International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-2 Issue-4, September 2013

IP- Apriori: Improved Pruning in Apriori for Association Rule Mining

Prince Verma, Dinesh Kumar

Abstract- Association rule mining which is of great importance and use is one of a vital technique for data mining. Main among the association rule mining techniques have been Apriori and many more approaches have been introduced with minute changes to Apriori but their basic concept remains the same i.e use of support and confidence threshold(s). According to best of our knowledge we came to know that no work has been done in the field of improving the pruning step of Apriori. This paper introduces a new algorithm IP-APRIORI i.e. 'Improved Pruning in Apriori'. This algorithm improves the pruning procedure of Apriori algorithm by using average support (sup_{avg}) instead of minimum support (sup_{min}), to generate probabilistic item-set instead of large item-set.

Keywords- Data Mining, KDD Process, Association Rule Mining, Pruning.

I. INTRODUCTION

Originally, "DATA MINING" is statistical term which means the exploitation of data to retrieve valid information. So, it's the process of discovering useful summaries of data. Data Mining is the process to discover the knowledge or hidden pattern from large databases [5]. Data mining involves six classes of tasks commonly- *Anomaly detection*, *Association rule mining*, *Clustering*, *Classification*, *Regression* and *Summarization*. In this paper we are introducing the algorithm for Association Rule Mining.

A. Association Rule Mining

Association Rule Mining (ARM) [6] is a function of data mining to discover the possibility of co-occurrence of items in transactional database. It's most important and one of the well researched techniques among data mining, which was introduced by Rakesh Agarwal in [6], which aims to extract interesting correlations or associations among item-sets in transaction databases or data repositories. The statement introduced by Agarwal [6] for ARM was:

If transaction database, D = $\{T_1, T_2, \ldots T_N\}$, where T $\subseteq$ I where I is item-set of m distinct attributes, I = $\{I_1, I_2, \ldots, I_m\}$ and there are two item-sets A and B, such that A $\subseteq$ T and B $\subseteq$ T. Then association rule, A=>B holds where A C I and B C I and A ∩ B = Ø.

Revised Manuscript Received on 30 September 2013.
* Correspondence Author
PG Scholar Mr. Prince Verma, M.tech Computer Science, Punjab Technical University, DAV Institute of Engg. & Tech., Jalandhar, India.
Mr. Dinesh Kumar, Associate Professor, Department of Information Technology, DAV Institute of Engg. & Tech., Jalandhar, India.

Here, A is called antecedent while B as consequent. There are two important and basic measures for association rules, support and confidence. Thresholds for support and confidence are predefined by users to eliminate the uninterested rules. Support for an association rule is defined by fraction or percentage of transactions in D that contain A∪ B. Support is calculated by the following formula:

$$sup(\mathrm{A} \cup \mathrm{B}) = \frac{count(\mathrm{A} \cup \mathrm{B})}{count(D)} \qquad (1)$$

Confidence is a measure for the strength of the association rules and it is defined by the fraction or percentage of the number of transactions in D that contain A also contains B. 'IF' component is called Antecedent and 'THEN' component called consequent i.e. here in example, A is antecedent while B is consequent. Confidence is calculated by dividing the probability of items occurring together to the probability of occurrence of antecedent. So, Confidence is calculated by the following formula:

$$conf(A => B) = \frac{\sup(A \cup B)}{\sup(A)} \qquad (2)$$

ARM finds the association rules that satisfy the user predefined thresholds sup_{min} and $conf_{min}$ from a given database [6].

For ARM certain approaches/algorithms many have been proposed. Various among these algorithms are AIS, SETM, Apriori and FP-tree. Many more approaches have been introduced in between these algorithms with minute changes. But algorithm which makes base for new upcoming algorithms is Apriori.

II. TYPICAL APRIORI ALGORITHM

The Apriori algorithm [6] generate the candidate item-sets in one pass by using only the large item-sets of the preceding pass without considering the transactions of database. The basic concept used here is that every subset of a large item-set must be large. The k^{th} pass candidate item-sets with k items is generated from previous pass by joining large item-sets of k-1 items (candidate generation), and deleting those item-sets that contain any subset that is not large (Pruning concept). This pruning process results in generation of a much smaller number of candidate item-sets.

```
1) Algorithm Apriori(large-1 item-sets)
2)  L1={large-1 item-sets};
3)  for(k=2; Lk-1≠Ø; k++) do begin
4)      Ck=apriori-gen(Lk-1);  //New candidates
5)      forall transactions t ∈ D do begin
6)          Ct=subset(Ck,t);    //Candidates contained in t
7)          forall candidates c ∈ Ct do
8)              c.count++;
```


Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

9) **end for**
10) $L_k=\{c \in C_k \mid c.count \geq minsup\}$
11) **end for**
12) Answer=$U_k L_k$

Algorithm 1: Apriori Algorithm [6]

Algorithm 1 is the Apriori algorithm. The first pass of algorithm finds the item occurrences to determine the large 1-item-sets. A pass, say k, has two phases.

First, the large item-sets, say L_{k-1} founded in the $(k-1)^{th}$ pass are used to generate the candidate item-set C_k, using the apriori-gen function Algorithm 2. The apriori-gen function takes an argument L_{k-1}, the set of all large (k-1)-item-set and returns a superset of the set of all large k-item-sets. The function in the join step joins one L_{k-1} with another matching L_{k-1}. And in the next prune step, we delete those item-sets c, where $c \epsilon C_k$ such that any (k-1)-subset of c is not in L_{k-1} of the previous pass.

Then, in the next phase the database is scanned for support of candidates in C_k.

1) Algorithm Apriori-gen(L_k)
2) **insert** into C_k
3) **select** $p.item_1.p.item_2,.....p.item_{k-1},q.item_{k-1}$
   from L_{k-1} p,L_{k-1} q
   where $p.item_1=q.item_1.....p.item_{k-2}= q.item_{k-2}$,
   $p.item_{k-1}<q.item_{k-1}$;
4) **forall** item-sets $c \in C_k$ **do**
5) **forall** (k-1)-subsets s of c **do**
6) **if**($s \notin L_{k-1}$) **then**
7) **delete** c from C_k;

Algorithm 2: Apriori-Gen function [6]

A. *Apriori Explanation*

As Apriori algorithm generate the candidate item-sets without considering the transactions in the database.

For Example; we are provided with a database D (Table I) with some set of transactions, sup_{min}=61% and $conf_{min}$=70%

Table I: Database D

Transaction	Items	Transaction	Items
11	C,D,E	21	A,B,C,D
12	A,B,C,D	22	C,D,E
13	E	23	A,B,C,D
14	A,B,C,D	24	E
15	A,B,C,E	25	A,B,C,D
16	A,B,D	26	A,B,C,D,E
17	A,B,C,D,E	27	A,B,D,E
18	C,D	28	C,D,E
19	A,B,C,D	29	A,B,C,D,E
20	B,C,D	30	A,B

Before creating large item-sets (L1, L2 & so on) and candidate item-sets (C2 & so on) candidate, firstly Candidate item-set, C1 is generated from database (i.e. Table II from Table I) and then, Large item-set, L1 is created from C1 using supmin=61% (i.e. Table III from Table II)

Table II: Candidate Item-set C1

Candidate Item-set, C1	Support
A	65%
B	70%
C	75%
D	80%
E	55%

Table III: Large Item-set L1

Large Item-set, L1	Support
A	65%
B	70%
C	75%
D	80%

Step 1: Candidate set is generated as follows:

•The candidate item-sets, C_k can be generated by joining large item-sets L_{k-1} items, and

•Deleting those that contain any subset that is not large.

In Example C_2 is generated from L_1 items by join procedure (i.e. Table IV from Table III) and those item-sets are deleted that have some (k-1) subset of c is not in L_{k-1} where $c \in C_k$.

*Step 2:*Large item-set, L_k is generated from candidate item-set, C_k using sup_{min}.

The above two steps are repeated until Large Item-set came to be empty.

Here from the candidate item-sets, C_2 elements with $sup \geq sup_{min}$, Large item-set, L_2 is created (Table V). As all candidate item-sets, C_3 has sup $\geq$ sup_{min}, so all became Large item-set, L_3 (Table V from Table IV).

Table IV: Candidate Item-set C_2

Candidate item-set, C2	Support
A,B	65%
A,C	50%
A,D	55%
B,C	60%
B,D	60%
C,D	70%

Table V: Large item-set L_2

Large Set L2	Support
A,B	65%
C,D	70%

Large Item-set: (A,B), (C,D) & Association Rules are:

Table VI: Association Rules

Large Set	Association Rules	Confidence
A,B	A=>B	65/65= 100%
	B=>A	65/70= 92.8%
C,D	C=>D	70/75= 93.3%
	D=>C	70/80= 87.5%

III. PROPOSED APPROACH (IP-APRIORI)

The Apriori algorithm needs user defined threshold values, i.e minimum support (sup_{min}) and minimum confidence ($conf_{min}$). Sup_m $_{in}$ is needed to generate the large item-set from candidate set and $Conf_{min}$ is required to generate required set of association rules from generated large item-sets.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The process of creating large item-set by removing the not so required candidate sets is called pruning. So, Pruning is dependent on Sup_{min} threshold provided by user. But some important rules get pruned because of this user-defined threshold as user doesn't know that at which threshold, valuable association rules get generated.

By literature review we came to know that no work has been done in the field of improving the pruning step of Apriori. So, an algorithm is proposed named *"IP-APRIORI"*. The proposed algorithm focuses in improving the pruning procedure of Apriori algorithm by using average support (sup_{avg}) instead of minimum support (sup_{min}), to generate probabilistic item-set instead of large item-set. Here average support is not user defined value but it is calculated using the formula:

$$sup_{avg} = \frac{\sum_{k=1}^{m}(\sup(k))}{m} \quad (3)$$

where m is the number of items involved.

So this formula would lead to better item-sets and this increases the number of better association rules that were left underestimated by Apriori algorithm.

The methodology used to implement the mentioned technique is defined using a flowchart provided in Fig. 1.

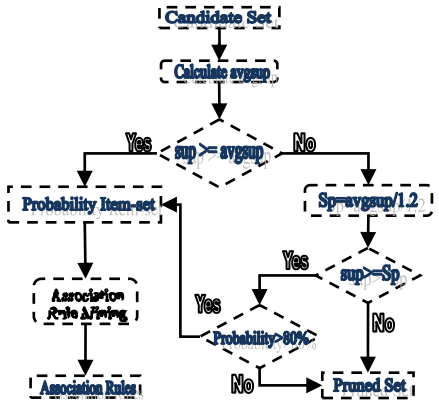

Fig. 1: Technique used in proposed approach "IP-APRIORI"

The above methodology can be illustrated in the form of an algorithm as follows:

1) Algorithm IP-Apriori(probability-1 item-sets)
2) P_1={probability-1 item-sets};
3) **for**(k=2; $P_{k-1}\neq\emptyset$; k++) **do begin**
4) C_k=IPApriori-candidate(P_{k-1}) //new candidates
5) **forall** transactions $t \in D$ **do begin**
6) C_t=subset(C_k,t); //candidates in t
7) **forall** candidates $c \in C_t$ **do**
8) c.count++;
9) **end for**
10) **if**(c.count≥avgsup)
11) P_k={$c \in C_k$};
12) **else**
13) S_p= sup_{avg} / 1.2;
14) **if**(c.count≥S_p and c.prob≥80%)
15) P_k={$c \in C_k$};
16) **end if**
17) **end**
18) Answer=$U_k P_k$

Algorithm 3: IP-Apriori Algorithm

1) Algorithm IPApriori-candidate(P_k)
2) **insert** into C_k
3) **select** s.$item_1$.s.$item_2$,.....s.$item_{k-1}$,t.$item_{k-1}$
from P_{k-1} s,P_{k-1} t
where s.$item_1$=t.$item_1$.....s.$item_{k-2}$= t.$item_{k-2}$, s.$item_{k-1}$<t.$item_{k-1}$;
4) **forall** item-sets $c \in C_k$ **do**
5) **forall** (k-1)-subsets s of c **do**
6) **if**(s $\notin P_{k-1}$) **then**
7) **delete** c from C_k;

Algorithm 4: IP-Apriori-candidate generation function[6]

Here in IPApriori-candidate algorithm (Algorithm 4) candidates are generated for creating probability item-set in IP-Apriori algorithm(Algorithm-3). So, IP-Apriori contains following steps:

1. Calculate sup_{avg} from given database D
   a. For item-sets with sup ≥ sup_{avg} are inserted into probability item-set.
   b. For item-sets with sup ≤ sup_{avg}, item-sets with sup ≥ S_p (where S_p = sup_{avg} / 1.2) and their probability of occurance >80% (i.e. items with sup greater than 80% of the maximum support of item-set having sup>S_p) are inserted into probability item-set, while others are pruned.
2. Above step is r
3. epeated for all probability item-sets and from this probability item-set, association rules are produced.

A. IP-APRIORI Algorithm Explanation

The proposed algorithm focuses in improving the pruning procedure of Apriori algorithm by using average support (sup_{avg}) instead of minimum support (sup_{min}), to generate probabilistic item-set instead of large item-set.

For Example; we are provided with a database D (Table I) with some set of transactions and $conf_{min}$=70%.

Before creating probability item-sets (P_1, P_2 & so on) and candidate item-sets (C_2, C_3 & so on) candidate, firstly Candidate item-set, C_1 is generated from database (i.e. Table VIII from Table VII).

Then, Would-be Probability item-set, P_1' is created from C_1 (i.e. Table IX from Table VIII). The item-sets with sup ≥ sup_{avg} are inserted into probability item-set. i.e.

$$\text{Using } sup_{avg} = \frac{65+70+75+80+55}{5} = 69\%$$

Table VII: Database D

Transaction	Items	Transaction	Items
11	C,D,E	21	A,B,C,D
12	A,B,C,D	22	C,D,E
13	E	23	A,B,C,D
14	A,B,C,D	24	E
15	A,B,C,E	25	A,B,C,D
16	A,B,D	26	A,B,C,D,E
17	A,B,C,D,E	27	A,B,D,E
18	C,D	28	C,D,E
19	A,B,C,D	29	A,B,C,D,E
20	B,C,D	30	A,B

Table VIII: Candidate Item-set C1

Candidate Item-set, C1	Support
A	65%
B	70%
C	75%
D	80%
E	50%

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table IX: Probability Item-set P1'

Probability Item-set, P1'	Support
B	70%
C	75%
D	80%

Now emphasis are performed on item-sets with sup ≤ supavg, item-sets and sup ≥ Sp (here Sp = supavg / 1.2 = 69 / 1.2 = 57.5% i.e. item A. So, items with sup ≥ Sp is {A} with sup=65%). From these, items with probability >80% are taken into Probability Item-set. i.e. If items with sup ≥ Sp have maximum sup, say Sp-max then items with support greater than (80% of Sp-max) are inserted into probability item-set, while others are pruned. (here Sp-max=65% and 80% of Sp-max=52%). So, Probability Item-set P1 is given in Table X.

Table X: Probability Item-set P1

Candidate Item-set, C2	Support
A,B	65%
A,C	50%
A,D	55%
B,C	60%
B,D	60%
C,D	70%

Candidate item-set C_2 is generated from P_1 (Table XI from Table X). Here from the candidate item-sets, C_2 elements with sup ≥ sup_{min}, Probability item-set, P_2 is created (Table XII from Table XI). And for probability item-set generation same process as above is repeated.

Table XI: Candidate Item-set C_2

Probability Item-set, P1	Support
A	65%
B	70%
C	75%
D	80%

Table XII: Probability Item-set P_2

Probability Item-set, P2	Support
A,B	65%
A,D	55%
B,C	60%
B,D	60%
C,D	70%

Table XIII: Candidate Item-set C_3

Candidate item-set, C3	Support
A,B,D	55%
B,C,D	50%

From here P_3 is created containing (A,B,D) as its sup >52% and (B,C,D) is pruned (see Table XIV).

Table XIV: Probability Item-set P_3

Probability Item-set,P3	Support
A,B,D	55%

Probability Item-set: {(A,B), (B,C), (B,D), (C,D),(A,B,D)}. So, (B,C), (B,D) are more in probability item-set than large item-set of Apriori algorithm in Section 2.1. And association Rules are prescribed in Table XV.

Table XV: Association Rules

Large Set	Association Rules	Confidence
A,B	A=>B	65/65= 100%
	B=>A	65/70= 92.8%
A,D	A=>D	55/70=78.5%
	D=>A	55/80=68.7%
B,C	B=>C	60/70= 85.7%
	C=>B	60/75= 80%
B,D	B=>D	60/70= 85.7%
	D=>B	60/80= 75%
C,D	C=>D	70/75= 93.3%
	D=>C	70/80= 87.5%
A,B,D	(A,B)=>D	55/65=84.6%
	(A,D)=>B	55/55=100%
	(B,D)=>A	55/60=91.6%

Here we also obtain those rules which were left underestimated by the Apriori Algorithm.

IV. EXPERIMENTAL RESULTS

To evaluate the efficiency of our proposed algorithm we have extensively studied the performance of our algorithm (IP-Apriori) in comparison to the Apriori algorithm. The algorithms are implemented using Weka in Java API and run on a 2.13 GHz Intel Core i3 CPU with 3 GB of RAM and 300 GB Hard Disk running the 32-bit Windows 7 Home Premium operating system. *The parameters for comparison between Apriori and IP-apriori would be*:

(i) Minimum, Maximum & Average confidence
(ii) Time consumed
(iii) Number of Rules generated
  a) Total
  b) With confidence≥0.9
  c) With confidence≥0.8
  d) With confidence≥0.7
  e) With confidence<0.7

The Test Database to be used by us for the purpose of comparison with the Apriori algorithm is Authentic and recognized i.e. Monk Problem Dataset, Fitting Contact Lenses Dataset and Breast Cancer Wisconsin Original Dataset. We use these datasets for the experiment.

(1). Monk's Problem Dataset-1

The Monk's Problems were the basis of first international comparison of learning algorithms. There are three Monk's Problems Datasets out of which we are using the first Monk's Problem Dataset having 7 attributes.

a) Data Set Information: The Monk's Problems were the basis of first international comparison of learning algorithms summarized in "The Monk's Problem- A Performance comparison of different Learning Algorithms" Technical Report, CS-CMU, Carnegie Mellon University, Pittsburgh, USA, Dec 1991, pp-91-197.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

b) Donor: Sebastian Thrun, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA-15213, USA.
c) Date Donated: 01-10-1992
d) Attribute Information
(i) a1: 1,2,3
(ii) a2: 1,2,3
(iii) a3: 1,2
(iv) a4: 1,2,3
(v) a5: 1,2,3,4
(vi) a6: 1,2
(vii) class: 0,1
e) Results:

Monk Problem Dataset Results are shown in Table XVI (Min-Max Confidence, Total Rules & Time required) & Table XVII (Rules with Conf≥90%, Conf≥80%, Conf≥70%, Conf<70%).

From the results (Fig. 2) we came to know that Minimum & Average confidence is better in case of IP-Apriori than any case of Apriori Algorithm. The rules generated in IP-Apriori are better as rules with Conf≥90% are much more than any case of Apriori. But the time consumed in IP-Apriori is more than Apriori. (i.e. out of 1653 total rules, 1653 are the rules with Conf≥90%.

Table XVI: Monk Problem Dataset Results

Monk Problem	*Apriori*				*IP-Apriori*
Dataset	***min sup=0.06***	***min sup=0.07***	***min sup=0.08***	***min sup=0.09***	***Avg Sup=0.0743***
confmin	0.1029	0.1323	0.147	0.1617	1
confmax	1	1	1	1	1
confavg	0.354	0.3767	0.3878	0.4001	1
Total Rules	2924	1406	1018	784	1653
Time Required (milisecond s)	320±10	245±10	230±10	210±10	1450±10

Table XVII: Number of Rules (Monk Problem Dataset)

Monk Problem	*Apriori*				*IP-Apriori*
Dataset	***min sup=0.06***	***min sup=0.07***	***min sup=0.08***	***min sup=0.09***	***Avg Sup=0.0743***
Rules with conf >= 0.9	44	20	14	12	1653
Rules with conf >= 0.8	33	13	7	6	0
Rules with conf >= 0.7	65	27	24	16	0
Rules with conf < 0.7	2782	1346	973	750	0

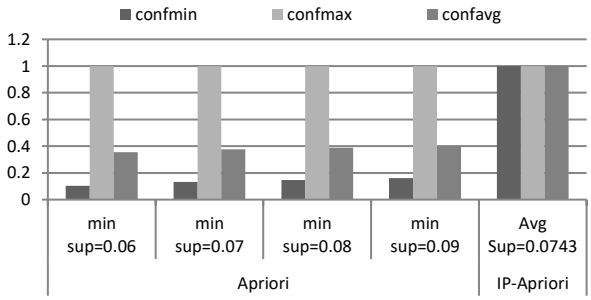

Fig. 2: Monk Problem: Comparison of Min, Max & Average Confidence of Rules in Apriori & IP-Apriori

(2). Fitting Contact Lenses Dataset

The Dataset helps to know that patient should be fitted with hard contact lenses, soft contact lenses or none.

a) Data Set Information: The Fitting Contact Lenses Problems is summarized in Cendrowska, J. "PRISM: An algorithm for inducing modular rules", International Journal of Man-Machine Studies, 1987, pp-349-370.
b) Donor: Benoit Julien
c) Date Donated: 01-08-1990
d) Attribute Information
(i) age: young, pre-presbyopic, presbyopic
(ii) spectacle-prescription: myope, hypermetrope
(iii) astigmatism: no, yes
(iv) tear-prod-rate: reduced, normal
(v) contact-lenses: soft, hard, none

e) *Results:*

Fitting Contact Lenses Dataset Results are shown in Table XVIII (Min-Max, Average Confidence, Total Rules & Time required) & Table XIX (Rules with Conf≥90%, Conf≥80%, Conf≥70%, Conf<70%).

From the results we came to know that Min-Max & Average confidence is same in case of IP-Apriori as some cases of Apriori Algorithm (with minsup = 0.07, 0.08, 0.09). The rules generated in IP-Apriori are also same as these above defined cases of Apriori. But in IP-Apriori is better in case than Apriori when minsup=0.06. Also we don't need to find the minimum support for finding the best rules in IP-Apriori as in the case of Apriori.

Table XVIII: Fitting Contact Lenses Dataset Results

Fitting contact Lenses	*Apriori*				*IP-Apriori*
Dataset	***min sup=0.06***	***min sup=0.07***	***min sup=0.08***	***min sup=0.09***	***Avg Sup=0.0644***
confmin	0.0666	0.1333	0.1333	0.1333	0.1333
Confmax	1	1	1	1	1
Confavg	0.4204	0.4204	0.4204	0.4204	0.4204
Total	2682	966	966	966	966

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Rules					
Time Required (milisecon ds)	285±10	195±10	195±10	195±10	395±10

Table XIX: Number of Rules (Fitting Contact Lenses Dataset)

Fitting contact Lenses	*Apriori*				*IP-Apriori*
Dataset	***min sup=0.06***	***min sup=0.07***	***min sup=0.08***	***min sup=0.09***	***Avg Sup=0.0644***
Rules with conf >= 0.9	233	83	83	83	83
Rules with conf >= 0.8	6	6	6	6	6
Rules with conf >= 0.7	19	19	19	19	19
Rules with conf < 0.7	2424	858	858	858	858

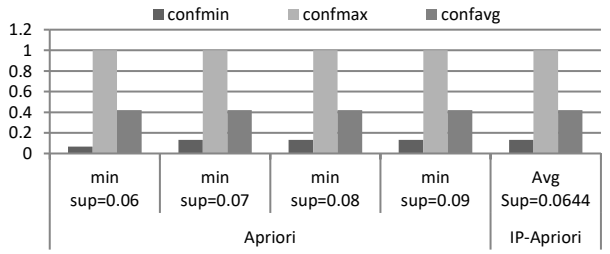

Fig. 3: Fitting Contact Lenses Dataset: Comparison of Minimum and maximum confidence of Rules in Apriori & IP-Apriori

(3). Breast Cancer Wisconsin Original Dataset

The Dataset contains the clinical cases of Dr. Wolberg for Breast Cancer.

a) Data Set Information: The Dataset is created by Dr. William H.Wolberg working as Physician in University of Wisconsin Hospitals, Madison, Wisconsin, USA. The Datasets contains information of the patients with breast Cancer in 10 attributes.

b) Donor: Olvi Mangasarian.

c) Date Donated: 15-07-1992

d) Attribute Information

(i) age: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99

(ii) menopause: lt40, ge40, premeno

(iii) tumor-size: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59

(iv) inv-nodes: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35,36-39

(v) node-caps: yes, no

(vi) deg-malig: 1, 2, 3

(vii) breast: left, right

(viii) breast-quad: left_up, left_low, right_up, right_low, central

(ix) irradiat: yes, no

(x) class: no-recurrence-events, recurrence-events

e) Results:

Breast Cancer Dataset Results are shown in Table XX (Min-Max Confidence, Total Rules & Time required) & Table XX1 (Rules with Conf≥90%, Conf≥80%, Conf≥70%, Conf<70%).

From the results we came to know that Min & Max confidence of IP-Apriori is same in some cases (when minsup=0.09, 0.1 in Apriori) but average confidence is better in those cases. And in one case (when minsup=0.2 in Apriori) IPApriori is better in Max confidence than Apriori Algorithm. The rules generated in IP-Apriori is more in a case (when minsup= 0.02) & also rules with Conf≥90% are also more in that case. And in case (when minsup=0.09,0.1) IP-Apriori also performs better as it limits the rules to 3129 while Apriori make rules 6000. Also we don't need to find the minimum support for finding the best rules in IP-Apriori as in the case of Apriori.

Table XX: Breast Cancer Dataset Results

Breast Cancer Dataset	*Apriori*			*IP-Apriori*
	min sup=0.09	***min sup=0.1***	***min sup=0.2***	***Avg Sup=0.1555***
Rules with conf >= 0.9	525	525	102	273
Rules with conf >= 0.8	752	752	164	371
Rules with conf >= 0.7	670	670	144	354
Rules with conf < 0.7	4253	4253	676	2151

Table XXI: Number of Rules (Breast Cancer Dataset)

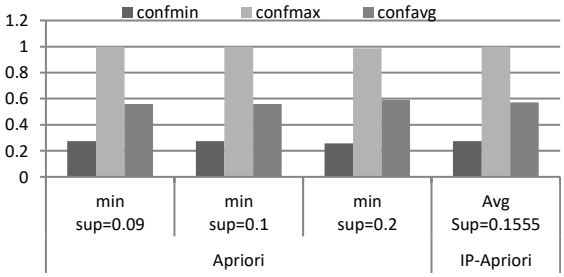

Fig. 4: Breast Cancer Dataset: Comparison of min-max & average confidence of Rules in Apriori & IP-Apriori

V. CONCLUSION & FUTURE SCOPE

This paper proposes and discussed the new algorithm IP-Apriori which is different in various ways than original apriori. In original Apriori, for better rules we always have to try various users' specified minimum supports (i.e. sup_{min}) to obtain the specified required number of rules. But in case of IP-Apriori, we obtain the required number of rules with the use of average support (i.e. sup_{avg}) in single go.

From the above readings we came to know that IP-Apriori algorithm is clearly better than the original Apriori algorithm either in case of confidence (max, min or average) or in case of number of rules (total rules, rules with confidence>90%).

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

But the results are becoming better at the sake of time also (as clear from the readings above).

In future we can implement some optimization method on the rules created by IP-Apriori to enhance the results.

REFERENCES

1. R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases" *ACM SIGMOD International Conference on Management of Data*, Washington, D.C., 1993, pp 207-216.
2. U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From data mining to knowledge discovery: an overview" *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, 1996, MA.
3. U. Fayyad, S. G. Djorgovski and N. Weir, "Automating the analysis and cataloging of sky surveys" *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, MA, 1996, pp. 471-94.
4. Technology Forecast, Price Waterhouse World *Technology Center*, Menlo Park, CA, 1997.
5. Sotiris Kotsiantis, Dimitris Kanellopoulos, "Association Rules Mining: A Recent Overview" *GESTS International Transactions on Computer Science and Engineering*, vol.32 (1), 2006, pp. 71-82.
6. Rakesh Agarwal, Ramakrishna Srikant, "Fast Algorithm for mining association rules" *VLDB Conference Santiago, Chile*, 1994, pp 487-499.
7. Suhani Nagpal, "Improved Apriori Algorithm using logarithmic decoding and pruning" *International Journal of Engineering Research and Applications*, vol. 2, issue 3, 2012, pp. 2569-2572.
8. Sang Jun Lee, Keng Siau, "A review of data mining techniques. Industrial Management and Data Systems", *University of Nebraska-Lincoln Press*, USA, 2001, pp 41-46.
9. Huan Wu, Zhigang Lu, Lin Pan, Rong Seng XU and Wenbao jiang, "An improved Apriori based algorithm for association rule mining" *IEEE Sixth international conference on fuzzy systems and knowledge discovery*, 2009, pp 51-55.
10. Farah Hanna AL-Zawaidah, Yosef Hasan Jbara and Marwan AL-Abed Abu-Zanana, "An improved Algorithm for mining Association Rule in large database" *World of Computer and Information technology*, vol. 1, no. 7, 2011, pp 311-316.
11. S. A. Abaya, "Association Rule Mining based on Apriori Algorithm in Minimizing Candidate generation" *International Journal of Scientific & Engineering Research*, vol-3, issue 7, 2012.
12. Zhuang Chen, Shibang Cai, Quilin Song, Chonglai Zhu, "An Improved Apriori Algorithm based on pruning Optimization and transaction reduction" *IEEE transactions on evolutionary computation*, 2011, pp 1908-1911.
13. M. S. Chen, J. Han, and P. Yu, "Data mining: an overview from a database perspective" *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, 1996, pp. 866-883.
14. R. Patel Nimisha, Sheetal Mehta, "A Survey on Mining Algorithms" *International Journal of Soft Computing and Engineering*, vol. 2, issue 6, 2013, pp 460-463.

AUTHOR PROFILE

Mr. Prince Verma, PG Scholar, pursuing M.Tech in Computer Science under Department of Computer Science, DAV Institute of Engineering & Technology, Jalandhar, India. He has completed his under graduate course B.Tech in Computer Science under Department of Computer Science from Malout Institute of Management & Information Technology, Malout, India. Areas of Interest are Data Mining and Optimization Techniques like genetic & Ant Colony optimization

Mr. Dinesh Kumar, Associate Professor & Head, Department of Information Technology, DAV Institute of Engineering & Technology. He has 13 years of Teaching Experience. He is pursuing his Ph.D from Punjabi University, Patiala. He has completed his M.Tech in Information Technology under Department of Information Technology from Punjabi University, Patiala. His Research Interests are Data mining and Speech Reorganization.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)