MINING CONCISE AND LOSSLESS LOCATION-AWARE NEWS FEED REPRESENTATION BY USING EFFICIENCY ALGORITHM

T.Santhana lakshmi[1], S.Nithyadhevi[2]

*[1]T.Santhana lakshmi, PG Scholar, Dept. of computer science,
SINCET, Nagapattinam, Tamil Nadu, India.
[2]S.Nithyadhevi, Head of Department, Dept. of computer science
SINCET, Nagapattinam, Tamil Nadu, India.*

Abstract **-** In a novel framework for mining closed+ high utility item sets (CHUIs), which serves as a compact and lossless representation of HUIs. Mining high utility item sets (HUIs) from databases is an important data mining task, which refers to the discovery of item sets with high utilities (e.g. high profits). However, it may present too many HUIs to users, which also degrades the efficiency of the mining process. To achieve high efficiency for the mining task and provide a concise mining Propose three efficient algorithms named A priori CH (A priori-based algorithm for mining High utility Closed+ item sets), A priori HC-D (A priori HC algorithm with Discarding unpromising and isolated items) and CHUD (Closed+ High Utility Item set Discovery) to find this representation. Further, a method called DAHU (Derive All High Utility Item sets) is proposed to recover all HUIs from the set of CHUIs without accessing the original database. Results on real and synthetic datasets show that the proposed algorithms are very efficient and that tries algorithm approaches achieve a massive reduction in the number of HUIs. In addition , where user tries to apply query from the item set. All possible route or prefix matches when the users query applied.

Key Words: **DAHU, HUIs, Route, Prefix Matches , Query Processing, Item set Miner , Map viewer.**

1. INTRODUCTION

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important in their data warehouses. Data mining tools predict future trends and behaviors allowing business to make proactive, knowledge driven decisions. The automated prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business question that traditionally were too time consuming to resolve. Databases for hidden patterns, finding predictive information that experts may miss because it lies outside expectations. Most companies already collect and refine massive quantities of data. Data mining techniques can be implementing rapidly on existing software and hardware platforms to enhance the value existing information resources, and can be integrated with new products and systems as brought online. Once installed on a device, each Android app lives in its own security.

The Android operating system is a multiuser Linux system in which each app is a different user. By default, the system assigns each app a unique Linux user ID. The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them. Each process has its own virtual machine, so an app's code runs in isolation from other apps. App components are the essential building blocks of an Android app. Each component is a different point through which the system can enter app. Not all components are actual entry points for the user and some depend on each other, but each one exists as its own entity and plays a specific role each one is a unique building block that helps define app's overall behavior.

2. LITERATURE SURVEY

C. F. Ahmed, et al., [1]Efficient Tree Structure For High Utility Pattern Mining. In Incremental Database presents high utility pattern mining is one of the most important research issues in data mining due to its ability to consider the no binary frequency values of items in transactions and different profit values for every item. On the other hand, incremental and interactive data mining provide the ability to use previous data structures and mining results in order to reduce unnecessary calculations when a database is updated, or when the minimum threshold is changed. Proposed three novel tree structures to efficiently perform incremental and interactive HUP mining . The first tree structure, Incremental HUP Lexicographic . Tree arranged according to an item's lexicographic order. It can capture the incremental data without any restructuring operation. The second tree structure is the IHUP transaction frequency tree which obtains a compact size by arranging items according to transaction .To reduce the mining time, the third tree, IHUP-transaction-weighted utilization tree is designed based on the TWU value of items in descending order.

R. Chan, et al., [2] Fast And Memory Efficient Mining Of High Utility Item sets In Data Streams examines efficient mining of high utility item sets has become one of the most interesting data mining tasks with broad applications. In this paper, we proposed two efficient one-pass algorithms, MHUI-BIT and MHUI-TID, for mining high utility item sets from data streams within a transaction-sensitive sliding window. Two effective representations of item information and an extended lexicographical tree-based summary data structure are developed to improve the efficiency of mining high utility item sets. Experimental results show that the proposed algorithms outperform than the existing algorithms for mining high utility item sets from data streams.

C.C. Chang, etal.,[3]Isolated Items Discarding For Discovering High Utility Item sets Propose utility mining has widely been discussed in the field of data mining. It finds high utility item sets by considering both profits and quantities of items in transactional data sets. However, most of the existing approaches are based on the principle of levelwise processing, as in the traditional two-phase utility mining algorithm to find a high utility item sets .An efficient utility mining approach that adopts an indexing mechanism to speed up the execution and reduce the memory requirement in the mining process. The indexing mechanism can imitate the traditional projection algorithms to achieve the aim of projecting sub-databases for mining. In addition, a pruning strategy is also applied to reduce the number of unpromising itemsets in mining. Finally, the experimental results on synthetic data sets and on a real data set show the superior performance of the proposed approach.

Hong.T.P , et al., [4] An Efficient Projection Based Indexing Approach For Mining High Utility Item sets Knowledge And Information Propose utility mining has widely been discussed in the field of data mining. It finds high utility item sets by considering both profits and quantities of items in transactional data sets. However, most of the existing approaches are based on the principle of level wise processing, as in the traditional two-phase utility mining algorithm to find a high utility item sets .An efficient utility mining approach that adopts an indexing mechanism to speed up the execution and reduce the memory requirement in the mining process. The indexing mechanism can imitate the traditional projection algorithms to achieve the aim of projecting sub-databases for mining. In addition, a pruning strategy is also applied to reduce the number of unpromising item sets in mining. Finally, the experimental results on synthetic data sets and on a real data set show the superior performance of the proposed approach.

J. Han et al., [5] Fast And Space Preserving Frequent Pattern Mining In Large Database presents a simple and novel data structure using hyper-links, H-strict, and a new mining algorithm, H-mine, which takes advantage of this data structure and dynamically adjusts links in the mining process. A distinct feature of this method is that it has a very limited and precisely predictable main memory cost and runs very quickly in memory-based settings. Moreover, it can be scaled up to very large databases using database partitioning. When the data set becomes dense, (conditional) FP-trees can be constructed dynamically as part of the mining process shows that H-mine has an excellent performance for various kinds of data, outperforms currently available algorithms in different settings, and is highly scalable to mining large databases. This also proposes a new data mining methodology, space-preserving mining, which may have a major impact on the future development of efficient and scalable data mining methodsa simple and novel data structure using hyper-links, H-struct, and a new mining algorithm, H-mine, which takes advantage of this data structure and dynamically adjusts links in the mining process. A distinct feature of this method is that it has a very limited and precisely predictable main memory cost and runs very quickly in memory-based settings. Moreover, it can be scaled up to very large databases using database partitioning. When the data set becomes dense,FP-trees can be constructed dynamically as part of the mining process shows that H-mine has an excellent performance for various kinds of data, outperforms currently available algorithms in different settings, and is highly scalable to mining large databases. This study also proposes a new data mining methodology, space-preserving mining, which may have a major impact on the future development of efficient and scalable data mining methods.

3. SYSTEM ANALYSIS

3.1 Existing System

Frequent item set mining is a fundamental research topic in data mining. one of its popular applications is market basket analysis, which refers to the discovery of sets of items (item sets) that are frequently purchased together by customers. however, in this application, the traditional model discover a large amount of frequent but low revenue item sets and lose the information on valuable item sets having low selling frequencies. these problems are caused by the facts that firm treats all items as having the same importance/unit profit/weight and it assumes that every item in a transaction appears in a binary form, i.e., an item can be either present or absent in a transaction, which does not indicate its purchase quantity in the transaction. hence, fim cannot satisfy the requirement of users who desire to discover item sets with high utilities such as high profits.

3.1.1 Drawback Of Existing System

Low efficiency, Discover a large amount of frequent but low revenue item sets. Lose the information on valuable item sets having low selling frequencies.

3.2 Proposed System Analysis

The proposed representation is lossless due to a new structure named utility unit array that allows recovering all HUIs and their utilities efficiently. The proposed representation is also compact. It reduces the number of item sets by several orders of magnitude, especially for datasets containing long high utility item sets (up to 800 times). Three efficient algorithms named A priori HC (A priori-based algorithm for mining High utility Closed+ item set), A priori HC-D (A priori HC algorithm with Discarding unpromising and isolated items) and CHUD (Closed+ High Utility item set Discovery) to find this representation. The A priori HC and A priori HC-D algorithms employs breadth-first search to find CHUIs and inherits some nice properties from the well-known A priori algorithm. The CHUD algorithm includes three novel strategies named REG, RML and DCM that greatly enhance its performance. Results show that CHUD is much faster than the state-of-the-art algorithms for mining all HUIs. In propose a top-down method named DAHU (Derive All High Utility item sets) for efficiently recovering all HUIs from the set of CHUIs. The combination of CHUD and DAHU provides a new way to obtain all HUIs and outperforms UP-Growth, one of the currently best methods for mining HUIs, Redundancy removed, High utility item sets from horizontal database, Efficiently recover all high utility item sets from closed^{+} high utility item sets, Improved efficiency.

4. MODULES

1. Admin module

2. End user Module

3. Closed Item set Matcher Module

4. Item set Miner Module

5. Map view Module

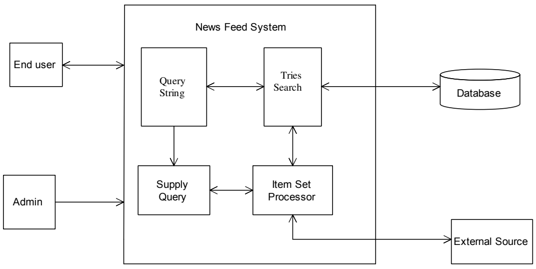

Fig -1: Architecture

4.1 Admin Module

The data owner uploads their data in the database for mined item set server. The information required relevant organizational users work location. All kind of location based information are gathered and mined as closed item set utility and persisted in the database server. The whole mining process controlled by admin where mining source information will be dynamic and get from external resource, the mining constraint will be like hotel, hospital, health care center, entertainment etc,.

4.2 End User Module

The data mining service provider manages the data to provide as per incoming query string which is supplied by end user. When the query string is received at mining service that tree based item sets are sorted out. The final results produced based on the location specific information only, the source of information is feed by admin where information are relatively closed to work location.

4.3 Closed Item Set Matcher Module

The first point that should discuss is how to incorporate the closed constraint into high utility item set mining. There are several possibilities. First, It can define the closure on the utility of item sets. In this case, a high utility item set is said to be closed if it has no proper superset having the same utility. However, this definition is unlikely to achieve a high reduction of the number of extracted item sets since not many item sets have exactly the same utility as their supersets in real datasets. Three efficient algorithms A priori HC (An A priori-based algorithm for mining High utility Closed+ item sets), A priori HC-D (A priori HC algorithm with Discarding less probability of item occurrence and isolated item set and isolated items) and CHUD (Closed+ High Utility item set Discovery) for mining CHUIs. Performance of item set matcher depends model's Transaction Weighted Utilization.

4.4 Item set Miner Module

Item set Miner module by using tries algorithm implementation resides at mining progress, where top level user tries to apply query from the item set. All possible route or prefix matches inspected when the users query applied. Based on most viewed results and maximum probability of item occurrences, the final result will be published as the result to end user.

The tires implementation consists of nodes. Each node has these fields:

Key:-Part of the string to be searched, inserted or deleted. Value:-The value associated with a string(dictionary could be the meaning of the word which searching). Neighbor Node Address:-It consists of the address of the neighboring node at the same level. Previous Neighbor Address:-It consists of the address of the previous node at the same level. Children Node Address:-It consists of the address of the child node at the current level. Parent Node Address:-It consists of the address of the parent node at the current node. Additional Nodes:-It consist parent and previous nodes are added to this implementation for making the search, and deletion easier.

4.5 Map View Module

The Module supplies an alternative result produced by using tries based searching technique, where the results are applied on map view to present graphical output .The map view result assist to an organizational users to get required news from their location surrounded information, it will assist the end user to navigate to specific information's locations

5. ALGORITHMS

5.1 The A priori Algorithm

The algorithm performs a database scan to compute the transaction utility of each transaction. At the same time, the TWU of each item is computed. Each item having a TWU no less than abs_min_utility is added to the set of 1-HTWUIs Ck. Then the algorithm proceeds recursively to generate item sets having a length greater than k. During the k-th iteration, the set of k-HTWUIs is used to generate (k+1)- candidates Ck+1 by using the A priori-gen function. Then the algorithm computes TWUs of item sets in Ck+1 by scanning the database D once.

5.2 Tries Algorithm

A **Tries**, also called **digital tree** and sometimes Rtree or **prefix tree** (as they can be searched by prefixes), is an ordered tree data structure that is used to store a dynamic set or association array where the keys are usually strings. Unlike a binary search tree, no node in the tree stores the key associated with that node; instead, its position in the tree defines the key with which it is associated. All the descendants of a node have a common prefix of the string associated with that node, and the root is associated with the empty string. Values are not necessarily associated with every node. Rather, values tend only to be associated with leaves, and with some inner nodes that correspond to keys of interest. For the space-optimized presentation of prefix tree, see compact tree.

6.Conclusion

To mine this representation, In proposed three efficient algorithms named Apriori HC, Apriori HC-D and CHUD. Apriori HC-D is an improved version of Apriori HC, which incorporates strategies DGU and IIDS for pruning candidates. Apriori HC and Apriori HC-D perform a breadth-first search for mining closed+ high utility item sets from horizontal database, while CHUD performs a depth-first search for mining closed+ high utility item sets from vertical database. The strategies incorporated in CHUD are efficient and novel. They have never been used for vertical mining of high utility item sets and closed+ high utility item sets. To efficiently recover all high utility item sets from closed+ high utility item sets, we proposed an efficient method named DAHU. Results on both real and synthetic datasets show that the proposed representation achieves a massive reduction in the number of high utility item sets on all real datasets.

7 FUTURE WORK

The Proposed system, addressed the problem of reduce, and best mining approach where can get closed item set without loss of possibilities. The key implementation in the system is done in object oriented approach, where item sets are linked one another. when concurrency process is applied in the system, that information mining and searching results occurrence on same time. Then index or position reorder will suffer the performance. To overcome this kind of issue, system environment may be customized into advance level into the cloud environment to handle number of simultaneous query processing, based on virtual machine overall performance can be enhanced.

REFERENCES

[1] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, "Efficient Tree Structures for High utility Pattern Mining in Incremental Databases," IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721,

[2] R. Chan, Q. Yang, and Y. Shen, "Mining High Utility Itemsets," in Proc. of the IEEE Int'l Conf. on Data Mining, pp. 19-26, 2003.

[3] K. Chuang, J. Huang, M. Chen, "Mining Top-K Frequent Patterns in the Presence of the Memory Constraint," VLDB Journal, Vol. 17, pp. 1321-1344, 2008.

[4] T.-P. Hong, V. S. Tseng, "An Efficient Projection-based Indexing Approach for Mining High Utility Itemsets.Knowledge and Information System, Vol. 38, Issue 1, pp. 85-107, 2014

[5] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," in Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, pp. 1-12, 2000.

[6] T. Hamrouni, S. Yahia, E. M. Nguifo, "Sweeping the Disjunctive Search Space Towards Mining New Exact Concise Representations of Frequent Itemsets," Data & Knowledge Engineering, Vol. 68, Issue 10, pp. 1091-1111, 2009.

[7] G.-C. Lan, T.-P. Hong, V. S. Tseng, "An Efficient Projection-based Indexing Approach for Mining High Utility Itemsets.Knowledge and Information System, Vol. 38, Issue 1, pp. 85-107, 2014.

[8] B. Le, H. Nguyen, T. A. Cao, and B. Vo, "A Novel Algorithm for Mining High utility Itemsets," in Proc. of the First Asian Conference on Intelligent Information and Database Systems, pp.13-17, 2009.

[9] H. Li, J. Li, L. Wong, M. Feng, Y. Tan, "Relative risk and odds ratio: a data mining perspective," in Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 368-377, 2005.

[10] K. Gouda and M. J. Zaki, "Efficiently Mining Maximal Frequent Itemsets," in Proc. of the IEEE Int'l Conf. onData Mining, pp. 163-170, 2001.

[11] T. Hamrouni, "Key Roles of Closed Sets and Minimal Generators in Concise Representations of Frequent Patterns," Intelligent Data Analysis. Vol. 16, Issue 4,pp. 581-631, 2012.

[12] T. Hamrouni, S. Yahia, E. M. Nguifo, "Sweeping the Disjunctive Search Space Towards Mining New Exact Concise Representations of Frequent Itemsets," Data & Knowledge Engineering, Vol. 68, Issue 10, pp. 1091-1111, 2009.