# An Efficient Three-Scan Approach for Mining High Utility Itemsets

Guo-Cheng Lan[1], Tzung-Pei Hong[2,3], and Vincent S. Tseng[1,4]

[1]Department of Computer Science and Information Engineering
National Cheng Kung University, 701, Taiwan
[2]Department of Computer Science and Information Engineering
National University of Kaohsiung, 811, Taiwan
[3]Department of Computer Science and Engineering
National Sun Yat-Sen University, 804, Taiwan
[4]Institute of Medical Informatics
National Cheng Kung University, 701, Taiwan

rrfoheiay@gmail.com, tphong@nuk.edu.tw, tsengsm@mail.ncku.edu.tw

**Abstract:** Utility mining finds out high utility itemsets by considering both the profits and quantities of items in transactions. In this paper, a three-scan mining approach is proposed to efficiently discover high utility itemsets from transaction databases. The proposed approach utilizes an itemset-generation mechanism to prune redundant candidates early and to systematically check the itemsets from transactions. Finally, the experimental results on a synthetic dataset show the superior performance of the proposed approach.

**Keywords:** Data mining, utility mining, high utility itemsets, the filtration mechanism.

## 1 INTRODUCTION

In the field of database knowledge, data mining techniques have been widely applied to various practical applications, such as supermarket promotions, biomedical data applications, multimedia data applications, and so forth. Association-rule mining [1] is one of the most important issues in data mining since the relationship among items in a database can be found by association-rule mining techniques. Traditional association rule mining [1], however, considers the occurrence of items in a transaction database but do not reflect any other factors, such as price or profit. Then, some product combinations with low-frequency but high-profit may not be found in association-rule mining. For example, assume there is a product combination like {DVD player, LCD TV}. The product combination may not have a high frequency in a transaction database, but may contribute a high utility in this database due to LCD TV. To handle this, a practical issue, namely utility mining, was thus proposed by Chan *et al.*, in 2003 [2]. In Chan *et al.*’s study [2], their proposed approach considered both the individual profits and quantities of products (items) in transactions, and used them to measure actual utility value for an itemset. The high utility itemsets, which had their utility values larger than or equal to a predefined minimum utility threshold, were then found as the desired.

However, since the downward-closure property in association-rule mining cannot directly be adopted to find high utility itemsets in utility mining. To deal with this, a new downward-closure property was designed by Liu *et al.*, in 2005 [6]. The property was called transaction-weighted utilization (abbreviated as *TWU*) model [6]. In addition, Liu *et al.* applied the model to their proposed two-phase utility mining algorithm (abbreviated as *TP*) [6] to complete the utility mining task. However, Liu *et al.*’s utility mining approach had to run level by level as the *Apriori* algorithm in association-rule mining, thus needing scanning database multiple times. As described above, it is quite urgent to efficiently complete the utility mining task.

In this study, we propose an efficient three-scan utility mining algorithm (abbreviated as *TSA*) for handling the problem of utility mining. Especially, a new filtration mechanism in the proposed algorithm is designed to effectively prune a large number of unpromising candidates for mining. Also, the proposed algorithm calculates both the transaction-weighted utility and the actual utility of each itemset at the same time. Finally, the experimental results show that the proposed *TSA* algorithm executes faster than the *TP* algorithm under various parameter settings.

The remaining parts of this paper are organized as follows. Some related works, the problem to be solved and related definitions are reviewed in Section 2. The proposed mining algorithm with a filtering strategy for finding high utility itemsets from a transaction database is stated in Section 3. The experimental results are shown in Section 4. Conclusions and future works are finally given in Section 5.

## 2 REVIEW OF RELATED WORKS

According to the principle of association-rule mining [1], only binary itemsets are considered. In practical applications, however, products bought in transactions usually contain both profits and quantities. Thus, for some high-profit products with low frequency, these items may not be found by the association-rule mining algorithms. For example, both jewel and diamond have high utility values but may not be a high frequency combination when compared to food and drink in a database. To deal with this, Chan *et al.* subsequently proposed utility mining to find high utility itemsets from a transaction database [2]. In this study [2], a utility itemset considers not only the quantities of the items in transactions, but also their individual profits.

However, traditional association-rule mining keep the downward-closure property, but utility mining does not. Liu *et al.* thus proposed a two-phase utility mining (*TP*) algorithm to handle this [6]. This approach used a new property, which was named as the transaction-weighted utilization (*TWU*) model, to find all high utility itemsets. It used the summation of utility values of all the items in a transaction as the upper bound of any itemset in that transaction to keep the downward-closure property. In Liu *et al.*'s study [6], their proposed algorithm could be divided into two phases. In the first phase, the possible candidates were found from a database by the *TWU* model. Then, in the second phase, an additional data scan was executed again to find the actual utility of each candidate, and then itemsets with actual utility values larger than or equal to a predefined minimum utility threshold were output. However, since the *TP* algorithm adopted the level-wise technique to find high utility itemsets from a database, the algorithm had to spend a great deal of time on data scan. Afterward, although some related studies about utility mining were published [3][5][7][8], but most of them were still based on the principle of the *TP* algorithm to find their desired interesting utility patterns. It is thus desirable to effectively find high utility itemsets from a database and reduce the number of candidate itemsets.

## 3 THE PROPOSED MINING ALGORITHM

In this section, the filtration mechanism in the proposed algorithm is first described below.

### 3.1 The Filtration Mechanism

In the proposed algorithm, we design an itemset-generation approach to reduce the number of database scans. In particular, a filtration mechanism in the itemset-generation approach is proposed to avoid producing a big number of unpromising candidate utility itemsets. The mechanism is based on the high transaction-weighted utilization (*HTWU*) *2*-itemsets and on Liu *et al.*'s approach to generate possible transaction-weighted utilization itemsets [6]. Thus, the high transaction-weighted utilization *2*-itemsets (abbreviated as $HTWU_2$) have to be first found, and then these itemsets are used to check a candidate itemset whether there exists at least a low transaction-weighted utilization *2*-itemset in the candidate itemset. If it does, the candidate itemset is identified as an unpromising candidate itemset, and then it is removed; otherwise, it is can be kept in the set of candidate itemsets. A simple example is first given below to illustrate the idea.

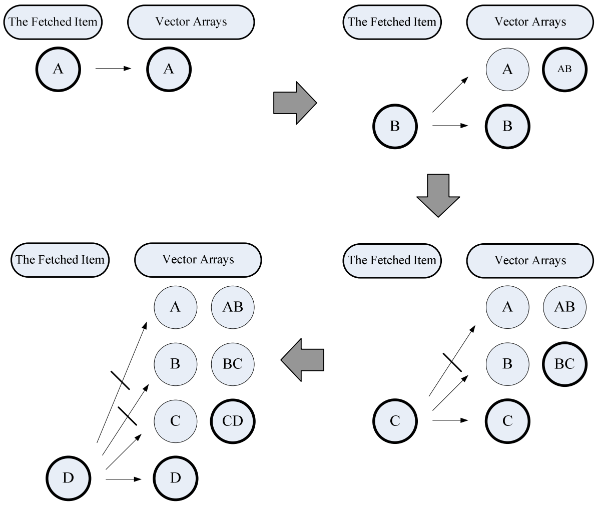

**Fig. 1.** The whole process of generating itemsets by using the filtration mechanism.

**Example 1:** Assume that a transaction *T* includes four items, 3*A*, 2*B*, 25*C* and 3*D*, where the numbers represent the quantities of the items. Also assume their profit values are 3, 10, 1 and 6, respectively, and the three itemsets {*AB*}, {*BC*}, and {*CD*} are all high transaction-weighted utilization *2*-itemsets, which have been found. Figure 1 shows the process of generating itemsets by using the filtration mechanism.

In Figure 1, the proposed algorithm first fetches the first item *A* in *T* and allocates it to the first row of the two dimensional vector array. The algorithm then fetches the second item *B* in *T* and allocates it to the second row of the vector array. Since there is only the item *A* in front of the fetched item *B*, the algorithm then checks whether items *A* and *B* have high transaction-weighted utilization relationship. In this example, {*AB*} is a high transaction-weighted utilization *2*-itemset. It is thus generated and put into the back of {*A*} in the first row because the first item in {*AB*} is *A*. The algorithm then continues to fetch the third item *C* and performs the same process. It puts {*C*} in the third row of the vector array, forms {*AC*}, and checks whether {*AC*} is a high transaction-weighted utilization *2*-itemset. In the example, {*AC*} is not, such that no combination of a subset in the first row with {*C*} is necessary. The algorithm then forms {*BC*} from the second row and finds it is a high transaction-weighted utilization *2*-itemset. {*BC*} is thus put in the back of {*B*} in the second row. Two new subsets {*BC*} and {*C*} are generated for the third item. Similarly, the fourth item *D* is fetched and the above process is repeated again. The two new subsets {*CD*} and {*D*} are derived and put into the suitable vector array.

As seen in this example, the filtering mechanism can be effectively adopted to reduce the number of unpromising candidate itemsets and speed up the execution efficiency.

### 3.2 The Three-scan Utility Mining Algorithm, *TSA*

The details of the algorithm are listed below.

INPUT: A set of items, each with a profit value; a transaction database, in which each transaction includes a subset of items with quantities; the minimum utility threshold.

OUTPUT: The final set of high utility itemsets (*HU*).

**Scan 1:**

STEP 1. For each *y*-th transaction $Trans_y$ in *D*, do the following substeps.

(a) Calculate the utility value $u_{yz}$ of each *z*-th item $i_{yz}$ in $Trans_y$ as:

$$u_{yz} = s_{yz} * q_{yz} ,$$

where $s_{yz}$ is the profit of item $i_{yz}$ and $q_{yz}$ is the quantity of $i_{yz}$ .

(b) Calculate the transaction utility $tu_y$ of $Trans_y$ as:

$$tu_y = \sum_z u_{yz} .$$

STEP 2. For each item *i* in *D*, calculate the transaction-weighted utility $twu_i$ of the item *i* as the summation of the transaction utility values of the transactions which include the item *i*. That is:

$$twu_i = \sum_{i \in Trans_y} tu_y .$$

STEP 3. For each item *i* in the set of candidate *1*-itemsets, if the transaction-weighted utility $twu_i$ of *i* is larger than or equal to the minimum utility threshold, put it in the set of high transaction-weighted utilization *1*-itemsets, $HTWU_1$.

**Scan 2:**

STEP 4. For each $Trans_y$ in *D*, do the following substeps.

(a) Check each item in $Trans_y$ whether it is a member in the set of $HTWU_1$. If it is, put it in the modified transaction $Trans_{y'}$; Otherwise, it is omitted.

(b) Check the number of items in the modified transaction $Trans_{y'}$ whether its number of items is larger than or equal to 2. If it is, keep the transaction $Trans_{y'}$ in the set of the modified transactions, and do the next step; Otherwise, remove the transaction $Trans_{y'}$, and do substep (a).

(c) Generate all possible *2*-itemsets in $Trans_{y'}$, and then put them in the set of $C_2$.

(d) Add the transaction utility $tu_y$ in the suitable transaction-weighted utility field values of the generated *2*-itemsets in the set of $C_2$.

STEP 5. Check whether the transaction-weighted utility $twu_x$ of each *2*-itemset *x* in the set of $C_2$ is larger than or equal to the minimum utility threshold , put it in the set of high transaction-weighted utilization *2*-itemsets, $HTWU_2$.

**Scan 3:**

STEP 6. Initialize the temporary itemset table as an empty table, in which each tuple consists of two fields: itemset and actual utility.

STEP 7. For each transaction $Trans_{y'}$ in the set of the modified transactions, do the following substeps.

(a) Generate all possible itemset *x* in $Trans_{y'}$ by using the set of $HTWU_2$, where the relationship between any two items in an itemset has to be a high transaction-weighted utilization relationship.

(b) Find the utility values $u_{yx}$ of the itemsets in $Trans_{y'}$, and then add in their actual utility field values in the temporary itemset table.

STEP 8. Check whether the actual utility $au_x$ of each *r*-itemset *x* in the temporary itemset table is larger than or equal to the minimum utility threshold , put it in the set of high utility *r*-itemsets, $HU_r$.

STEP 9. Output the final set of high utility itemsets, *HU*.

## 4 EXPERIMENTAL EVALUATION

In the experiment, the public *IBM* data generator was used in our experiment to produce the dataset [4]. Since our purpose was to find out high utility itemsets, we thus developed a simulation model, which was similar to that used in Liu *et al*. [6], to generate the quantities of the items in the transactions. In addition, the parameters used in the *IBM* data generator [4] were *T*, *I*, *N* and *D*, which represented the average length of items per transaction, the average length of maximal potentially frequent itemsets, the total number of different items, and the total number of transactions, respectively. All the algorithms were implemented in J2SDK 1.5.0 and executed on a PC with 3.0 GHz CPU and 1 GB memory.

Figure 2 then showed the execution time comparisons of the two algorithms for the synthetic T10I4N4KD200K dataset with various minimum utility thresholds, *min_util*.

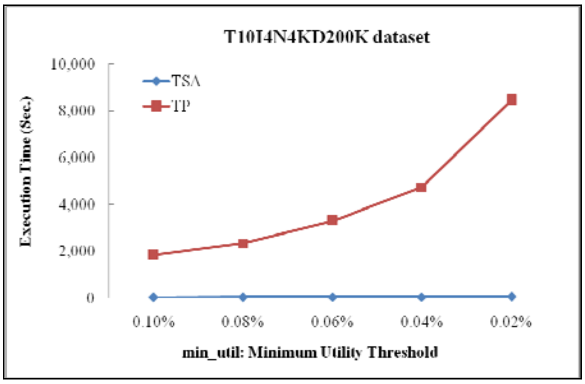

**Fig. 2.** Execution time of the two algorithms along with various minimum utility thresholds

It could be seen in the figure that the efficiency of the *TSA* algorithm was better than that of the *TP* algorithm, especially when the minimum utility threshold decreased. With the filtration mechanism, the *TSA* algorithm could thus effectively reduce a great number of unpromising candidates to achieve the goal of finding high utility itemsets. For the *TP* algorithm, since it adopted a level-wise technique to handle the problem of utility mining, a huge number of candidates had to be generated for mining. Thus, the candidate itemset requirement of the *TSA* algorithm was smaller than that needed by the *TP* algorithm.

## 5 CONCLUSIONS

In this paper, we have proposed an efficient three-scan utility mining approach (*TSA*) to find high utility itemsets from a transaction database. Especially, the proposed algorithm only needs three data scan to achieve the utility mining task. Also, the filtration mechanism adopts in this work can effectively skip unpromising candidate itemsets and thus further save time. The experimental result shows that the proposed mining algorithm is able to execute faster than the traditional *TP* algorithms for the synthetic datasets generated by the public *IBM* data generator.

## REFERENCES

[1] R. Agrawal, T. Imielinksi, and A. Swami (1993), Mining association rules between sets of items in large database. The ACM SIGMOD International Conference on Management of Data, pp. 207-216

[2] R. Chan, Q. Yang, and Y. Shen (2003), Mining high utility itemsets. The 3rd IEEE International Conference on Data Mining, pp. 19-26

[3] C. J. Chu, V. S. Tseng, and T. Liang (2008), Mining temporal rare utility itemsets in large databases using relative utility thresholds. International Journal of Innovative Computing, Information and Control, 4(8):2775-2792

[4] IBM Quest Data Mining Project, "Quest synthetic data generation code," Available at (http://www.almaden.ibm.com/cs/quest/syndata.html)

[5] G. C. Lan, T. P. Hong, and V. S. Tseng (2011), Discovery of high utility itemsets from on-shelf time periods of products. Expert Systems with Application, 38(5):5851-5857

[6] Y. Liu, W. Liao, and A. Choudhary (2005), A fast high utility itemsets mining algorithm. The Utility-Based Data Mining Workshop, pp. 90-99

[7] H. Yao and H. J. Hamilton (2006), Mining itemset utilities from transaction databases. *Data & Knowledge Engineering*, 59(3):603-626

[8] J. S. Yeh, C. Y. Chang, and Y. T. Wang (2008), Efficient algorithms for incremental utility mining. *The International Conference on Ubiquitous Information Management and Communication*, pp. 229-234