Temporal Fuzzy Utility Mining with Multiple Minimum Utility Thresholds

Wei-Ming Huang
National Sun Yat-sen University
Kaohsiung 804, Taiwan
granthill168@gmail.com

Tzung-Pei Hong*
National University of Kaohsiung
Kaohsiung 811, Taiwan
tphong@nuk.edu.tw

Jerry Chun-Wei Lin
Harbin Institute of Technology
Shenzhen Graduate School
Shenzhen 518055, China
jerrylin@ieee.org

Ming-Chao Chiang
National Sun Yat-sen University
Kaohsiung 804, Taiwan
mcchiang@cse.nsysu.edu.tw

ABSTRACT

Although recent work on fuzzy utility mining has taken into account transaction times, using a single minimum utility threshold for all items does not adequately reflect the true properties of items. We propose a multiple minimum-utility approach which considers the different properties of items for temporal fuzzy utility mining to assist users in determining the proper standards for an itemset in mining when its items have different standards.

KEYWORDS

Data mining, fuzzy set, temporal fuzzy utility mining, multiple thresholds

*Corresponding author: Tzung Pei Hong

1 INTRODUCTION

One fundamental research area in data mining is using association rules to mine frequent itemsets, as this can be used to identify implicit relationships among items in a dataset [1][2][3]. One drawback, however, is that we cannot take for granted that the quantity of an item is not binary, and we cannot consider the profit of an item as an essential factor. As such, the mining of frequent itemsets is easy, in contrast to that of high-profit itemsets with quantity information. Thus utility mining [13] has been proposed, which considers both the quantities sold of items and their prices (profits) in transactions, recognizing the actual utility values of itemsets and thus discovering high-utility itemsets in a transaction database. However, as quantitative values are still not easily understood by decision makers, fuzzy-utility mining was proposed [21]. By using fuzzy concepts to convert quantitative values in transactions into fuzzy regions, with the minimum operator for the intersection of fuzzy sets, users easily comprehend the implications of quantity.

Most work uses only a minimum threshold for data-mining tasks. However, a minimum threshold may not accurately reflect the property of items in real-life applications [4][5][6][7][8][10], such as the difference in significance between emerging items and aged ones. For this problem, each item can be assigned an individual minimum threshold. In this paper, we will consider the temporal fuzzy utility mining with multiple minimum utility thresholds. That is, the minimum utility threshold itself should also depend on the item. We present an effective framework with multiple minimum utility thresholds to find interesting highly temporal fuzzy-utility itemsets in a quantitative database.

2 RELATED WORKS

Agrawal et al. [1][2] proposed Apriori, a well-known method which finds frequent itemsets, although it cannot discover high-profit itemsets. Utility mining, however, can achieve this if a transaction includes the item's quantity and price. However, most such work in the last two decades has been on new utility-mining methods, including how to decrease memory consumption or devise better utility-mining algorithms. Transactional databases record quantitative item values and their profit (prices) along, making it easy to discover high-utility itemsets. However, the high utility itemsets that are discovered can be difficult for users to comprehend. The transaction "six oranges sold" in the supermarket can represent an actual quantity sold – they do not include the semantic meaning needed by decision makers. Indeed, fuzzy quantifiers can express the implicit meaning of actual facts [11][12]. Thus, methods for fuzzy utility mining were extended to capture the implied meaning of the quantity sold. For instance, Lan et al. [38] present a framework to discover high fuzzy-utility itemsets by using fuzzy concepts to convert the quantity value of the item in a quantitative transactional dataset into fuzzy regions. Wang et al. [9] define fuzzy utility mining as the problem of identifying practical itemsets while taking into account both whole (i.e., 3 apples) and fractional (i.e., 2.1 pounds) quantities.

The methods mentioned above are based on the assumption that the timestamp is not taken into account while discussing high-utility or fuzzy-utility itemsets. Because of this, short-sale or time-limited items in the supermarket may be obscured in mining. To overcome the limitation, Huang et al. [14] in 2017 introduced fuzzy utility mining that takes into consideration transactional time. Also, we note that sometimes – such as during summer or winter vacations and so on – supermarkets use different sales strategies, promoting unfixed products. However, if all products on the shelves are viewed as being of the same time, it is not possible to effectively explore the various combinations or products.

The use of a single standard (minimum utility threshold) in mining is applied in [9][14]. Nevertheless, existing approaches employ a single threshold for all items as a measuring standard by which to mine high actual-value itemsets. However, considering all items to be of the same importance does not reflect real-world practice. Thus, threshold requirement should depend on the item. For example, people will consider the utility of emerging items and aged ones in different criteria. An emerging item may not currently get much profit, but it may have a high potential to earn much money in the future. Hence, using a minimum-utility threshold for all items in utility mining or fuzzy utility mining is not a fair measurement. Although taking into account multiple item thresholds better reflects real-world needs, it is also more challenging.

3 PRELIMINARIES AND PROBLEM STATEMENT

Let $P = \{p_1, p_2, ..., p_m\}$, where p_i denotes the i-th time period and m is the number of time periods, be a set of continuous disjoint time periods with a designated time granularity. A set I contains n distinct items $\{i_1, i_2, ..., i_n\}$. A quantitative transaction database is a set of transactions $QTD = \{t_1, t_2, ..., t_z\}$, in which each transaction $t_y \in QTD$ $(1 \leq y \leq z)$ is a subset of I and has a unique identifier TID. In addition, each item i_g in each transaction t_y possesses a quantitative value sold of the item i_g (denoted as v_{yg}). Based on the membership functions of item i_g, the fuzzy set f_{yg} of the value v_{yg} of i_g in the quantitative transaction t_y is represented as $f_{yg} = \left(\frac{f_{yg1}}{R_{g1}} + \frac{f_{yg2}}{R_{g2}} + \cdots + \frac{f_{ygl}}{R_{gl}} + \cdots + \frac{f_{ygh}}{R_{gh}}\right)$, where h is the number of membership functions for i_g, R_{gl} is the l-th linguistic term of i_g, and f_{ygl} is the fuzzy membership value of v_{yg} in R_{gl}. An utility table $= \{S_1, S_2, ..., S_n\}$ denotes profit value S_g of each item i_g. An itemset X with k distinct items $\{i_1, i_2, ..., i_k\}$ such that $X \subseteq I$ is called a k-itemset, where k is the length of the itemset. Itemset X is said to be contained in a transaction t_y if $X \subseteq t_y$.

Table 1: Sample dataset

Period	TID	*A*	*B*	*C*	*D*	*E*	*F*	*G*
P1	*t1*	2	0	2	3	5	1	0
	t2	0	1	25	0	0	0	0
	t3	0	0	0	0	10	1	0
	t4	0	1	12	0	0	0	0
P2	*t5*	4	0	8	0	10	0	0
	t6	0	0	4	3	0	1	0
	t7	0	0	2	3	0	0	0
P3	*t8*	6	2	0	0	10	3	0
	t9	4	0	0	4	0	0	3
	t10	0	0	4	0	10	0	0

To illustrate the proposed method, the sample dataset is shown in Table 1. Table 2 represents the utility table, which defines the profit value of each item. Table 3 is the minimum utility threshold table, which defines the minimum threshold value of each item. The membership functions for each item are shown in Fig. 1.

Table 2: Utility table

Item	*Profit*
A	3
B	10
C	1
D	6
E	5
F	2
G	20

Table 3: Minimum utility threshold of each item

Item	*Minimum threshold*
A	0.25
B	0.4
C	0.3
D	0.5
E	0.25
F	0.25
G	0.3

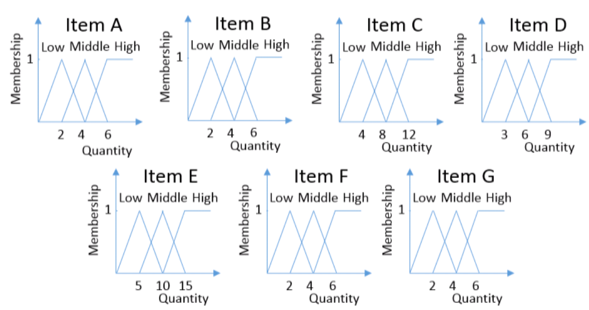

Figure 1: Membership function of each item

To explain temporal fuzzy utility mining, we offer the following definitions [14]:

Definition 1: Utility value fu_{ygl} is defined as $fu_{ygl} = f_{ygl} * v_{yg} * S_g$. Value f_{ygl} indicates that it is the l-th fuzzy region for an item i_g in t_y, the quantity sold v_{yg} is for item i_g in t_y, and the utility S_g is for item i_g.

Using item C in t_1 as an example, the quantity and the profit of item C in t_1 are 2 and 1 (Tables 1 and 2). From Fig. 1, its quantity membership in $R_{C.Low}$ is 0.5, $R_{C.Middle}$ is 0, and $R_{C.High}$ is 0. Thus,

$fu_{1,C.Low}$ = 0.5 * 2 * 1 = 1, $fu_{1,C.Middle}$ = 0 * 2 * 1 = 0 and $fu_{1,C.High}$ = 0 * 2 * 1 = 0. The results are shown in Table 4.

Table 4: Fuzzy region utilities for example

TID	A			B			C			D			E			F			G		
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
	fu value																				
t1	6	0	0	0	0	0	1	0	0	18	0	0	25	0	0	1	0	0	18	0	0
t2	0	0	0	5	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	1	0	0	0	0	0
t4	0	0	0	5	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	12	0	0	0	0	0	8	0	0	0	0	0	50	0	0	0	0	0	0	0
t6	0	0	0	0	0	0	4	0	0	18	0	0	0	0	0	1	0	0	18	0	0
t7	0	0	0	0	0	0	1	0	0	18	0	0	0	0	0	0	0	0	18	0	0
t8	0	0	12	20	0	0	0	0	0	0	0	0	0	50	0	3	3	0	0	0	0
t9	0	12	0	0	0	0	0	0	0	15.84	7.92	0	0	0	0	0	0	0	15.84	7.92	0
t10	0	0	0	0	0	0	4	0	0	0	0	0	0	50	0	0	0	0	0	0	0

Definition 2: The total utility of t_y is represented as tfu_y. That is, $tfu_y = \sum_{i_g \subseteq t_y \wedge t_y \subseteq QTD} fu_{yz}$. Take for example t_8 in Table 4: $tfu_8 = fu_{8,\{A.High\}} + fu_{8,\{B.Low\}} + fu_{8,\{E.Middle\}}$ = 12 + 20 + 50 = 82.

Definition 3: The utility of a fuzzy itemset X in t_y is represented as fu_{yX}. Thus, $fu_{yX} = f_{yX} * \sum_{R_{gl} \subseteq X} v_{yg} * S_g$, where f_{yX} is the membership value of X in t_y and can be calculated as $Min_{R_{gl \in X}} f_{ygl}$; v_{yg} is the quantity sold of i_g in t_y; and S_g is the profit value of i_g.

For instance, we calculate the fuzzy region values of *A.Middle* and *D.Low* in t_9 as 1 and 0.66 by applying the membership functions given in Fig. 1. Therefore, the common membership value $f_{y,\{A.Middle,\ D.low\}}$ is *Min*(1, 0.66) = 0.66 according to the minimum operator as the intersection. The utility $fu_{y,\{A.Middle,\ D.low\}}$ is then calculated as 0.66*((4*3) + (4*6)), which is 23.76.

Definition 4: The first-started transactional period of an item i_g in the given quantitative transaction database *QTD* is denoted as $st(i_g)$. For instance, the *st* values of the two items *D* and *G* in Table 1. are P_1 and P_3, respectively.

Definition 5: Itemset *X* may include one or more items. If $|X| \geq 2$, the interval of transaction periods of the itemset, denoted LTP_X, is defined from the common transaction period during which each subitem of the itemset at the same time is purchased to the end. If $|X| = 1$, the itemset only includes one item, and LTP_X is represented as from the first occurring transaction period of the item to the end.

Note the *LTP* of the itemset {*D*} and the itemset {*DG*} in Table 1. The former is denoted as from P_1 to the end, which is P_3, and the latter is denoted as from P_3 to P_3.

Definition 6: Denoted as mfu_{yg}, the maximal fuzzy utility of item i_g in t_y is represented as the highest utility value in the *l*-th fuzzy region R_{gl} for the item.

For instance, the three fuzzy values *Low*, *Middle*, and *High* for item *D* in t_9 are 15.84, 7.92, and 0, respectively. Value $mfu_{9,\{D\}}$ of item {*D*} in t_9 is represented as 15.84.

Definition 7: The maximal fuzzy utility of a transaction t_y (denoted $mtfu_y$) is computed as $mtfu_y = \sum_{i_g \subseteq t_y} mfu_{yg}$, where mfu_{yg} is the maximal fuzzy utility value of item i_g in t_y.

For example, in t_9 from Table 4, $mtfu_9 = mfu_{9,\{A\}} + mfu_{9,\{D\}} + mfu_{9,\{G\}}$ = 12 + 15.84 + 30 = 57.84. The same process is used for the other transactions in Table 4.

Definition 8: The first-started transactional period of all items *I* in *QTD* (denoted *STP(I)*), is considered as the latest first-started transactional period of the items in *I*, that is, $STP(I) = \min\{st(i_g) \mid i_g \in I, I \in QTD\}$.

For example, in Table 1, the occurring transaction period of each item of all the items *I* in *QTD* except item G is P_1; item G starts with P_3. Therefore, *STP(I)* is P_3.

Definition 9: Let *LTP(I)* represent a duration of time periods from *STP(I)* to the last time period of *QTD*. The temporal fuzzy utility upper-bound ratio of the fuzzy region of an itemset *X* (denoted $tfuubr_X$), is computed as $tfuubr_X = \sum_{X \in t_y \cap t_y \in LTP_x} mtfu_{yx} \ / \ \sum_{t_y \in LTP(I)} tfu_y$.

For example, the *tfuubr* of itemset {*B.Low*} in Table 4 is calculated as $tfuubr_{\{B.Low\}} = (mtfu_2 + mtfu_4 + mtfu_8) / (tfu_1 + tfu_2 + tfu_3 + tfu_4 + tfu_5 + tfu_6 + tfu_7 + tfu_8 + tfu_9 + tfu_{10})$ = 138 / 504.76, which is 0.2733.

Definition 10: An itemset X is referred to as a high temporal fuzzy utility upper-bound itemset if $tfuubr_X \geq$ threshold .

For example, if the threshold is set to 0.26, the temporal fuzzy 1-itemset {*B.Low*} in the above example is upper-bound 1-itemset.

As above mentioned, in real-world applications, a single standard may not suffice to reflect the importance of items. To solve this problem, below we use different thresholds based on each item as the significance of each item.

Definition 11: Let λ_i be the minimum utility threshold of an item i_g. The minimum threshold λ_X of itemset *X*, which includes $|X|$ 1-sub-itemsets (1-item) is selected as $\min\{\lambda_1, \lambda_2, \ldots, \lambda_{|X|}\}$. Hence, the itemset is said to be a high temporal fuzzy utility upper-bound itemset based on the minimum constraint (abbreviated as $HTFUUBMC_X$) if $tfuubr_X \geq \lambda_X$.

Definition 12: Let itemset *X* be a high temporal fuzzy utility upper-bound itemset based on the minimum constraint (abbreviated as $tfurmc_X$). The temporal fuzzy utility ratio based on the minimum-constraint fuzzy region for an itemset *X* in *QTD* is computed as $tfurmc_X = \frac{\sum_{X \in t_y \cap t_y \in LTP_X} fu_{yX}}{\sum_{t_y \in LTP_X} tfu_y}$, where fu_{yX} is the sum of the utility values of all the fuzzy regions on *X* in t_y on $t_y \in LTP_X$ and tfu_y is the sum of the total utility of a t_y on $t_y \in LTP_X$.

If *B.Low* and {*C.Low*, *D.Low*} are both *HTFUUBMC* in Table 4, take for example $tfurmc_{\{B.Low\}}$ and $tfurmc_{\{C, D\}}$ in Table 4 since their $LTP_{\{B.Low\}}$ and $LTP_{\{C.Low, D.Low\}}$ are both from its first occurring time period, P_1, to the last time period of the whole database, P_3: the total of the *tfu* values from the ten transactions is calculated as 504.76 (= 51 + 30 + 51 + 17 + 70 + 23 + 19 + 94 + 95.76 + 54). Additionally, since item *B.Low* appears in t_2, t_4, and t_8, the fuzzy utility value of {*B.Low*} is calculated as 30 (= 5 + 5 + 20). The temporal fuzzy utility ratio $tfur_{\{B.Low\}}$ is thus calculated as 30 / 504.76, which is 0.0594. In addition, since itemset {*C.Low*, *D.Low*} appears in t_1, t_6, and t_7, according to Definition 3, the fuzzy value of {*C.Low*, *D.Low*} is calculated as 42 (= 0.5*(2*1+3*6) + 1*(4*1+3*6) + 1*(2*1+3*6)). The temporal fuzzy utility ratio based on the minimum-constraint *tfurmc* value is thus 0.0832 (= 42 / 504.76).

Definition 13: An itemset *X* is referred to a high temporal fuzzy utility itemset *X* based on minimum constraint (abbreviated as $HTFUI_X$) if $tfurmc_X \geq \lambda_X$.

Problem Statement: The problem to be solved is based on the above mentioned concept: find all itemsets with actual values no less than the multiple thresholds from *QTD* under multiple minimum thresholds.

4 THE PROPOSED ALGORITHM

The proposed method mines high temporal fuzzy utility itemsets under multiple minimum thresholds in two main phases as follows. In the first phase, the promising itemsets with the upper-bound values *tfuubr* are found. This algorithm determines the upper-bound itemsets (*HTFUUBMC*) for which the *tfuubr* values are not less than the corresponding minimum thresholds. Based on the first phase, an additional data scan in the second phase was then performed to find the actual fuzzy-utility value of each promising itemset and found the ones that have actual fuzzy-utility values larger than or equal to their corresponding thresholds, which are derived from the minimum of the individual thresholds of the items contained in the itemsets. Then, the algorithm determines the complete set of high temporal fuzzy utility itemsets (*HTFUIs*) satisfying multiple minimum constraints.

INPUT:

QTD: quantitative transaction database in which each transaction (*t*) includes the quantity of item i_g
i_g: each item in *QTD*
S_g: profit value of item i_g
λ_g: minimum threshold of item i_g

OUTPUT:

A set of *HTFUIs*.

Phase 1: Find *HTFUUBMC*

Step1. Execute each transaction in *QTD*, convert its appearance time into a time period.

Step2. Acquire corresponding $st(i_g)$ information for each item i_g in *QTD* based on first transaction's occurrence time of item i_g.

Step3. Rearrange items in *QTD* according to minimum threshold values of all items in *QTD* in descending order: term this *SQTD.*

Step4. For each item i_g in *SQTD*, convert its value v_{yg} into a fuzzy set f_{yg}, which can be represented as $\frac{f_{yg1}}{R_{g1}} + \frac{f_{yg2}}{R_{g2}} + \cdots + \frac{f_{ygh}}{R_{gh}}$, according to the membership functions for quantities of items, where *h* is the total number of regions for i_g, R_{gl} is the *l*-th linguistic term of i_g, and f_{ygl} is the membership of v_{yg} in R_{gl}.

Step5. Create an empty table for the fuzzy utility of specialized time (*FUS*), in which each tuple has two fields: identification of specialized time (*IST*) and fuzzy utility of specialized time (*FUST*).

Step6. Execute the following substeps for each *y*-th transaction t_{jy} in each time period t_j of *SQTD*:

(a) Determine fu_{jygl} value of *l*-th fuzzy region on item i_g for t_{jy}.
(b) Find mfu_{jyg} value of item i_g in t_{jy} according to (a).
(c) Determine $mtfu_{jy}$ value of a transaction in t_{jy} according to (b).
(d) Sum each fu_{jygl} in t_{jy} to calculate fuzzy utility tfu_{jy} of each t_{jy} according to (a).
(e) Examine whether time period t_j has been in *FUS* table. If yes, insert fuzzy utility value tfu_{jy} of t_{jy} into corresponding fuzzy utility of specialized time t_j in *FUS* table; otherwise, simultaneously put time period t_j and its fuzzy utility tfu_{jy} of t_{jy} in *FUS* table.

Step7. Initialize temporary *1*-itemsets (TI_1) table as empty with two fields: fuzzy itemset and upper-bound value of fuzzy itemset.

Step8. For each t_{jy}, check whether item *i* in t_{jy} has been in TI_1 table. If yes, add only $mtfu_{jy}$ (maximal fuzzy utility of t_{jy}) to its corresponding upper-bound field value to table; if not, simultaneously add item i and its $mtfu_{jy}$ value to table.

Step9. Establish first-started transactional period *STP(I)* of whole item.

Step10. Find *LTP(I)*, which represents a duration of time periods from *STP(I)* to last time period of *QTD*.

Step11. Determine minimum value (denoted as $\lambda_{min,1}$ in first pass) among minimum utility thresholds of all items for *QTD*.

Step12. Execute following substeps for each item i in TI_1:

(a) Derive upper-bound value $tfuubr_i$ of item i by a period of time periods *LTP(I)*.

(b) Check whether value of $tfuubr_i$ is equal to or larger than threshold of $\lambda_{min,1}$. If so, item i is then inserted in $HTFUUBMC_1$; otherwise, item i is omitted.

Step13. Set variable r to 1, where r is used to count the total number of linguistic terms of a fuzzy itemset to be processed.

Step14. Initially set table $TI_{(r+1)}$ of temporary (r+1)-itemsets as empty, with two fields: fuzzy itemset and upper-bound value of fuzzy itemset.

Step15. Use Apriori algorithm to derive candidate (r+1)-itemsets from $HTFUUBMC_r$, and keep these in $TI_{(r+1)}$ table. If no candidates are derived, execute Step 21; else, execute Step 16.

Step16. Carry out following substeps for each candidate (r+1)-itemset X in $TI_{(r+1)}$.

(a) Check if all of its r-sub-itemsets exist in $HTFUUBMC_r$. If not, itemset X is deleted from $TI_{(r+1)}$.

Step17. For each itemset X in $TI_{(r+1)}$, scan all transactions in *SQTD* to determine its required upper-bound value by fuzzy utility of the transactions including X in *SQTD*.

Step18. Determine minimum value (denote as $\lambda_{min,r+1}$ in (r+1)-th pass) among minimum utility thresholds of all distinct items in $TI_{(r+1)}$.

Step19. Execute following substeps for each itemset X in $TI_{(r+1)}$:

(a) Derive $tfuubr_X$ value of X by *LTP(I)*.

(b) Determine whether $tfuubr_X$ value of X satisfies threshold ($\lambda_{min,r+1}$). If so, put X in $HTFUUBMC_{(r+1)}$; otherwise, omit it.

Step20. If $HTFUUBMC_{(r+1)}$ is not null, set r = r +1 and repeat Steps 13 to 20; else, execute next step.

Phase 2: Find *HTFUIs*

Step21. Prepare new, empty table (*HTFUUBMCs*) with three fields: fuzzy itemset, fuzzy utility value of fuzzy itemset, and temporal fuzzy utility ratio of fuzzy itemset. Put in all upper-bound itemsets from each *HTFUUBMC* set, *HTFUUBMCs*.

Step22. Scan transactions in *SQTD* to determine fuzzy utility value of each itemset X in *HTFUUBMCs*.

Step23. Execute following substeps for each itemset X in *HTFUUBMCs*:

(a) Find LTP_X of first-started transactional periods of X according to first-started transactional period information *st* of corresponding items of terms in X.

(b) Among all items in X determine minimum value (denoted as λ_i) as corresponding utility threshold λ_X of X.

(c) Establish total specialized time of fuzzy utility $tstfu_X$ of X using its LTP_X from *FUS* table, and then calculate temporal fuzzy utility ratio $tfurmc_X$ of X using LTP_X of X. Determine whether $tfurmc_X$ of X satisfies threshold (λ_i). If so, keep X in high temporal fuzzy utility itemsets (*HTFUIs*); if not, remove it.

Step24. Output *HTFUIs* to users.

5 EXPERIMENTAL RESULTS

We implemented the proposed algorithm to evaluate its execution efficiency. The operating system was Microsoft Windows 7. All programs were implemented in Java, and run in J2SDK 1.8.0 and executed on a PC with 3.2 GHz CPU and 8GB memory. Experiments were made on the synthetic T10I4N4KD200K and T10I4N4KD300K datasets to evaluate the performance of the proposed approach. The experimental results are described, showing the execution times on the dataset with the minimum threshold λ_{LB} varying from 1% to 5%. The minimum threshold of each item is produced from λ_{LB} to 1. The results are shown in Fig. 2 and Fig. 3.

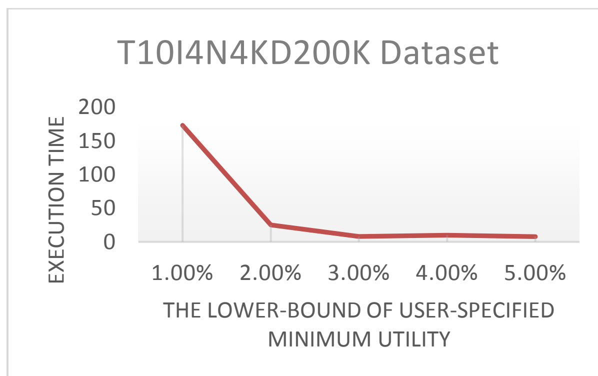

Figure 2. Execution time of the proposed method on the T10I4N4KD200K dataset

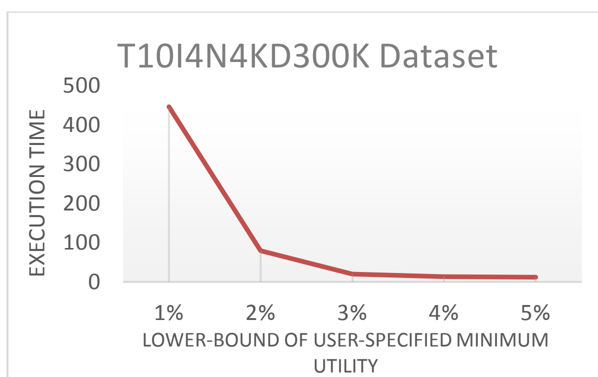

Figure 3. Execution time of the proposed method on the T10I4N4KD300K dataset

6 CONCLUSIONS

We present temporal fuzzy utility mining with multiple minimum constraints, a new function which uses multiple minimum constraints to define the minimum utilities of itemsets in a database when items are better judged using different standards. In addition, the proposed method uses an upper-bound function to maintain downward closure with multiple minimum constraints. The experimental results show that the approach's multiple constraints yield good performance.

REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. 1993. Mining association rules between sets of items in large databases. *ACM SIGMOD International Conference on Management of data*, 207–216.

[2] R. Agrawal and R. Srikant. 1994. Fast algorithms for mining association rules in large databases. *The International Conference on Very Large Data Bases,* 487–499.

[3] F. Berzal, J. C. Cubero, N. Marín, and J. M. Serrano. 2001. TBAR: An efficient method for association rule mining in relational databases. *Data and Knowledge Engineering*, Vol. 37, 47–64.

[4] C. H. Chen, T. P. Hong, V. S. Tseng, and C. S. Lee. 2008. A genetic-fuzzy mining approach for items with multiple minimum supports. *The International Journal of the ACM on Soft Computing*, Vol. 13, 521–533.

[5] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. 1996. Mining optimized association rules for numeric attributes. *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 182–191.

[6] Y. H. Hu and Y. L. Chen. 2006. Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism. *Decision Support Systems*, Vol. 42, no. 1, 1–24.

[7] Y. H. Hu and I. C. Chiang. 2011. Mining cyclic patterns with multiple minimum repetition supports. *The Eighth International Conference on Fuzzy Systems and Knowledge Discovery*, 1545–1549.

[8] H. Jiang, Y. Zhao, C. Yang, and X. Dong. 2008. Mining both positive and negative weighted association rules with multiple minimum supports. *The International Conference of Computer Science and Software Engineering*, 407–410.

[9] C. M. Wang, S. H. Chen, and Y. F. Huang. 2009. A fuzzy approach for mining high utility quantitative itemsets. *IEEE International Conference on Fuzzy Systems*, DOI: 10.1109/FUZZY.2009.5277408.

[10] B. Liu, W. Hsu, and Y. Ma. 1999. Mining association rules with multiple minimum supports. *The International Conference on Knowledge Discovery and Data Mining*, 337–341.

[11] Q. Li. 2013. A novel Likert scale based on fuzzy sets theory. *Expert Systems with Applications*, Vol. 40, no. 3, 1609–1618.

[12] I. Glöckner and A. Knoll. 2001. Fuzzy quantifiers: a natural language technique for data fusion. *The Fourth International Conference on Information Fusion.*

[13] H. Yao, H. J. Hamilton, and C. J. Butz. 2004. A foundational approach to mining itemset utilities from databases. *The 4th SIAM International Conference on Data Mining*, 482–486.

[14] W. M. Huang, T. P. Hong, G. C. Lan, M. C. Chiang, and J. C. W. Lin. 2017. Temporal-based fuzzy utility mining. *IEEE Access*, Vol. 5, 26639–26652.