IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 4, June 2014.
www.ijiset.com
ISSN 2348 – 7968

Performance Based Study of Association Rule Algorithms On Voter DB

K.Padmavathi[1], R.Aruna Kirithika[2]
[1]*Department of BCA, St.Joseph's College, Thiruvalluvar University, Cuddalore, Tamil Nadu, India,* sivapadma2009@gmail.com

[2]*Department of BCA, St.Joseph's College, Thiruvalluvar University, Cuddalore, Tamil Nadu, India,* ashassss79@gmail.com

Abstract
Data mining is widely done in recent times to predict results from the existing wide range of data that fits our research. On this basis a voting data base is studied to find out the interest of the voters among the attributes given using the some Associative Rule data mining algorithms. The Association Rule algorithm studies the frequent items that are being used in the data base. A comparative study of the Associate Rule (FP-Growth & Apriori) algorithms is used on the voter data set and the results are compared in this paper. The results suggest the reliability of FP-Growth algorithm of Association rule than the Apriori algorithm in terms of time consumed when dealing with huge number of instances, minimum number of scans used on the complete dataset and Minimum utilization of memory.

Keywords: *frequent items, associate rule, attributes etc.*

1. Introduction

Analysis of the voter data set with the association rule algorithms included about 17 key attributes and more than 600 instances. The Data set vote.arff file contains records that are taken online after having surveyed of the attitude of the parties that go for elections.
Associate rules are used to mine the frequent items that are used in the database. In order to find the frequent items from the database the associations rules are performed all these rules are useful for the decision maker in order to know which items occurs more frequently in the database based on rules by specifying minimum threshold specified all these items can be extracted. The different algorithms like Apriori and fp-growth are used. In order to determine the minimum threshold value for the Apriori and fp-growth we calculate the support and confidence and by using these values the minimum threshold value is set. The vote data is taken as input and by using the open source machine learning software like weka. In this weka explorer the stages like preprocessing and the association algorithm is applied in the preprocessing stage selection of attributes is carried so that the irrelevant data has been removed from the database by selecting particular type in which the user wants to discover the items.
Association rule mining finds a correlation relationship among a large set of data items according to the relationship between the items the frequent items are discovered and these rules are mined by using the minimum threshold value. Let I= $\{i_1,i_2\ldots i_d\}$ be the set of all items in a vote data and T= $\{t_1,t_2\ldots t_N\}$ be the set of all transactions. Each transaction t_i contains a subset of items chosen from I. In association analysis, a collection of zero or more items is termed an itemset. If an itemset contains k items, it is called an k-itemset. The null set is an itemset that does not contain any items [1].
An association rule is an implication expression of the form X→Y, where X and Y are disjoint item sets i.e. $X \cap Y = \phi$. The strength of an association rule can be measured in terms

of its support and confidence. Support determines how often a rule is applicable to a given data set, while confidence determines how frequently items in Y appear in transactions that contain X. the formal definitions of these metrics are

$$\text{Support, } s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$
$$\text{Confidence } ((X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Where support is an important measure because a rule that has very low support may occur simply by chance.
Where confidence measures the reliability of the inference made by a rule.
Association analysis results should be interpreted with caution. The inference made by an association rule does not necessarily imply causality instead, it suggests a strong co-occurrence relationship between items in the antecedent and consequent of the rule. Causality, on the other hand, requires knowledge about the casual and effect attributes in the data and typically involves relationships occurring over time (e.g., ozone depletion leads to global warming).
A common strategy adopted by many association rule mining algorithms is to decompose the problem into two major subtasks:

- **Frequent Itemset Generation,** whose objective is to find all the item-sets that satisfy the minsup threshold. These itemsets are called frequent itemsets.

- **Rule Generation,** whose objective is to extract all the high- confidence rules from the frequent items called strong rules.

3. Association rule algorithms

Association rules are used to find the frequent pattern, association or correlation in transaction database. Association rule mining can be used in Basket Data Analysis, Educational Data mining, Classification, Clustering etc. Association rule algorithms are Apriori, Sampling, Partitioning and Parallel algorithm.

3.1 Apriori Association Rule

Apriori association rule is used to mine the frequent patterns in database. Support and confidence are the normal method used to measure the quality of association rule.

- Support for the association rule X->Y is the percentage of transaction in the database that contains X⋃Y.
- Confidence for the association rule id X->Y is the ratio of the number of transaction that contains X⋃Y to the number of transactions that contains X.

Terms relates to this algorithm are as follows:

- Frequent Itemsets: The set of item which has minimum support & it is denoted by L_i for i^{th} itemset.
- Apriori Property: Any subset of frequent itemset must be frequent [2].
- Join Operation: To find L_k , a set of candidate K-itemsets is generated by joining L_{k-1} with itself.
- Join Step: Candidate item C_k is generates by joining L_{k-1} with itself [2].
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

The Apriori association algorithm is given below:

Algorithm: Apriori Association Rule Algorithm

Purpose: To find subsets which are common to at least a minimum number C (Confidence Threshold) of the itemsets.

Input: Database of Transactions D={t_1,t_2,….,t_n} Set of items I={I_1,I2,….,I_k} Frequent (Large) Itemset L Support, Confidence.

Output: Association Rule satisfying Support &confidence

Method:

```
1. C1 = Itemsets of size one in I;
2. Determine all large itemsets of size 1, L1;
3. i = 1;
4. Repeat
5. i = i + 1;
6. Ci = Apriori- Gen (Li-1);
7. Apriori -Gen (Li-1)
1. Generate candidates of size i+1 from large itemsets of size i.
2. Join large itemsets of size i if they agree on i-1.
3. Prune candidates who have subsets that are not large.
8. Count Ci to determine Li;
9. until no more large itemsets found;
```

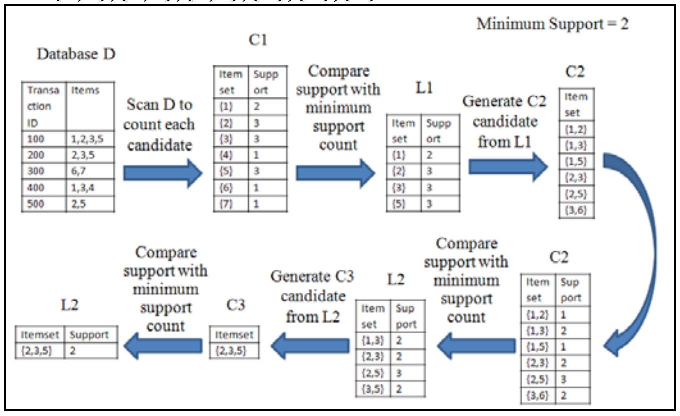
Figure 1 show the generation of itemsets & frequent itemsets where the minimum support count is 2 [2].

To generate the association rule from frequent itemset we use the following rule:

- For each frequent itemset L, find all nonempty subset of L
- For each nonempty subset of L, write the association rule S → (L-S) if support count of L/support count of S >= Minimum Confidence

The best rule from the itemset L= {2, 3, 5} are calculated as follows:

Consider the minimum support is 2 & minimum confidence is 70%. All nonempty subset of {2, 3, and 5} are: {2,3},{2,5},{3,5},{2},{3},{5}.


Minimum Support = 2
Database D
C1
Scan D to count each candidate
Compare support with minimum support count
L1
Generate C2 candidate from L1
C2
C2
Compare support with minimum support count
L2
Generate C3 candidate from L2
C3
Compare support with minimum support count
L2

Database D

Transaction ID	Items
100	1,2,3,5
200	2,3,5
300	6,7
400	1,3,4
500	2,5

C1

Item set	Support
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3
{6}	1
{7}	1

L1

Item set	Support
{1}	2
{2}	3
{3}	3
{5}	3

C2

Item set
{1,2}
{1,3}
{1,5}
{2,3}
{2,5}
{3,6}

C2

Item set	Sup port
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,6}	2

L2

Item set	Sup port
{1,3}	2
{2,3}	2
{2,5}	3
{3,5}	2

C3

Itemset
{2,3,5}

L2

Itemset	Support
{2,3,5}	2

Fig -1: Generation of itemsets & frequent itemsets

Rule 1: {2, 3} → {5} Confidence = Support Count of ({2, 3, 5})/ Support Count of ({2, 3}) = 2/2 = 100%
Rule 2: {2, 5} → {3} Confidence = Support Count of ({2, 3, 5})/ Support Count of ({2, 5}) = 2/3 = 67%
Rule 3: {3, 5} → {2} Confidence = Support Count of ({2, 3, 5})/ Support Count of ({3, 5}) = 2/2 = 100%
Rule 4: {2} → {3, 5} Confidence = Support Count of ({2, 3, 5})/ Support Count of ({2}) = 2/3 = 67%
Rule 5: {3} → {2, 5} Confidence = Support Count of ({2, 3, 5})/ Support Count of ({3}) = 2/3 = 67%
Rule 6: {5} → {2, 3} Confidence = Support Count of ({2, 3, 5})/ Support Count of ({5}) = 2/3 = 67%

Hence the accepted rules are Rule 1 & Rule 3 as the confidence of these rules is greater than 70% [10].

In this paper, we use Apriori Association algorithm to find out the result i.e. best combination of attributes, after the preprocessing of data stage. In Weka the option available with Apriori association rule algorithm are car, class Index, delta, lower bound minimum support, metric type, minimum metric, number of rules, output itemsets, remove all missing columns, significance level, upper bound minimum support, verbose[2].

3.2 FP- Growth Algorithm

The FP- growth algorithm encodes the data set using a compact data structure called an FP-tree and extracts frequent itemsets directly from this structure [4].

3.2.1 FP-Tree Representation

An FP-tree is a compressed representation of the input data. It is constructed by reading the data set one transaction at a time and mapping each transaction onto a path in the FP-tree. As different transactions can have several items in common, their paths may overlap [4]. The more the paths overlap with one another, the more compression we can achieve using the FP- tree structure. If the size of the FP-tree is small enough to fit into main memory, this will allow us to extract frequent item sets directly from the structure in memory instead of making repeated passes over the data stored on disk [3].

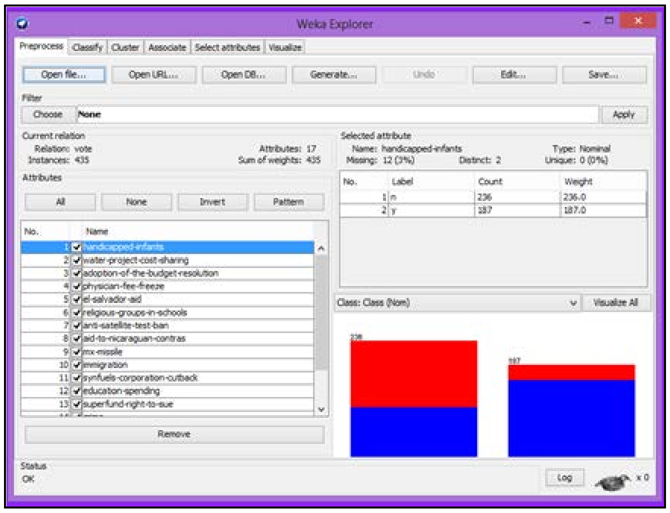
3.2.2 Pseudo code for FP- Growth:

```
Input: A transactional database and a minimum support threshold €
Output: Its frequent pattern tree, FP-tree.


Method: call FP-growth (FP-tree, null)
Procedure FP-growth (Tree A)
{

If Tree contains a single path P
        then for each combination of the nodes in the path P do
        generate pattern B⋃A with support =minimum support of nodes in B
else for each ai in the header of the Tree do
{
        Generate pattern B=ai⋃A with support =ai.support;
        Construct B’s conditional pattern base and B’s conditional FP-tree

Tree B;
        If Tree B≠ϴ
        Then call Fp- growth (Tree B, B)
        }
        }
```

4. Experimental results:

By using weka tool the data set selected is vote.arff data is taken as the input and the results obtained is as follows

The results obtained after preprocessing state and by visualizing all the classes are shown in fig 2 and fig 3



Filter: None

Current relation
Relation: vote
Instances: 435
Attributes: 17
Sum of weights: 435

Selected attribute
Name: handicapped-infants
Type: Nominal
Missing: 12 (3%)
Distinct: 2
Unique: 0 (0%)

No.	Label	Count	Weight
1	n	236	236.0
2	y	187	187.0

Attributes

No.	Name
1	handicapped-infants
2	water-project-cost-sharing
3	adoption-of-the-budget-resolution
4	physician-fee-freeze
5	el-salvador-aid
6	religious-groups-in-schools
7	anti-satellite-test-ban
8	aid-to-nicaraguan-contras
9	mx-missile
10	immigration
11	synfuels-corporation-cutback
12	education-spending
13	superfund-right-to-sue

Class: Class (Nom)

Status
OK

Fig -2: preprocessing state of the db

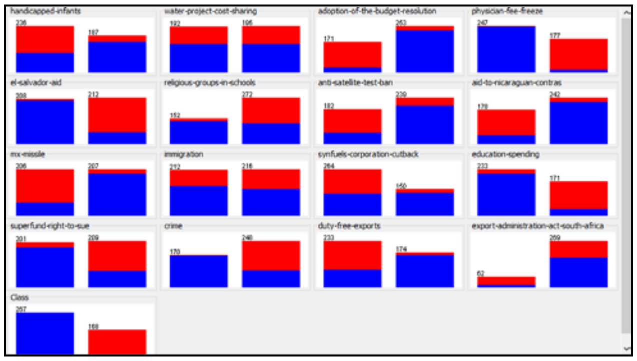
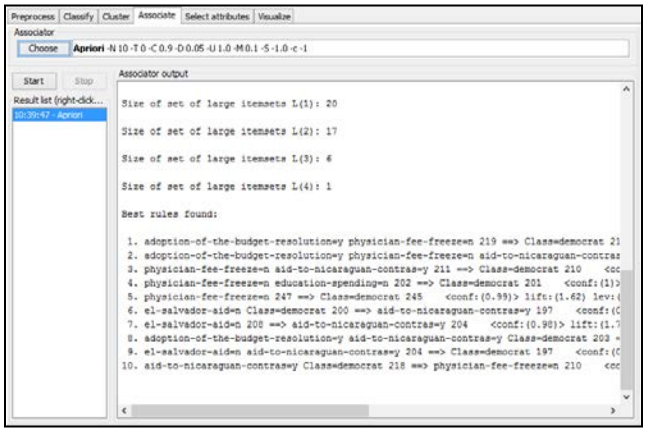


Fig -3: Visualization of results.

The frequent rules items obtained by using the Apriori algorithm on vote data shown in fig 4

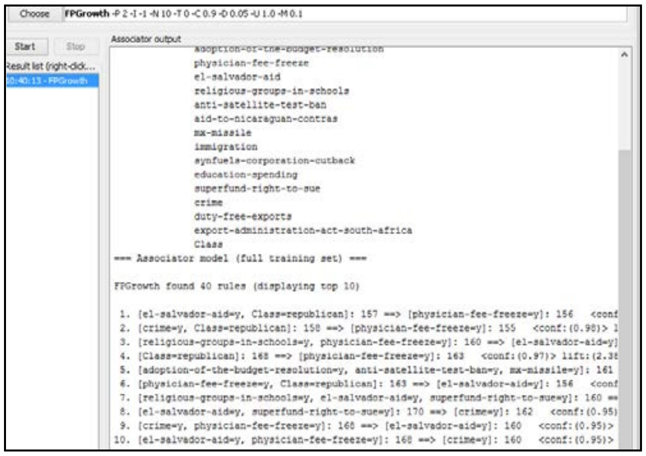


Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

```
Size of set of large itemsets L(1): 20

Size of set of large itemsets L(2): 17

Size of set of large itemsets L(3): 6

Size of set of large itemsets L(4): 1

Best rules found:

 1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 21
 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-contras
 3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210    <cc
 4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201    <conf:(1)>
 5. physician-fee-freeze=n 247 ==> Class=democrat 245    <conf:(0.99)> lift:(1.62) lev:(
 6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197    <conf:(C
 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204    <conf:(0.98)> lift:(1.7
 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 203 =
 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197    <conf:(C
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210    <cc
```

Fig -4: Apriori algorithm on vote data

The frequent rules items obtained by using the FP-Growth algorithm on vote data shown in fig 5



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1

```
              adoption-of-the-budget-resolution
              physician-fee-freeze
              el-salvador-aid
              religious-groups-in-schools
              anti-satellite-test-ban
              aid-to-nicaraguan-contras
              mx-missile
              immigration
              synfuels-corporation-cutback
              education-spending
              superfund-right-to-sue
              crime
              duty-free-exports
              export-administration-act-south-africa
              Class
=== Associator model (full training set) ===

FPGrowth found 40 rules (displaying top 10)

 1. [el-salvador-aid=y, Class=republican]: 157 ==> [physician-fee-freeze=y]: 156   <conf
 2. [crime=y, Class=republican]: 158 ==> [physician-fee-freeze=y]: 155   <conf:(0.98)> l
 3. [religious-groups-in-schools=y, physician-fee-freeze=y]: 160 ==> [el-salvador-aid=y]
 4. [Class=republican]: 168 ==> [physician-fee-freeze=y]: 163   <conf:(0.97)> lift:(2.38
 5. [adoption-of-the-budget-resolution=y, anti-satellite-test-ban=y, mx-missile=y]: 161
 6. [physician-fee-freeze=y, Class=republican]: 163 ==> [el-salvador-aid=y]: 156   <conf
 7. [religious-groups-in-schools=y, el-salvador-aid=y, superfund-right-to-sue=y]: 160 ==
 8. [el-salvador-aid=y, superfund-right-to-sue=y]: 170 ==> [crime=y]: 162   <conf:(0.95)
 9. [crime=y, physician-fee-freeze=y]: 168 ==> [el-salvador-aid=y]: 160   <conf:(0.95)>
10. [el-salvador-aid=y, physician-fee-freeze=y]: 168 ==> [crime=y]: 160   <conf:(0.95)>
```

Fig -5: FP-Growth algorithm on vote data

5. Comparative results:

Apriori and FP- growth for associative rules algorithms.

The Experimental results obtain using voter dataset using the Weka tool shows the following results.

5.1 Technique Used

a. Apriori algorithm --- uses join and prune property

b. FP-Growth ----------- Constructs conditional frequent pattern tree and conditional from database with minimum support.

5.2 Memory Utilization

a. Apriori ---------- requires large memory space for large number of candidates.

b. FP-Growth ----- requires less memory since no candidate generation is done.

5.3 No. Of Scan

a. Apriori --------- does multiple scans for generating candidate sets.

b. FP- Growth --- scans only twice the database.

5.4 Time

a. Apriori ------ Execution time is more as time is wasted in producing candidates every time.

b. Fp-growth – consumes less time than the apriori algorithm.

6. Conclusion:

Given the vote data set, the performance of the Apriori and FP-growth algorithms are analyzed under the association rule algorithm. The results show that the FP-Growth algorithm consumes less time even for large number of instances than the apriori algorithm. The FP-growth algorithm constructs more frequent patterns than the apriori algorithm. Scanning the complete database is done only twice in Fp-Growth but the apriori algorithm does multiple times. Both apriori and Fp-Growth are aiming to find out complete set of patterns but, FP-growth is more efficient than Apriori in respect to long patterns proving the reliability of Fp-Growth algorithm towards Apriori algorithm.

7. References

[1] Alex Tze Hiang Sim, Maria Indrawan, Samar Zutshi, Member,IEEE, and Bala Srinivasan,"*Logic-Based Pattern Discovery*," Ieee transactions on knowledge and data engineering, vol. 22,no.6,2010.

[2] R. Agrawal, T. Imielinski, and A. Swami, "*Mining Association Rules between Sets of Items in Large Databases*," SIGMOD Record, vol. 22, pp. 207-216, 1993.

[3] Mohammed J. Zaki, "*Scalable algorithms for association mining*," IEEE Transactions on Knowledge and Data Engineering, 12(3):372-390, May/June 2000.

[4] Frequent Pattern Growth (FP-Growth) Algorithm An Introduction Florian Verhein fverhein@it.usyd.edu.au School of Information Technologies, The University of Sydney, Australia Copyright 2008 Florian Verhein.

K.Padmavathi has completed her M.Sc., and M.Phil. degrees in Computer science., from Thiruvalluvar University, Vellore, Tamil Nadu, India, during the year 2010 and 2011 respectively. She has been working with St.Joseph's college of Arts and Science College, Cuddalore, Tamil Nadu, India as Assistant

Professor for the last 3 years. She has attended number of national/ International conferences in and around Tamil Nadu, India ever since her completion of the degrees. Presently interested and working with the data mining algorithms (Association, clustering etc…).She has also published in international journal IJCSMA on the data mining V2I409.

R.Aruna Kirithika has completed her M.C.A., and M. Phil., degrees in Computer science, from Annamalai University, Chidambaram and Alagappa University, Tamil Nadu, India, during the year 2002 and 2008 respectively. She has worked as guest faculty in Pondicherry University in the Bioinformatics center for 3 years from 2006-2009. She has been working with St.Joseph's college of Arts and Science College, Cuddalore, Tamil Nadu, India as Assistant Professor for the last 5 years. She has attended number of national/ International conferences in and around Tamil Nadu, India ever since her completion of the degrees. Presently interested and working with the data mining algorithms (Association, clustering etc…).She has also published in international journal IJCSMA on the data mining V2I409.