MINING HIGH UTILITY ITEMSETS IN DATA STREAMS BASED ON THE WEIGHTED SLIDING WINDOW MODEL

Pauray S.M. Tsai

Department of Computer Science and Information Engineering, Minghsin University of Science and Technology, Hsin-Feng, Hsinchu 304, Taiwan

ABSTRACT

Most of researches on mining high utility itemsets focus on the static transaction database, where all transactions are treated with the same importance and the database can be scanned more than once. With the emergence of new applications, data stream mining has become a significant research topic. In the data stream environment, online data stream mining algorithms are restricted to make only one pass over the data. However, present methods for mining high utility itemsets still cannot meet the requirement. In this paper, we propose a single pass algorithm for high utility itemset mining based on the weighted sliding window model. The developed algorithm takes advantage of reusing stored information to efficiently discover all the high utility itemsets in data streams.

KEYWORDS

Data Stream Mining, Weighted Sliding Window, Frequent Itemset, High Utility Itemset

1. INTRODUCTION

Frequent itemset mining is an important research topic in data mining communities. The well-known Apriori algorithm [1,2] is a kind of generation-and-test approach which needs scanning the database multiple times. The traditional frequent itemset mining cannot find profitable itemsets because the purchased quantity and the unit profit of an item are not considered. To fulfill the requirement of finding the profitable itemsets, more and more researches have been performed on high utility itemset mining [4,9,12,13,18,20,24].

The goal of high utility itemset mining is to find all the itemsets with utilities higher than the user-specified threshold. In the transaction database, there are two types of utilities for items, internal utility and external utility. The internal utility of an item represents the importance of an item in the transaction, for example, the quantity of an item purchased in the transaction. The external utility of an item is defined according to user objectives, for example, the unit profit value of an item, which is not available in the transaction.

Based on the definitions of high utility itemstes [23] , Liu, Liao, and Choudhary [18] proposed the Two-Phase algorithm to discover high utility itemstes. The transaction-weighted utilization

DOI : 10.5121/ijdkp.2014.4202

was defined and proved to satisfy the downward closure property. In the first phase, multiple database scans are required. For the kth database scan, k-element transaction-weighted utilization itemsets are found and used to generate candidate (k+1)-element transaction-weighted utilization itemsets. In the second phase, one extra database scan is performed to determine the actual high utility itemstes. The approach is especially suitable for the sparse database with short patterns. Erwin, Gopalan, and Achuthan [8] proposed the CTU-Mine algorithm based on the pattern growth approach [11]. The algorithm is more efficient than the Two-Phase method in the dense database with long patterns.

Most of research on high utility mining focuses on static databases. With the emergence of new applications, the data processed may be in the continuous dynamic data stream [7,14,15]. Examples include network traffic analysis, Web click stream mining, network intrusion detection, and on-line transaction analysis. Because the data in streams come with high speed and are continuous and unbounded, each item in a stream could be examined only once and the mining result should be generated as fast as possible. The traditional mining methods for static data usually read the database more than once. However due to the consideration of performance and storage constraints, online data stream mining algorithms are restricted to make only one pass over the data. Thus, traditional approaches of mining high utility itemsets [3,16,22], which require to scan databases more than once, cannot be directly applied to data stream mining.

The time models for data stream mining mainly include the landmark model [19], the tilted-time window model [10] and the sliding window model [5]. The landmark model considers all the data from a specified point of time (usually the time the system starts) to the current time. All the data considered are treated equally. The tilted-time window model is a variation of the landmark model. It also considers data from the start of a stream up to the current moment, but the time period is divided into multiple time slots. Different from the landmark model, the sliding window model focuses on the recent data from the current moment back to a specified time point. The size of the window could be defined to be a fixed time period or a given number of transactions.

Chu, Tseng, & Liang [6] first proposed a method, named THUI-Mine algorithm, for mining temporal high utility itemsets from data streams. THUI-Mine divides the database into several partitions, and used the filtering threshold and the database reduction method to reduce the number of candidate itemsets. The THUI-Mine algorithm still requires reading the database more than once. Li, Yeh, & Chang [17] proposed two algorithms, MHUI-BIT and MHUI-TID, for mining high utility itemsets from data streams within a transaction-sensitive sliding window. However, the data in a sliding window need to be scanned twice, which cannot really meet the requirement of stream mining.

Existent models such as the landmark model and the tilted-time window model consider the data generated from the starting time up to the current moment. As to the traditional sliding window model, only the data in one window is considered at each time point. We proposed a new framework for data stream mining, called weighted sliding window model [21]. The model allows the user to specify the number of windows, the size of a window, and the weight for each window. The approach of allowing the user to specify higher weights to more significant windows could make the mining result be closer to the user's requirement.

In this paper, we propose an approach for mining high utility itemsets in data streams based on the weighted sliding window model. The rest of this paper is organized as follows. In Section 2, we describe the motivation of adopting the weighted sliding window model. In Section 3, algorithm HUI_W is proposed for efficient generation of high utility itemsets based on the weighted sliding windows. An example is given to illustrate the processing of the algorithm in Section 4. Finally we conclude in Section 5.

2. MOTIVATION

In this section, we illustrate how the weighted sliding window model can be used to effectively find high utility itemsets in data streams.

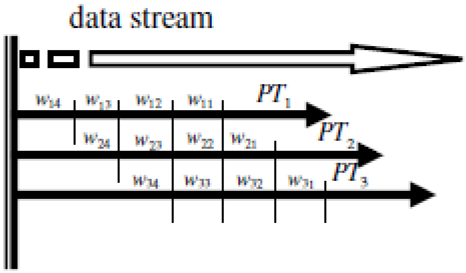


Figure 1. The weighted sliding window model

The weighted sliding window model, shown in Figure 1, has the following two features:

(1) The size of a window is defined by time, not the number of transactions. The purpose is to avoid the case where the lengths of intervals that cover the same number of transactions at different time points vary dramatically.

(2) The number of windows considered for mining is specified by the user. Moreover, the user can assign different weights to different windows according to the importance of data in each section. For example, the data nearer to the current moment may be more influential in the mining and could be given a higher weight.

We give examples to explain the essence of the weighted sliding window model. Assume that the number of sliding windows is 4 and the time covered by each window is *t*. (We call *t* the size of a window.) The weight α_j for window w_{ij} at the mining point PT_i is as follows: $\alpha_1 = 0.4$, $\alpha_2 = 0.3$, $\alpha_3 = 0.2$, $\alpha_4 = 0.1$, $\sum_{j=1}^{4} \alpha_j = 1$. Table 1 shows the transaction data at time point PT_1, where *Tid* represents the identifier of a transaction and (*x*,*y*) indicates an item *x* with a purchased quantity *y*. The profit for each item is shown in Table 2.

Table 1. The transaction data at time point PT_1

w_{14}		w_{13}		w_{12}		w_{11}	
Tid	itemset	*Tid*	itemset	*Tid*	itemset	*Tid*	itemset
T_1	{(a,1),(d,2),(e,1)}	T_4	{(c,6),(d,2)}	T_8	{(a,2),(b,3)(c,2)}	T_{10}	{(d,1),(e,2)}
T_2	{(a,3),(b,5)}	T_5	{(a,2),(b,5) (c,1),(d,1)}	T_9	{(c,1),(d,2)}	T_{11}	{(a,1),(b,1),(d,5)}
T_3	{(a,1),(b,2)}	T_6	{(d,3),(e,2)}			T_{12}	{(c,2),(d,1),(e,5)}
		T_7	{(b,3),(c,1)}				

Table 2. Profit table

item	**profit(per unit)**
a	1
b	5
c	9
d	5
e	3

Related definitions for utility mining are described as follows.

Definition 2.1. The *internal utility* $q(x,T)$ represents the quantity of item x in transaction T.

Definition 2.2. The *external utility* (profit per unit) $P(x)$ is the profit of item x.

Definition 2.3. The *utility of item x in transaction T* is defined by $u(x,T)=q(x,T)\times P(x)$.

Definition 2.4. The *utility of itemset X in transaction T* is defined by

$u(X,T)=\sum_{x\in X} u(x,T)$.

Definition 2.5. An itemset X is called a *high utility itemset* if its utility is no less than a user-specified *minimum utility threshold* (denoted as *min_utility*). Conversely, an itemset Y is called a *low utility itemset* if its utility is less than *min_utility*.

Definition 2.6. The *utility of itemset X in window* w_{ij} is defined by

$u(X, w_{ij})=\sum_{X\subseteq T\wedge T\in w_{ij}} u(X,T)$.

Definition 2.7. The *weighted utility of itemset X in window* w_{ij} is defined by

$wu(X, w_{ij})=u(X, w_{ij})\times \alpha_j$.

Definition 2.8. The *weighted utility of itemset X at the mining point* PT_i is defined by

$wu(X)=\sum_{1\le j\le n} wu(X, w_{ij})$, n is the number of sliding windows.

Definition 2.9. The *minimum weighted utility threshold* (denoted as *min_wu*) is defined as *min_utility*/the number of sliding windows.

Definition 2.10. An itemset X is called a *high weighted utility itemset* (HWU_itemset) if its weighted utility is no less than a user-specified *min_wu*. Otherwise, it is called a *low weighted utility itemset*.

By Table 1, we use the following two cases to show the effects of weighted sliding windows on utility mining.

Case 1. Sliding windows without weight consideration

Assume *min_utility* is 90. The utilities of itemset $\{b\}$ and itemset $\{d\}$ are evaluated, respectively, as follows.

$u(\{b\})=u(\{b\},w_{11})+u(\{b\},w_{12})+u(\{b\},w_{13})+u(\{b\},w_{14})=1\times5+3\times5+8\times5+7\times5=95$

$u(\{d\})=u(\{d\},w_{11})+u(\{d\},w_{12})+u(\{d\},w_{13})+u(\{d\},w_{14})=7\times5+2\times5+6\times5+2\times5=85$

In this case, itemset $\{b\}$ is a high utility itemset, but itemset $\{d\}$ is a low utility itemset.

Case 2. Sliding windows with weight consideration

Assume *min_utility* is 90. By Definition 2.9, *min_wu*=90/4=22.5. The weighted utilities of itemset $\{b\}$ and itemset $\{d\}$ are evaluated, respectively, as follows.

$wu(\{b\})=wu(\{b\},w_{11})+wu(\{b\},w_{12})+wu(\{b\},w_{13})+wu(\{b\},w_{14})=5\times0.4+15\times0.3+40\times0.2+35\times0.1=18$.

$wu(\{d\})=wu(\{d\},w_{11})+wu(\{d\},w_{12})+wu(\{d\},w_{13})+wu(\{d\},w_{14})=35\times0.4+10\times0.3+30\times0.2+10\times0.1=24$.

In this case, itemset $\{b\}$ becomes a low weighted utility itemset, while itemset $\{d\}$ becomes a high weighted utility itemset. It can be seen that the weights of windows could effectively affect the determination of high weighted utility itemsets. Even if the total utility of an itemset is large, if its utility in the window with a high weight is very low, it may not become a high weighted utility itemset. Thus, the consideration of assigning a higher weight to a nearer window is helpful to make the mining result be closer to user's requirements.

3. MINING HIGH WEIGHTED UTILITY ITEMSETS OVER WEIGHTED SLIDING WINDOWS

In this section, we introduce an algorithm for mining high weighted utility itemsets based on the weighted sliding window model. Table 1, Table 2, and assumed weights for windows in Section 2 are used for the following examples. We first define some terminologies and then introduce the proposed algorithm, HUI_W.

Definition 3.1. An itemset of size k is called a k-itemset.

Definition 3.2. The *transaction utility* of transaction T_s is defined by $TU(T_s)=u(T_s, T_s)$.

For example, $TU(T_1)=u(\{\mathrm{a}\}, T_1)+u(\{\mathrm{d}\}, \mathrm{T}_1)+ u(\{\mathrm{e}\}, T_1)=14$.

Definition 3.3. The *transaction-weighted utility* of transaction T_s is defined by $TWU(T_s)= TU(T_s)\times\alpha_j$, $T_s\in w_{ij}$.

For example, $TWU(T_1)= TU(T_1)\times\alpha_4$ =14×0.1=1.4.

Definition 3.4. The *transaction-weighted utilization* of itemset X is defined by $TWU(X)= \sum_{X\subseteq T_s\wedge T_s\in w_{ij}} TU(T_s)\times\alpha_j$.

For example, $TWU(\{a,b\})=TU(T_2)\times0.1+TU(T_3)\times0.1+TU(T_5)\times0.2+TU(T_8)\times0.3+TU(T_{11})\times0.4$

=28×0.1+11×0.1+41×0.2+35×0.3+31 ×0.4=35.

Definition 3.5. An itemset X is a *high transaction-weighted utilization itemset* (*HTWU_itemset*) if $TWU(X)\geq min\_wu$.

By Definitions 3.4 and 3.5, we obtain the following property.

Property 3.1. Transaction-weighted downward closure. For any itemset X, if X is not an *HTWU_itemset*, any superset of X is a low weighted utility itemset.

Assume the number of windows is n, the size of a window is t, the current time point is PT_i, and the weight for window w_{ij} at the mining point PT_i is α_j ($1\leq j\leq n$). *x.tid* and *x.qty* denote the identifier and the quantity of item x, respectively. $ST_{ij}(X)$ represents the set of identifiers of transactions containing itemset X in window w_{ij} and the associated utilities. Assume that $ST_{ij}(X)=\{st_1, st_2,\ldots, st_p\}$, $st_k.tid$ and $st_k.uty$ ($1\leq k\leq p$) denote the identifier of the transaction containing itemset X and the utility of X in the transaction, respectively. $ST_{ij}(X).uty$ represents the utility of itemset X in window w_{ij}, namely, $ST_{ij}(X).uty$ is equal to $u(X, w_{ij})$. In the mining, we can evaluate the transaction utility for each transaction when windows are scanned. Thus the transaction-weighted utilization (*TWU*) for item x can be computed and used to determine whether $\{x\}$ is an *HTWU*_itemset.

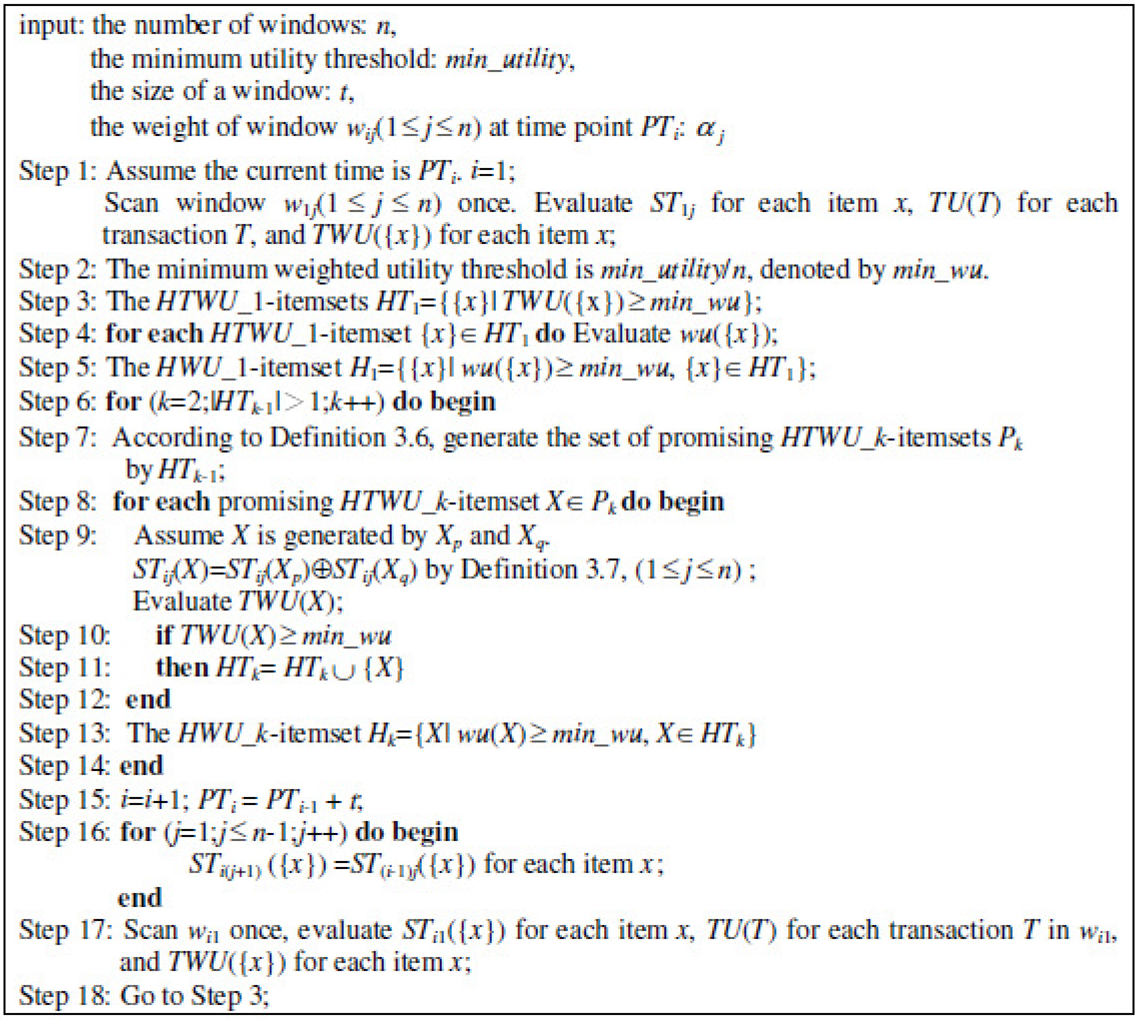
According to Property 3.1, if itemset $\{x\}$ is not an *HTWU*_itemset, any of its superset will be a low weighted utility itemset. In the case, itemset $\{x\}$ need not be considered in the following processing. In the mining process, we first consider all the *HTWU*_1-itemsets. For each *HTWU*_1-itemset $\{x\}$, its weighted utility can be evaluated by $ST_{ij}(X).uty$ to determine whether it is a high weighted utility itemset. According to Property 3.1, we use the combination of *HTWU*_(k-1)-itemsets to generate the *promising HTWU_k*-itemsets, which is similar to the method of Apriori [2] using frequent (k-1)-itemsets to generate candidate k-itemsets.

Let $X[1]$,$X[2]$,…, and $X[k-1]$ be the k-1 items in *HTWU_*(k-1)-*itemset* X and $X[1]<X[2]<\ldots<X[k-1]$. HT_{k-1} represents the set of all *HTWU_*(k-1)-*itemsets*.

Definition 3.6. The set of promising *HTWU_k*-itemsets($k\geq2$), P_k, is defined as

$P_k=\{\{X_p[1],X_p[2],\ldots,X_p[k-1],X_q[k-1]\}|X_p\in \mathrm{HT}_{k-1}$and $X_q\in \mathrm{HT}_{k-1}$ and $X_p[u]=X_q[u](1\leq u\leq k-2)$, and $X_p[k-1]<X_q[k-1]\}$

Assume that promising *HTWU_k*-itemset Y is generated by *HTWU_*(k-1)-*itemsets* X_p and X_q. $ST_{ij}(Y)$ can be evaluated by the combination of $ST_{ij}(X_p)$ and $ST_{ij}(X_q)$



```
input: the number of windows: $n$,
       the minimum utility threshold: $min\_utility$,
       the size of a window: $t$,
       the weight of window $w_{ij}(1 \le j \le n)$ at time point $PT_i$: $\alpha_j$
Step 1: Assume the current time is $PT_i$. $i$=1;
        Scan window $w_{1j}(1 \le j \le n)$ once. Evaluate $ST_{1j}$ for each item $x$, $TU(T)$ for each
        transaction $T$, and $TWU(\{x\})$ for each item $x$;
Step 2: The minimum weighted utility threshold is $min\_utility/n$, denoted by $min\_wu$.
Step 3: The $HTWU\_1$-itemsets $HT_1=\{\{x\} | TWU(\{x\}) \ge min\_wu\}$;
Step 4: for each $HTWU\_1$-itemset $\{x\} \in HT_1$ do Evaluate $wu(\{x\})$;
Step 5: The $HWU\_1$-itemset $H_1=\{\{x\} | wu(\{x\}) \ge min\_wu, \{x\} \in HT_1\}$;
Step 6: for ($k$=2;$|HT_{k-1}| > 1$;$k$++) do begin
Step 7:   According to Definition 3.6, generate the set of promising $HTWU\_k$-itemsets $P_k$
          by $HT_{k-1}$;
Step 8:   for each promising $HTWU\_k$-itemset $X \in P_k$ do begin
Step 9:     Assume $X$ is generated by $X_p$ and $X_q$.
            $ST_{ij}(X)=ST_{ij}(X_p) \oplus ST_{ij}(X_q)$ by Definition 3.7, $(1 \le j \le n)$ ;
            Evaluate $TWU(X)$;
Step 10:    if $TWU(X) \ge min\_wu$
Step 11:    then $HT_k = HT_k \cup \{X\}$
Step 12:  end
Step 13:  The $HWU\_k$-itemset $H_k=\{X | wu(X) \ge min\_wu, X \in HT_k\}$
Step 14: end
Step 15: $i=i+1$; $PT_i = PT_{i-1} + t$;
Step 16: for ($j$=1;$j \le n-1$;$j$++) do begin
             $ST_{i(j+1)}(\{x\}) = ST_{(i-1)j}(\{x\})$ for each item $x$;
         end
Step 17: Scan $w_{i1}$ once, evaluate $ST_{i1}(\{x\})$ for each item $x$, $TU(T)$ for each transaction $T$ in $w_{i1}$,
         and $TWU(\{x\})$ for each item $x$;
Step 18: Go to Step 3;
```

Figure 2. Algorithm HUI_W

Definition 3.7. Let X_p and X_q be *HTWU_*(k-1)-itemsets, $ST_{ij}(X_p)=\{st_{p1}, st_{p2},\ldots, st_{pm}\}$, and $ST_{ij}(X_q)=\{st_{q1}, st_{q2},\ldots, st_{qn}\}$. The combination of $ST_{ij}(X_p)$ and $ST_{ij}(X_q)$ is defined by $ST_{ij}(X_p) \oplus ST_{ij}(X_q)=\{st_{c1}, st_{c2},\ldots, st_{cr}\}$, $\{st_{c1}.tid, st_{c2}.tid,\ldots, st_{cr}.tid\}=\{st_{p1}.tid, st_{p2}.tid,\ldots, st_{pm}.tid\} \cap \{st_{q1}.tid, st_{q2}.tid,\ldots, st_{qn}.tid\}$, $st_{cu}.uty=st_{pv}.uty+st_h.uty$ $(st_h \in ST_{ij}(\{X_q[k\text{-}1]\})$ and $st_h.tid=st_{cu}.tid=st_{pv}.tid)$，$1 \le u \le r$，$1 \le v \le m$.

The *TWU* value of a promising *HTWU_k*-itemset X can be computed by the transaction utilities of transactions containing X, and used to determine whether itemset X is an *HTWU_k*-itemset. For each *HTWU_k*-itemset X, the weighted utility of X is evaluated and used to determine whether it is an *HWU_k*-itemset. The detailed algorithm is shown in Figure 2.

We can easily maintain all the high weighted utility itemsets by *HUI_W* algorithm. For example, consider Figure 1. At time point PT_2, for each item x, $ST_{24}(\{x\})=ST_{13}(\{x\})$, $ST_{23}(\{x\})=ST_{12}(\{x\})$, $ST_{22}(\{x\})=ST_{11}(\{x\})$. We only need to scan the data in w_{21} once to get $ST_{21}(\{x\})$. Once $ST_{2j}(\{x\})$ for each item x in w_{2j} $(1 \le j \le 4)$ are obtained, all the *HTWU_*1-itemsets can be generated and *HWU_*1-itemsets determined. By Step 6 in algorithm *HUI_W*, we can find all the high weighted utility k_itemsets ($k \geq 2$) at time point PT_2.

4. EXAMPLE

Assume the number of windows is 4 and the size of a window is 50 minutes. Namely, the interval between two mines is 50 minutes. The weight α_j of window $w_{ij}(1 \leq j \leq 4)$ is as follows: $\alpha_1 = 0.4$，$\alpha_2 = 0.3$，$\alpha_3 = 0.2$，$\alpha_4 = 0.1$. $\sum_{j=1}^{4} \alpha_j = 1$. Table 1 shows the transaction data at time point PT_1 and Table 2 is the profit for each item. Assume the minimum utility threshold is 80, and the minimum weighted utility threshold is 20. In the following, we illustrate the process of mining high weighted utility itemsets based on the weighted sliding window model.

By Step 1 of algorithm *HUI_W*, all the transaction data in each window are scanned once first at time point PT_1. ST_{1j} for each item, the transaction utility for each transaction T, and the transaction weighted utility for each item are evaluated as shown in Table 3, Table 4, and Table 5, respectively.

Table 3. ST_{1j} for each item.

ST_{1j} item	j=4	j=3	j=2	j=1
{a}	{(1,1),(2,3),(3,1)}	{(5,2)}	{(8,2)}	{(11,1)}
{b}	{(2,25),(3,10)}	{(5,25),(7,15)}	{(8,15)}	{(11,5)}
{c}	∅	{(4,54),(5,9),(7,9)}	{(8,18),(9,9)}	{(12,18)}
{d}	{(1,10)}	{(4,10),(5,5),(6,15)}	{(9,10)}	{(10,5),(11,25),(12,5)}
{e}	{(1,3)}	{(6,6)}	∅	{(10,6),(12,15)}

By Step 3, the set of *HTWU*_1-itemsets HT_1={{a},{b},{c},{d},{e}}. Then the weighted utility for each *HTWU*_1-itemset is computed as follows.

$wu(\{a\})=ST_{11}(\{a\}).uty\times0.4+ST_{12}(\{a\}).uty\times0.3+ST_{13}(\{a\}).uty\times0.2+ST_{14}(\{a\}).uty\times0.1=1\times0.4+2\times0.3+ 2\times0.2+5\times0.1=1.9$.

$wu(\{b\})=ST_{11}(\{b\}).uty\times0.4+ST_{12}(\{b\}).uty\times0.3+ST_{13}(\{b\}).uty\times0.2+ST_{14}(\{b\}).uty\times0.1=5\times0.4+15\times0.3+ 40\times0.2+35\times0.1=18$.

Table 4. Transaction utility (*TU*) for each transaction.

Tid	*TU*
T_1	14
T_2	28
T_3	11
T_4	64
T_5	41
T_6	21
T_7	24
T_8	35
T_9	19
T_{10}	11
T_{11}	31
T_{12}	38

Table 5. Transaction weighted utility (*TWU*) for each item.

TU_{1j} item	*j*=4	*j*=3	*j*=2	*j*=1	*TWU*
$\{a\}$	53	41	35	31	36.4
$\{b\}$	39	65	35	31	39.8
$\{c\}$	0	129	54	38	57.2
$\{d\}$	14	126	19	80	64.3
$\{e\}$	14	21	0	49	25.2

$wu(\{c\})=ST_{11}(\{c\}).uty\times0.4+ST_{12}(\{c\}).uty\times0.3+ST_{13}(\{c\}).uty\times0.2+ST_{14}(\{c\}).uty\times0.1=18\times0.4+27\times0.3+72\times0.2+0\times0.1=29.7.$

$wu(\{d\})=ST_{11}(\{d\}).uty\times0.4+ST_{12}(\{d\}).uty\times0.3+ST_{13}(\{d\}).uty\times0.2+ST_{14}(\{d\}).uty\times0.1=35\times0.4+10\times0.3+30\times0.2+10\times0.1=24.$

$wu(\{e\})=ST_{11}(\{e\}).uty\times0.4+ST_{12}(\{e\}).uty\times0.3+ST_{13}(\{e\}).uty\times0.2+ST_{14}(\{e\}).uty\times0.1=21\times0.4+0\times0.3+6\times0.2+3\times0.1=9.9.$

By Step 5, the set of *HWU*_1-itemsets $H_1=\{\{c\},\{d\}\}$. In Step 7, the set of promising *HTWU*_2-itemsets P_2 is generated by HT_1. ST_{1j} and *TWU* for each promising *HTWU*_2-itemset are evaluated in Step 9, as shown in Table 6 and Table 7, respectively. By Step 11, the set of

*HTWU*_2-itemsets $HT_2=\{\{a,b\},\{a,d\},\{b,c\},\{b,d\},\{c,d\},\{d,e\}\}$. Then the weighted value for each *HTWU*_2-itemset is computed as follows.

$wu(\{a,b\})=ST_{11}(\{a,b\}).uty\times0.4+ST_{12}(\{a,b\}).uty\times0.3+ST_{13}(\{a,b\}).uty\times0.2+ST_{14}(\{a,b\}).uty\times0.1=$ $6\times0.4+17\times0.3+ 27\times0.2+39\times0.1=16.8$.

$wu(\{a,d\})= 26\times0.4+0\times0.3+7\times0.2+11\times0.1=12.9$

$wu(\{b,c\})= 0\times0.4+33\times0.3+ 58\times0.2+0\times0.1=21.5$.

$wu(\{b,d\})= 30\times0.4+0\times0.3+30\times0.2+0\times0.1=18$.

$wu(\{c,d\})= 23\times0.4+19\times0.3+78\times0.2+0\times0.1=30.5$.

$wu(\{d,e\})= 31\times0.4+0\times0.3+ 21\times0.2+13\times0.1=17.9$.

Table 6. ST_{1j} for each promising *HTWU*_2-itemset.

ST_{1j} itemset	j=4	j=3	j=2	j=1
$\{a,b\}$	{(2,28),(3,11)}	{(5,27)}	{(8,17)}	{(11,6)}
$\{a,c\}$	∅	{(5,11)}	{(8,20)}	∅
$\{a,d\}$	{(1,11)}	{(5,7)}	∅	{(11,26)}
$\{a,e\}$	{(1,4)}	∅	∅	∅
$\{b,c\}$	∅	{(5,34),(7,24)}	{(8,33)}	∅
$\{b,d\}$	∅	{(5,30)}	∅	{(11,30)}
$\{b,e\}$	∅	∅	∅	∅
$\{c,d\}$	∅	{(4,64),(5,14)}	{(9,19)}	{(12,23)}
$\{c,e\}$	∅	∅	∅	{(12,33)}
$\{d,e\}$	{(1,13)}	{(6,21)}	∅	{(10,11),(12,20)}

By Step 13, the set of *HWU*_2-itemsets H_2={{b,c},{c,d}}. Similarly, the set of promising *HTWU*_3-itemsets P_3 is generated by HT_2. ST_{1j} and *TWU* for each promising *HTWU*_3-itemset are evaluated in Step 9, as shown in Table 8 and Table 9, respectively. By Step 11, the set of *HTWU*_3-itemsets HT_3 is $\{\{a,b,d\}\}$. $wu(\{a,b,d\})=ST_{11}(\{a,b,d\}).uty\times0.4+ST_{12}(\{a,b,d\}).uty\times0.3$ $+ST_{13}(\{a,b,d\}).uty\times0.2+ST_{14}(\{a,b,d\}).uty\times0.1=31\times0.4+0\times0.3+32\times0.2+0\times0.1=18.8$. The set of *HWU*_3-itemsets is empty. No promising *HTWU*_4-itemsets can be generated, and the process of data mining at time point PT_1 terminates.

As shown in Figure 1, window w_{21} represents the period of 50 minutes after PT_1. Table 10 shows the transaction data at time point PT_2. $w_{24}=w_{13}$, $w_{23}=w_{12}$, $w_{22}=w_{11}$. We need only to scan window w_{21} once. Then ST_{21} value for each item can be obtained as shown in Table 11. Transaction utilities for new transactions T_{13} and T_{14} are shown in Table 12. *TWU* for each item is evaluated as shown in Table 13. By Step 3, the set of *HTWU*_1-itemsets $HT_1=\{\{a\},\{b\},\{c\},\{d\}\}$. The weighted value for each *HTWU*_1-itemset is computed as follows.

$wu(\{a\})=ST_{21}(\{a\}).uty\times0.4+ST_{22}(\{a\}).uty\times0.3+ST_{23}(\{a\}).uty\times0.2+ST_{24}(\{a\}).uty\times0.1=$
$3\times0.4+1\times0.3+2\times0.2+2\times0.1=2.1.$

$wu(\{b\})=ST_{21}(\{b\}).uty\times0.4+ST_{22}(\{b\}).uty\times0.3+ST_{23}(\{b\}).uty\times0.2+ST_{24}(\{b\}).uty\times0.1=$

$35\times0.4+5\times0.3+15\times0.2+40\times0.1=22.5.$

$wu(\{c\})=ST_{21}(\{c\}).uty\times0.4+ST_{22}(\{c\}).uty\times0.3+ST_{23}(\{c\}).uty\times0.2+ST_{24}(\{c\}).uty\times0.1=$
$9\times0.4+18\times0.3+27\times0.2+72\times0.1=21.6.$

$wu(\{d\})=ST_{21}(\{d\}).uty\times0.4+ST_{22}(\{d\}).uty\times0.3+ST_{23}(\{d\}).uty\times0.2+ST_{24}(\{d\}).uty\times0.1=$
$15\times0.4+35\times0.3+10\times0.2+30\times0.1=21.5.$

Table 7. Transaction weighted utility (*TWU*) for each promising *HTWU*_2-itemset.

TU_{1j} itemset	j=4	j=3	j=2	j=1	*TWU*
$\{a,b\}$	39	41	35	31	35
$\{a,c\}$	0	41	35	0	18.7
$\{a,d\}$	14	41	0	31	22
$\{a,e\}$	14	0	0	0	1.4
$\{b,c\}$	0	65	35	0	23.5
$\{b,d\}$	0	41	0	31	20.6
$\{b,e\}$	0	0	0	0	0
$\{c,d\}$	0	105	19	38	41.9
$\{c,e\}$	0	0	0	38	15.2
$\{d,e\}$	14	21	0	49	25.2

Table 8. ST_{1j} for each promising *HTWU*_3-itemset.

ST_{1j} itemset	j=4	j=3	j=2	j=1
$\{a,b,d\}$	∅	{(5,32)}	∅	{(11,31)}
$\{b,c,d\}$	∅	{(5,39)}	∅	∅

Table 9. Transaction weighted utility (*TWU*) for each promising *HTWU*_3-itemset.

TU_{1j} itemset	j=4	j=3	j=2	j=1	*TWU*
{a,b,d}	0	41	0	31	20.6
{b,c,d}	0	41	0	0	8.2

Table 10：The transaction data at time point PT_2

W_{24}		W_{23}		W_{22}		W_{21}	
Tid	itemset	*Tid*	itemset	*Tid*	itemset	*Tid*	itemset
T_4	{(c,6),(d,2)}	T_8	{(a,2),(b,3) (c,2)}	T_{10}	{(d,1),(e,2)}	T_{13}	{(a,3),(b,2)}
T_5	{(a,2),(b,5) (c,1),(d,1)}	T_9	{(c,1),(d,2)}	T_{11}	{(a,1),(b,1),(d,5)}	T_{14}	{(b,5),(c,1),(d,3)}
T_6	{(d,3),(e,2)}			T_{12}	{(c,2),(d,1),(e,5)}		
T_7	{(b,3),(c,1)}						

Table 11. ST_{2j} for each item.

ST_{2j} item	j=4	j=3	j=2	j=1
{a}	{(5,2)}	{(8,2)}	{(11,1)}	{(13,3)}
{b}	{(5,25),(7,15)}	{(8,15)}	{(11,5)}	{(13,10),(14,25)}
{c}	{(4,54),(5,9),(7,9)}	{(8,18),(9,9)}	{(12,18)}	{(14,9)}
{d}	{(4,10),(5,5),(6,15)}	{(9,10)}	{(10,5),(11,25),(12,5)}	{(14,15)}
{e}	{(6,6)}	∅	{(10,6),(12,15)}	∅

By Step 5, the set of *HWU*_1-itemsets H_1={{b},{c},{d}}. In Step 7, the set of promising *HTWU*_2-itemsets P_2 is generated by HT_1. ST_{2j} and *TWU* for each promising *HTWU*_2-itemset are evaluated in Step 9, as shown in Table 14 and Table 15, respectively.

By Step 10, the set of *HTWU*_2-itemsets HT_2={{a,b},{b,c},{b,d},{c,d}}. The weighted value for each *HTWU*_2-itemset is computed as follows.

$wu(\{a,b\})=ST_{21}(\{a,b\}).uty\times0.4+ST_{22}(\{a,b\}).uty\times0.3+ST_{23}(\{a,b\}).uty\times0.2+ST_{24}(\{a,b\}).uty\times0.1=$ $13\times0.4+6\times0.3+ 17\times0.2+27\times0.1=13.1$.

$wu(\{b,c\})=ST_{21}(\{b,c\}).uty\times0.4+ST_{22}(\{b,c\}).uty\times0.3+ST_{23}(\{b,c\}).uty\times0.2+ST_{24}(\{b,c\}).uty\times0.1=$ $34\times0.4+0\times0.3+ 33\times0.2+58\times0.1=26$.

$wu(\{b,d\})=ST_{21}(\{b,d\}).uty\times0.4+ST_{22}(\{b,d\}).uty\times0.3+ST_{23}(\{b,d\}).uty\times0.2+ST_{24}(\{b,d\}).uty\times0.1=$ $40\times0.4+30\times0.3+ 0\times0.2+30\times0.1=28$.

$wu(\{c,d\})=ST_{21}(\{c,d\}).uty\times0.4+ST_{22}(\{c,d\}).uty\times0.3+ST_{23}(\{c,d\}).uty\times0.2+ST_{24}(\{c,d\}).uty\times0.1=$ $24\times0.4+23\times0.3+19\times0.2+78\times0.1=28.1$.

Table 12. Transaction utility (*TU*) for each transaction.

Tid	*TU*
T_4	64
T_5	41
T_6	21
T_7	24
T_8	35
T_9	19
T_{10}	11
T_{11}	31
T_{12}	38
T_{13}	13
T_{14}	49

Table 13. Transaction weighted utility (*TWU*) for each item.

TU_{2j} item	*j*=4	*j*=3	*j*=2	*j*=1	*TWU*
{*a*}	41	35	31	13	25.6
{*b*}	65	35	31	62	47.6
{*c*}	129	54	38	49	54.7
{*d*}	126	19	80	49	60
{*e*}	21	0	49	0	16.8

By Step 13, the set of *HWU*_2-itemsets H_2={{*b*,*c*},{*b*,*d*},{*c*,*d*}}. Similarly, the set of promising *HTWU*_3-itemsets P_3 is generated by HT_2. ST_{2j} and *TWU* for each promising *HTWU*_3-itemset are evaluated in Step 9, as shown in Table 16 and Table 17, respectively. By Step 11, the set of *HTWU*_3-itemsets HT_3={{*b*,*c*,*d*}}. *wu*({*b*,*c*,*d*})=ST_{21}({*b*,*c*,*d*}).*uty*×0.4+ST_{22}({*b*,*c*,*d*}).*uty*×0.3+ ST_{23}({*b*,*c*,*d*}).*uty*×0.2+ST_{24}({*b*,*c*,*d*}).*uty*×0.1=49×0.4+0×0.3+0×0.2+39×0.1=23.5. The *HWU*_3-itemsets H_3={{*b*,*c*,*d*}}. No promising *HTWU*_4-itemsets can be generated, and the process of data mining at time point PT_2 terminates.

Note that only the shadowed values in Table 11~Table 17 need to be computed. The other values have been obtained at time point PT_1.

Table 14. ST_{2j} for each promising *HTWU*_2-itemset.

ST_{2j} / itemset	*j*=4	*j*=3	*j*=2	*j*=1
{*a*,*b*}	{(5,27)}	{(8,17)}	{(11,6)}	{(13,13)}
{*a*,*c*}	{(5,11)}	{(8,20)}	∅	∅
{*a*,*d*}	{(5,7)}	∅	{(11,26)}	∅
{*b*,*c*}	{(5,34),(7,24)}	{(8,33)}	∅	{(14,34)}
{*b*,*d*}	{(5,30)}	∅	{(11,30)}	{(14,40)}
{*c*,*d*}	{(4,64),(5,14)}	{(9,19)}	{(12,23)}	{(14,24)}

Table 15. Transaction weighted utility (*TWU*) for each promising *HTWU*_2-itemset.

TU_{2j} / itemset	*j*=4	*j*=3	*j*=2	*j*=1	*TWU*
{*a*,*b*}	41	35	31	13	25.6
{*a*,*c*}	41	35	0	0	11.1
{*a*,*d*}	41	0	31	0	13.4
{*b*,*c*}	65	35	0	49	33.1
{*b*,*d*}	41	0	31	49	33
{*c*,*d*}	105	19	38	49	45.3

Table 16: ST_{2j} for each promising *HTWU*_3-itemset.

ST_{2j} itemset	*j*=4	*j*=3	*j*=2	*j*=1
{*b*,*c*,*d*}	{(5,39)}	∅	∅	{(14,49)}

Table 17. Transaction weighted utility (*TWU*) for each promising *HTWU*_3-itemset.

TU_{2j} itemset	*j*=4	*j*=3	*j*=2	*j*=1	*TWU*
{*b*,*c*,*d*}	41	0	0	49	23.7

5. CONCLUSIONS

In the data stream environment, online data stream mining algorithms are restricted to make only one pass over the data and requested to generate the result efficiently. However, present methods for mining high utility itemsets still cannot meet the requirements. In this paper, we propose a single pass algorithm, *HUI_W*, for high utility itemset mining based on the weighted sliding window model. Using the model, the proposed algorithm takes advantage of reusing the stored information to efficiently discover all the high weighted utility itemsets over data stream.

REFERENCES

[1] Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Databases. In Proceedings of ACM SIGMOD, pp. 207-216.

[2] Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In Proceedings of the VLDB Conference, pp. 487-499.

[3] Ahmed, C. F., Tanbeer, S. K., Jeong, B. S., & Lee, Y. K. (2009). Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases. IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 12, pp. 1708-1721.

[4] Chan, R., Yang, Q., & Shen, Y. (2003). Mining High Utility Itemsets. In Proceedings of IEEE International Conference on Data Engineering, pp. 19-26.

[5] Chi, Y., Wang, H., Yu, P. S., & Muntz, R. R. (2006). Catch the Moment: Maintaining Closed Frequent Itemsets over a Data Stream Sliding Window. Knowledge and Information Systems, Vol. 10, No. 3, pp. 265-294.

[6] Chu, C. J., Tseng, V. S., & Liang, T. (2008). An Efficient Algorithm for Mining Temporal High Utility Itemsets from Data Streams. The Journal of Systems and Software, Vol. 81, No. 7, pp. 1105-1117.

[7] Domingos, P., & Hulten, G. (2000). Mining High-Speed Data Streams, In Proceedings of ACM SIGKDD, pp. 71-80.

[8] Erwin, A., Gopalan, R. P., & Achuthan, N. R. (2007). CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach. In Proceedings of IEEE International Conference on Computer and Information Technology, pp. 71-76.

[9] Erwin, A., Gopalan, R. P., & Achuthan, N. R. (2008). Efficient Mining of High Utility Itemsets from Large Datasets. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 554-561.

[10] Giannella, C., Han, J., Pei, J., Yan, X., & Yu, P. S. (2003). Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In H. Kargupta, A. Joshi, K.Sivakumar, & Y. Yesha (Eds.), Next generation data mining, AAA/MIT, pp. 191-210.

[11] Han, J., Pei, J., & Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. In Proceedings of ACM SIGMOD, pp. 1-12.

[12] Hong, T. P., Lee, C. H., & Wang, S. L. (2009). An Incremental Mining Algorithm for High Average-Utility Itemsets. In Proceedings of International Symposium on Pervasive Systems, Algorithms, and Networks, pp. 421-425.

[13] Hong, T. P., Lee, C. H., & Wang, S. L. (2011). Effective Utility Mining with the Measure of Average Utility. Expert Systems with Applications, Vol. 38, No. 7, pp. 8259–8265.

[14] Jiang, N., & Gruenwald, L. (2006). Research Issues in Data Stream Association Rule Mining. SIGMOD Record, Vol. 35, No. 1, pp. 14-19.

[15] Lee, C. H., Lin, C. R., & Chen, M. S. (2001). Sliding-Window Filtering: An Efficient Algorithm for Incremental Mining. In Proceedings of International Conference on Information and Knowledge Management, pp. 263-270.

[16] Li, H. F., Huang, H. Y., Chen, Y. C., Liu, Y. J., & Lee, S. Y. (2008). Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams. In Proceedings of IEEE International Conference on Data Mining, pp. 881-886.

[17] Li, Y. C., J., Yeh, J. S., & Chang, C. C. (2008). Isolated Items Discarding Strategy for Discovering High Utility Itemsets. Data and Knowledge Engineering, Vol. 64, No. 1, pp. 198-217

[18] Liu, Y., Liao, W. K., & Choudhary, A. (2005). A Two Phase Algorithm for Fast Discovery of High Utility Itemsets. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 689-695.

[19] Manku, G., & Motwani, R. (2002). Approximate Frequency Counts over Data Streams. In Proceedings of the VLDB Conference, pp. 346-357.

[20] Pillai, J., & Vyas, O. P. (2010). Overview of Itemset Utility Mining and its Applications. International Journal of Computer Applications, Vol. 5, No. 11, pp. 9-13.

[21] Tsai, P. S. M. (2009). Data Stream Mining Using the Weighted Sliding Window Model. Expert Systems With Applications, Vol. 36, No. 9, pp. 11617-11625.

[22] Tseng, V. S., Wu, C. W., Shie, B. E., & Yu, P. S. (2010). UP-Growth: An Efficient Algorithm for High Utility Itemset Mining. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 253-262.

[23] Yao, H., Hamilton, H. J., & Butz, C. J. (2004). A Foundational Approach to Mining Itemset Utilities from Databases. In Proceedings of SIAM International Conference on Data Mining, pp. 482-486.

[24] Yao, H., & Hamilton, H. J. (2006). Mining Itemset Utilities from Transaction Databases. Data & Knowledge Engineering, Vol. 59, No. 3, pp. 603-626.