Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 13, Issue 1, January 2022: 87-94

An Implentation Frame Work for Generating Closed Frequent Item Set Objects

Mr. Dr. U. Mohan Srinivas, Associate Professor,Dept Of Master Of Computer Applications In Qis College Of Engineering & Technology From JNTUK.,(A.P),India .
Mr. Mr.Sk. Anjaneyulu Babu, Assistant Professor Dept Of Master Of Computer Applications In Qis College Of Engineering & Technology From JNTUK.,(A.P),India .
Miss. R.Lahari ,Pg Scholar, Dept Of Master Of Computer Applications In Qis College Of Engineering & Technology From JNTUK.,(A.P),India .
Mr.D. Sandeep,Pg Scholar, Dept Of Master Of Computer Applications In Qis College Of Engineering & Technology From JNTUK.,(A.P),India .
Miss.J.Sai Sree Archana,Pg Scholar, Dept Of Master Of Computer Applications In Qis College Of Engineering & Technology From JNTUK.,(A.P),India .
Miss.K.Yamini,Pg Scholar, Dept Of Master Of Computer Applications In Qis College Of Engineering & Technology From JNTUK.,(A.P),India .

ABSTRACT :
In terms of item counts and income, high-utility itemset mining (HUIM) has emerged as a significant step in the pattern mining process. A number of approaches for mining high-utility itemsets have been developed (HUIs). Because these algorithms usually produce a significant number of detected patterns, a more compact and lossless version has been devised. The newly described closed high utility itemset mining (CHUIM) algorithms were designed to work with certain datasets (e.g., those without probabilities). Real-world data bases may include items or itemsets associated with probability values. Several methods for rapidly mining frequent patterns from uncertain data bases have been developed, however there is no technique to mine CHUIs from this kind of database. CPHUI-List is a new and efficient method for mining closed prospective high-utility itemsets (CPHUIs) from uncertain datasets without developing candidates, as described in this study. The proposed method is based on DFS and takes use of non-CPHUIs' downward closure characteristic, as well as a high transaction-weighted probabilistic mining top run. The experiment findings show that the proposed technique beats the CHUI Miner in terms of execution time and memory use.
Keywords *:* High-utility itemset mining, closed high utility itemset mining (CHUIM).

I INTRODUCTION

Mining frequent itemsets (FI) from databases is a key difficulty in knowledge discovery. There have been several efficient approaches for FIs (e.g., Apriori FP-Growth), frequent sequence patterns (e.g., GSP), data streams, and high-utility itemsets (HUIs) -. The concept of HUIM was first proposed in and was inspired by the issue of frequent itemset mining (FIM). A HUI is defined as an itemset with a utility value greater than or equal to a user-specified threshold. HUIM's purpose is to uncover a set of patterns that generate a lot of money (utility). HUIM is considered a more challenging assignment than FIM since the downward closure characteristic does not hold for utility measurements. In addition, the number of HUIs received and candidates formed throughout the HUIM process is usually considerable, necessitating a significant amount of runtime and memory. To tackle this difficulty, Tseng et al. proposed closed high-utility itemsets (CHUIs), which boost the mining process' efficiency and enable the extraction of a full set of HUIs. It's a CHUI if an itemset's utility value exceeds a user-specified minimum utility threshold and there are no supersets with the same support. As previously noted, data gained in real-world applications, such as locations retrieved through RFID or GPS or shopping habits gleaned from e-commerce websites, may be suspect. Traditional pattern mining algorithms (such as those for mining FIs or HUIs) are unable to function or provide incorrect results when applied to input data that is either incomplete or includes erroneous values. There are a number of methods for extracting useful information from ambiguous datasets. Chuietal demonstrated how to mine FIs from uncertain datasets using the UApriori approach, which uses a generate-and-test and breadth-first search strategy. Later, Leung et al. offer the UFP-Growth approach, which mines uncertain FIs without creating candidates using a tree structure called the UFP-tree. The CUFP tree was proposed by Lin and Hong as a means to simplify ambiguous frequent patterns. In addition, a strategy for mining uncertain FIs from the constructed tree was described based on this tree structure. By extracting just common patterns from uncertain data, PUF-Growth reduces runtime (i.e., no false negatives or positives). Linet al. recently presented the PU-List approach for mining probable HUIs (PHUIs) from unclear datasets.

II. LITERATURE SURVEY

A Time-Saving Approach to Mining Closed Itemsets using High Usefulness Mining closed high utility itemsets (CHUIs) requires determining a representative set of HUIs that is often less than the whole number of HUIs but capable of creating the complete set of HUIs without compromising information. As a consequence, several approaches for mining CHUIs have been proposed, with CHUI Miner and EFIM-Closed being the two most efficient algorithms. However, when mining CHUIs from sparse datasets, they have performance issues. In this paper, we offer a method for mining CHUIs in both dense and sparse datasets. We first modify the H-Miner algorithm's compact utility list structure to reduce mining time, and then devise backward and forward checking approaches based on the most recently researched CHUIs, which we combine with what we learnt while developing the subsequent tiers.

PHUIMUS: A High-Value Itemset Mining Algorithm Based on Stream Uncertainty Data Mining high utility itemsets (HUIs) has recently been a popular topic, since it can be used to mine profitable itemsets while taking both quantity and profit into account. Until far, academics have focused their attention on HUIs that mine unpredictable datasets and data streams. However, to our knowledge, the topic of HUIs mining over unpredictably large data streams has received little attention. This paper introduces the PHUIMUS (potential high utility itemsets mining over uncertain data stream) approach, which uses sliding windows to mine prospective high utility itemsets (PHUIs), which represent itemsets with high utilities and existential possibilities, over an uncertain data stream. To actualize the strategy, a potential utility list over uncertain data stream (PUS-list) is used to mine PHUIs without scanning the investigated uncertain data stream. The TWPUS-tree is a transaction-weighted probability and utility tree (TWPUS-tree) that is utilised by the PHUIs method to minimise the number of candidate itemsets created. Extensive investigations are carried out on real-life and synthetic data bases in terms of run-time, the number of recognised PHUIs, memory utilisation, and scalability. The results imply that our suggested method for extracting meaningful PHUIs from ambiguous data streams is both fair and acceptable.

III SYSTEM ANALYSIS

EXISTING SYSTEM

HUIs' mining techniques may generate a significant number of outputs and candidates. Algorithms based on a two-phase model may aid in this reduction. For example, all HUIs may be efficiently mined in this manner. Lan et al. introduced a projection-based index approach. It does, however, use a lot of memory and takes a long time to finish. To speed up the discovery of HUIs by taking item co-occurrences into consideration, Fournier-Viger et al. developed the FHM technique, which featured a structure called EUCS (Estimated Utility Co-occurrence Structure) and a pruning methodology called Estimated Utility Co-occurrence Pruning (EUCP). The HUP-Miner technique was created by Krishnamoorthy, and it extends the HUI-Miner approach by splitting the input data bases to minimise the search space. It also employed a look ahead strategy to expedite the creation of utility lists. For effectively mining HUIs, Zida et al. recently devised a technique called EFIM. To substantially trim the search space and decrease database scans, the authors developed two new and tighter upper limits, namely sub-tree utility and local utility, as well as two efficient algorithms, high-utilitydatabaseprojection (HDP) and high-utilitytransaction merging (HTM).

PROPOSED SYSTEM

The present study proposes an approach for mining closed prospective high-utility itemsets based on the tuple uncertainty database model (CPHUI). The proposed model is equivalent to the projected support model. Following that, a technique known as the CPHUI List is suggested. It uses the PEU-List (potential extended utility) format to mine CPHUIs. This research is a continuation of the approach presented in. The following are the main contributions of this paper:

I It introduces the CPHUI pattern, as well as the PEU-List data structure for mining CPHUIs.

(ii) A pruning approach termed Pr-Trim is presented to prune the search space and lower the cost of database scans using the proposed PEU-List.

(iii) To directly mine CPHUIs from uncertain datasets, an effective method named CPHUI-List algorithm is created based on the suggested PEU-List, Pr-Prunestrategy.

IV IMPLEMENTATION

The purpose of the design phase is to come up with a plan for fixing the problem described in the requirement document. Completing this phase is the first step in moving from the problem domain to the solution domain. To put it another way, we start with what we need and move backwards to see how design may help us satisfy those needs. The design of a system is possibly the most essential factor impacting software quality; it has a big impact on the stages after that, particularly testing and maintenance. This stage results in the design document. This document acts as a solution blueprint and is used in the implementation, testing, and maintenance of the solution. System Design and Detailed Design are the two stages of the design process. System design, also known as top-level design, is concerned with determining which modules should be included in the system, their needs, and how they interact with one another in order to accomplish the desired results. At the end of the system design, all of the key data structures, file formats, output formats, and significant modules in the system, as well as their requirements, are picked. Detailed Design determines the internal logic of each of the modules described in the system design. System design focuses on identifying the modules, while detailed design focuses on describing the logic for each module. System design focuses on what components are necessary in other works, while detailed design focuses on how the components may be implemented in software. The goal of design is to identify software components and their interactions. By outlining software structure and giving a blueprint, you may provide a blueprint for the document phase. Modularity is one of the desirable characteristics of large systems. It indicates that the system is broken down into multiple parts. Developers utilise this method to bridge the gap between the requirements specification established during elicitation and analysis and the system delivered to the user. Design is the place where brilliance is fostered through growth. The process of translating requirements into a visual representation of software is known as software design.

Module Description:

Data Owner:

The data owner must first register with the cloud server in this module (CS1, CS2, CS3, and CS4). The data owner will then log in to the cloud server that has been allocated to him. A file will be encrypted and sent to the cloud server by the data owner (CS1, CS2, CS3, CS4) The data owner double-checks that the file he uploaded is safe. The data owner may see the number of files that have been uploaded to the cloud servers (CS1, CS2, CS3, CS4) The data owner will send the file to the trust manager, who will store it on the cloud servers that are linked to it (CS1, CS2, CS3, CS4).

Cloud Server:

Data owners encrypt their files and upload them to the cloud, where cloud users may access them. Data consumers must first download and decode encrypted data files from the cloud to have access to the shared data files.

Trust Manager:

The trust manager is accessible to both the data owner and the end user, allowing them to log in. All cloud statuses are accessible to it. All positive and negative feedback submitted by the end user may be seen and listed by the trust manager. The trust management system determines the number of users in cloud services (IAAS, PAAS, and SAAS). The Trust Manager may view the attackers on cloud servers (CS1, CS2, CS3, and CS4), as well as the quantity of time attacks.

Cloud Consumer:

The cloud consumer must first register with the cloud server he will be utilising (CS1, CS2, CS3, and CS4). To begin, the cloud subscriber must log in to the cloud to which he has subscribed. Customer comments on cloud-based data (positive or negative feedback) The attacker will have access to both registered users and cloud files. Attacks against collaboration with the goal of deceiving the public about the cloud When a person conducts a significant number of transactions in a single day, this is known as a Sybil Attack (Exceeds the limit which is assigned by the Trust Manager).

V RESULT AND DISCUSSION

Fig 1 : user login

Fig 2: Admin login

Fig 3: Products with ranks

Fig 4:Use reviews

Fig 5: Exit page

VI CONCLUSION

This research suggested a method for extracting CPHUIs from uncertain data sets without having to generate candidates. HUIs and high-probability itemsets are mined using the CPHUI-List algorithm. The PEU-List structure and the set-enumeration tree serve as the foundation for the algorithm. It mines CPHUIs directly using a DFS-based technique. In terms of runtime and consumed memory, experimental findings for actual databases demonstrate that CPHUI-List outperforms CHUI-miner. This is the initial stage in extracting CPHUIs from shaky databases. We will modify the structure of PEU-List in order to increase the efficiency of the CPHUI-List method in the future, and we will expand our work to mine maximal-potential HUIs and top k-potential HUIs.

VII REFERENCES

[1] V. S. Tseng, C.-W. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient algo rithms for mining the compact and lossless representation of high utility itemsets," IEEE Trans. Knowl. Data Eng., vol. 27, no. 3, pp. 726–739, August 2015.

[2]."A study of uncertain data algorithms and applications," IEEE Trans. Knowl. Data Eng., vol. 21, no. 5, pp. 609–623, May 2009.

[3] C. C. Aggarwal and P. S. Yu, "A survey of uncertain data algorithms and applications," IEEE Trans. Knowl. Data Eng., vol. 21, no. 5, pp. 609–623, May 2009.

[4]."Mining uncertain data with probabilistic assurances," in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 273–282.

[5]"AT-Mine: An effective method of frequent itemset mining on uncertain dataset," J. Comput., vol. 8, no. 6, pp. 1417– 1426, May 2013.

[6]"Mining frequent itemsets from uncertain data," C.-K. Chui, B. Kao, and E. Hung, in Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining, Jun. 2007, pp. 47–58.

.

[7]. "A new mining technique for uncertain databases using CUFP trees," Expert Syst. Appl., vol. 39, no. 4, pp. 4084–4093, Mar. 2012. C.-W. Lin and T.-P. Hong, "A new mining strategy for uncertain databases using CUFP trees," Expert Syst. Appl., vol. 39, no. 4, pp. 4084–4093, Mar. 2012.

[8]. "PUF-tree: A compact tree structure for frequent pattern mining of uncertain data," in Proc. 17th Pacific–Asia Conf. Knowl. Discovery Data Mining, 2013, pp. 13–25. C. K.-S. Leung and S. K. Tanbeer, "PUF-tree: A compact tree structure for frequent pattern mining of uncertain data," in Proc. 17th Pacific–Asia Conf. Knowl. Discovery Data Mining

[9]. "Effi cient algorithms for mining high-utility itemsets in uncertain datasets," J. C.-W. Lin, W. Gan, P. Fournier-Viger, T.-P. Hong, and V. S. Tseng," Mar. 2016, vol. 96, pp. 171–187, Knowl.-Based Syst."Efficient query evaluation on probabilistic databases," VLDB J., vol. 16, no. 4, pp. 523–544, 2007.

[10]. D. Nilesh and S. Dan, "Efficient query evaluation on probabilistic databases," VLDB J., vol. 16, no. 4, pp. 523–544, 2007.

[11]"Mining frequent item- sets over uncertain databases," Proc. VLDB Endowment, vol. 5, no. 11, pp. 1650– 1661, Jul. 2012. Y. Tong, L. Chen, Y. Cheng, and P. S. Yu, "Mining frequent item- sets over uncertain databases," Proc. VLDB Endowment, vol. 5, no. 11, pp. 1650– 1661, Jul. 2012.

[12] "Mining closed high utility itemsets in uncertain databases," N. Bui, B. Vo, V.-N. Huynh, C.-W. Lin, and L. T. T. Nguyen, in Proc. 7th Symp. Inf. Commun. Technol., Ho Chi Minh City, Vietnam, 2016, pp. 7–14.

[13] "Mining closed high utility itemsets without candidate generation," C.-W. Wu, P. Fournier-Viger, J.-Y. Gu, and V. S. Tseng, in Proc. Conf. Technol. Appl. Artif. Intell. (TAAI), Nov. 2015, pp. 187–194.

[14]"BAHUI: Fast and memory efficient mining of high utility itemsets based on bitmap," Int. J. Data Warehousing Mining, vol. 10, no. 1, pp. 1–15, Jan. 2014. W. Song, Y. Liu, and J. Li, "BAHUI: Fast and memory efficient mining of high utility itemsets based on bitmap," Int. J. Data Warehousing Mining, vol.

[15]"UP-growth: An efficient approach for high utility itemset mining," in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 253–262.

[16]"Fast identification of high utility itemsets from candidates," Information, vol. 9, no. 5, p. 119, May 2018. [16]. J.-F. Qu, M. Liu, C. Xin, and Z. Wu, "Fast identification of high utility itemsets from candidates," Information, vol. 9, no. 5, p. 119, May 2018.

[17]"An efficient projection-based indexing strategy for mining high utility itemsets," G.-C. Lan, T.-P. Hong, and V. S. Tseng," Jan. 2014, vol. 38, no. 1, pp. 85–107, Knowledge Inf. Syst.

[18]"FHM: Faster high utility itemset mining using estimated utility co-occurrence pruning," in Proc. 21st Int. Symp. Methodol. Intell. Syst., P. Fournier-Viger, C.-W. Wu, S.