International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 9 5484 - 5487

A Novel Algorithm for Discovering Frequent Closures and Generators

G.Usha Rani
Research Scholar
JNTUH
Hyderabad, India
ushajntuhphd@gmail.com

P. Premchand
Professor
O.U. College of Engineering
Osmania University, Hyderabad
profpremchand.p@gmail.com

A.Govardhan
Director, SIT
JNTU Hyderabad
Hyderabad, India
govardhan_cse@yahoo.co.in

***Abstract*-**The Important construction of many association rules needs the calculation of Frequent Closed Item Sets and Frequent Generator Item Sets (FCIS/FGIS). However, these two odd jobs are joined very rarely. Most of the existing methods apply level wise Breadth-First search. Though the Depth-First search depends on different characteristics of data, it is often better than others. Hence, in this paper it is named as FCFG algorithm that combines the Frequent closed item sets and frequent generators. This proposed algorithm (FCFG) extracts frequent itemsets (FIs) in a Depth-First search method. Then this algorithm extracts FCIS and FGIS from FIs by a level wise approach. Then it associates the generators to their closures. In FCFG algorithm, a generic technique is extended from an arbitrary FI-miner algorithm in order to support the generation of minimal non-redundant association rules. Experimental results indicate that FCFG algorithm performs better when compared with other level wise methods in most of the cases.

Keywords*—*FCIS, FGIS, FCFG, Frequent Itemset.

*****

I. INTRODUCTION

The discovering the meaningful association rules is an important data mining task [1]. Any association rule miner proceeds in two steps. First one is extracting all frequent patterns X from a database and second one is breaking each pattern X into two parts like a premise Y and a conclusion X \ Y form a rule Y → X \ Y. The measures Support and Confidence are applied to prune the set of extracted association rules. However, the number of the remaining rules may still be way too high to be practical. As a remedy, various concise representations of the family of valid association rules have been proposed. A good survey on association mining can be found in. This paper focuses on the computation of frequent closed itemsets (FCIS) and frequent generator Item Sets (FGIS), which bring the Minimal Non-Redundant association rules (MNR). There are rules with the form P → Q \ P, where P ⊂ Q, where P is a (minimal) generator (a.k.a. key-sets or free- sets) and Q is a closed itemset. In other terms, in such rules the premise is minimal and the conclusion is maximal. It is known that MNR is a lossless, sound and informative representation of all valid association rules. Moreover, further restrictions can be imposed on the association rules in MNR, leading to more compact representations such as the generic basis or the proper basis. In computation, constructing MNR or its sub-structures requires the Set of frequent closed itemsets (FCIS) and their generators (FGIS), and it is possibly the precedence order between FCIS. There are some methods related to FCFI and FGFI published in the literature. FCIS/FGIS miners are exclusively applied in level based strategies, Even though the level wise itemset miners are formed better by Depth-First search methods on a high range of dataset profiles especially on dense ones. That's why FCFG algorithm is designed. The algorithm that is proposed here splits the association rule mining task into three sub steps. First, it is applied on vertical algorithm FCFG to extract the set of Frequent Itemsets FIs. Second, it processes the FIs in a level wise manner to filters FCIS and FGIS. This is why FCFG is said to be a hybrid algorithm. Finally, the algorithm associates FGIS to their closures (FCIS) to provide the necessary starting point for the production of MNR.

Experimental results show that FCFG algorithm performs well than two other algorithms A-Close and Zart. The FCFG algorithm, due to its Depth First nature, provides the FIs in a completely unordered way. However, the level wise post-processing steps require the FIs in ascending order by length. It is managed to solve this problem with a special file indexing that proves to be efficient, generic and gives no memory overhead at all. As it can be seen that the idea of FCFG can be generalized and used for arbitrary FI-mining algorithm, either breadth-first or depth-first.

The main contribution of this paper is a general way of extending frequent item set miners to calculate minimal non-redundant association rules. In this paper, a new method has presented for storing frequent items in the file system if frequent items are not provided in ascending order by its length. The file indexing technique do not requires any additional memory space. FIs can be sorted in a lengthwise manner. Once itemsets are available in this order, FCFG technique can be used to generate closed itemsets and associating generators to their closures.

II. EXISTING METHODS

A. *Closed itemset and Generator*

An itemset X is closed (generator) if it has no proper superset (subset) with the same support (respectively).

The closure of an itemset X is thus the largest itemset in the equivalence class of X. For example, in dataset D, the sets AB and AC are generators and their closures are ABE and AC respectively (That is, the equivalence class of AC is a singleton). In this approach, it rely on the following two properties.

Property 1. A closed itemset cannot be the generator of a larger itemset.

Property2. The closure of a frequent non-closed generator *g* is the smallest proper superset of *g* in the set of frequent closed itemsets.

An association rule r: $P_1 \rightarrow P_2$ involves two itemsets P1, $P_2 \subseteq$ A, such that $P_1 \cap P_2 = \Phi$ and $P_2 \neq \Phi$. The support of a rule r is supp(r) = supp ($P_1 \cup P_2$) and its confidence conf(r) = supp ($P_1 \cup P_2$)/supp (P_1). Frequent rules are defined in a way similar to frequent itemsets, whereas confident rules play an equivalent role for the confidence measure. A valid rule is both frequent and confident. Finding all valid association rules in a dataset is the target of a typical association rule mining task. As their number may grow up to exponential, reduced sub-families of valid rules are defined, which nevertheless convey the same information (lossless). Associated expansion mechanisms allow for the entire family to retrieve from the reduced ones without any non-valid rules to be mixed. The minimal non-redundant association rule family (MNR) is made of rules $P \rightarrow Q \setminus P$, where $P \subset Q$, P is a (minimal) generator and Q is a closed itemset. A more restricted family arises from the additional constraint of P and Q belonging to the same equivalence class, i.e. $P'' = Q$. It is known as the generic basis for exact (100% confidence) association rules. Here, the basis refers to the non-redundancy of the family with respect to a specific criterion. Inexact rule bases can also be defined by means of generators and closures, e.g. the informative basis, which further involves the inclusion order between closures.

B. *Vertical Frequent Itemset Mining*

The frequent itemset mining methods from the literature can be roughly divided into breadth-first and depth-first miners. Apriori-like [1] level wise breadth-first algorithms exploit the anti-monotony of frequent itemsets in a straight forward manner: they advance one level at a time, generating candidates for the next level and then computing their support upon the database. In contrast, Depth-First algorithms organize the search space in a tree. Typically using a sorted representation of the itemsets, they factor out common prefixes and hence limit the computing effort.

C. *Eclat*

Eclat is a plain FI-miner traversing the IT-tree in a depth-first manner in a pre-order way, from left-to-right.

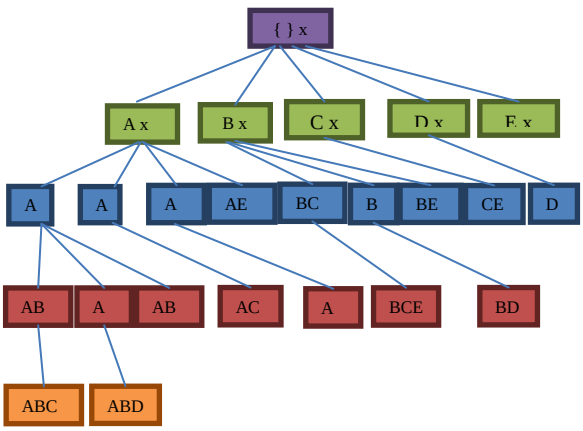

Fig. 1. IT-tree: Itemset-Tidset Search tree of dataset D

At the beginning, the IT-tree is reduced to its root (empty itemset). Eclat extends the root one level downwards by adding the nodes of all frequent 1-itemsets. Then, each of the new nodes is extended similarly: first, candidate descendant nodes are formed by adding to its itemset to the itemset of each right sibling; second, the tidsets are computed by intersection and the supports are established; and third, the frequent itemsets are added as effictive descendant nodes of the current node. Using Figure.1, the execution of Eclat on dataset D with min_supp = 20% is illustrated. Initially, the IT-tree comprises only the root node whose support is 100%. Frequent items with their TID sets are then added under the root. Each of the new nodes is recursively extended, following a left-to-right order and processing the corresponding sub-trees in a pre-order fashion. For instance, the subtree of A comprises all frequent itemsets starting with A. Thus, at step two, all 2-long supersets of A are formed using the right siblings of A (frequent 1-itemsets). As AB, AC, AD and AE are all frequent, they are added as descendant nodes under the node of A. The extended procedure is then recursively called on AB and the computation goes one level deeper in the IT-tree. All frequent itemsets are discovered when the algorithm stops.

III. PROPOSED ALGORITHM-FCFG

The proposed FCFG is a hybrid algorithm that combines the vertical FI-miner Eclat with an original level wise extension. Eclat finds all FIs that are saved in the file system. Then, this file is processed in a level wise manner, i.e. itemsets are read in ascending order by length, generators and closed itemsets are filtered. Finally, generators are associated to their closures. In the following subsection, the algorithm is presented in detail.

A. *Processing Itemsets in Ascending Order by Length*

In the previous subsection, the first part of the algorithm has been presented, i.e. how to get frequent itemsets in ascending order by their length, even if they are produced in an unordered way. This subsection continues with the second part namely how to associate generators to their closures, once FIs are available in a good order. The main block is shown in Algorithm 1. Two kinds of tables are used, namely Fi for i-long frequent and Zi for i-long frequent closed itemsets. The *readTable* function is an in-charge of reading frequent itemsets of a given length. If this algorithm which produces FIs in an unordered way like Eclat is used, then "*readTable"* reads FIs from the binary file as explained previously. The function returns FIs in Fi table. Fields of the table are initialized: itemsets are marked as *keys* and *closed*. However, these values may change during the post-processing step. Frequent attributes (frequent 1-itemsets) represent a special case. If they are present in each object of the dataset, then they are not generators, because they have a smaller subset with the same support, namely the empty set. In this case the empty set is a useful generator with respect to rule generation. The "*findKeysAndClosedItemsets"* procedure is an in-charge of filtering FCIs and FGs among FIs. The filtering procedure is based on Def. 1. The *Find-Generators* procedure takes Zi table

ISSN: 2321-8169
5484 - 5487

as an input. The method functions as follows: For each frequent closed itemset z in Zi, it finds its proper subsets in the global list FG, registers them as generators of z, deletes them from FG, and adds non-closed generators from Fi to FG. Properties 1 and 2 guarantee that whenever the subsets of an FCI are looked up in the list FG, only its generators are returned.

IV. Experimental Results and Discussion

FCFG has been evaluated against "Zart" and A-Close algorithms. The algorithms have been implemented in Java under the "Coron" data mining platform. The experiments have been carried out on a I3-processor Intel Linux with 4 GB RAM. For the experiments, the following benchmark datasets have been used: T20I6D100K, C20D10K and Mushrooms. The T20I6D100K6 is a sparse dataset. It has been constructed according to the properties related to market basket data analysis that are typical weakly correlated data. The census dataset is part of the PUMS sample files C20D10K, while the Mushroom data describes 7 mushrooms characteristics. The last two are highly correlated datasets. Table1 contains the experimental results of "FCFG" compared with Zart algorithm and A-Close algorithms. All the times reported are obtained from the UNIX time command between input and output. Zart and A-Close algorithms are chosen because they represent two efficient algorithms that produce exactly the same output as FCFG. Both Zart and A-Close algorithms are level wise algorithms.

Table1. Response times of FCFG and other statistics

Dataset	Minimum Support in %	Execution Time (Seconds)		
		FCFG	Zart	A-Close
T20I6D-100	1	4.11	6.58	24.06
	0.75	3.31	12.39	29.44
	0.50	5.82	34.61	72.88
	0.25	24.55	121.03	204.69
C20D10K	30	1.07	6.27	11.27
	20	1.71	11.32	20.77
	10	5.17	23.99	40.70
	5	20.24	49.29	62.64
Mush-room	30	0.82	2.87	5.86
	20	3.36	7.72	11.68
	10	37.46	46.37	29.43
	5	368.03	391.97	50.20

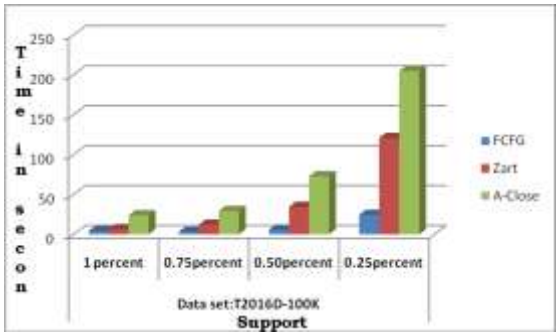

Fig. 2. Comparison of Response times of FCFG, Zart, A-Close for Dataset T2016D-100K

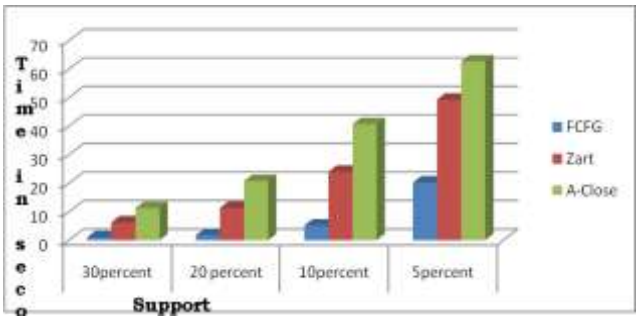

Fig. 3. Comparison of Response times of FCFG, Zart, A-Close for Dataset C20D10K

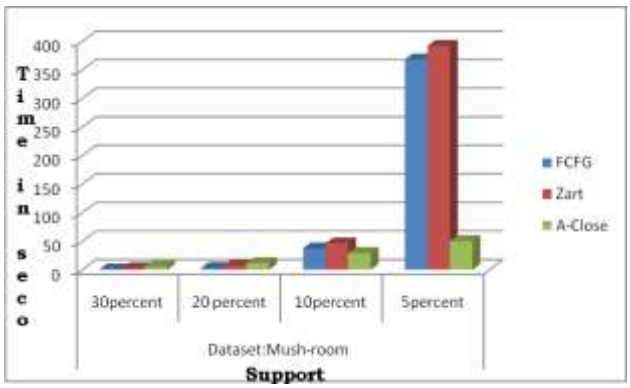

Fig. 4. Comparison of Response times of FCFG, Zart, A-Close for Dataset Mushroom

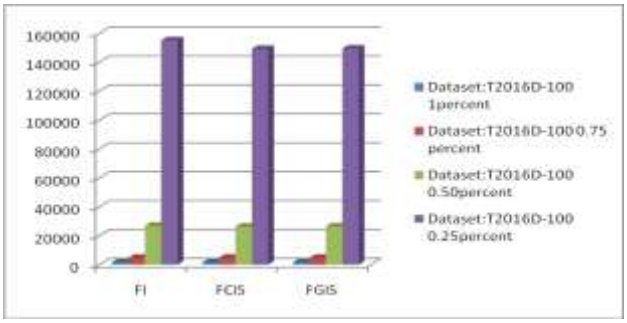

Fig. 5. Comparison of FI, FCIS,FGIS for 1%, 0.75%, 0.50%, 0.25% for the Dataset T2016D-100

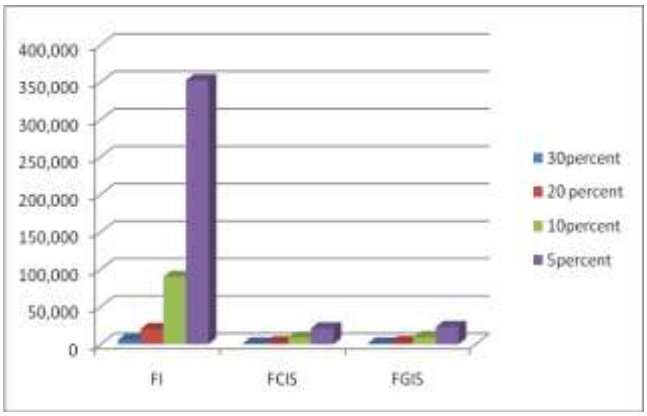

Fig. 6. Comparison of FI, FCIS,FGIS for 30%, 20%, 10%, 5% for the Dataset C20D10K

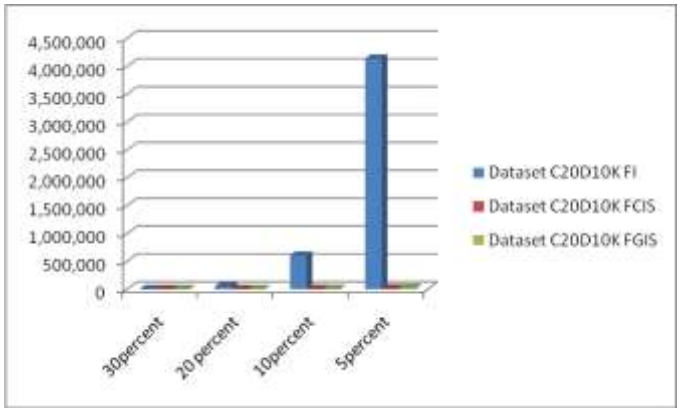

Fig. 7. COMPARISON OF FI, FCIS,FGIS FOR 30%, 20%, 10%, 5% FOR THE DATASET MUSHROOM.

Zart algorithm is an extension of Pascal algorithm, i.e. first it finds all FIs using pattern-counting inference, then it filters FCIs and finally the algorithm associates FGs to their closures. A-Close algorithm reduces the search space to FGs only and then it calculates the closure itemsets for each generator. The way A-Close algorithm computes the closure itemsets of generators is quite expensive because of the huge number of intersection operations. In the sparse dataset T20I6D100K, almost all frequent itemsets are closed and generators at the same time. It means that most equivalence classes are singletons, thus A-Close cannot reduce the search space significantly. Since the computation of closure itemsets in A-Close algorithm is quite expensive, FCFG algorithm performs much better than A-Close. Zart and FCFG algorithms are similar in the sense that both algorithms extract FIs initially. FCFG is based upon Eclat algorithm while Zart algorithm is based on Pascal. The better performance of FCFG algorithm is due to the better performance of its FI-miner *engine*. In datasets C20D10K and Mushrooms, the number of FGIS is considerably less than the total number of FIs. In this case, Zart algorithm can take advantage of its pattern counting inference technique and A-Close algorithm can benefit from its search space reduction. Despite these optimizations, FCFG algorithm still forms better than two algorithms in most of the cases. However, if the number of FGs is much less than the number of FIs (for instance in Mushrooms by min_supp = 5%), A-Close algorithm gives better response time. As a summary, it can be stated that FCFG algorithm clearly forms better than its level wise competitors on sparse datasets and it also performs very well on dense, highly correlated datasets if the minimum support threshold is not set too low.

V. CONCLUSION

In this Paper, a generic algorithm called FCFG algorithm that identifies FCIS and their associated generators has been presented. From the experimental output, numerous concise representations of valid association rules can be readily derived.

FCFG algorithm splits the FCIS/FGIS-mining problem into three tasks: (1) FI-mining, (2) Filtering FCIS and FGIS, and (3) associating FGIS to their closures (FCIS). The FI-mining part is solved by a well known depth-first algorithm, Eclat algorithm. However, the challenge to be faced with Eclat algorithm is: it produces itemsets in an unordered way. This issue has been solved by using a special file indexing technique and it has been managed to solve this issue in an efficient way, thus steps (2) and (3) can post-process FIs in a level wise manner. As seen, the idea of the proposed hybrid algorithm called FCFG algorithm can be generalized and used for any FI-mining algorithm, be it breadth-first or depth-first. Experimental results prove that FCFG algorithm is highly efficient and outperforms its level wise competitors in most of the cases. The study led to a range of exciting questions that are currently investigated. FCFG algorithm is highly efficient, but first it traverses the whole set of FIs. It causes no problem for sparse datasets. However, it may be a drawback in dense datasets with very low minimum support. It would be interesting to combine the search space reduction of A-Close algorithm with the efficiency of FCFG algorithm. Further, challenge also lies in the computation of the FCIS precedence order that underlies some of the association rule bases from the literature.

REFERENCES

[1] Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. *In Proc. of the 20th Intl. Conf. on Very Large Data Bases (VLDB '94)*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1994), pp. 487–499

[2] Frank, A. & Asuncion, A. UCI Machine Learning Repository *[http://archive.ics.uci.edu/ml]*. Irvine, CA: University of California, School of Information and Computer Science. 2010.

[3] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly Detection: A Survey." *To Appear in ACM Computing Surveys, 2009*.

[4] K.F. Jea and M.Y. Chang. "Discovering frequent itemsets by support approximation and itemset clustering." *Data and Knowledge Engineering*, Volume 65, No.1, 2008, pp. 90-107.

[5] A. Koufakou, M. Georgiopoulos, and G.C. Anagnostopoulos. "Detecting Outliers in High-Dimensional Datasets with Mixed Attributes." *WORLDCOMP International Conference on Data Mining DMIN*, 2008, pp. 427-433.

[6] A. Koufakou, J. Secretan, J. Reeder, K. Cardona, and M. Georgiopoulos. "Fast parallel outlier detection for categorical datasets using Map Reduce." *IEEE World Congress on Computational Intelligence International Joint Conference on Neural Networks IJCNN*, 2008, pp. 3298-3304.

[7] M. Biba, F. Esposito, S. Ferilli, N. Di Mauro, and T.M.A. Basile. "Unsupervised Discretization using Kernel Density Estimation." *In Proceedings International Conference on Artificial Intelligence*, Hyderabad, India, 2007, pp. 696-701.

[8] A. Koufakou, E.G. Ortiz, M. Georgiopoulos, G.C. Anagnostopoulos, and K.M.Reynolds. "A Scalable and Efficient Outlier Detection Strategy for Categorical Data." *IEEE International Conference on Tools with Artificial Intelligence ICTAI*, 2007, pp. 210-217.

[9] T. Hu, Q. Xu, H. Yuan, J. Hou, and C. Qu. "Hyperclique Pattern Base Off-Topic Detection." *Lecture Notes in Computer Science*, 2007, pp. 374.

[10] T. Calders and B. Goethals. "Non-Derivable Itemset Mining." *Data Mining and Knowledge Discovery*, Volume1, February 2007, pp.171-206.

[11] C.T. Chu, S.K. Kim, Y.A. Lin, Y.Y. Yu, G. Bradski, A.Y. Ng, and K. Oluko-tun. "Map-reduce for machine learning on multi core." *Advances in Neural Information Processing Systems*, 2007, pp. 281.

[12] D. Borthakur. The hadoop distributed file system: Architecture and design. *http://lucene.apache.org/hadoop*, 2007.