

A New Approach to Evaluate Best-k High Beneficial Itemsets

Kumaraswamy G , Nagamalleswararao Tadikonda
ASST.PROFESSOR, Department of CSE GITAM Institute of Technology,
kumaraswamy521@gmail.com

Department of Computer Science, Programmer Analyst, Trans IT mpower labs pvt.ltd, Bangalore, Karnataka-560029, India.
nagamalleswararao@mpowerglobal.com

Abstract:

The main purpose of this work is to develop a superior structure to Evaluate Best-k high beneficial itemsets. Here K is the picked portion of high beneficial itemsets that is to be established. High beneficial item set tunneling is surely a prominent study in data mining but the factors for setting minimum utility margin is definitely a difficult task. In this work, a novel approach to evolve Best-k high beneficial itemsets named Enhanced Best-k high beneficial item set tunneling (EBKU) is proposed. EBKU uses B+ Tree data structure instead of using a Utility Pattern Tree (UP-Tree) data structure that is used in existing Best-k high beneficial itemsets tunneling (TKU) method. Although TKU helps to reduce the time taken for the process of tunneling by reducing the total number of database scans to two, the complexity lies in the UP-Tree traversal for obtaining potential Best-k high beneficial itemsets. B+ Tree used in EBKU does not have data associated with interior nodes so that more keys can fit into the memory. The leaf nodes of B+ Tree are linearly linked, so a full scan of a tree requires only one linear pass through all the leaf nodes.

Keywords

Best-k high beneficial itemsets tunneling, transaction weighted utilization, minimum utility, B+ Tree, UP-Tree, frequent itemsets mining

Introduction

Data mining is stated as the policy to bring out the useful information from vast quantity of data. The technique for uncovering the data to identify unrevealed relationships and project future trends has been a prolonged history. Occasionally data mining is also mentioned as knowledge discovery in databases. Data mining is a process of discovering peculiarities, patterns and correlations in huge sets of data to forecast the end result. By using a wide variety of techniques, this information could be used to raise revenues, reduce cost and enhance customer relationships.

Frequent itemsets mining is an elementary research issue in data mining. Frequent itemsets mining is one of the foremost well known and most accepted data mining procedures. It was actually introduced for market basket analysis but later on it is used for almost any task which needs to identify regularities among variables. It focuses on identifying the regularities in the purchase behavior of the buyer of supermarkets, mail order companies and online stores. Especially, it attempts to find sets of items that are often purchased together.

Utility mining appears as a key concept in data mining. In utility mining, every item is assigned with a value and count of existence in each transaction. The effectiveness of an item set shows its significance which can be measured

regarding weight, worth, volume or other statistics depending upon the user's interest. An item set is called high beneficial item set if its value is greater than the user specified minimum utility margin. In most of the cases, discovering a good minimum utility margin by hit and miss method is truly a laborious process. If minimum utility value is inadequate then great number of high beneficial itemsets may originate and that will be genuinely ineffective. However, if minimum utility is excessive then no high beneficial itemsets are going to appear. To overcome the above mentioned consequences, a state-of-the-art scheme to find the Best-k high beneficial itemsets is proposed in this work. Here K is the selected count of high beneficial itemsets that is to be established.

Using a constant K alternative to minimum utility margin is advisable for numerous applications. For instance, to examine consumer purchase behavior, Best-k high beneficial itemsets tunneling serves as an optimistic result for the users who wish to know "What are the Best-k sets of products that are bringing huge profits to the company?"

However, reducing the search space for finding high beneficial itemsets is hard because a superset of low beneficial itemsets can be high beneficial itemsets. In order to face this issue, the transaction weighted utilization (TWU) model was brought to smoothen the functioning of mining task. An itemsets is said to be high transaction weighed beneficial itemsets (HTWBI) if its TWU is greater than the minimum utility.

A traditional TWU representation based algorithm undergoes in two stages. In the first stage, called stage 1, the whole set of

HTWBIs are originated. In the second stage, called stage 2, by computing the accurate utility values of HTWBIs with a database scan, the resultant high beneficial itemsets are acquired.

Two active methods named TKU and TKO are implemented in [1] for drilling such itemsets without setting minimum utility margin. TKU is the two-stage method for tunneling Best-k high beneficial item sets, which integrates five strategies namely pre-evaluation (PE), elevating the margin by node utilities (NU), elevating the margin by minimum utility values of descendants (MD), elevating the margin by considering minimum utility of candidates (MC) and elevating the margin by classifying and computing exact utilities of candidates (SE) to successfully lift the border minimum utility margin and moreover reduces the search space.

The TKO is a one-stage method used for tunneling Best-k high beneficial itemsets, which combines three strategies namely raising the margins by utilities of the candidates (RUC), reducing the approximate utility values by using Z-elements (RUZ) and exploring most promising branches first (EPB) significantly enhance its performance.

Related Work

Joyful Yin et al. [2] presented an effective method named Best-k high utility sequence tunneling (TUS) for digging Best-k high beneficial sequential patterns out of utility-based sequence databases. The method promises that no sequence is missed during the digging process. A new sequence border and a corresponding pruning strategy are introduced to eliminate the unpromising candidates. Pre-insertion and sorting

methods are included to lift the minimum utility margin.

Tran Minh Quant et al. [3] designed a new method named Examiner for tunneling Best-k frequent patterns. The sew-BOMA approach is a fusion of seq-ExMiner method and the idea of “build once mine anytime” feature. It permits users to dig Best-k frequent patterns where user need not specify any minimum support margin value.

Cheng Wei Wu et al. [4] presented a new scenario for tunneling high beneficial episode in complex event sequences. To find out the utility of episodes, both internal utility and external utility of events are considered. For tunneling high beneficial episodes containing simultaneous events, the framework of complex event sequences is taken into consideration. A novel method named utility episodes tunneling by spanning prefixes (UP-span) is proposed for tunneling complete set of high beneficial episodes. The TWU model is extended to episode tunneling and episode weighted utilization (EWU) model is proposed to smoothen tunneling process. The UP-span method is included with two efficient strategies named discarding global unpromising events and discarding local unpromising events for reducing the candidates count in tunneling process and to improve the performance of tunneling task.

Vincent S. Tseng et al. [5] says digging high beneficial itemsets from a transactional database mentions finding the itemsets with high utility. An efficient algorithm named Utility Pattern Growth is proposed for tunneling high beneficial itemsets. The information of high beneficial itemsets is organized in a special data structure called Utility Pattern Tree so that candidate

itemsets will be generated with two database scans.

Proposed Method

A novel approach to dig Best-k high beneficial itemsets using EBKU is proposed. The selection of data structure and search policy will affect the effectiveness of Best-k high beneficial itemsets tunneling approach regarding execution time and memory. Crucial contributions for this approach are outlined as follows:

An improved version of TKU named Enhanced Best-k beneficial itemsets tunneling (EBKU) is implemented for tunneling the entire group of Best-k high beneficial itemsets in databases without the requirement to state the minimum utility threshold. Figure 1 represents the structures of the EBKU method. The EBKU method adopts B+ Tree to preserve the information of transactions and utilities of itemsets. EBKU inherits functional characteristics from the TWU model and contains the two stages. In stage 1, potential Best-k high beneficial itemsets (PKHBIs) are produced. In stage 2, Best-k high beneficial itemsets are recognized from the set of PKHBIs identified in stage 1.

The existing TKU method uses UP-tree. Another existing method TKO uses a list-based design named utility list to reserve the utility information of the itemsets in the database. Vertical data representation approach is used to identify the Best-k high beneficial itemsets in a single stage. The proposed EBKU uses B+ tree.

The methods used for high beneficial itemsets tunneling can be commonly classified into two types: two stage and one stage methods. The chief characteristic of a

two stage method is it contains two stages. In the first stage, a set of candidates called PKHBIs are produced. In the second stage, the precise utility of every candidate identified in the first stage are calculated to discover the high beneficial itemsets. The characteristic of one stage method is that high beneficial itemsets are discovered using a single stage and it does not generate any candidates. High beneficial itemsets digger considers a database in a vertical format and converts it into utility lists. The utility list format used in high beneficial itemsets digger (HBI-Digger) permits to directly calculate the utility of produced itemsets in the main memory by eliminating the original database scan. In this paper, a two stage method is used for EBKU as that in existing TKU but using B+ tree.

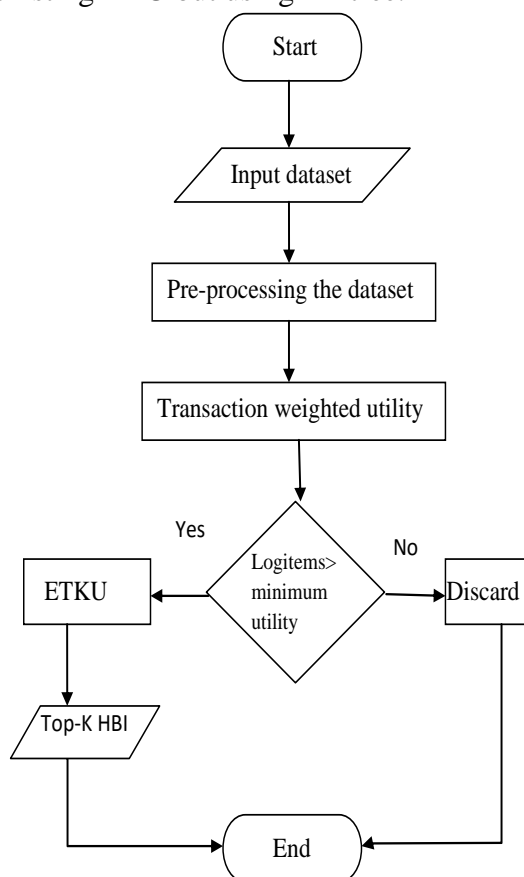
Figure 1. Proposed structure

1.1. EBKU Structure

TKU is an efficient method to discover Best-k high beneficial itemsets without setting minimum utility. In the existing systems, although the existing algorithms help to speed up the process of rule mining by reducing the total number of database scans to two, the complexity lies in the frequent UP-Tree traversal for obtaining potential high beneficial item sets. Tree Traversal of UP-Tree structure requires various number of iterations.

In this work, B+ Tree data structure is proposed instead of UP-tree to optimize the solution. Because B+ Tree does not have data associated with interior nodes and more keys can fit on a page of memory. Therefore, fewer cache misses are required in order to access data that is present on a leaf node. The leaf nodes of B+ Tree are linked, so doing a full scan of all objects in a tree requires just one linear pass through all the leaf nodes.

The chief objective of a B+ Tree is rapid traversal and expeditious search. Any record in the B+ Tree can be searched quickly since all the nodes are kept at the same distance and the balance of the tree is maintained properly. As the record count in the database increases, the intermediate nodes and leaf nodes are needed to be separated and spread widely to maintain the balance of the tree. Since the nodes are widely spread, the time taken for searching a record becomes faster. As the branches of the tree are widely spread, it takes less I/O on the disk to retrieve the record. Records that are to be retrieved are retrieved in logarithmic fraction of time.



If the file grows in size in the database, the functioning of B+ Tree remains constant. This is due to the maintenance of the records at the leaf nodes and all the nodes are at equal distance from the root. Moreover, if overflow situation arises then the structure of the tree is automatically reorganized. The reorganization of the tree does not affect the performance. B+ Tree has good space utilization because all the intermediate nodes contain pointer to the records and the records are only held by leaf nodes. The space utilized by the pointers is very less when compared to the space utilized by the records.

The EBKU method adopts B+ Tree data structure to retain the information of the transactions and Best-k high beneficial itemsets. The EBKU method is implemented in three steps. In the first step, a B+ Tree is constructed. In the second step, potential Best-k high beneficial itemsets are generated from the B+ Tree and in the third

step Best-k high beneficial itemsets are discovered from the set of PKHBIs.

It requires two database scans to construct a B+ Tree. The tree construction is explained with an example. Table 1 is an example database which contains five transactions. Let I^* be the set of distinct items $I^* = \{I_1, I_2, \dots, I_n\}$, here in the considered example, $I^* = \{P, Q, R, S, T, U, V\}$. Let database $D = \{T_1, T_2, \dots, T_m\}$, here $D = \{T_1, T_2, T_3, T_4, T_5\}$ is a set of transactions and each transaction in the database is a subset of I^* . Every item is associated with internal utility and external utility. The internal utility of an item is the count of occurrence of that item in a particular transaction. External utility is the profit assigned to that item. Table 2 contains the external utilities of all the items in the Table 1 example database. Table 3 contains the occurrence count of each item in the database.

Table 1. Example Database

Tno	Transaction	Transaction Utility (TU)
T1	(P,1) (R,1) (S,2)	10
T2	(P,1) (Q,4) (R,1) (S,6) (T,1) (U,4)	25
T3	(P,2) (R,6) (T,2) (V,2)	34
T4	(Q,2) (R,1) (T,1) (V,2)	11
T5	(Q,4) (R,1) (S,1) (T,3)	16

Table 2. Profit Table

Item	P	Q	R	S	T	U	V
Unit Profit	6	1	2	1	3	1	2

Table 3. Items and their occurrence count

Item	P	Q	R	S	T	U	V
Occurrence count	4	10	10	9	7	4	4

During the first database scan, the tree construction is done by calculating the occurrence count of each item in all the transactions. The process of inserting an item into the tree is as follows:

- 1) Detect the correct leaf position X1. The search starts at the root node and the key comparisons will direct it to a leaf.
- 2) Try to insert the node into the position X1
 - a) If X1 has enough space for new item, then insert the item.
 - b) Else, split X1 (into X1 and a new node X2)
 - i) Reassign X1 entries evenly between X1 and X2
 - ii) Duplicate the middle key, i.e., recursively insert the middle key into the parent of X1 and add a pointer from X1's parent to X2
- 3) While inserting a new node into an internal node Y1
 - a) If Y1 has enough space, then insert the new item.
 - b) Else, split Y1 (into Y1 and a new node Y2)
 - i) Reassign Y1 entries evenly between Y1 and Y2.
 - ii) Move up the middle key.
- 4) Splits spread the tree by making it broad. When the tree splits, the height of the tree gets increased by one.

For Table 1, the B+ Tree structure after the first database scan is shown below:

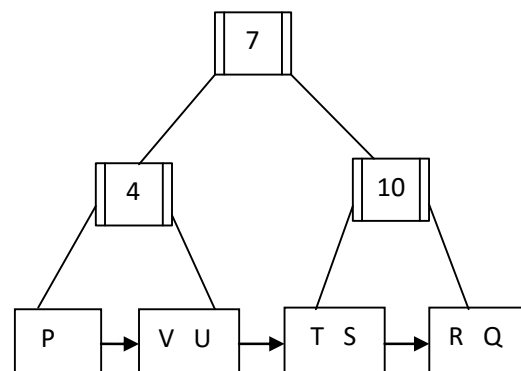


Figure 2. B+ Tree structure after first database scan

Before the second database scan, the items with less occurrence count and less unit profit can be discarded in the second database scan. During the second database scan, the tree is reorganized based on the transaction weighted utility (TWU) values of the items in the database. The TWU value of each item is calculated by using the following formulae.

Definition 1: Total utility of an item I_k in a transaction T_s is defined as $EU(I_k, T_s) = EX(I_k, D) * IN(I_k, T_s)$, here EX and IN are external and internal utilities respectively.

Definition 2: Total utility of an itemsets X in a transaction T_s is defined as $EU(X, T_s) = \sum_{I_k \in X} (I_k, T_s)$

Definition 3: Transaction Utility (TU) of a transaction T_s in a database D is defined as $TU(T_s) = EU(T_s, T_s)$.

The Transaction Utility of T_1 in the Table 1 is $TU(T_1) = EU(P, T_1) + EU(R, T_1) + EU(S, T_1) = (6*1 + 2*1 + 1*2) = 10$. The transaction utilities of all the transactions are calculated in the same manner and are mentioned in the Table 1.

Definition 4: The transaction weighted utility (TWU) of an itemsets X is the sum of all transaction utilities of the transactions in the database containing X .

Definition 5: An itemsets X is called Best-k high beneficial itemsets in a database D if there are less than K itemsets whose utility values are greater than $EU(X)$ in $f_{HBI}(D, O)$

Definition 6: An itemsets is called potential Best-k high beneficial itemsets (PKHBI) if its TWU and maximum utility are greater than minimum utility margin.

The TWU of an item P in Table 1 is $TWU(P) = TU(T_1) + TU(T_2) + TU(T_3) = 10 + 25 + 34 = 69$. Table 3 contains the TWU of all the items that are present in the database.

Table 3. Items and their TWUs

Item	P	Q	R	S	T	U	V
TWU	69	52	96	51	70	25	45

The leaf nodes of the B+ Tree are reorganized based on the TWU values and the reorganized tree for the above example is shown below.

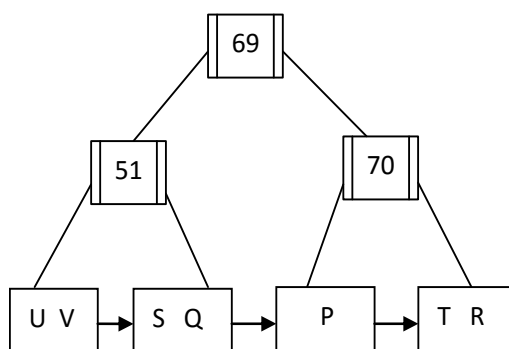


Figure 3. B+ Tree structure after second database scan

The items are retrieved from the B+ Tree by a doing a full scan and then they must be arranged in the descending order of their transaction utilities. To discover the PKHBIs, EBKU uses an internal variable

named as minimum utility margin which is set to 0 initially and it is lifted dynamically after identifying certain count of itemsets with high utilities. Once the PKHBIs are discovered then EBKU computes the utility

of PKHBIs by scanning the original database once to discover the Best-k high beneficial itemsets.

Conclusion

In this paper, the issue of Best-k high beneficial itemsets tunneling is studied, where K is the picked portion of high beneficial itemsets that is to be established. A novel method named EBKU is proposed using a B+ Tree data structure. B+ Tree does not have data associated with interior nodes so that more keys can fit on a page of memory. Therefore, fewer cache misses are required to access data that is present on a leaf node. In a B+ Tree, a full scan of all objects in a tree requires just one linear pass through all the leaf nodes. The main objective of a B+ Tree used in EKTU is rapid traversal and expeditious search. The EBKU is a two stage method for tunneling Best-k high beneficial itemsets. The proposed EBKU method takes less time and more effective than TKU method for obtaining Best-k high beneficial itemsets.

References

- [1] Tseng, Vincent S., Cheng-Wei Wu, Philippe Fournier-Viger, and S. Yu Philip. "Efficient algorithms for mining Best-k high utility itemsets." *IEEE Transactions on Knowledge and Data Engineering* 28, no. 1 (2016): 54-67.
- [2] Yin, Junfu, Zhigang Zheng, Longbing Cao, Yin Song, and Wei Wei. "Efficiently mining Best-k high utility sequential patterns." In *13th International Conference on Data Mining (ICDM)*, pp. 1259-1264. IEEE, 2013.
- [3] Quang, Tran Minh, Shigeru Oyanagi, and Katsuhiko Yamazaki. "ExMiner: An efficient algorithm for mining Best-k frequent patterns." In *International Conference on Advanced Data Mining and Applications*, pp. 436-447. Springer, Berlin, Heidelberg, 2006.
- [4] Wu, Cheng-Wei, Yu-Feng Lin, Philip S. Yu, and Vincent S. Tseng. "Mining high utility episodes in complex event sequences." In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 536-544. ACM, 2013.
- [5] Tseng, Vincent S., Cheng-Wei Wu, Bai-En Shie, and Philip S. Yu. "UP-Growth: an efficient algorithm for high utility itemset mining." In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 253-262. ACM, 2010.