FPC+: Algorithm for Mining of Frequent Closed Itemsets

Darshan Modi[1] Namrata Shroff[2]
[1]M.E. Student [2]Assistant Professor
[1,2]Department of Computer Engineering
[1,2]Government Engineering College, Gandhinagar, Gujarat, India

Abstract— In this paper, we presents a new algorithm for finding closed frequent item sets, which is an abridged and lossless demonstration of every frequent itemsets that able to mine from a transactional database. We used a divide-and-conquer method and follow a specific visiting and partitioning method of the search space found on a real theoretic framework, which describes the difficulty of closed itemsets mining in very much detailing. The algorithm takes some exaggeration targeted to minimize time in finding itemset supports and their closures. We propose a pruning technique using Lattice method and Linkage Disequilibrium method, which, dislike other past methods, does not demand the full set of closed patterns mined so long and to be hold in the main memory. This algorithm also passes every passed partition of the search space to be mined freely in any position and in parallel also. We present our new algorithm called FPC+ for frequent closed itemsets and compare it with existing approaches like Apriori, FP-Growth, Closet+ and BCTFI in this paper.

Key words: Data mining, frequent closed itemset mining, frequent closed itemsets, Association rules

I. INTRODUCTION

Data mining, the lineage of invisible predictive flea in ear from wealthy databases, is a powerful polished technology by the whole of great weight to throw in one lot with companies intensifies on the close but no cigar important flea in ear in their front page new warehouses. Data mining tools expect future trends and behaviors, allowing businesses to the way one sees it proactive, knowledge-driven decisions.

Association rules generally standardize one of the significant procedures of data mining. Association rule mining finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories [13]. The volume of data is expanding drastically as the information created by daily practices. So, mining association rules from huge amount of data in the database is fascinated for some ventures which can help out in numerous business basic leadership procedures, for instance, promotion assortment, cross-marketing and Basket data study. The systems for finding association rules from the data have customarily centered on distinguishing relations between things letting several know part of humanoid nature, normally purchasing nature for deciding things that customer buys at the same time. All tenets of this sort show a such local pattern. The aggregation of association rules can be effortlessly interpreted and imparted.

There have been a more number of methods created for finding frequent itemsets in very big databases. The size of the database decides effectiveness of algorithm. There are two general strategies taken by these algorithms: the one is an efficient pruning technique to lower the combinatorial search space of candidate itemsets (Apriori techniques). The other technique is to take a compressed data representation to simplify in main itemsets processing (FP-tree techniques).

BCTFI represent the original transaction data by binary 0/1 representation. Using this representation, BCTFI is able to transform the data to decimal number. As in algorithm it can merge transactions and then the frequent itemsets can be obtained from constructing count Table for all items in the merged transactions. The merge process in BCTFI is based on decimal number representation of each transaction.

In the past numerous years, more number of researches have proposed very fast quick algorithms for mining of frequent closed itemsets, such as A-close, CLOSET, CHARM and CLOSET+.

In our study, we systematically study on the search methods and make an algorithm FPC+. FPC+ uses the benefits of the previous proposed efficient methods as well as several ones lately created here. A complete functional practice on real and synthetic data sets has given the benefits of the methods and the betterment of FPC+ above current mining algorithms, along Apriori, BCTFI, FP-Growth and CLOSET+, on base of scalability, memory usage and runtime.

II. PROPOSED SYSTEM

Let I be the set of items that are available for sale. An itemset A = $\{i_1, i_2, \ldots i_k\}$ is a set of items defined over the whole item set I, and is called a k-itemset if it has k number of items.

- A transaction is single sale information and it consists of a transaction ID (TID) and all the items that are included in the sale.
- Support value of an itemset A, denoted as sup(A), is the propo
- rtion of the transactions which include all the items in the set A. Formally, this is represented as:
- sup(A) = Number of transactions containing itemset A / Total number of transactions
- Itemsets satisfying the minimum support value are said to be frequent. As an example, a support value of 40% for an itemset such as A = {milk, butter} simply means that two out of five transactions in the database contain these two items.
- Confidence value of a rule such as B $\subset$ A is dened as:
- conf (A $\rightarrow$ B) = sup(A U B) / sup(A)
- Here the support value of the union of item sets A and B is divided with the support value of only the item set A. This gives us how likely it is that items in B exist given that items in A exist in a transaction.

Further used Linkage disequilibrium produces strongly linked variants tightly correlated generating cost savings for association rules practices.

As in the support case, a minimum confidence value, also called confidence threshold, is used to prune those rules having less confidence value than the minimum. A high confidence value shows a high correlation between the antecedent and the consequent.

The metric shows how frequent an itemset occurs in the database. A method named as "ScalingNextClosure" is created and called which generates multiple dataset (combinational)

Generally, association rules having high confidence values are desired because the high correlation value between items translates into higher probability of these items bought together.

We present a new improved algorithm called as FPC+ using Apriori, FP-Growth and Closet+ algorithms.
FPC+ has two phases as given below:

- First phase: A compact representation of the database using FP-tree structure is built.
- Second phase: Commences where the FP-tree is mined and frequent patterns are found.

FPC+ makes use of several novel techniques for pruning the search space and thus increasing the mining speed.

- The items after removal of min sup will be sorted in decreasing support values.
- The other remaining items will be added to fp-tree in sorted order and pruned removed.
- We can create fp-tree at each process and at the end, will merge to represent the whole database…
- While merging, if the same node is added only the count will be increased and if required will be added from the root to trees in dfs order.

We need to experiment with different databases. We will create its lattice for better representation and traversal. We will use sparse and dense datasets to experiment.

FPC+ algorithm is improved on the base of association-based classification, clustering and dependency/linkage analysis in very big databases.

Our proposed algorithm will add association based classification and will try to implement with large databases with improved results. Our proposed algorithm will add conditional FP tree based algorithm for better results with classification in association mining.

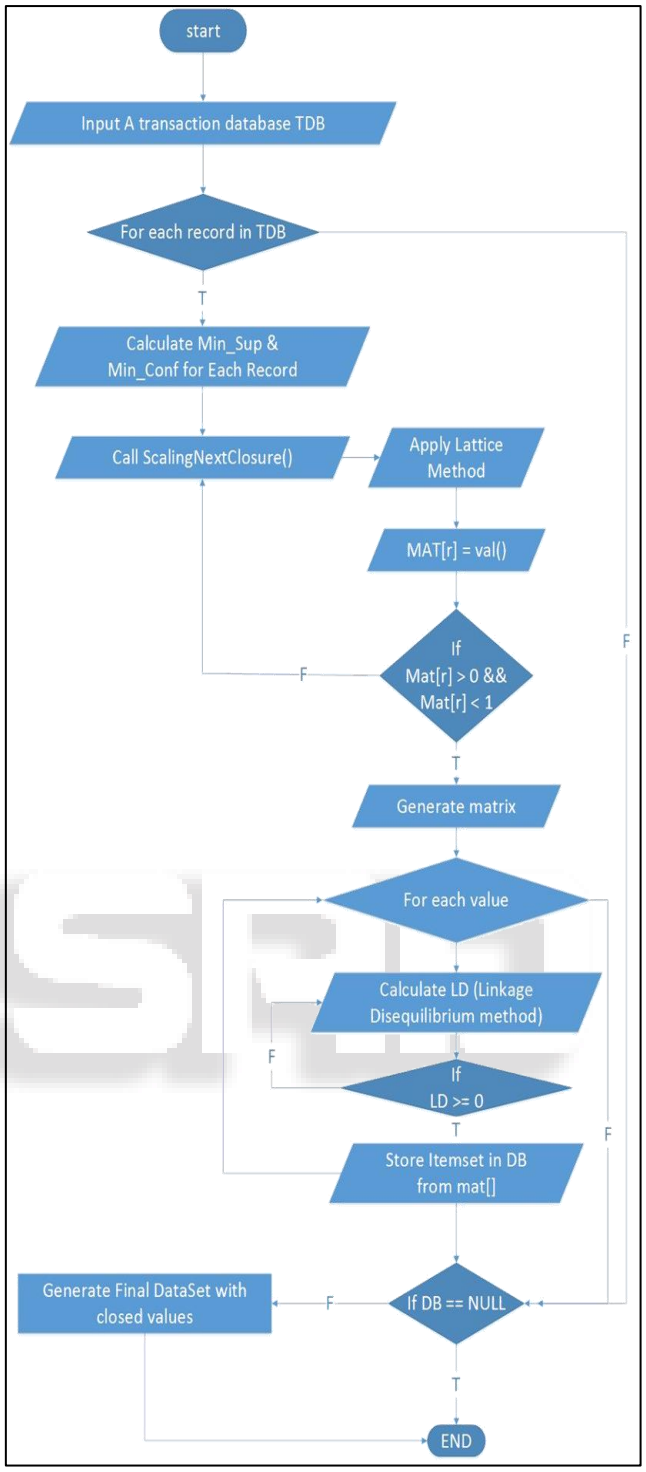

Fig. 1: Flow Chart of FPC+ algorithm

A. Proposed Algorithm:

Input: A Transaction Database D, MST - Minimum support Threshold

Output: Set of frequent closed itemsets.

Step 1: Scan the transaction database TDB to find out the number of occurrences of all size 1 item sets.

Step 2: Input Min Support & Min Confidence & stored in a data structure

Step 3: Count each item's support by using compressed data structure

Step 4: Call ScalingNextClosure() to generate combinational dataset in a multiple subsets.

i. Linkage Disequilibrium equation is performed to find nearby dataset

$$p_{AB} \neq p_A p_B$$
$$p_{Ab} \neq p_A p_b = p_A(1-p_B)$$
$$p_{aB} \neq p_a p_B = (1-p_A)p_B$$
$$p_{ab} \neq p_a p_b = (1-p_A)(1-p_B)$$

ii. Remove if common sets generated, add each subset (unique) to set

Step 5: Generate a conditional dataset based on min_sup and min_conf of set

i. Lattice method use for conditions

Step 6: Repeat until all sets checked

i. Perform the equation on all unique sets

ii. Store it in Separate Table tb2.

Step 7: Display final results from tb2.

Step 8: Exit

III. Experiments & Results

Compare base paper method BCTFI with our new approach FPC+ for various datasets like sparse datasets, dense datasets for frequent closed itemset with space and runtime complexity. We prove our approach is comparatively best with respect to previous approaches.

Dataset	Total records / attributes	F.C.I	Time efficiency
PUMSB (census)	32661/8	129	O(n)
Mushroom	8124/22	24	O(n)

Table 1: Number of frequent closed item sets found & time efficiency achieved

Above table is generated for our new approach to show performance achieved in term of time efficiency. The dataset is taken from http://fimi.ua.ac.be/data/ and https://archive.ics.uci.edu/ml/datasets/Mushroom.

To implement the Frequent Closed Itemset minimum support is required. For the research purpose publicly available dataset pumsb(census), mushroom, chess, T10I4D100K are used. Each Dataset have number of records and attributes. For all these datasets Frequent Closed Itemsets are found and running time is measured. For research work we selected different dataset properties, to improve the effectiveness of the algorithm.

A. Comparison Analysis

As a output of the experiments, we discovered the performance of our method FPC+ with the BCTFI and Closet+. The time calculated to mine the frequent itemsets is called the running time. The experiential analysis of running time is given in Table 2 and Figure 2 gives that the new method FPC+ surpasses BCTFI and Closet+.

Datasets	BCTFI	Closet+	FPC+
Chess	10.22	12.24	9.84
Pumsb	10.48	19.06	10.08
mashroom	0.5	0.79	0.44
T10I4D100K	0.25	0.42	0.2

Table 2: Running time comparison of finding F.C.I for algorithms

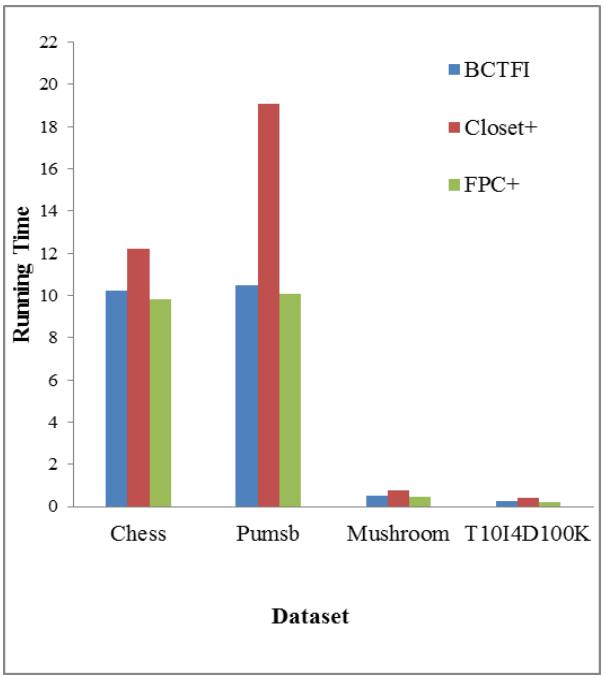

Fig. 2: Running time comparison of finding F.C.I for algorithms

The above database has taken in consideration to compare it with Closet+ & BCTFI algorithm. As this algorithm has been used to prove the efficiency, we have opted to experiment our results with the same databases. The Databases has been taken from standard websites like UCI machine learning repository and FIMI.

B. Results Comparison for Different Algorithm

Datasets	Apriori	FP-Growth	BCTFI	FPC+
Mushroom	4500	2975	2840	2567
Chess	856	570	258	249
Pumsb	14114	7938	5784	4332
T40I10D100K	5998	4320	2533	2193

Table 3: Comparison of F.C.I found

Graphical representation will clearly specify the results.

1) Mushroom Dataset

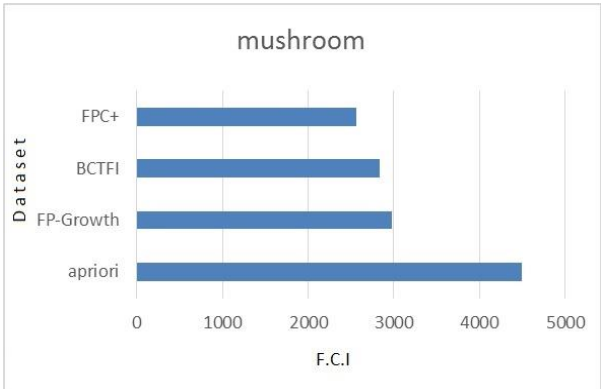

Fig. 3: Finding F.C.I for Mushroom Dataset

This information set includes pictures of speculative samples comparing to 23 kinds of gilled mushrooms in the Agaricus and Lepiota Family. Every species is reputed as unquestionably eatable, certainly harmful, or of obscure edibility and not prescribed. This final category was collected with the harmful one. The instruction unmistakably exhibits that there is no direct run for judging the edibility of a mushroom.

2) Chess Dataset

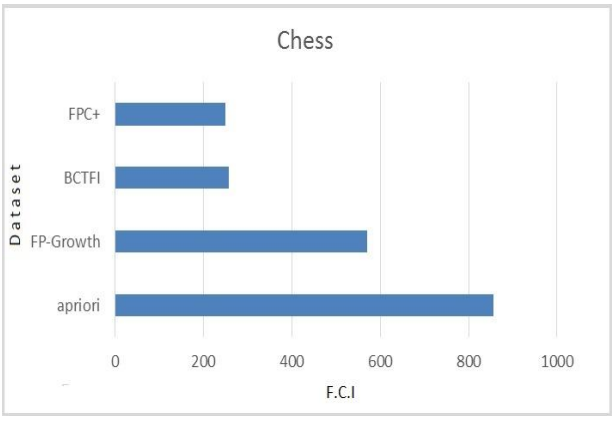

Fig. 4: Finding F.C.I for Chess Dataset

The dataset format is described as white can win and white cannot win. Note: the format of this database was modified on 2/26/90 to conform with the format of all the other databases in the UCI repository of machine learning databases.

C. PUMSB Dataset

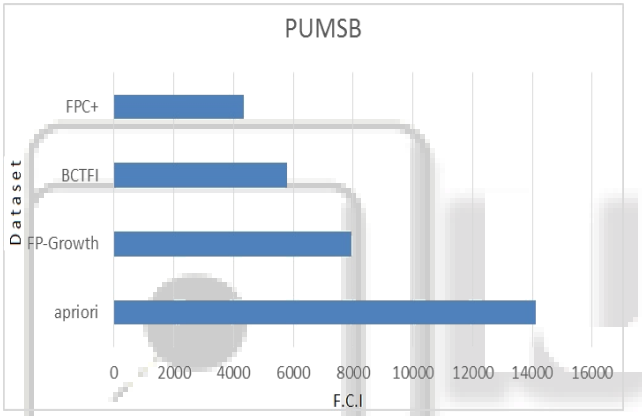

Fig. 5: Finding F.C.I for PUMSB Dataset

By 22 straight out qualities, for example, shape, shading, scent, and so on. There is a class name portraying if a mushroom is noxious or eatable, and there are 2,480 missing qualities altogether. At last, the third dataset, evaluation has been extricated from the registration authority database, and it contains demographic data on 32,561 individuals in the US. There are 8 all out properties, (for example, instruction, and occupation.

D. T40I10D100K Dataset

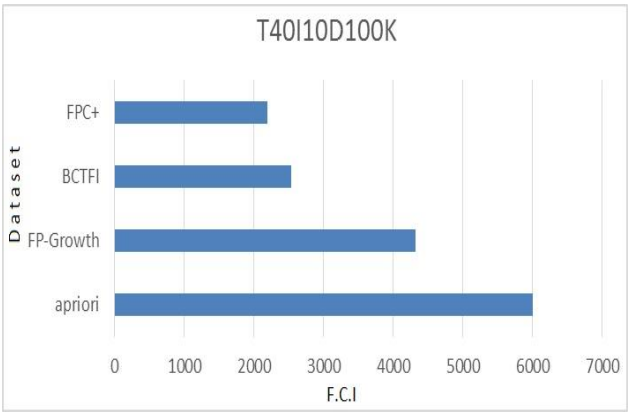

Fig. 6: Finding F.C.I for Transaction Dataset

Dataset downloaded from UCI mainly used for market basket analysis. This dataset was generated using the generator from the IBM Almaden Quest research group. It is having 30 attributes and around 100000 records.

The results above show that the proposed algorithm has proven itself more efficient in two types of comparisons. 1 for time efficiency and 2 for clearer frequently closed item set. The other results to compare are taken from base paper and the timing efficiency was noted after running the proposed algorithm.

IV. CONCLUSIONS

In this paper, we acknowledged the determined norms for making our method, which are the run-time consumption; these norms are impressed by the way for searching the frequent closed itemsets.

Efficiently generating the frequently closed dataset is the aim of the research, where we save time and frequent closed itemset both. Work has been completed to make an algorithm which is an betterment over BCTFI and Closet+ for a transactional database.

On bases of our study, the operation of the algorithms are clearly fully relies on the characteristic of the data sets and the support levels.

We give a new efficient pruning method, which, does not request for the full set of closed patterns mined so long. Every called partitioning of the search space to be mined freely in any sequence and, also in parallel allowed by this method also.

In this paper, we review some existing algorithms for frequent closed item set mining and present a proposal of our new approach. We perform tests on many publicly available data sets and compare with our new approach. For data sets, the memory consumption and run time of our new algorithm surpassed BCTFI and Closet+.

The memory utilization is also roughly similar as the BCTFI at higher support and performance done better at lower support.

REFERENCES

[1] Marghny H. Mohamed, Mohammed M. Darwieesh, "Efficient mining frequent itemsets algorithms" Springer-Verlag Berlin Heidelberg 2013
[2] Anurag Choubey, Dr. Ravindra Patel, Dr. J.L. Rana, "Graph Based New Approach For Frequent Pattern Mining" International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 1, Feb 2012
[3] Richa Mathur, Virendra Kumar, "A Fast & Memory Efficient Technique for Mining Frequent Item Sets from a Data Set" IOSR Journal of Computer Engineering (IOSR-JCE) , Volume 16, Issue 4, Ver. III (Jul – Aug. 2014), PP 112-115
[4] Bay Vo, Frans Coenen, Bac Le, "A new method for mining Frequent Weighted Itemsets based on WIT-trees" Elsevier, Expert Systems with Applications 40 (2013) 1256–1264
[5] Show-Jane Yen, Yue-Shi Lee, Chiu-Kuang Wang, "An efficient algorithm for incrementally mining frequent closed itemsets" Springer Science+Business Media New York 2013
[6] Wang J, Han J, Pei J (2003), "CLOSET+: searching for the best strategies for mining frequent closed itemsets."

In: Proc of 9th ACM SIGKDD international conference on knowledge discovery and data mining, pp 236–245

[7] Sujatha Dandu, B.L.Deekshatulu, Priti Chandra, "Improved Algorithm for Frequent Item sets Mining Based on Apriori and FP-Tree" Global Journal of Computer Science and Technology Software & Data Engineering, Volume 13 Issue 2 Version 1.0 Year 2013

[8] Hannu Toivonen, Paivi Onkamo, Kari Vasko, Vesa Ollikainen, Petteri Sevon, Heikki Mannila, Mathias Herr and Juha Kere, "Data Mining Applied to Linkage Disequilibrium Mapping", The American Journal of Human Genetics 67.1 (2000): 133-145.

[9] Aggrawal.R, Imielinski.t, Swami.A. "Mining Association Rules between Sets of Items in Large Databases". In Proc. Int'l Conf. of the 1993 ACM SIGMOD Conference Washington DC, USA.

[10] Agrawal.R and Srikant.R. "Fast algorithms for mining association rules". In Proc. Int'l Conf. Very Large Data Bases (VLDB), Sept. 1994, pages 487–499.

[11] Savasere, E. Omiecinski, and S. Navathe. "An efficient algorithm for mining association rules in large databases". In Proc. Int'l Conf. Very Large Data Bases (VLDB), Sept. 1995, pages 432–443.

[12] Pasquier N, Bastide Y, Taouil R, Lakhal L, (1999) "Discovering frequent closed itemsets for association rules." In: Proc of 7th international conference on database theory, pp 398–416

[13] Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, "Data mining Concepts and Techniques", 2006.

[14] J. Pei, J. Han, and R. Mao., "CLOSET: An efficient algorithm for mining frequent closed itemsets." In DMKD'00, May 2000.

