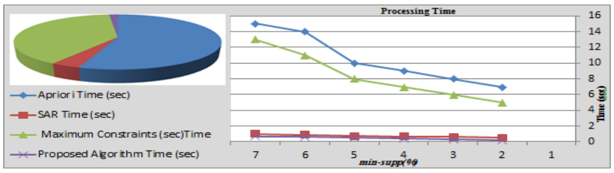


Fig 4: Comparison of processing time between Basic Apriori, SAR, Maximum Constraints and Proposed Algorithm of AdventureWorksDW

In Figure (4) proposed algorithm takes the minimum measure of time among other algorithms and Basic Apriori takes the longest time. The time increases when the min-supp decreases because the number of elementsets and rules produced increment.

B. Accuracy Test, Interesting Measurements:

Subsequent to building a mining model, must choose the model validity. The data must be haphazardly isolated into two isolated datasets (preparing and testing). The preparation dataset is utilized to construct the model, and the testing dataset is utilized to test the exactness of the model. This is a portion of the software engineering cycle to test numerous algorithms that take care of the same issue then test their proficiency in taking care of the issue. The two datasets are haphazardly isolated to test the four algorithms that have been mentioned in this paper. The isolation was in the proportion of 10% and 90%. The four algorithms are implemented on the preparation and testing datasets at diverse estimations of min-supp with constant min-conf, follow the comparison methods (procedures) Listed in section IV to specify the min-supp that ought to be given to every individual component while testing the multiple supports algorithm. The specification is a percentage of the min-supp assigned to the original dataset, so the algorithms don't produce the same number of rules and the time is computed as the time taken to create every rule. Subsequent to performing the tests, the outcomes are gathered to figure the precision of every algorithm by applying the idea of exactness index where:

$$accuracyindex = \frac{\sum_n accuracy}{\max(accuracy)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. (1)$$

Equation (1) demonstrates that if an algorithm has the accuracy of 100%, it is the most astounding exactness among alternative algorithms. It doesn't imply that it has 100% total accuracy. The steps of the accuracy test are:

Partition the data into datasets haphazardly.

1. Mine each dataset independently by utilizing weka instrument.

2. Compare the rules produced from both dataset and discover their intersection, then compute the percentage of this intersection. Applying the time index on the outcomes indicated in Table (3) where:

$$timeindex = \frac{\sum_n time}{\max(time)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (2)$$

Equation (2) demonstrates that if an algorithm takes 100% time, it takes the longest measure of time among other algorithms. Case in point, in the wake of isolating the data into 90% and 10%. The 90% dataset produces 50 rules and the 10% dataset creates 60 rules. If the amount of rules intersections are 20. Then the percentage on first part is

20/50*100=40% and the percentage on second part is 20/60*100=33.33%. Likewise, implementation the previous step for each of the 20% and 80% and after that 30% and 70%. Tables (4) and (5) show the results of the time and accuracy indices of AdventureWorksDW and these outcomes are outlined in Figure (5) and Figure (6).

Table 4. Time indices of AdventureWorksDW

	Apriori Time (sec)	SAR Time (sec)	Maximum Constraints Time(sec)	Proposed Algorithm Time (sec)
Time index%	100.00	97.77	83.45	81.40

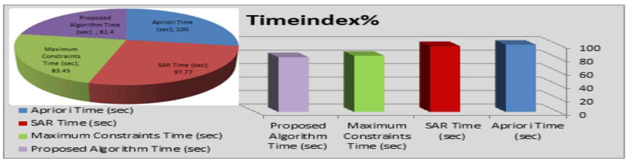


Fig 5: Time index of the four algorithms of AdventureWorksDW

Table 5. Accuracy indices of AdventureWorksDW

	Apriori Time (sec)	SAR Time (sec)	Maximum Constraints Time(sec)	Proposed Algorithm Time (sec)
Accuracy index%	97.89	99.50	98.25	100.00

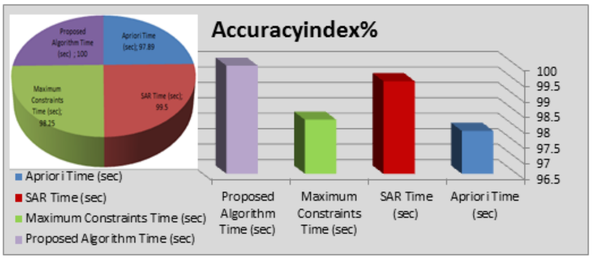


Fig 6: Accuracy index of the four algorithms of AdventureWorksDW

Figure (5) shows that the accuracy of the SAR is the best among other algorithms and the distinction between the precision of simple association rules and proposed algorithm is in the scope of 0.5%. That implies the proposed algorithm is speedier than any other algorithms and the accuracy is the best when implemented on dataset without influencing the number of rules created in light of the fact that it produces less elementsets because of the two conditions which is added to the algorithm (it utilizes the idea of multiple min-supp that gives the client adaptability in determining a min-supp for every element and making in repeat child node pruning step). Figure (6) shows that proposed algorithm is quicker than other algorithms. Basic Apriori takes the longest time and proposed algorithm takes the shortest.

8. CONCLUSION

Data mining has been broadly utilized as a part of business, commercial enterprises and engineering. In this paper, it has been offering another perspective about characterizing the minimum supports of elementsets when elements have minimum supports. In this paper, a simple rule set has been introduced which holds all information in the genuine rule set, yet has a littler size. The simulation results of the proposed algorithm (mining simple association rules with multiple minimum supports utilizing maximum constraints) are speedier than any other algorithms and the accuracy is the best when implemented on dataset without influencing the number of rules produced. As it is indicated in the accompanying points:

1. It has been demonstrated that all qualified association rules can be taken from the SAR.
2. Proposed algorithm which is a blend between the simple association's rules taken from essential Apriori Algorithm with the multiple minimum support algorithms.
3. The new algorithm take less time than the SAR and the accuracy isn't influenced in a greater percentage when implemented on a dataset that has an immense number off odd elements.
4. The rules created from the algorithms are 100% interest when implemented to AdventureWorksDW dataset.
5. In this new algorithm the client is given the flexibility to specify a different min-supp for every single element this choice beats the issue of rare elements.
6. The maximum constraint is utilized, which has been all around clarified and may be suitable to some mining fields.
7. The new algorithm consumes almost the same time as the maximum constraint algorithm and the accuracy is influenced in a greater rate when implemented on dataset that has colossal number of odd elements.
8. Producing simple rules decreases the processing time when implemented on dataset that has minimal number of odd elements.
9. This algorithm has demonstrated their points Compared with traditional methods regarding of the number of candidate rules.
10. Besides, the granular computing technique of bit strings can easily be used to speed up the proposed data mining algorithm.

9. REFERENCES

[1] Vincent, M., BSc. 2012. Introducing a Data Mining Process Framework to Enable Consultants to Determine Effective Data Analytics Tasks, Delft University of Technology, Delft University of Technology.

[2] Yun-S. C., Min-H. K., 2004., A Study of mining support value in association analysis –from Bayesian viewpoint, Yuan-Ze University, 135 Yuan-Tung Road, Chung-Li 32026, Taiwan, R.O.C., Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference.

[3] Jure L., Anand R., Jeffrey D. U. 2014. Mining of Massive Datasets.

[4] Ian H. W., Eibe F., Mark A. H., 2011. Data Mining Practical Machine Learning Tools and Techniques. Third Edition, MA 01803, USA

[5] Margaret H. D., Yongqiao X., Le G., Zahid H. A Survey of association rules. Southern Methodist University-Dallas, Texas 75275-0122, University of Oklahoma-Norman, OK 73019 Dallas, Texas 75275-0122 Norman, OK 73019.

[6] Jeremy M., Jun W. L. 2005. Mining Association Rules in Spatio- Temporal Data: An Analysis of Urban

Socioeconomic and Land Cover Change. Urban Studies Temple University, University of Colorado, USA.

[7] María N. M., Saddys S., Vivian F. L. 2005. Association Rules: Problems, Solutions and New Applications. Universidad de Salamanca, Plaza Merced S/N, 37008, Salamanca.

[8] Yeong-C., L., Tzung-P. H. Wen-Y. L. 2004. Mining Association Rules with Multiple Minimum Supports Using Maximum Constraints. I-Shou University, Kaohsiung, 840, Taiwan, R.O.C., National University of Kaohsiung, 811, Taiwan, R.O.C.

[9] B. Senthil Kumar. P. Rajapandi, K. Sridar, D. Shanthi, 2015. Mobile Based Medical Diagnosis System Using Ann- Arm For The Diabetes Mellitus.

[10] Rakesh A., Ramakrishnan S. 1994. Fast Algorithms for Mining Association Rules. IBM Almaden Research Center 650 Harry Road, San Jose, CA 95120.

[11] Mingjun S., Sanguthevar R. 2005. A Transaction Mapping Algorithm for Frequent Elementsets Mining. University of Connecticut, Storrs, IEEE Transaction on Knowledge and Data Mining.

[12] Tsau Y. L., 2000. Data mining and machine oriented modeling: a granular computing approach. Journal of Applied Intelligence, Vol. 13, No 2, pp. 113-124.

[13] Tzung-P. H., Chan-S. K., Sheng-C. C. 1999. Mining association rules from quantitative data.

[14] Rakesh A., Tomasz I., Arun S. 1993. Mining association rules between sets of elements in massive databases.

[15] Rakesh A., Ramakrishnan S. 1995. Mining sequential patterns. The Eleventh IEEE International Conference on Data Engineering, , pp. 3-14.

[16] Vidisha H. Z., Leena H. P. 2015. Literature Survey on Formation of Association Rule using Secure Mining. International Journal of Scientific &Technology research volume 4, Issue 01.

[17] Rakesh A., Tomasz I., Arun S. 1993. Mining association rules between sets of elements in large Database, The ACM SIGMOD Conference, pp.207-216.

[18] William J. F., Gregory P. S., Christopher M. 1991. Knowledge discovery in databases: an Overview. The AAAI Workshop on Knowledge Discovery in Databases.

[19] Heikki M., Hannu, A. Inkeri V. 1994. Efficient algorithm for discovering association rules. The AAAI Workshop on Knowledge Discovery in Databases, pp.181-192.

[20] Ramakrishnan S., [15] Rakesh A. 1995. Mining generalized association rules.

[21] K. Wang, Y. H, J. Han, 2000. Mining frequent item sets using support constraints.

[22] Sotiris. K., Dimitris K. 2006. Association Rules Mining: A Recent Overview. University of Patras.

[23] Ayush. J. Jordan W., Khulood A., Sultanah A. The Use of Evolutionary Algorithms in Data Mining.

[24] Cornelia. G., Robert G., Stefan H. A Comparative Study of Association Rules Mining Algorithms. University of Oradea, University of Timisoara.

[25] Urvi Y. B., Pratik A. P. A. 2014. Recent Overview: Rare Association Rule Mining. International Journal of Computer Applications (0975 – 8887) Volume 107 – No 18, December, India.

[26] G. Johnny, 2009.Developing A-priori Algorithm for Fast Mining Association Rules.

[27] Medhat H. A. A., Sara G, E. 2012. Aggregate FunctionBased Enhanced Apriori Algorithm for Mining Association Rules.

[28] W. J. Samaraweera, S. Vasanthapriyan, K.S. Oza. 2014. Designing a Multi-Level Support based Association Mining Algorithm..

[29] A. R. Ingle1, A. A. Ambekar, S. M. Chawre, S. N. Kedar, L. S. Mundafale. 2014. Implementation of an Algorithm for Mining Association Rules.

[30] Paresh T., Yogesh G. 2014. Using Apriori with WEKAfor Frequent Pattern Mining.