Mining High Utility Itemsets in Uncertain Databases

Yuqing Lan, Yang Wang, Shengwei Yi, Dan Yu, and Simin Yu

School of Computer Science and Engineering, Beihang University, China
{lanyuqing, yangwang, yishengwei, yudan, siminyu}@buaa.edu.cn

Abstract. Recently, with the growing popularity of Internet of Things (IoT) and pervasive computing, a large amount of uncertain data, i.e. RFID data, sensor data, real-time monitoring data, etc., has been collected. As one of the most fundamental issues of uncertain data mining, the problem of mining uncertain frequent itemsets has attracted much attention in the database and data mining communities. Although some efficient approaches of mining uncertain frequent itemsets have been proposed, most of them only consider each item in one transaction as a random variable following the binary distribution and ignore the unit values of items in the real scenarios. In this paper, we focus on the problem of mining probabilistic high utility itemsets in uncertain databases (MPHU), in which each item has a unit value. In order to solve the MPHU problem, we propose a novel mining framework, called UUIM, which not only includes an efficient mining algorithm but also contains an effective pruning technique. Extensive experiments on both real and synthetic datasets verify the effectiveness and efficiency of proposed solutions.

1 Introduction

Recently, with the growing popularity of Internet of Things (IoT) and pervasive computing, a large amount of uncertain data, i.e. RFID data, sensor data, real-time monitoring data, etc., has been collected. As one of the most fundamental issues of uncertain data mining, the problem of mining uncertain frequent itemsets has attracted much attention in the database and data mining communities. Although some efficient approaches of mining uncertain frequent itemsets have been proposed, most of them only consider each item in one transaction as a random variable following the binary distribution and ignore the unit values of items in the real scenarios. In recommender systems (e.g. music, video) analysis, mining frequent itemsets from an uncertain database refers to discovery of the itemsets which may frequently appear together in the records (transactions). However, the unit values (profits) of items are not considered in the framework of uncertain frequent itemsets mining. Hence, it cannot satisfy the requirement of the user who is interested in discovering the itemsets with high enjoyment values. For example, consider the table of song data shown in Table 1. A and B in the table have different scores which depend on the evaluation of most users. The table of record is shown in the Table 2. Jack had heard A (Big Big World). The probable value that Jack enjoys Big Big World is 0.8. When songs are recommended, not only the frequency of songs in different users record should be considered, but the popularity of songs also ought to be taken into account. In view of this, utility mining will emerges as an important topic in data mining for discovering the itemsets with high utility like values (profits) in uncertain databases.

Table 1. A Table of Songs

ID	Song Name	Score
A	Big Big World	3
B	Someone Like You	4
C	Dont You Remember	1
D	My Love	1
E	Time to say goodbye	8
F	Right Here Waiting	2
G	Can You	1

Table 2. A Table of Records

User	Record and matching analysis
Jack	(A, 0.8) (B, 0.8) (C, 0.7) (D, 0.7)
Rose	(A, 0.9) (C, 0.8) (G, 0.8)
Tom	(A, 0.6) (B, 0.7) (C, 0.6) (D, 0.5) (E, 0.8) (F, 0.5)
Peter	(B, 0.7) (C, 0.8) (D, 0.5)
Linda	(B, 0.8) (C, 0.7) (F, 0.4)

Mining high utility itemsets from the databases refers to finding the itemsets with high utilities. The basic meaning of utility is the ~~interestedness~~ / importance / profitability of ~~items~~ to the users. Before finding high utility itemsets over uncertain databases, the definition of the high utility itemset is the most essential issue. In deterministic data, the utility of items in a transaction database consists of two aspects: (1) the importance of distinct items, which is called external utility, and (2) the importance of the items in the transaction, which is called internal utility. The utility of an itemset is defined as the external utility multiplied by the internal utility. An itemset is called a high utility itemset if its utility is no less than a user-specified threshold. However, different from the deterministic case, the utility of items in an uncertain transaction database only contain the importance of distinct items, which is called external utility in deterministic database. The definition of a high utility itemset over uncertain data has two different semantic explanations: expected support-based high utility itemset and probabilistic high utility itemset. However, the two definitions are different on using the random variable to define high utility itemsets. In the definition of the expected support-based high utility itemset, the expectation of the utility of an itemset is defined as the measurement, called as the expected utility of this itemset. In this definition, an itemset is high utility if and only if the expected utility of such itemset is no less than a specified minimum expected utility threshold, min ~~e~~util. In the definition of probabilistic utility itemset, the probability that the utility of an itemset is no less than the threshold is defined as the measurement, called as the high utility probability of an itemset, and an itemset is high utility if and only if the high utility probability of such itemset is larger than a given probabilistic threshold.

Mining high utility itemsets from uncertain databases is an important task which is essential to a wide range of applications such as ~~most of~~ recommender system~~s~~ analysis, Internet of Things (IoT) and ~~even~~ biomedical applications. The definition of probabilistic high utility itemset includes the complete probability distribution of the utility of an itemset. Although the expectation is known as an important statistic, it cannot show the

complete probability distribution. Hence, we mainly discuss mining probabilistic high utility itemsets in this paper.

To sum up, we make the following contributions:

- To the best of our knowledge, this is the first work to formulate the problem of mining probabilistic high utility itemsets in uncertain databases (MPHU).
- Due to the challenges from utility constraints, we propose a novel mining framework, called UUH-mine, which not only includes an efficient mining algorithm but also contains an effective pruning technique.
- We verify the effectiveness and efficiency of the proposed methods through extensive experiments on real and synthetic datasets.

The rest of the paper is organized as follows. Preliminaries and our problem formulation are introduced in Section 2. In Section 3, we present a novel mining framework, called UUH-mine. Based on this framework, an efficient mining algorithm and an effective pruning technique is devised in Section 4. In Section 5, experimental studies on both real and synthetic datasets are reported. In Section 6, we review the existing works. Finally, we conclude this paper in Section 7.

2 Preliminaries and Problem Definitions

In this section, we first introduce some basic concepts and then define the problem of mining probabilistic high utility itemsets in uncertain databases.

2.1 Preliminaries

We first review the classical problem of frequent pattern mining.

Given a finite set of items I = $\{i_1, i_2,...,i_n\}$. An itemset X is a subset of items, i.e., X $\subseteq$ I. For the sake of brevity, an itemset X = $\{i_1, i_2,...,i_m\}$ is also denoted as X = $i_1 i_2 ... i_m$. A transaction T = (tid, Y) is a 2-tuple, where tid is a transaction-id and Y is an itemset. A transaction T = (tid, Y) is said to contain itemset X if and only if X $\subseteq$ Y. A transaction database D is a set of transactions. The number of transactions in D containing itemset X is called the support of X, denoted as sup(X). Given a transaction database D and a support threshold min_sup, an itemset X is a frequent pattern, or a pattern in short, if and only if sup(X) $\geq$ min_sup.

The problem of frequent pattern mining is to find the complete set of frequent patterns in a given transaction database with respect to a given support threshold.

In the problem of high utility pattern mining, each item $i_p (1 \leq p \leq n)$ has a unit profit $p(i_p)$, and each item i_p in the transaction T_d is associated with a quantity q(i_p,T_d), that is, the purchased number of i_p in T_d. The utility of an item i_p in the transaction T_d is denoted as u(i_p,T_d)and defined as $p(i_p) \times q(i_p, T_d)$. The utility of an itemset X in T_d is denoted as $u(X, T_d)$ and defined as $\sum_{i_p \in X \wedge X \subseteq T_d} u(i_p, T_d)$. The utility of an itemset X in D is denoted as $u(X)$ and defined as $\sum_{X \subseteq T_d \wedge T_d \in D} u(X, T_d)$.An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold which is denoted as min_util. Otherwise, it is called a low utility itemset.

Given a transaction database D and a user-specified minimum utility threshold min_util, mining high utility itemsets from the transaction database is equivalent to ~~discover~~ from D all itemsets whose utilities are no less than min_util.

All above mentioned problems are based on the deterministic databases. When mining high utility itemsets in uncertain databases, there will be lots of differences.

2.2 Problem Definitions

In this subsection, we give several basic definitions about mining high utility itemsets over uncertain databases.

Let $I = \{i_1, i_2, ..., i_n\}$ be a set of distinct items. Each item i_p has a unit value $v(i_p)$. We name a non-empty subset, X, of I as an itemset. For brevity, we use $X = x_1x_2...x_n$ to denote itemset $X = \{x_1, x_2, ...x_n\}$. X is a l - itemset if it has l items. Given an uncertain transaction database UD, each transaction is denoted as a tuple $< tid, Y >$ where tid is the transaction identifier, and $Y = \{y_1(p_1), y_2(p_2), ..., y_n(p_n)\}$. Y contains m units. Each unit has an item y_i and probability, p_i, denoting the possibility of item y_i appearing in the tid tuple. The number of transactions containing X in UD is a random variable, denoted as $sup(X)$. Given UD, the expected support-based high utility itemsets and probabilistic high utility itemsets are defined as follows.

$Definition1$. (Expexted Support) Given an uncertain transaction database UD which includes N transactions, and an itemset X, the expected support of X is:

$esup(X) = \sum_{i=1}^{N} p_i(X)$

Table 3. An Uncertain Database

TID	Transaction
T_1	(A, 0.8) (B, 0.8) (C, 0.7) (D, 0.7)
T_2	(A, 0.9) (C, 0.8) (G, 0.8)
T_3	(A, 0.6) (B, 0.7) (C, 0.6) (D, 0.5) (E, 0.8) (F, 0.5)
T_4	(B, 0.7) (C, 0.8) (D, 0.5)
T_5	(B, 0.8) (C, 0.7) (F, 0.4)

Table 4. Value Table

Item	A	B	C	D	E	F	G
Value	3	4	1	1	8	2	1

~~Differ~~ from deterministic databases, the utility of an itemset X in T_d is denoted as $u(X, T_d)$ and defined as $\sum_{i_p \in X} v(i_p)$ in uncertain databases.

Table 5. The Probability Distribution of sup(A)

sup(A)	0	1	2	3
Probability	0.008	0.116	0.444	0.432

$Definition2$. (Expected Utility) Given an uncertain transaction database UD which includes N transactions, and an itemset X, the expected utility of X is:

$EU(X) = U(X, T_d) \times esup(X)$

$Definition3$. (Expected Support-based High Utility Itemset) Given an uncertain transaction database UD which includes N transactions, and a minimum expected utility ratio, min_eutil, an itemset X is an expected support-based high utility itemset if and only if $EU(X) \geq N \times min\_eutil$

$Definition4$. (High Utility Probability) Given an uncertain transaction database UD which includes N transactions, a minimum utility ratio min_util, and an itemset X, X's high utility probability, denoted as Pr(X), is shown as follows:

$Pr(X) = Pr\{sup(X) \times U(X, T_d) \geq N \times min\_util\}$

$Definition5$. (Probabilistic High Utility Itemset) Given an uncertain transaction database UD which includes N transactions, a minimum utility ratio min_util, and a probabilistic high utility threshold put, an itemset X is a probabilistic high utility itemset if X's high utility probability is larger than the probabilistic high utility threshold, namely,

$Pr(X) = Pr\{sup(X) \times U(X, T_d) \geq N \times min\_util\} \geq put$

$Example$ (Probabilistic High Utility Itemset) Given an uncertain database in Table 5, min_util = 1 and probabilistic high utility threshold put = 0.9, $sup(A) \geq min\_util \times N \div U(A, T_d)$. So, the high utility probability of A is: $Pr(A) = Pr\{sup(A) \geq 5 \times 1 \div 3\} = Pr\{sup(A) \geq 2\} = Pr\{sup(A) = 2\} + Pr\{sup(A) = 3\} = 0.444 + 0.432 = 0.876 < 0.9 = put$. Thus, A is not a probabilistic high utility itemset.

We are now able to specify the Mining Probabilistic High Utility Itemsets (MPHU) problem as follows: Given an uncertain transaction database UD, a minimum utility threshold min_util and a probabilistic high utility threshold put. ~~The~~ problem of MPHU is to find all probabilistic high utility itemsets in uncertain databases.

3 UUH-mine Framework

In order to solve the MPHU problem, we propose a novel mining framework called UUH-Mine which is ~~also~~ based on the divide-and-conquer framework and the depth-first search strategy. The algorithm was extended from the H-Mine algorithm which is classical algorithm in deterministic frequent itemset mining.

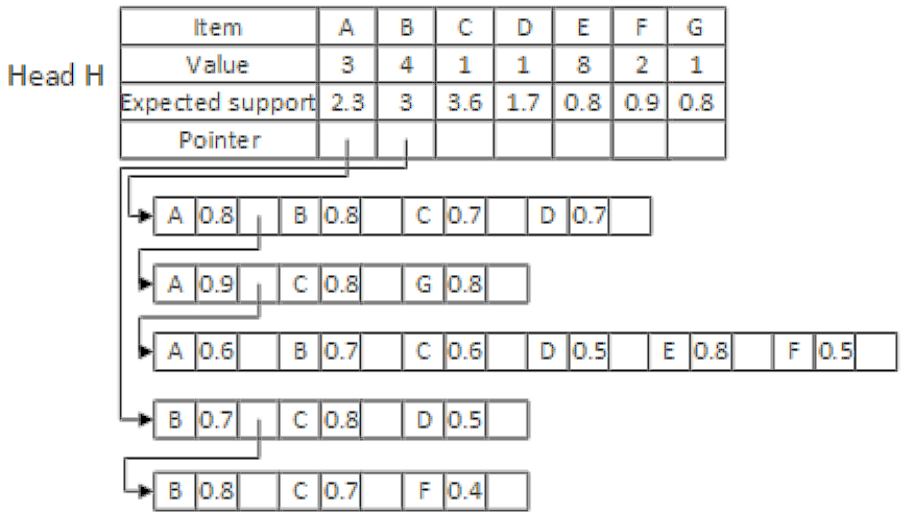

Fig. 1. UUH-Struct Generated from Table 3

UUH-Mine algorithm can be outlined as follows. Firstly, it scans the uncertain database and finds all items. Then, the algorithm builds a head table which contains

all items. For each item, the head table stores four elements: the label of this item, the value of this item, the expected support of such item, and a pointer domain. After building the head table, the algorithm inserts all transactions into the data structure, UUH-Struct.

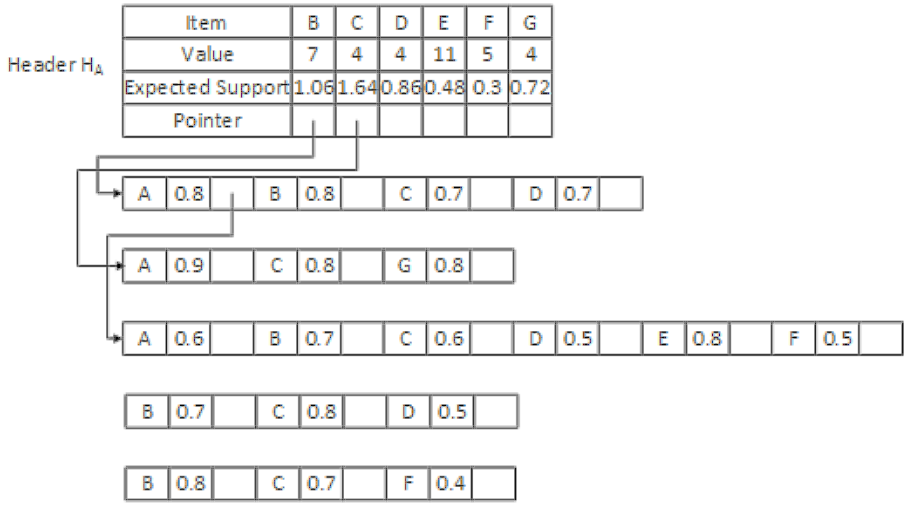

Fig. 2. UUH-Struct of Head Table of A

In this data structure, each item is assigned with its label, its value, its appearing probability and a pointer. The UUH-Struct of Table 3 and Table 4 is shown in Figure 1. After building the global UUH-Struct, the algorithm uses the depth-first strategy to build the head table in Figure 2 where A is the prefix. Then, the algorithm recursively builds the head tables where different itemsets are prefix and generates all the itemsets.

We use UUH-Mine framework to find all high utility itemsets from database through depth first search. However, this UUH-Mine framework needs to traverse all 2^n itemsets to find the result, so we must optimize it.

4 Optimization Strategies and Algorithms

In this section, we first introduce some optimization Strategies to improve the UUH-Mine framework. Then, we illustrate our UUIM algorithm, which is called Uncertain Utility Itemsets Mining algorithm.

4.1 Optimization Strategies

$Definition6$. For the given uncertain transaction T, its transaction maximum expected utility equals to the max expected utility of itemsets it contains, denoted as $MU(T)$:

$MU(T) = max\{EU(X)|X \subset T\}$

For example, the maximum expected utility of transaction 1 is 4.48, and the maximum expected utility of all 16 itemsets contained by transaction 1 is A, B, refer to Table 4.

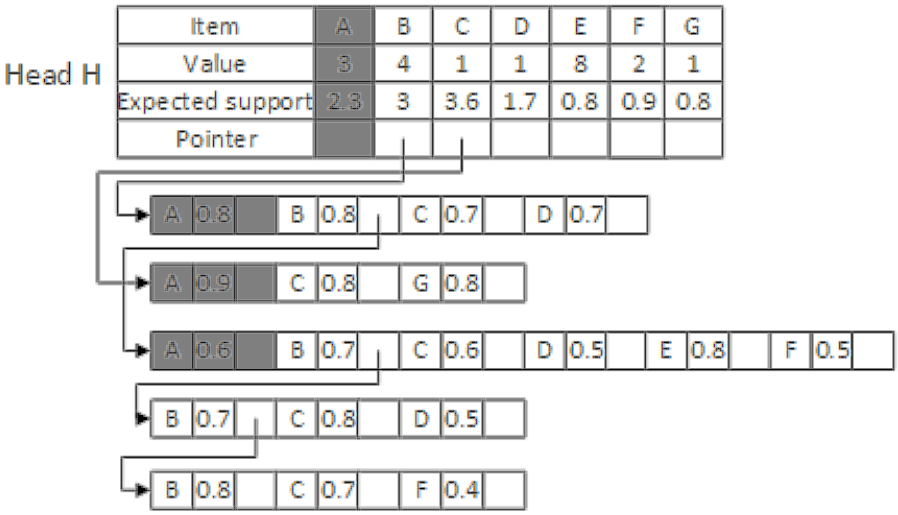

Fig. 3. UUH-struct after removing item A

Table 6. Uncertain Database with Maximum Transaction Expected Utility

TID	Transaction	MU
T_1	(A, 0.8) (B, 0.8) (C, 0.7) (D, 0.7)	4.48
T_2	(A, 0.9) (C, 0.8) (G, 0.8)	2.88
T_3	(A, 0.6) (B, 0.7) (C, 0.6) (D, 0.5) (E, 0.8) (F, 0.5)	6.72
T_4	(B, 0.7) (C, 0.8) (D, 0.5)	2.8
T_5	(B, 0.8) (C, 0.7) (F, 0.4)	3.2

In this example, the biggest transaction contains only 6 items, so we can calculate maximum expected utility of every transaction through exhaustive method. However, in practical problems, a transaction may contains many items, so the exhaustive method is not efficient. Here we will introduce a fast way:

Given a transaction T ~~which~~ length (the number of item it contains) is L, items in T is $\{i_1, i_2, ..., i_L\}$, probability of each item is $\{p_1, p_2, ..., p_L\}$. We can ~~give~~ the sub problem $S_{X,j}$ (X represent an itemset), which means the maximum expected utility in set which contains itemsets derived from itemset X and the last j items in T. Obviously, we have EU(T) = $S_{\phi,L}$ and $S_{I,0}$ = EU(X,T) as well as recursive relation:

$S_{X,j} = max\{S_{X\cup i_{L-j+1},j-1}, S_{X,j-1}\}$

We can get EU(T) from that recursive relation, but it still need to consider all 2^L itemsets. So we can optimize it by the theorem:

$Theorem1$. If there exist itemset X_1 and $X_2 = X_1 \cup \{i_j\}$ (X_2 represents a super-itemset which has one more item than X_1), and EU(X_1,T) ¿ EU(X_2,T), then expected utility of X_2 and all super-itemsets of X_2 cannot be the maximum expected utility of T.

$Rationale1$. The expected utility of X_2 obviously cannot be the maximum expected utility of T, so we will prove that is also true for X_2's super-itemsets $X_3 = X_2 \cup X'$. For one of X_2's super itemset, we can construct a new itemset $X_4 = X_1 \cup X'$, then we have:

$EU(X_3,T) = U(X_3)\times P(X_3,T) = (U(X_2)+U(X'))\times P(X_2,T)\times P(X',T) = (EU(X_2,T) + U(X')) \times P(X_2,T) \times P(X',T)$

$EU(X_4,T) = U(X_4)\times P(X_4,T) = (U(X_1)+U(X'))\times P(X_1,T)\times P(X',T) = (EU(X_1,T) + U(X')) \times P(X_1,T) \times P(X',T)$

Due to $EU(X_1,T) > EU(X_2,T)$ and $P(X_1,T) \geq P(X_2,T)$, we can know $EU(X_3,T) < EU(X_4,T)$, namely, the expected utility of X_3 cannot be the maximum expected utility of T.

$$S_{X,j} = \begin{cases} S_{X,j} = max\{S_{X\cup i_{L-j+1},j-1}, S_{X,j-1}\} & (EU(X,T) < EU(X\cup\{i_{L-j+1}\},T)) \\ S_{X,j-1} & (EU(X,T) \geq EU(X\cup\{i_{L-j+1}\},T)) \end{cases}$$

Thus, we can greatly speed up the calculation of transaction maximum expected utility.

$Definition7$. For itemset X and uncertain transaction database UD, the transaction maximum expected utility of itemset X is MU:

$MU(X) = \sum_{X\subseteq T\wedge T\in D} MU(T)$

which means the sum of transaction maximum expected ~~utility~~ of transactions containing itemset X.

We can derive Lemma 1 through definitions above:

$Lemma1$. The transaction maximum expected utilities of X's super-itemsets are not more than the transaction maximum expected utilities of X. For $X \subseteq X'$, always $MU(X') \geq MU(X)$.

According to Chernoff bound, suppose $X_1, X_2, ..., X_n$ be independent random variables taking values in {0,1}. Let X denote their sum and let $\mu = E[X]$ denote the sum's expected value. For any $\delta > 0$ it holds that

$Pr(X > (1+\delta)\mu) < (\frac{e^\delta}{(1+\delta)^{(1+\delta)}})^\mu$ ($\delta > 0$)

In the problem of mining probabilistic high utility itemsets, for one itemset X, ~~it's appear~~ in one uncertain transaction T can be seen as an independent Possion experiment, and the real support of X, i.e. the sup(X) is the sum of many Possion experiments, so the expect of that variable is expected suppprt count of X. The utility probability of X is:

$Pr(sup(X) \times U(X) \geq min\_util) = Pr(sup(X) \geq \frac{min\_util}{U(I)})$

When $esup(X) < min\_util/U(X)$, we can let $(1+\delta)esup(X) = min\_util/U(X)$, so we can get

$\delta = \frac{min\_util}{U(X)esup(X)} - 1 = \frac{min\_util}{EU(X)-1}$

$Pr(sup(X) \geq \frac{min\_util}{U(X)}) < (\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}})^{esup(X)}$

While $\delta = min\_util/EU(X) - 1$. The right side of inequality is a decreasing function of δ, so when δ decreases, the original inequality still holds. Hence we can get the following Lemma:

$Lemma2$. For itemset X, given uncertain transaction database D, utility threshold min_util and probabilistic utility threshold put , if $MU(X) < min\_util$ and $(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}})^{esup(X)} < put$, then itemset X and all of its super-itemsets cannot be utility.

Lemma 2 can greatly reduce the search space so that the algorithm can be efficient.

We can use it to optimize the UUH-mine framework mentioned in last section, as shown in Fig.4. Because the transaction maximum expected utility of itemset G is 2.88, cannot pass the check in Lemma 2, ~~so~~ G and all of its super itemsets cannot be high utility itemsets and we ~~dont~~ need to check them.

~~Of course,~~ we can use the same method to optimize the other header tables generated by projection databases (such as H_A, H_AB, etc).

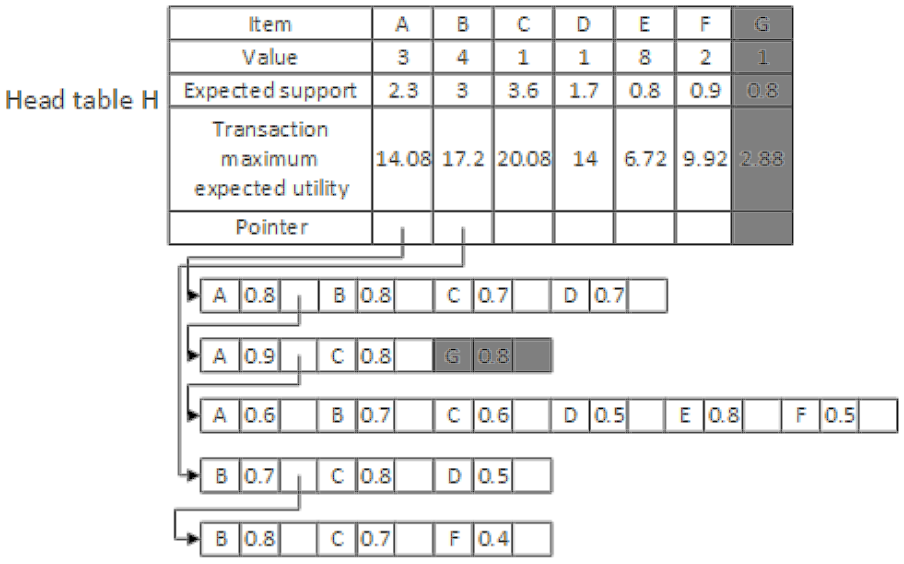

Fig. 4. UUH-mine data structure with optimization

4.2 UUIM Algorithm

In this section, we will introduce UUIM (Uncertain Utility Itemsets Mining) algorithm in detail. First, we will give the overall outline of this algorithm, which divides the mining process into two phases and briefly describe them. Then we will explain ~~that~~ two phases in detail.

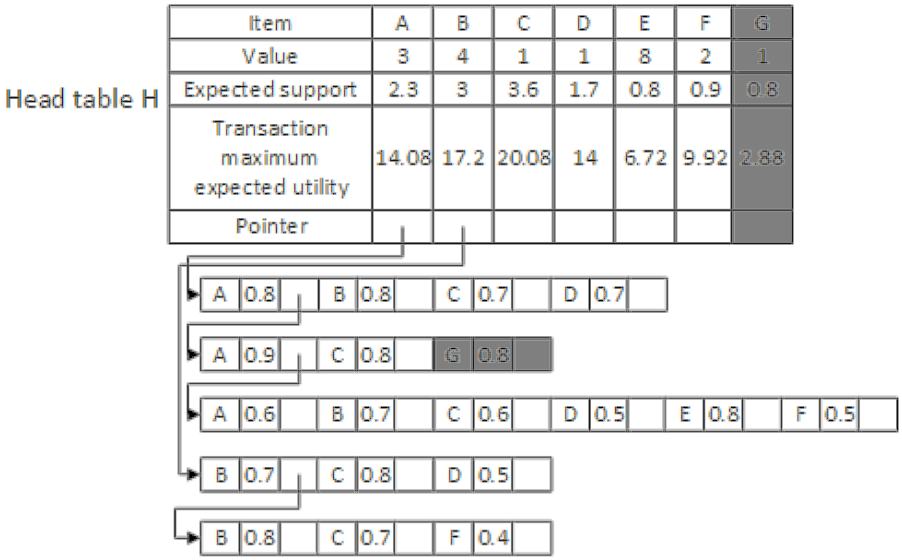

Fig. 5. UUH-mine data structure with optimization

Algorithm 1(UUIM) is the algorithm framework of mining utility itemsets. This algorithm ~~is purposed~~ to mine all utility itemsets UIS from the given uncertain transaction database UD through the given utility threshold min_util and probabilistic high utility threshold put. Line 1 is used to initialize the result set UIS; line 2 creates the initial header table H of the UUH-mine framework through function Initialize Header; line 3 use the key function Recursion search each items in depth first way; the last line return the calculation results which are all utility itemsets. According to this we can know that the main part of this algorithm consists of two parts, one for creating header table which is explained above in detail; the other is the recursive function Recursion which is used to traverse all probabilistic high utility itemsets.

Next, we will introduce how the key function Recursion works. In algorithm 2(Recursion), line 1 traverses each itemset in header table; line 2 and 3 check whether the new itemset is utility itemset ~~if~~ true it will be added to result; line 4 checks whether that new itemset can pass the $Pr(sup(X) \geq \frac{min\_util}{U(X)}) < (\frac{e^{\delta}}{(1+\delta)(1+\delta)})^{esup(X)}$, ~~if passed~~ a header table will be created in line 5 and traversed in line 6.

In summary, this section explains the probabilistic utility itemset mining algorithm, UUIM. First we introduce a new data structure UUH-struct, which ~~lay~~ the foundation

of the algorithms efficiency; then we describe several algorithm optimization methods such as global bound and local pruning in order to speed up the algorithm.

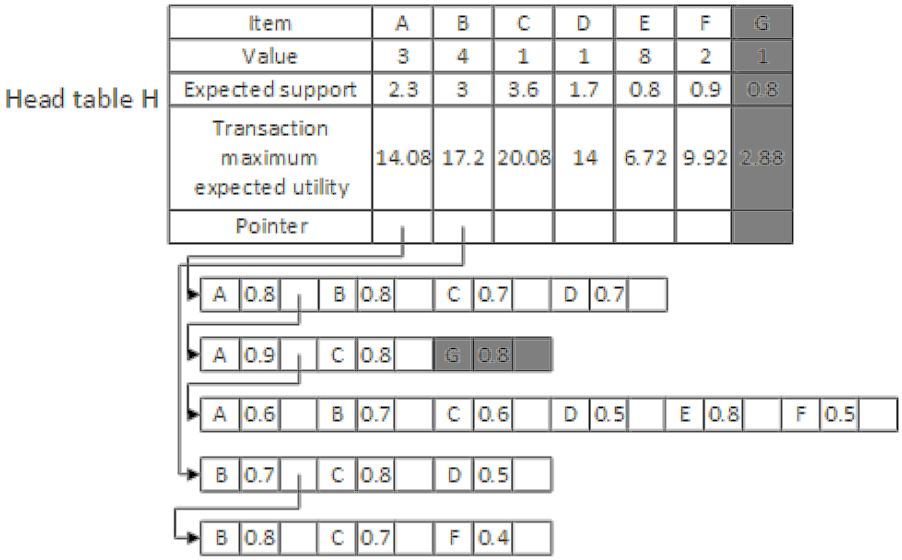

Fig. 6. UUH-mine data structure with optimization

5 Performance Evaluations

In this section, we will report and analyze our experiment results. All the experiments are ~~proceeded~~ on a PC with CPU Inter(R) Core(TM)i7-2600, frequency 3.40GHz, memory 8.00GB, hard disk 500GB. The Operation System is Microsoft Windows 7 Enterprise Edition. The development software is Microsoft Visual Studio 2010, using ~~language~~ C++ and its standard template library (STL).

6 Conclusion

In this paper, we formulate a new type of problem of mining uncertain frequent itemsets, called *mining probabilistic high utility itemsets in uncertain databases* (MPHU), where each item has an unit profit and likely appears multiple times in one transaction. In order to solve the *MPHU* problem, we propose a novel mining framework, called *UUIM*, which not only includes an efficient mining algorithm but also contains an effective pruning technique. Extensive experiments on both real and synthetic datasets verify the effectiveness and efficiency of proposed solutions.

References