Volume 3, Issue 3, March 2018 ISSN (online): 2456-0006
International Journal of Science Technology Management and Research
Available online at: www.ijstmr.com

Discover Useful Itemset Over Large and Dynamic Data Set

Soniya Pare
M Tech (CSE) IV semester
Lord Krishna College of Technology
Indore M.P. India
soniya.pare05@gmail.com

Vijay Kumar Verma
Assistant professor CSE dept.
Lord Krishna College of Technology
Indore M.P. India
vijayvermaonline@gmail.com

Abstract: ***Pattern mining algorithms can be applied on various types of data such as sequence databases, streams, strings, spatial data, graphs etc. There are several types of pattern exist in data mining. Useful pattern can be frequent pattern, multilevel pattern, multidimensional pattern, domain based pattern etc. Useful pattern some time known as margin pattern in term of profit. In this paper we proposed a new approach to discover useful pattern by more correct pruning approach which increase performance of the reduces size and complexity.***

Keywords: itemsets, performance, complexity, data mining, pattern

I. INTRODUCTION

The term data mining is often used to apply to the two separate processes of knowledge discovery and prediction. Knowledge discovery provides explicit information that has a readable form and can be understood by a user. Forecasting, or predictive modeling provides predictions of future events and may be transparent and readable in some approaches (e.g. rule based systems) and opaque in others such as neural networks. Moreover, some data mining systems such as neural networks are inherently geared towards prediction and pattern recognition, rather than knowledge discovery.Data Mining is most frequently used for Customer Relationship Management applications these applications can contribute significantly to the bottom line. Rather than contacting a prospect or customer through a call center or sending mail, only prospects that are predicted to have a likelihood of responding to an offer are contacted. More sophisticated methods may be used to optimize across campaigns so that we can predict which channel and which offer an individual is most likely to respond to - across all potential offers[1,2]

II. BACKGROUND AND LITERATURE REVIEW

In 2012 Cheng Wei et al "Mining Top-K Item set s" Mining Item set s from databases is an emerging topic in data mining, which refers to the discovery of Item set s with utilities er than a user-specified minimum threshold . Although several studies have been carried out on this topic, setting an appropriate minimum threshold is a difficult problem for users. If is set too low, too many Item set s will be generated, which may cause the mining algorithms to become inefficient or even run out of memory. On the other hand, if min util is set too , no Item set will be found. Setting appropriate minimum thresholds by trial and error is a tedious process for users. They solve this problem by proposing a new framework named top-k Item set mining, where k is the desired number of Item set s to be mined. An efficient algorithm named TKU (Top-K Item set s mining) is proposed for mining such Item set s without setting min util. Several features were designed in TKU to solve the new challenges raised in this problem, like the absence of anti-monotone property and the requirement of lossless results. Moreover, TKU incorporates several novel strategies for pruning the search space to achieve efficiency[3].

In 2013 Arumugam P and Jose Proposed "Advance Mining of Item set s in Transactional Data". The white good industry domain is a dynamic and unpredictable field. Several analysis and algorithms provide investors with some technical tools for managing their stocks and predicting their market field. But these techniques are not enough to produce all the discovery possibilities. The sales executives plan their yearly, month wise target and their historical analysis, data mining approach used extensively in the markets and help in association analysis. It is useful for discovering interesting relationships hidden in large datasets. The uncovered relationships can be represented in the form of association rules. Traditional Apriori algorithm takes more time, space and memory for candidate generation process. They proposed the novel algorithm for transactional item set mining approach. This make to find association and correlation can generate less number of sets. So the sales person can use this item set transaction for their stocks planning distributor/dealer month wise, product wise, model wise target setting[4].

In 2014 D. Usha Nandini, Ezil Sam Leni, M. Maria Nimmy Proposed "Mining of Item set s from Transactional Databases". Efficient discovery of Item set s from transactional databases crucial task in data mining. UP-Growth and UP-Growth+ algorithms are proposed for mining Item set s. They also proposed a compact tree structure, called pattern tree (UP-Tree) and it maintains the information of Item set s. Previously we proposed FP-Growth algorithm for mining only large number of frequent Item set s, but not generate the Item set s. They have the issue of producing large number of candidate Item set s and probably it degrades mining performance in terms of speed and space requirement. However, our previous study needs more space and execution time. Many

algorithms are used to show the performance of UP-Growth and UP-Growth+. UP-Growth and UP-Growth+ becomes more efficient since database contain long transactions and generate fewer number of sets than FP-Growth. The experimental results and comparison validate its effectiveness[11,12].

In 2014 More Rani N. and Anbhule Reshma V "Mining Item sets From Transaction Database" Mining item sets from a transactional database means to retrieve item sets from database. Here, item sets are the item sets which have est profit. In existing system number of Algorithm's have been proposed but there is problem like it generate huge set of candidate Item sets for Item sets. If database contains large number of Transactions then it degrades the performance of mining in terms of execution time and space requirement. In proposed system, Efficient Algorithm for Mining Item sets From Transactional Database i.e. UP-Growth Algorithm. For that algorithm information of item sets is maintained in tree based data structure named Pattern Tree. With the help of UP-Tree candidate item sets can be generated with only two scans of database. In first scan, Transaction of each transaction is calculated. At the same time Transaction Weighted of each single item is also calculated. In second scan, transaction is inserted into UP Tree. Proposed algorithm, not only reduce number of candidate item sets but also work efficiently when database contains lot's of long transactions. They propose a tree-based algorithm, called UP-Growth, for efficiently mining item sets from transactional databases. We take Data Structure UP-Tree for maintaining the information of Item set s and four effective strategies, DGU, DGN, DLU and DLN, to reduce search space and the number of sets for mining. PHUIs can be efficiently generated from UP-Tree with only two database scans. UP Growth Algorithm is faster than existing algorithms when database contains lots of long transactions[5,6].

In 2014 G. Saranya and A .Deepak Kumar proposed "Implementation of Efficient Algorithm for Mining Item set s in Distributed and Dynamic Database" Association Rule Mining (ARM) is finding out the frequent Item set s or patterns among the existing items from the given database. Pattern Mining has become the recent research with respect to data mining. The proposed work is Pattern for distributed and dynamic database. The traditional method of mining frequent Item set mining embrace that the data is astride and sedentary, which impose extreme communication overhead when the data is distributed, and they waste calculation resources when the data is dynamic. To overcome this, Pattern Mining Algorithm is proposed, in which Item set s are maintained in a tree based data structure, called as Pattern Tree, and it generates the Item set without store the entire database, and has sparse communication overhead when mining with respect to distributed and dynamic databases. A quick update incremental algorithm is used which scans only the incremental database as well as collects only the support count of newly generated frequent Item set s. Incremental Mining Algorithm not only includes new Item set into a tree but also discard the infrequent Item set from a pattern tree structure. Hence it provides faster execution, minimal communication and cost when compared to the existing methods [7,8].

III. PROBLEM STATEMENTS

A basic approach is based on following formula

$$u'(I^k) = \frac{\sup_{\min}(I^k)}{k-1} \sum_{i=1}^{m} \frac{u(I_i^{k-1})}{\sup(I_i^{k-1})} + \frac{k-m}{k-1} \times \varepsilon$$

Ii^{k-1} is a (k-1)-itemset such that $Ii^{k-1} = I^k - \{i\}$, i.e. Ii k-1 includes all the items except item i. sup(I) is the support of item set I, which is the percentage of all the transactions that contain item set I. The minimum support among all the (k-1) subsets of I^k is given as

$$\sup_{\min}(I^k) = \min_{\forall I_i^{k-1} \subset I^k, (1 \le i \le m)} \{\sup(I_i^{k-1})\}$$

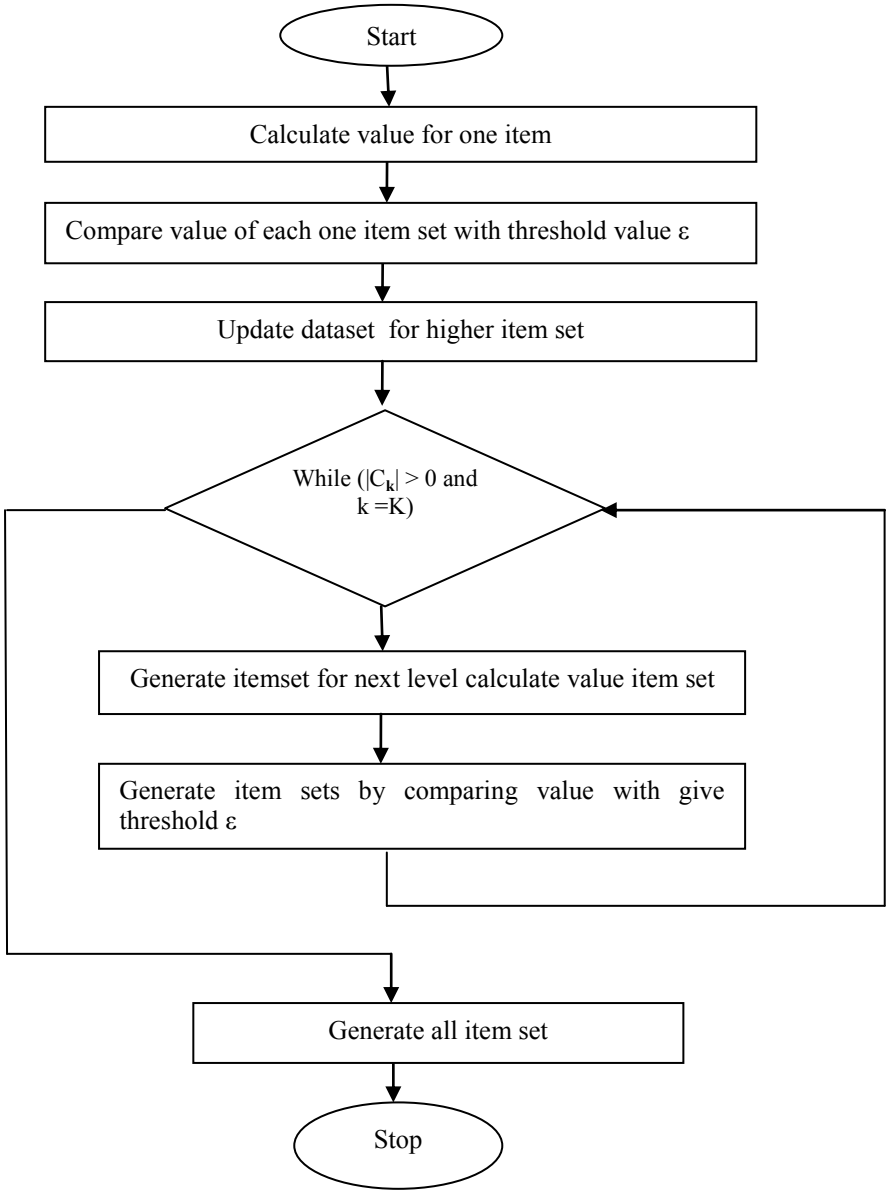
For each Ik, there are k (k-1)-subsets. m is the number of itemsets among the (k-1) subsets where Iik-1 ($1 \le i \le m$) are itemsets, and Iik-1 ($m+1 \le i \le k$) are itemsets. This prediction is based on the support boundary property which states that the support of an item set always decreases as its size increases. Thus, if the support of an item set is zero, its superset will not appear in the database at all. This approach uses the itemsets at level (k-1) to calculate the expected utility value for level k and the threshold ε to substitute the low utility itemsets[9,10].

IV. PROPOSED ALGORITHMS

Input: Database DB {Set of Transactions}
Transaction T ∈DB i=1,k=1, i_p value of item S: item set
C_k: Candidate's item set Minimum value threshold
Output: Itemsets S
[1] For each T ∈ DB // scan data DB
[2] Compute the value ∀ single item set
[4]For each $i_p \in C_k$ // scan data DB and generate candidate set
[5] Accumulate ∀ item set
[6] If value (i_p) ≥ threshold // high utility
[7] S.add (C) ;
[8] If valuel(i_p) ≤ threshold
[9] $C_k := C_k - i_p$ //delete useless item set
[10] End
[11] End
[12] End

[13] return (S);
[14] While (Ck !=Null)

V. FLOWCHART OF PROPOSED ALGORITHMS

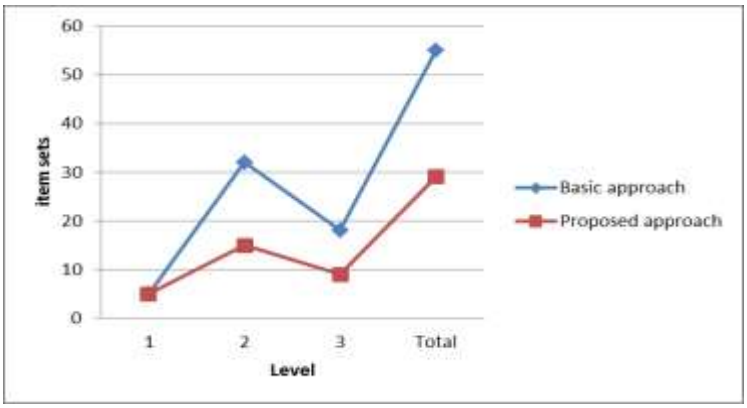


VI. COMPARATIVE ANALYSIS

We evaluate the performance of proposed algorithm and compare it with basic approach. The experiments were performed on i3 processor (2.5GHz Intel Processor with 4M cache memory), 2GB main memory and 400 GB secondary memory , and running on Windows XP. The algorithms are implemented in using C# Dot Net Framework language version 4.0.1. Both synthetic datasets are used to evaluate the performance of the algorithms. Number of item sets generated at different level using basic approach and proposed approach shown in table

Level No	Basic approach	Proposed approach
1	5	5
2	32	15
3	18	9
Total	55	29

Table 1 item set at different level



V. CONCLUSION

we have proposed a new an efficient approach to discover useful item set form large dataset proposed approach reduce size and also improve performance of basic approach . Proposed approach also reduce complexity of basic approach

REFERENCE

1. J. Han, et al "Mining Closed Patterns without Minimum Support," In Proc. of ICDM, 2002.
2. Y. Hirate et al "Growth: An Efficient Algorithm for Mining item set without any Thresholds", In Proc. of ICDM 2004.
3. H.-F. Li, H.-Y et al "Fast and Memory Efficient Mining Itemsets in Data Streams" In Proc. of the 8th IEEE Int'l Conf. on Data Mining, pp. 881-886, 2008.
4. Y. Liu, W. Liao, et al "A fast item sets mining algorithm". In Proc. of the Data Mining Workshop, 2005.
5. Y.-C. Li, et al "Isolated Items Discarding Strategy for Discovering Itemsets". In Data & Knowledge Engineering, Vol. 64, Issue 1, pp. 198-217, 2008.
6. S. Ngan, T. Lam, at al "Mining N-most Interesting Itemsets without Support Threshold" by the COFI-Tree, Int. J. Business Intelligence & Data Mining, Vol. 1, No. 1, pp. 88-106, 2005.
7. T. M. Quang, et al "An Efficient Algorithm for Mining Top-K Frequent Patterns", ADMA 2006, LNAI 4093, pp. 436 – 447, 2006.
8. L. Shen, H. Shen, et al "Finding the N Largest Itemsets" in Proc. Int'l Conf. on Data Mining, pp. 211-222, 1998.
9. B.-E. Shie, V. S. Tseng, et al "Online Mining of Temporal Itemsets from Data Streams". In Proc. of the 25th Annual ACM Symposium on Applied Computing (ACM SAC 2010), 2010.
10. V. S. Tseng, at al "UP-Growth: an efficient algorithm for item set mining". In Proc. of Int'l Conf. on ACM SIGKDD, pp. 253–262, 2010.
11. V. S. Tseng et al "Efficient mining of temporal high-utility itemsets from data streams. In ACM KDD Workshop on Utility-Based Data Mining Workshop, 2006".
12. B. Vo, H. Nguyen at al "Parallel Method for Mining High-utility Itemsets from Vertically Partitioned Distributed Databases. In KES 2009, Part I, LNAI 5711, pp. 251-260, 2009".