International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 3, Issue 1 (Jan-Feb 2015), PP. 31-34

A FLEXIBLE APPROACH TO MINE HIGH UTILITY ITEMSETS FROM TRANSACTIONAL DATABASES USING UP-GROWTH+: A SURVEY

Mr. Ramesh S. Yevale[1], Prof. Vinod S. Wadne[2]
[1]Department of Computer Engineering, [2]Assistant Professor
ICOER, Wagholi, Pune, Maharashtra, India
[1]ryevale33@gmail.com

Abstract- Present day, mining of high utility itemsets especially from transactional databases is required task to process many transactional operations quick. There are many methods that are presented for mining high utility itemsets from transactional datasets are subjected to some serious limitations such as performance of this methods needs to be investigated in low memory based systems for mining high utility itemsets from large transactional datasets and hence needs to address further as well. Further limitation includes these methods cannot overcome the screenings as well as overhead of null transactions; hence, performance degrades eventually. We are analyzing the new approaches to overcome these limitations such as distributed programming model for mining business-oriented transactional datasets, which overcomes the limitations and main memory-based computing, but also unexpectedly highly scalable in terms of increasing database size. We have used this approach with existing UP-Growth and UP-Growth+ with aim of improving their performances further.

Keywords: Data Mining, Frequent Itemset, Itemset Utility, UP-Growth, UP-Growth+

I. INTRODUCTION

A high utility itemset is defined as: A group of items in a transaction database is called itemset. This itemset in a transaction database consists of two aspects: First one is itemset in a single transaction is called internal utility and second one is itemset in different transaction database is called external utility. The transaction utility of an itemset is defined as the multiplication of external utility by the internal utility. By transaction utility, transaction weight utilizations (TWU) can be found. To call an itemset as high utility itemset only if its utility is not less than a user specified minimum support threshold utility value; otherwise itemset is treated as low utility itemset. To generate these high utility itemsets mining recently in 2010, UP-Growth (Utility Pattern Growth) algorithm was proposed by Vincent S. Tseng et al. for discovering high utility itemsets and a tree based data structure called UP-Tree (Utility Pattern tree) which efficiently maintains the information of transaction database related to the utility patterns. Four strategies (DGU, DGN, DLU, and DLN) used for efficient construction of UP-Tree [11] and the processing in UP-Growth [11]. By applying these strategies, can not only efficiently decrease the estimated utilities of the potential high utility itemsets (PHUI) but also effectively reduce the number of candidates. But this algorithm takes more execution time for phase II (identify local utility itemsets) and I/O cost.

In this paper, the existing UP-Growth algorithm is improved to generate high utility itemsets efficiently for large datasets and reduce execution time in phase II compared with existing UP-Growth algorithm. In the experimental section, experiments are conducted on our improved algorithm and existing algorithm with variety of synthetic and real-time datasets.

II. PROBLEM DEFINITION

In this section we describe the concepts of regular frequent pattern mining and define the basic definitions of the problem to obtain complete set of regular frequent patterns in incremental transaction databases.

Let I = {i1, i2, . . . , in} be a set of items. A set X = {ij, . . . ,ik} ⊆ I, where $j \leq k$ and j, k ∈ [1, n] is called a pattern or an itemest. A transaction t = (tid, Y) is a couple where tid is a transaction-id and Y is a pattern. Let size (t) be the size of t, i.e., the number of items in Y. A transaction database DB over I is a set of transactions T = {t1, . . . ,tm}, m = | DB | is the size of DB, i.e., the total number of transactions in DB. If X ⊆ Y, which means that t contains X or X occurs in t and denoted as tjX, j∈[1, m]. Therefore, TX = {tjX, . . . ,tkX}, $j \leq k$ and j, k ∈[1, m] is the set of all transactions where pattern X occurs in DB.

A. Definition 1 (frequent pattern X):
The total number of transactions in a DB that contains pattern X is called the support of X i.e., Sup(X). Hence Sup(X) = | TX|, where | TX | is the size of TX. The pattern X is said to be frequent if its support is greater than or equal to user given minimum support threshold i.e., $Sup(X) \geq min\_sup(\delta)$.

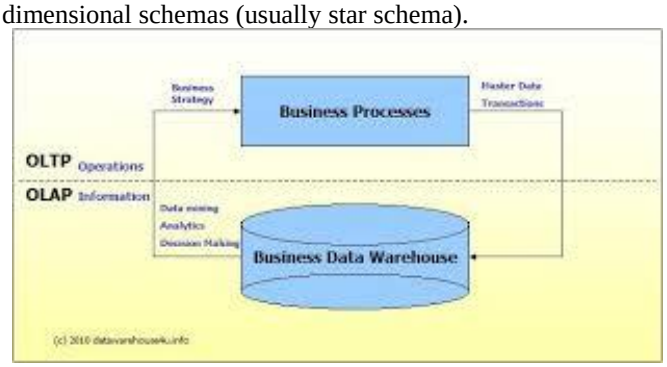
B. Definition 2 (regularity of frequent pattern X)
Let tXj+1 and tjX, j∈[1, (m - 1)] be two successive transactions where frequent pattern X appears. The variation between these two successive transactions can be defined as a period of X, say pX (i.e., p= tXj+1 – tX, j∈[1, (m -)]). For ease, to calculate the period of a pattern, we consider the first transaction in the DB as null i.e., tf = 0 and the last transaction is the mth transaction i.e., tl = tm. Let for a TX, PX be the set of all periods of X i.e., PX = {p1X, . . . ,prX}, where r is the total number of periods in PX. Then the regularity of a frequent pattern X can be denoted as Reg(X) = max{p1X, . . . ,prX}. A frequent pattern X is said to be regular frequent if its regularity is less than or equal to user given maximum regularity threshold i.e., λ.

III. DATA SOURCES

A. OLAP

OLTP (On-line Transaction Processing) is characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is theentity model(usually3NF).

OLAP (On-line Analytical Processing) is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema).



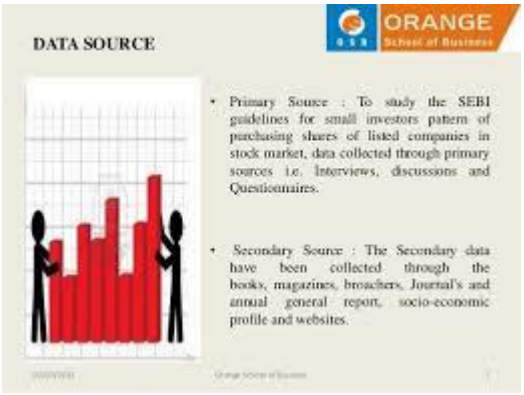
B. Big Data

Big Data describes the process of extracting actionable intelligence from disparate, and often times non-traditional, data sources. These data sources may include structured data such as databases, sensor, click stream and location data, as well as unstructured data like email, HTML, social data and images. The actionable data may be represented visually (e.g. in a graph), but it is often distilled down to a structured format, which is then stored in a database for further manipulation.

C. Stock Market

The goal of this article is to introduce the concepts, terminology and code structures required to develop applications that utilise real-time stock market data (e.g. trading applications). It discusses trading concepts, the different types of market data available, and provides a practical example on how to process data feed events into a market object model.

The article is aimed at intermediate to advanced developers who wish to gain an understanding of basic financial market data processing. I recommend that those who are already familiar with trading terminology skip ahead to the Market Data section.



D. Datasets

Real world data sets Accidents and Chess are obtained from FIMI Repository [4]; Chain-store is obtained from NU-MineBench 2.0 [5]; Foodmart is acquired from Microsoft foodmart 2000 database. In the above data sets, except Chain-store and Foodmart, unit profits for items in utility tables are generated between 1 and 1,000 by using a log-normal distribution and quantities of items are generated randomly between 1 and 10. The two real data sets Chain-store and foodmart already contain unit profits and purchased quantities. Total utilities of the two data sets are 26,388,499.8 and 120,160.84, respectively.

IV. TECHNIQUES USED FOR HIGH UTILITY ITEMSETS MINING

A. Mining Regular Frequent Patterns

In this section we describe the mining process of regular frequent patterns in incremental transactional databases using vertical data format requires only one database scan. To generate length-1 itemset our algorithm constructs an item header table called RFPID-table consists of four fields (Itemset,Tid, Sup, Reg). Itemset is an item name, T_{id} is the transaction list where the item occurs in various transactions, Sup is the support of the itemset and Reg is the regularity of an itemset. Each itemset consists of its own array to accommodate Tids and other intermediate results. Let Table 1 be the transactional database DB in horizontal format which is somewhat similar to the database in [9]. Convert the above horizontal database into vertical database with one database scan to store all length-1 items with respective t_{id}s, support and regularity. For example, Let us consider the minimum support threshold value, $\delta = 5$ and maximum.

TABLE I
Transactional Database DB

Tid	Transaction
1	a, d, e, c
2	d, e, f, a, c
3	a, e, c
4	d, e, c
5	e, c, a, f
6	b, f
7	d, c, e, b
8	b, c, d, e
9	a, d, c, b

B. Frequent Pattern Mining Tree:Design and Construction

Let I = a1,a2,….a_n be a set of items, and a transaction database DB = hT1,T2,….Tni,where Ti (i….[1::n]) is a transaction which contains a set of items in I. The support1 (or occurrence frequency) of a pattern A, which is a set of items, is the number of transactions containing A in DB. A is a frequent pattern if A's support is no less than a predefined minimum support threshold. Given a transaction database DB and a minimum support threshold, , the problem of finding the complete set of frequent patterns is called the frequent pattern mining problem.

C. UP-Growth Algorithm

The UP-Growth [11] is one of the efficient algorithms to generate high utility itemsets depending on construction of a global UP-Tree. In phase I, the framework of UP-Tree follows three steps: (i). Construction of UP-Tree [11]. (ii). Generate PHUIs from UP-Tree. (iii). Identify high utility itemsets using PHUI.The construction of global UP-Tree [11] is follows, (i). Discarding global unpromising items (i.e., DGU strategy) is to eliminate the low utility items and their utilities from the transaction utilities. (ii). Discarding global node utilities (i.e., DGN strategy) during global UP-Tree construction. By DGN strategy, node utilities which are nearer to UP-Tree root node are effectively reduced[15]. The PHUI is similar to TWU, which compute all itemsets utility with the help of estimated utility. Finally, identify high utility itemsets (not less than min_sup) from PHUIs values. The global UP-Tree contains many sub paths. Each path is considered from bottom node of header table. This path is named as conditional pattern base (CPB).

D. Improved UP-Growth

Although DGU and DGN strategies are efficiently reduce the number of candidates in Phase 1(i.e., global UP-Tree). But they cannot be applied during the construction of the local UP-Tree (Phase-2). Instead use, DLU strategy (Discarding local unpromising items) to discarding utilities of low utility items from path utilities of the paths and DLN strategy (Discarding local node utilities) to discarding item utilities of descendant nodes during the local UP-Tree construction. Even though, still the algorithm facing some performance issues in phase-2. To overcome this, maximum transaction weight utilizations (MTWU) are computed from all the items and considering multiple of min_sup as a user specified threshold value as shown in algorithm. By this modification, performance will increase compare with existing UP-Tree construction also improves the performance of UP-growth algorithm. An improved utility pattern growth is abbreviated as IUPG.

V. LIMITATIONS OF FREQUENT ITEMSETS MINING

A. Frequent Itemset Mining is Uncertain Transaction databases semantically and had significant *drawbacks* which led to misleading results.

B. Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all $2^{|S|} - 1$of its proper subsets.

VI. APPLICATIONS OF FREQUENT ITEMSETS MINING

A. Methodology/Principal Findings

The claims datasets of 1 million nationally representative people within Taiwan's National Health Insurance in 2005 were used to calculate the number of patients with one-stop visits. The frequent itemsets mining was applied to compute the combination patterns of specialties in the one-stop visits. Among the total 13,682,469 ambulatory care visits in 2005, one-stop visits occurred 144,132 times and involved 296,822 visits (2.2% of all visits) by 66,294 (6.6%) persons. People tended to have this behavior with age and the percentage reached 27.5% (5,662 in 20,579) in the age group ≥80 years. In general, women were more likely to have one-stop visits than men (7.2% vs. 6.0%). Internal medicine plus ophthalmology was the most frequent combination with a visited frequency of 3,552 times (2.5%), followed by cardiology plus neurology with 3,183 times (2.2%). The most frequent three-specialty combination, cardiology plus neurology and gastroenterology, occurred only 111 times.

B. Association Rule Learning

Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases.

VII. CONCLUSION

In this paper we have analyzed new enhanced frameworks of recently presented algorithms namely UP-Growth and UP-Growth+ with aim of improving the processing time

performance and mining performance under the less system memory environment as well. We have seen the concept of UP-Growth and UP-Growth+. The systems presented the work done so far over the previous approaches with the datasets used. In the future completely evaluate this proposed architecture and compare its performance against existing methods in order to claim the effectiveness and efficiency of this proposed network

VIII. ACKNOWLEDGEMENT

I express great many thanks to Prof. Vinod S. Wadne for his great effort of supervising and leading me, to accomplish this fine work. To college and department staff, they were a great source of support and encouragement. To my friends and family, for their warm, kind encourages and loves. To every person who gave me something too light along my pathway. I thanks for believing in me.

REFERENCES

[1] S. J. Yen and Y. S. Lee.: Mining high utility quantitative association rules. In *Proc. of 9th Int'l Conf. on Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science 4654*, pp. 283-292, Sep., 2007.

[2] Frequent itemset mining implementations repository, http://fimi.cs.helsinki.fi/

[3] Vincent. S. Tseng, C. W. Wu, B. E. Shie, and P. S. Yu.: UP-Growth: An Efficient Algorithm for High Utility Itemset Mining. In Proc. of ACM-*KDD*, Washington, DC, USA, pp. 253-262, July 25–28, 2010.

[4] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.

[5] C.H. Lin, D.Y. Chiu, Y.H. Wu, and A.L.P. Chen, "Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window," Proc. SIAM Int'l Conf. Data Mining (SDM '05), 2005.

[6] Y. Liu, W. Liao, and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm," Proc. Utility-Based Data Mining Workshop, 2005.

[7] F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining Using Weighted Support and Significance Framework," Proc. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '03), pp. 661-666, 2003

[8] H. Dutta, and J. Demme, "Distributed Storage of Large Scale Multidimensional EEG Data using Hadoop/HBase," Grid and Cloud Database Management, New York City: Springer; 2011.

[9] G. Y. Ming, W. Zhi-jun. A Vertical format algorithm for mining frequent itemsets. IEEE Transactions, pp. 11-13 (2010).

[10] M. J. Zaki, G. Karam. Fast Vertical Mining Using Diffsets, ACM SIGKDD. pp. 24-27 (2003).

[11] M. G. Elfeky, W. G. Aref, A. K. Elmagarmid. Periodicity Detection in Time Series Databases. IEEE Transactions on Knowledge and Data Engineering 17(7), pp. 875-887 (2005).

[12] A. Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Data Sets," Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), pp. 554-561, 2008.

[13] R. Chan, Q. Yang, and Y. Shen, "Mining High Utility Itemsets," Proc. IEEE Third Int'l Conf. Data Mining, pp. 19-26, Nov. 2003.

[14] U. Yun and J.J. Leggett, "WIP: Mining Weighted Interesting Patterns with a Strong Weight and/or Support Affinity," Proc. SIAM Int'l Conf. Data Mining (SDM '06), pp. 623-627, Apr. 2006.

[15] U. Yun, "An Efficient Mining of Weighted Frequent Patterns with Length Decreasing Support Constraints," Knowledge-Based Systems, vol. 21, no. 8, pp. 741-752, Dec. 2008.