Discovery of High Utility Sequential Patterns with Consideration of On-shelf Time Periods of Products

Guo-Cheng Lan[a*], Tzung-Pei Hong[b, c], Jen-Peng Huang[d], and I-Wei Lin[b]

[a]Department of Computer Science and Information Engineering
National Cheng Kung University, 701, Taiwan
[b]Department of Computer Science and Information Engineering
National University of Kaohsiung, 811, Taiwan
[c]Department of Computer Science and Engineering
National Sun Yat-Sen University, 804, Taiwan
[d]Department of Information Management
Southern Taiwan University of Science and Technology, 701, Taiwan
*rrfoheiay@gmail.com, tphong@nuk.edu.tw, jehuang@mail.stust.edu.tw,
yatweilin@gmail.com

Abstract

This work presents a new research issue named high on-shelf utility sequential pattern mining, which considers not only quantities and profits of items in sequences but also the common on-shelf time periods of the items. Since items in stores are not always on shelf, it may have a bias for the actual utilities of subsequences in a sequence database with multiple time periods. In addition, we propose a two-phase mining algorithm to effectively and efficiently handle this problem. At last, the experimental results also show the proposed approach has good performance.

1 Introduction

The main purpose of data mining is to extract various useful rules [1] or patterns [2] from data sets. Sequential pattern mining [2], which considered not only of the frequency relationship of items in the pattern but also the order relationship of the items according to the time stamps of the items, was proposed to find trend analysis or regularity patterns from a set of data with time. However, the patterns found by traditional sequential pattern mining techniques do not reflect any other factors, such as cost, price or profit. For example, assume there exists one pattern <(Diamond), (Necklace)> in a sequence database, and assume it is not a high-frequency pattern when compared with another pattern <(computers), (Printers>. But, it may contribute a large portion to the overall profit of the shop due to its high profit. Accordingly, some products with high profit but low frequency may not be discovered in sequences databases by using the traditional sequential pattern mining approaches.

To address this, Ahmed *et al*. [3] proposed a new research issue, namely utility sequential pattern mining, which considered not only quantities and time-stamps of items in a sequence but also individual profits of items in a quantitative sequence database. In addition, there existed three kinds of conditions for calculating the utility of a pattern in a sequence. Afterward, Lan *et al*. presented a new maximum utility function to simplify the calculation for the utility of a subsequence in a sequence [6]. The main principle of the maximum utility function [6] is derived from the concept of sequential pattern mining that the count of any subsequence in a sequence must be the value of 1 to find most customers' buying behaviors.

However, in reality, a sequence database may consist of the sequences of multiple different user-specified time periods. For example, a sequence database in a shop can be composed of sequences of four seasons of a year when the user-specified time period is set at "season" unit. However, some products in shops are not always on the shelf for sale, such as new products, and also the products may be put on the shelf and taken off multiple times, such as seasonal products. Hence, some biases may exist in the discovered patterns.

Based on the above reasons, this work presents a new research issue named on-shelf-based high utility sequential pattern mining, for discovering high utility sequential patterns with the consideration of the on-shelf timer periods of items in a quantitative sequence database. To our best knowledge, this is the first work on mining high utility sequential patterns with considering both of the on-shelf time periods in the field of utility mining. In addition, the two-phase approach for mining high on-shelf utility sequential patterns (abbreviated as *TP-HOUS*) is proposed to effectively handle the problem. At last, the experimental results show the proposed *TP-HOUS* has better performance under various parameters.

The remaining parts of this paper are organized as follows. Some related works are described in Section 2, and the problem to be solved and its related definitions are illustrated in Section 3. The proposed *TP-HOUS* is stated in Section 4. The experimental evaluation is showed in Section 5. Conclusions are finally given in Section 6.

2 Related Works

Different from association-rule mining, Yao *et al*. presented a new research issue named utility mining [10], which considered not only the quantities of items but also their profits in a set of transactions. By using these two kinds of information, the actual utility of an item in a transaction database can be more accurately recognized. Afterward, to effectively reduce the search space of candidates in mining, Liu *et al.* then proposed a two-phase mining algorithm for discovering high-utility itemsets from a database by adopting a new downward-closure property [8] called the transaction-weighted utilization (*TWU*) model. Several studies have extended utility mining to various practical problems, such as utility mining in stream environments [9], incremental process of utility mining [7], etc.

However, these studies did not consider the on-shelf time periods of products in stores since a product might be put on a shelf and taken off it multiple times. Hence, Lan *et al*. then proposed a new issue named on-shelf utility mining, which considered the on-shelf periods of items [5] in addition to the quantities and profits of items. In addition, they proposed a two-phase approach to find high-on-shelf-utility itemsets from a temporal database [5].

Although utility mining can be used to find high utility product combinations (or itemsets) from a set of transactions, its relevant techniques cannot be applied to discover regularity knowledge with high-profit in a time-series data (called sequence data). To address this problem, Ahmed *et al*. then proposed a new issue, namely high utility sequential pattern mining, which was extended from the principle of utility mining [8]. Different from utility mining [8][10], their study [3] not only considered quantities and profits of items in sequences but also the order relationship of items. In addition, they designed three utility calculation ways to identify the utility of a pattern in a sequence under corresponding conditions of the three ways.

Afterward, to simplify the utility calculation of subsequences in sequences, Lan *et al*. proposed a simple utility measure [6], which was called maximum utility measure, to achieve this goal. The main principle of the maximum utility measure was derived from the concepts of traditional sequential pattern mining [2] and profit maximization. Besides, the effective sequence utility upper-bound (abbreviated as *SUUB*) model [6] was designed to keep the property, and thus a mining approach (called *PHS*) was proposed for the required patterns in sequence databases.

As mentioned previously, however, a quantitative sequence database may consist of quantitative sequences of multiple user-specified time periods, such as the sets of sequence data in different seasons, and also some products are not always on the shelf for sale throughout the whole time periods. To effectively obtain more accurate utilities of subsequences in databases, it is then necessary for considering the individual on-shelf time periods of products instead of the whole time period.

Based on the above reasons, this motivates our attempt to find high on-shelf utility sequential patterns from quantitative sequence databases.

3 Problem Statement and Definitions

To clearly explain the problem of on-shelf utility sequential pattern mining, consider nine sequences, each of which consists of three features, namely sequence identification (*SID*), sequence corresponding time period, and items purchased. As an example, the data after the transformation shown in Table 1 are used for mining, where the numbers represent the purchased quantities. In addition, there are six items, respectively denoted as *A* to *F*, and their profits are 3, 10, 1, 6, 5 and 2. The on-shelf time period information of each item is then showed in Table 2.

By using the three tables, a set of terms related to on-shelf utility sequential pattern mining can be defined as follows.

Definition 1. $T = \{t_1, t_2, ..., t_j, ..., t_n\}$ is a set of mutually disjoint time periods, where t_j denotes the *j*-th time period in the whole set of periods, *T*.

Definition 2. An itemset *X* is a set of items, $X \subseteq I$. If $|X| = k$, the itemset *X* is called a *k*-itemset. $I = \{i_1,$

i_2, ..., i_m} is a set of items, which may appear in sequences.

Table 1: Nine sequences for this example

Period	***SID***	***Quantitative Sequences***
t_1	Seq_1	<a(1), a(2), e(1), f(2)>
	Seq_2	<a(1), {b(1), c(15)}, c(3)>
	Seq_3	<e(3), c(2), e(1)>
t_2	Seq_4	<{b(2), c(9)}, c(5)>
	Seq_5	<a(3), {a(2), c(8)}, d(2)>
	Seq_6	<b(1), c(4), d(3)>
t_3	Seq_7	<{a(3), d(2)}, c(2), e(2), f(3)>
	Seq_8	<a(3), f(1)>
	Seq_9	<a(1), a(2), e(1), f(2)>

Table 2: On-shelf time periods of the items

Period / ***Item***	t_1	t_2	t_3
a	1	1	1
b	1	1	0
c	1	1	1
d	0	1	1
e	1	0	1
f	1	0	1

Definition 3. Seq = <X_1, X_2, ..., X_m> is a list of itemsets sorted in ascending order according to their occurrence time. Each itemset appears in a sequence Seq is called an element of the sequence. For brevity, the brackets are omitted if an itemset only includes an item.

Definition 4. A temporal quantitative sequence database $TQSD$ is composed of sequence sets of several time periods. That is, $TQSD$ = {$Seq_{1.1}$, $Seq_{1.2}$, …, $Seq_{j.y}$, …, $Seq_{n.m}$}, where $Seq_{j.y}$ is the y-th sequence in the j-th time period. Moreover, all sequences in the j-th time period t_j can be identified as $TQSD_j$. For example, in Table 1, $TQSD_1$ includes Seq_1, Seq_2, and Seq_3 in t_1.

Definition 5. The local sequence utility lsu_{yj} of an item i_{yj} in a sequence Seq_y is the quantity of the j-th item i_{yj} in Seq_y. For example, in Table 1, there are two elements a and f in Seq_8, and their local sequence utility values are 3 and 1, respectively.

Definition 6. The external utility s_i of an item i is the corresponding utility value of the item in the utility table. In this example, s_a = 3.

Definition 7. The utility u_{yj} of an item i_j in a sequence Seq_y is the external utility s_i of the item i_j in the utility table multiplied by the local sequence utility lsu_{yj} of the item i_j in the sequence Seq_y, and can be defined as $u_{yj} = s_i * lsu_{yj}$. Note that the maximum value among the utility values of the item in the sequence is regarded as the utility of the item in that sequence by the proposed maximum utility measure.

For example, in Table 1, the item a appears two times in Seq_1, and the utility values of the item a in the two times can be calculated as 3 (= 3* 1) and 6 (= 3*2), respectively. Since the maximum utility value is 6, the value of 6 is regarded as the utility of the item a in Seq_1.

Definition 8. The utility $u_{y,S}$ of a subsequence S in a sequence is the maximum value among utility values of all combinations in the sequence.

For example, in Table 1, there are two combinations of <ac> in Seq_2, and the two utility values are 18 and 6, respectively. The value of 18 is then regarded as the utility of <ac> in Seq_2 by the maximum measure.

Definition 9. The sequence utility su_y of a sequence Seq_y is the sum of the utility values of all items contained in Seq_y. That is,

$$su_y = \sum_{i_j \in Seq_y} u_{yj} .$$

Definition 10. The periodical utility $pu(S, t_j)$ of a subsequence S is the sum of the actual utility values of S in all sequences including S within the j-th period t_j. That is:

$$pu(X, t_j) = \sum_{Seq_{j,y} \subseteq SQD_j \wedge S \subseteq Seq_{j,y}} u(S, Seq_{j,y}) .$$

For example, in Table 1, pu(<ab>, t_1) = u(<ab>, $Seq_{1.2}$) = (3*1 + 10*1) = 13.

Definition 11. The periodical total sequence utility $ptsu(t_j)$ is the sum of the sequence utilities of all sequences within the j-th time period t_j. That is:

$$ptsu(t_j) = \sum_{Seq_{j,y} \subseteq D_j} su(Seq_{j,y}) .$$

For example, $ptsu(t_1) = su(Seq_{1.1}) + su(Seq_{1.2}) + su(Seq_{1.3})$ = 18 + 31 + 22 = 71.

Definition 12. The periodical utility ratio $pur(S, t_j)$ of a subsequence S is the periodical utility $psu(S, t_j)$ of X within the j-th period t_j over the periodical total utility $ptsu(t_j)$ of the time period t_j. That is:

$$pur(X, t_j) = pu(X, t_j) \big/ ptsu(t_j) .$$

For example, pur(<ab>, t_1) = pu(<ab>, t_1)/$ptsu(t_1)$ = 13/71 = 18.31%.

Definition 13. Let λ be a pre-defined minimum on-shelf utility threshold. A subsequence S is called a high-periodical-utility subsequence ($HPUS$) if $pur(S) \geqq \lambda$. For example, if λ is set at 25%, then <bc> is not a $HPUS$.

Here, discovery of $HPUSs$ in a time period is the same as that of high utility sequential pattern mining [3][6] in a quantitative sequence database. The above definitions are further extended to on-shelf utility sequential pattern mining.

Definition 14. The on-shelf time periods of an item i, $op(i)$, are for sale on the shelf within a set of time periods. For example, in Table 3, the on-shelf time periods of item b, $op(b)$, are t_1 and t_2.

Definition 15. The common on-shelf time periods of all items in a subsequence *S*, *op(S)*, are for sale on the shelf within a set of time periods. For example, in Table 3, the common on-shelf time periods of <*bc*>, *op(*<*bc*>*)*, include t_1 and t_2.

Definition 16. The on-shelf utility of an item *i*, *ou(i)*, is the sum of utilities of *i* within the union of on-shelf time periods of *i*. For example, in Table 1, *ou(*<*b*>*)* = *pu*(<*b*>, t_1) + *pu*(<*b*>, t_2) = (10*3 + 10*1) + (10*2 + 10*1) = 10 + 30 = 40.

Definition 17. The on-shelf utility of a subsequence *S*, *opu(S)*, is the sum of utilities of *S* within the union of on-shelf periods of *S*. For example, in Table 1, *ou(*<*bc*>*)* = *pu*(<*bc*>, t_1) + *pu*(<*bc*>, t_2) = (25) + (34+14) = 73.

Definition 18. The on-shelf utility ratio of a subsequence *S*, *our(S)*, *is* the sum of utilities of *S* within all on-shelf time periods of *S* over the sum of all the sequence utilities within the union of the on-shelf time periods of *S*. For example, in the three tables, *our(*<*bc*>*)* = *ou(*<*bc*>*)*/(*ptsu*(t_1) + *ptsu*(t_2)) = (73) /(71 + 89) = 25.3%.

Definition 19. A subsequence *S* is called a high on-shelf utility sequential pattern (*HOUS*) if *our*(*S*) $\geqq \lambda$. For example, if λ is set at 25%, <*bc*> is a *HOUS*.

The downward-closure property in high on-shelf utility sequential pattern mining cannot be maintained. For example in Table 1, if λ = 25%, then *our*(<*c*>) = 21.05%<λ; <*c*> is thus a low on-shelf utility subsequence. However, its superset <*bc*> is a *HOUS*. To handle this, the concept of the *SUUB* model [6] is extended to avoid information losing in mining. That is, if a subsequence is a *HPUS* within at least one of its on-shelf time periods, it is possible to a *HOUS*; otherwise, it is impossible to be a *HOUS*. The relevant terms are defined as follows.

Definition 20. The periodical utility upper-bound *puu*(*S*, *t*) of a subsequence *S* within one time period *t* is the sum of the sequence utility values of all sequences including *S* within *t*. That is:

$$puur(S,t) = \sum_{Seq_{j,y} \subseteq QSD_j \wedge S \subseteq Seq_{j,y}} su(Seq_{j,y}).$$

For example, *puu*(<*c*>, t_1) = *su*($Seq_{1.1}$) + *su*($Seq_{1.2}$) = (31 + 22) = 53.

Definition 21. The periodical utility upper-bound ratio *puur*(*S*, *t*) of a subsequence *S* is the sum of the sequence utility values of all sequences including *S* over the sum of all the sequence utility values within one time period *t*. That is:

$$puur(S,t) = puu(S,t) / pttu(t).$$

For example, *puur*(<*c*>, t_1) = 53 / 71 = 74.65%.

Definition 22. A subsequence *S* is called a promising on-shelf utility subsequence (*POUS*), if *S* is a *HPUS* within at least one of its on-shelf time periods. For example, <*c*> is a *POUS* since it is a *HPUS* within its two on-shelf periods, t_1 and t_2.

Problem Statement: Based on the above definitions, the problem to be solved is to find the subsequences whose actual on-shelf utility values within the union of their on-shelf time periods are larger than or equal to a predefined minimum on-shelf utility threshold. To solve this problem, this work presents a two-phase mining algorithm for finding *HOUSs* in temporal quantitative sequence databases *TQSD*.

4 Proposed Approach

In this section, the details of the proposed two-phase approach for finding high on-shelf utility sequential patterns (*TP-HOUS*) based on Lan *et al.*'s upper-bound model are then stated as follows.

INPUT: A temporal quantitative sequence database *TQSD* with *n* sequences, each of which consists of sequence identification, sequence occurring time period and items purchased, *m* items in *TQSD*, each with a profit value, an on-shelf table of items with *k* desired time periods, and the minimum utility threshold λ.

OUTPUT: The set of high on-shelf utility sequential patterns (*HOUSs*).

Phase 1: Discovery of *POUSs*

STEP 1. Initialize the PTSU (Periodical Total Sequence Utility) table as a zero table, in which the row number is the time period number and each entry in the table is set as 0.

STEP 2. For each *y*-th transaction Seq_{jy} in the set of sequences $TQSD_j$ in each time period t_j, do the following substeps.

(a) Calculate the utility value u_{jyz} of each *z*-th item I_{jyz} in Seq_{jy}.

(b) Calculate the sequence utility su_{jy} of the sequence Seq_{jy}.

STEP 3. Calculate the periodical total sequence utility $ptsu_j$ in each time period t_j.

STEP 4. Find in the PTSU table the entry which has the same occurring time period as each t_j, and add $ptsu_j$ to the entry.

STEP 5. Find the subsequences with high utility values in each time period t_j by the *Finding-Individual-HUS* procedure. Let the set of returned high utility subsequences as HUS_j and the union of all HUS_j's for all time periods as *POUS.*

Phase 2: Discovery of *HOUSs*

STEP 6. Initially set the set of high on-shelf utility sequential patterns *HOUS* as empty.

STEP 7. For each subsequence *S* existing in *POUS*, find its actual on-shelf utility by the following substeps:

(a) Use the AND operation to obtain the set of common on-shelf periods (COS_S) of all the items in *S* from the *OS* table.

(b) Sum the actual utilities $u(S)^{Appearing}$ of *S* appearing in *POUS* as follows:

$$u(S)^{Appearing} = \sum_{S \in HUU \wedge t_j \in COS_S} u_j(S),$$

where $u_j(S)$ is the actual utility of subsequence *S* in the time period t_j.

STEP 8. For each subsequence *S* in *POUS*, scan the database to find its actual utility $u(S)_j^{scan}$ in a time period $t_j \in COS_S$ if *S* does not appear in that time period.

STEP 9. Calculate the actual on-shelf utility $u(S)^{actual}$ of each subsequence *S* in *POUS* as:

$$u(S)^{actual} = u(S)^{Appearing} + \sum_{S \in HUS \wedge S \notin HUS_j \wedge t_j \in COS_S} u(S)_j^{scan}.$$

STEP 10. If $u(S)^{Actual} / ptsu(S) \geqq \lambda$, then *X* is a high on-shelf utility subsequence; set $HOUS = HOUS \cup S$ and $POUS = POUS - X$; Otherwise, set $POUS = POUS - S$.

STEP 11. Output the set of high on-shelf utility sequential pattern in *HOUS*.

After STEP 11, all the *HOUSs* are found. The *Finding-Individual-HUS* procedure used in STEP 5 is described below. Based on Lan *et al*.'s sequence-utility upper-bound (*SUUB*) model, the *HUSs* in each time period can be found. The procedure is then stated below.

The Finding-Individual-HUS procedure:

Input: The set of temporal quantitative sequences $TQSD_j$ in a time period t_j.

Output: The high utility subsequences in t_j, HUS_j.

PSTEP 1: Set $r = 1$, where *r* represents the number of items in the current set of candidate sequence-utility upper-bound subsequences (C_{jr}) to be processed.

PSTEP 2: Initially set C_{jr} as the set of all the items in the time period t_j.

PSTEP 3: For each *r*-subsequence S_{jz}, do the following substeps:

(a) Calculate the sequence-utility upper-bound $suub_{Sjz}$ of S_{jz} in the time period t_j as the sum of the sequence utilities of S_{jz} in all the sequences in t_j.

(b) Calculate the actual utility u_{jz} of S_{jz} in t_j.

PSTEP 4: Check whether the sequence-utility upper-bound $suub_{Sjz}$ of each candidate *r*-subsequence S_{jz} in t_j exceeds or equals to the threshold of $\lambda * ptsu_j$ within t_j. If it is, put it into the set of high sequence-utility upper-bound *r*-subsequences $HSUUB_{jr}$ for t_j. That is:

$$HSUUB_{jr} = \{ S_{jz} \mid suub_{Sjz} \geq \lambda * ptsu_j \text{ for the time period } t_j \}.$$

PSTEP 5: Check whether the actual utility u_{jz} of each candidate *r*-subsequence S_{jz} in t_j exceeds or equals to the threshold of $\lambda * ptsu_j$ within t_j. If it is, put it with its actual utility u_{jz} into the set of HUS_{jr}. That is:

$$HUS_{jr} = \{ S_{jz} \mid u_{Sjz} \geq \lambda * ptsu_j \text{ for the time period } t_j \}.$$

PSTEP 6: Generate the candidate set $C_{j(r+1)}$ from $HSUUB_{jr}$ in the current time period t_j. The *r*-subsequences in each candidate in $C_{j(r+1)}$ must exist in $HSUUB_{jr}$.

PSTEP 7: If $HSUUB_r$ is not null, set $r = r + 1$ and repeat PSTEPs 1 to 8; otherwise, return the set of high utility subsequences HUS_j with their actual utility values.

5 Experimental Evaluation

In the experiments, the public *IBM* data generator was used in our experiments to produce the sequences data [4]. To fit the problem of high on-shelf utility sequential mining, we also developed a simulation model, which was similar to that used in Lan *et al*.'s study [6], to generate the quantities and profits of the items in the sequences. In addition, an integer value from 1 to the total number of given time periods was randomly assigned to each transaction in the generated datasets [5].

Experiments were then made to evaluate the efficiency of the proposed *TP-HOUS*, and Figure 1 showed execution time of the *TP-HOUS* algorithm for the datasets with the varied minimum on-shelf utility threshold (*min_outil*) under the three time periods, *P5*, *P10*, and *P50*.

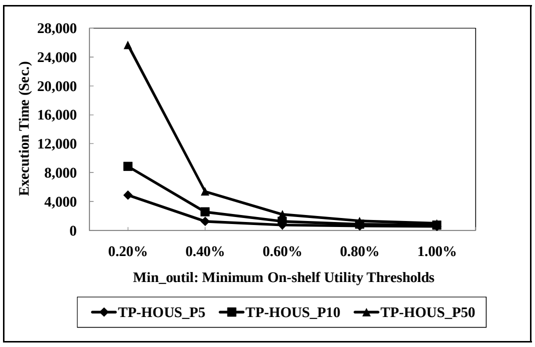

Figure 1. Efficiency of the proposed *TP-HOUS* algorithm for various thresholds.

The figure shows that the execution efficiency of the proposed *TP-HOUS* algorithm has good performance in handling the problem of on-shelf utility sequential pattern mining.

6 Conclusion

This work has presented a new research issue named high on-shelf utility sequential pattern mining, which consider the common on-shelf time periods for items. In addition, we have proposed an effective two-phase mining approach (*TP-HOUS*) to effectively find the desired patterns from quantitative sequence databases. The experimental results show that the proposed *TP-HOUS* has good performance.

References

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

[2] R. Agrawal and R. Srikant, “Mining sequential patterns,” The 11th International Conference Data Engineering, pp. 3-14, 1995.

[3] C. F. Ahmed, S. K. Tanbeer, and B. S. Jeong, “A novel approach for mining high-utility sequential patterns in sequence databases,” *ETRI Journal*, Vol. 32, No. 5, pp. 676-686, 2010.

[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 Vincent S. Tseng, “Discovery of high utility itemsets from on-shelf time periods of products,” *Expert Systems with Applications*, Vol. 38, No. 5, pp. 5851-5857, 2011.

[6] G. C. Lan, T. P. Hong, and Vincent S. Tseng, “Applying the Maximum Utility Measure in High Utility Sequential Pattern Mining,” accepted and to appear in *Expert Systems with Applications*, , 2014.

[7] C. W. Lin, G. C. Lan, and T. P. Hong, “An incremental mining algorithm for high utility itemsets,” *Expert Systems with Applications*, Vol. 39, No. 8, pp. 7173-7180, 2012

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

[9] V. S. Tseng, C. J. Chu, and T. Liang, “Efficient mining of temporal high utility itemsets from data streams,” *The ACM KDD Workshop on Utility-Based Data Mining*, pp. 1105-1117, 2006.

[10] H. Yao, H.J. Hamilton, and C.J. Butz, “A foundational approach to mining itemset utilities from databases,” The Third SIAM International Conference on Data Mining, pp. 482-486, 2004.

[11] U. Yun, John J. Leggett, and T. J. Ong “Mining weighted sequential patterns based on length-decreasing support constraints,” The International Conference on Intelligence and Security Informatics, pp. 650-651, 2006.