INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 3)

Available online at: www.ijariit.com

Comparative study on Apriori and FP growth algorithms in big data

Reena Lobo
reenalobo321@gmail.com
Alva's Institute of Engineering and Technology, Mangaluru, Karnataka

Venkatesh
venkateshbhat2007@gmail.com
Alva's Institute of Engineering and Technology, Mangaluru, Karnataka

ABSTRACT

A dataset is a collection of data. Today a huge amount of the data is being captured by information sensing devices such as mobiles, computers, sensors etc. This huge amount of the data are now called as big data. Frequent Itemset mining is a tool for identifying the frequently occurring items together. There are many frequent Itemset mining algorithms like apriori, Eclat and FP growth. In the proposed work we use FP growth algorithm and compare it with Apriori algorithm and we show that FP growth algorithm is better when compared to the apriori algorithm.

Keywords: *Dataset, FP growth, Apriori.*

1. INTRODUCTION

Big data is a collection of the huge dataset that cannot be processed using traditional computing technique. Big data involves huge volume, huge velocity and variety of data. The data present in the big data will be of 3 types they are structured data, semi-structured and unstructured data. Frequent Itemset is to determine what items are typically brought together. For example which items customers typically buy together in a database of online shopping transactions? There are many frequent Itemset algorithms they are Apriori, Eclat, and FP growth etc. These algorithms have the same input and same output, the input is transaction database and a minimum support threshold. The difference between these algorithms is the way in which they generate the output. A minimum support threshold is applied to find all frequent Itemset in the database.

2. RELATED WORK

In [1], authors have introduced the problem of "mining" a large collection of basket type transaction for association rules between sets of items with the same threshold.
In [2], authors have implemented the parallel apriori algorithms based on MapReduce, which makes it applicable to new association rules from a large database of transactions.
In [3], authors have presented the problem of frequent Itemset mining and techniques to explore the search space of Itemset employed by Itemset mining algorithm.

3. PROPOSED WORK

In the proposed technique we study the apriori algorithm and FP growth algorithms and we conclude that the FP growth algorithm is best when compared to apriori algorithm to generate the frequent patterns.

A. APRIORI ALGORITHM

Apriori algorithm is used to mine the frequent Itemset in the transactional database. The algorithm was discovered by Indians R Agrawal and R.Srikant in 1994. Two properties used in apriori algorithm are,

- Downward closure property: This property of frequent pattern says that subset of any frequent Itemset must be frequent.
- Apriori pruning principal: If any Itemset which is infrequent its super set should not be tested or generated.ie if the subset is infrequent then super set must also be infrequent. It follows the rule of set theory known as antigenotoxicity.

These two properties will reduce the time complexity in the algorithm.

How Apriori Works

1. Find all frequent Itemsets:
- Get frequent items:
✓ Items whose occurrence in the database is greater than or equal to the min.support threshold.
- Get frequent itemsets:
✓ Generate candidates from frequent items.
✓ Prune the results to find the frequent itemsets.
2. Generate strong association rules from frequent itemsets
    - Rules which satisfy the min.support and min.confidence threshold.

Example:

Original table:

Transaction ID	Items Bought
T1	{M, O, N, K, E, Y }
T2	{D, O, N, K, E, Y }
T3	{M, A, K, E}
T4	{M, U, C, K, Y }
T5	{C, O, O, K, I, E}

Step 1: Count the number of transactions in which each item occurs, Note 'O' is bought 4 times in total, but, it occurs in just 3 transactions.

Item	No of transactions
M	3
O	3
N	2
K	5
E	4
Y	3
D	1
A	1
U	1
C	2
I	1

Step 2: Now remember we said the item is said frequently bought if it is bought at least 3 times. So in this step, we remove all the items that are bought less than 3 times from the above table and we are left with

Item	Number of transactions
M	3
O	3
K	5
E	4
Y	3

This is the single items that are bought frequently. Now let's say we want to find a pair of items that are bought frequently. We continue from the above table (Table in step 2)

Step 3: We start making pairs from the first item, like MO, MK, ME, MY and then we start with the second item like OK,OE,OY. We did not do OM because we already did MO when we were making pairs with M and buying a Mango and Onion together is same as buying Onion and Mango together. After making all the pairs we get,

Item pairs
MO
MK
ME
MY
OK
OE
OY
KE
KY
EY

Step 4: Now we count how many times each pair is bought together. For example, M and O is just bought together in {M, O, N, K, E, Y}

While M and K is bought together 3 times in {M,O,N,K,E,Y}, {M,A,K,E} AND {M,U,C, K, Y}
After doing that for all the pairs we get

Item Pairs	Number of transactions
MO	1
MK	3
ME	2
MY	2
OK	3
OE	3
OY	2
KE	4
KY	3
EY	2

Step 5: Golden rule to the rescue. Remove all the item pairs with a number of transactions less than three and we are left with

Item Pairs	Number of transactions
MK	3
OK	3
OE	3
KE	4
KY	3

These are the pairs of items frequently bought together.
Now let's say we want to find a set of three items that are brought together.
We use the above table (table in step 5) and make a set of 3 items.

Step 6: To make the set of three items we need one more rule (it's termed as self-join),
It simply means, from the Item pairs in the above table, we find two pairs with the same first Alphabet, so we get

- OK and OE, this gives OKE
- KE and KY, this gives KEY

Then we find how many times O,K,E are bought together in the original table and same for K,E,Y and we get the following table

Item Set	Number of transactions
OKE	3
KEY	2

While we are on this, suppose you have sets of 3 items say ABC, ABD, ACD, ACE, BCD and you want to generate item sets of 4 items you look for two sets having the same first two alphabets.

- ABC and ABD -> ABCD
- ACD and ACE -> ACDE

And so on … In general, you have to look for sets having just the last alphabet/item different.

Step 7: So we again apply the golden rule, that is, the item set must be bought together at least 3 times which leaves us with just OKE, Since KEY are bought together just two times.
Thus the set of three items that are bought together most frequently are O,K,E.

B. FP GROWTH ALGORITHM

FP-Growth: allows frequent Item set discovery without candidate Itemset generation. Two-step approach:

- Step 1: Build a compact data structure called the FP-tree
  - Built using 2 passes over the data-set.
- Step 2: Extracts frequent Itemset directly from the FP-tree.

FP tree compress database into a tree. In this we don't scan the database, again and again, It will reduce the time.Consider the following example, the original table is given below and the support count is 3.

Transaction ID	Items Bought
T1	{M, O, N, K, E, Y }
T2	{D, O, N, K, E, Y }
T3	{M, A, K, E}
T4	{M, U, C, K, Y }
T5	{C, O, O, K, I, E}

Steps to follow:

- Analyze the data based on min support.

Item	Number of transactions
M	3
O	3
N	2
K	5
E	4
Y	3
D	1
A	1
U	1
C	2
I	1

- Now write the set L ie frequent item set.It is the one whose min support is>min_sup and writes down the support count decreasing order

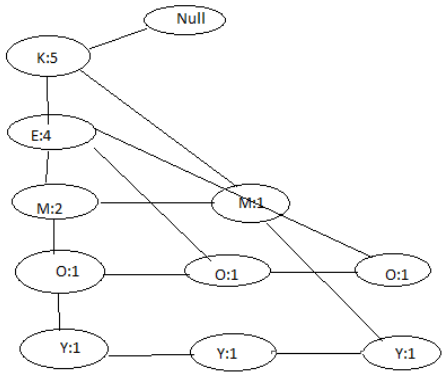
Item	Number of transactions
K	5
E	4
M	3
O	3
Y	3

- Construct FP tree: we required ordered Itemset based on a frequent pattern or Itemset or L.We just scan each transaction in dB in L,

TID	Itemsets	Ordered itemset
T1	M,O,N,K,E,Y	K,E,M,OY
T2	D,O,N,K,E,Y	K,E,O,Y
T3	M,A,K,E	K,E,M
T4	M,U,C,K,Y	K,M,Y
T5	C,O,O,K,I,E	K,E,O

- Construct tree

Item	Number of transactions
K	5
E	4
M	3
O	3
Y	3



Now write the items in L in the reverse order of frequency based on each item we find in conditional pattern base.

Item	Conditional pattern base
Y	{K,E,M,O:1},{K,E,O:1},{K,M:1}
O	{K,E,M:1},{K,E:2}
M	{K,E:2},{K:1}
E	{K:4}
K	-

- Write the conditional FP tree based on conditional pattern base

Item	
Y	K:3
O	KE:3
M	K:3
E	K:4
K	-

- Generate frequent pattern from Y,O,E,M
  Y-{K, Y:3},
  O-{K.O:3},
  {E,O:3},{O,E,K:3}
  M-{M,K:3}
  E-{E,K:3}
- Find Out The Association Rules
  ie from the above pattern {K,Y}
  There are two association rules ie either k implies y or y implies k. If the minimum support is provided in the question then we can find strong rules from association rules

4. CONCLUSION

The main drawback of apriori algorithm is at each step we have to calculate candidate sets of the pattern and we have to repeatedly scan the database in apriori algorithm which requires more times. These drawbacks can be overcome by using FP growth algorithm in which we scan the database only twice and it uses a divide and conquers method.

5. REFERENCES

[1] Rakesh Agrawal, Tomasz Imielinski, Arun SwamiS "Mining Association Rules between Sets of Items in Large Databases". IBM Almaden Research Center 650 Harry Road, San Jose, CA 95120.

[2] Ning Li*, Li Zeng, Qing He, Zhongzhi Shi "Parallel Implementation of Apriori Algorithm Based on MapReduce" International Journal of Networked and Distributed Computing, Vol. 1, No. 2 (April 2013), 89-96 .

[3] Viger ∗, Jerry Chun-Wei Lin†, Bay Vo‡§ , Tin Truong Chi¶, Ji Zhangk, Hoai Bac Le," A Survey of Itemset Mining".

[4] Gole, Sheela, and Bharat Tidke. "Frequent itemset mining for Big Data in social media using ClustBigFIM algorithm", 2015 International Conference on Pervasive Computing (ICPC), 2015.

