Pushing Frequency Constraint to Utility Mining Model

Jing Wang, Ying Liu, Lin Zhou, Yong Shi, and Xingquan Zhu

Data Technology and Knowledge Economy Research Center, Chinese Academy of Sciences
Graduate University of Chinese Academy of Sciences
Beijing, China 100080
{jingw04, zhoulin05}@mails.gucas.ac.cn,
{yingliu, yshi}@gucas.ac.cn, xqzhu@cse.fau.edu

Abstract. Traditional association rules mining (ARM) only concerns the frequency of itemsets, which may not bring large amount of profit. Utility mining only focuses on itemsets with high utilities, but the number of rich-enough customers is limited. To overcome the weakness of the two models, we propose a novel model, called general utility mining, which takes both frequency and utility into consideration simultaneously. By adjusting the weight of the frequency factor or the utility factor, this model can meet the different preferences of different applications. It is flexible and practicable in a broad range of applications. We evaluate our proposed model on a real-world database. Experimental results demonstrate that the mining results are valuable in business decision making.

Keywords: general utility, utility mining, association rules mining, weighted association rules mining.

1 Introduction

Traditional association rules mining (ARM) [2] is to identify frequently occurring patterns of itemsets. ARM model treats all the items in the database equally by only considering if an item is present in a transaction or not. However, frequent itemsets may only contribute a small portion of the overall profit to the business and generate huge amount of inventory cost, labor cost, transportation cost.

In order to overcome the weakness of traditional association rules mining, utility mining model was proposed in [3]. Intuitively, utility is a quantitative measure of how "useful" (i. e. "profitable") an itemset is. The definition of utility of an itemset *X*, *u(X)*, is the sum of the utilities of *X* in all the transactions containing *X*.

Can we have a more general model which takes both frequency and utility into consideration simultaneously?

We propose a *general utility mining* model which is a linear combination of utility and frequency *gu*(*X*): $\lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{u(X)}{U}$, where $\frac{\sup(X)}{S}$ denotes the frequency of itemset *X* in the database, $\frac{u(X)}{U}$ denotes the fraction of the utility of itemset *X* out of total utility, λ is the weight of frequency, and (1-λ) is the weight of

Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489, pp. 685–692, 2007.

utility. A user specified threshold ε is used to measure the "usefulness" of itemset *X*. High utility itemsets with low supports may be filtered out by our model; popular itemsets that generate very low utility may also be filtered out.

Table 1 is an example transaction database where the total utility is 400. The number in each transaction in Table 1(a) is the sales volume of each item, and the subjective value of each item is listed in Table 1(b). For instance, let's set ε=15% and λ=0, *gu*({B,C,E}) = 0.18 > ε, {B,C,E} is a high utility itemset. Although {B,C,E} generates $72 profit, it occurs only once in the database, which may potentially incur overstocking problem. If we set λ=0.4, *gu*({B,C,E}) = 0.148 < ε, thus {B,C,E} is not interesting to the marketing professionals.

Table 1. A transaction database

(a) Transaction table.

ITEM TID	A	B	C	D	E
T_1	0	0	18	0	1
T_2	0	6	0	1	1
T_3	2	0	1	0	1
T_4	1	0	0	1	1
T_5	0	0	4	0	2
T_6	1	1	0	0	0
T_7	0	10	0	1	1
T_8	3	0	25	3	1
T_9	1	1	0	0	0
T_{10}	0	6	2	0	2

(c)The support and profit for all itemsets

Item-sets	Supp-ort	Profit ($)	Itemsets	Supp-ort	Profit ($)
A	5	24	BE	3	240
B	5	240	CD	1	43
C	5	50	CE	5	85
D	4	36	DE	4	56
E	4	50	ACD	1	52
AB	2	26	ACE	2	51
AC	2	41	ADE	2	46
AD	2	36	BCE	1	72
AE	3	33	BDE	2	182
BC	1	62	CDE	1	48
BD	2	172	ACDE	1	57

(b) Subjective value table. The right column displays the profit of each item per unit in dollars.

ITEM	PROFIT ($)(per unit)
A	3
B	10
C	1
D	6
E	5

(d)Transaction utility (TU) of the transaction database.

TID	TU	TID	TU
T_1	23	T_6	13
T_2	71	T_7	111
T_3	12	T_8	57
T_4	14	T_9	13
T_5	14	T_{10}	72

The difficulty of general utility mining is that the model does not follow "*downward closure property*" (*anti-monotone property*), that is, a high general utility itemset may consist of some low general utility sub-itemsets. Without this property, the number of candidates generated at each level increases exponentially. We push the frequency factor into Two-Phase algorithm proposed in [1], which maintains a *Transaction-weighted Downward Closure Property*. We apply our proposed general utility mining model on a real world database and the observations demonstrate the significance of general utility mining.

The rest of this paper is organized as follows. Section 2 overviews the related work. In Section 3, we introduce the technical terms in utility mining model. In Section 4, we propose the *general utility mining model*. Section 5 presents the experimental results and we summarize our work in Section 6.

2 Related Work

A number of ARM algorithms and optimizations have been proposed in the past ten years. The common assumption is that each item in a database is equal in weight and the sales quantity is 0 or 1. These algorithms exploit the “downward closure property” as disclosed in Apriori [2] (all subsets of a frequent itemset must be frequent).

Researches that assign different weights to items have been proposed in [4, 5, 6, 7]. These weighted ARM models are special cases of utility mining.

A utility mining algorithm is proposed in [9], which captures the semantic significance of itemsets at the transaction level. It focuses on mining the top-K high utility closed patterns that directly support a given business objective.

An alternative formal definition of utility mining and theoretical model was proposed in [3], where the utility is defined as the combination of objective information in each transaction and additional resources. Since this model cannot rely on “downward closure property” to restrict the number of itemsets to be examined, a heuristic is used to predict whether an itemset should be added to the candidate set.

An efficient utility mining algorithm, *Two-Phase algorithm* is proposed in [1]. It proposes the concept of “transaction-weighted utilization” and maintains “Transaction-weighted Downward Closure Property”. *Two-Phase algorithm* finds out itemsets with high transaction-weighted utilization first, and then find out itemsets with high utility. It is scalable and the memory cost as well as the computation cost is efficiently reduced.

3 Utility Mining

We start with the definition of a set of terms that leads to the formal definition of utility mining problem. The same terms are given in [1].

- $I = \{i_1, i_2, \ldots, i_m\}$ is a set of items.
- $D = \{T_1, T_2, \ldots, T_n\}$ *is* a transaction database where each transaction $T_i \in D$ is a subset of I.
- $o(i_p, T_q)$, *objective value*, represents the value of item i_p in transaction T_q.
- $s(i_p)$, *subjective value,* is the specific value assigned by a user to express the user’s preference.
- $u(i_p, T_q)$, *utility of an item i_p in transaction T_q,*, is defined as $o(i_p, T_q) \times s(i_p)$.

- *$u(X, T_q)$, utility of an itemset X in transaction T_q*, is defined as $\sum_{i_p \in X} u(i_p, T_q)$, where $X = \{i_1, i_2, \ldots, i_k\}$ is a k-itemset, $X \subseteq T_q$ and $1 \leq k \leq m$.
- *$u(X)$, utility of an itemset X,* is defined as $\sum_{T_q \in D \wedge X \subseteq T_q} u(X, T_q)$.
- *$tu(T_q)$, the transaction utility of transaction T_q*, is the sum of the utilities of all the items in T_q: $tu(T_q) = \sum_{i_p \in T_q} u(i_p, T_q)$.
- *$twu(X)$, the transaction-weighted utilization of an itemset X,* is the sum of the transaction utilities of all the transactions containing X: $twu(X) = \sum_{X \subseteq T_q \in D} tu(T_q)$.
- *$sup(X)$, the support count of an itemset X,* is the count of all the transactions containing X.
- *U, the total utility of all the transactions: $U = \sum_{T_q \in D} tu(T_q)$.*
- *S, the total number of transactions.*

X is *a high utility itemset* if $u(X) \geq \varepsilon$, where $X \subseteq I$ and ε is the minimum utility threshold, otherwise, it is *a low utility itemset.* For example $u(\{A, D, E\}) = u(\{A, D, E\}, T_4) + u(\{A, D, E\}, T_8) = 46$. If $\varepsilon = 120$, {A, D, E} is a low utility itemset.

4 General Utility Mining

In order to push frequency into utility mining model, we propose a general utility mining model which combines both frequency and utility linearly. We define *General Utility* and *General Transaction Utility*, and propose an extension to the Two-Phase algorithm in [1]. In addition, we discuss the universality and flexibility of this model.

4.1 Definitions and Theorems

Definition 1. (General Utility) The *General Utility of itemset X,* denoted as *gu(X),* is the linear combination of frequency and utility:

$$gu(X) = \lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{u(X)}{U} \quad \textbf{(3.1)}$$

where λ $(0 \leq \lambda \leq 1)$ is the weight assigned by users to adjust the contribution of frequency and utility. As $sup(X) \leq S$, $u(X) \leq U$, $0 \leq \frac{\sup(X)}{S} \leq 1$, $0 \leq \frac{u(X)}{U} \leq 1$, so $0 \leq gu(X) \leq 1$.

Definition 2. (High General Utility Itemset) For a given itemset *X, X* is a high general utility itemset if $gu(X) \geq \varepsilon$, where $\varepsilon (0 \leq \varepsilon \leq 1)$ is the minimum threshold.

Definition 3. (General Transaction-weighted Utilization) The *general transaction -weighted utilization of itemset X,* denoted as *tgu(X),* is the combination of transaction -weighted utilization and frequency: $tgu(X) = \lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{twu(X)}{U}$ **(3.2)**

Definition 4. (High General Transaction-weighted Utilization Itemset) For a given itemset *X, X* is a *high general transaction-weighted utilization itemset* if $tgu(X) \geq \varepsilon'$, where ε' $(0 \leq \varepsilon' \leq 1)$ is the minimum threshold.

Theorem 1. (General Transaction-weighted Downward Closure Property) Let I^k be a *k*-itemset and I^{k-1} be a (*k*-1)-itemset such that $I^{k-1} \subset I^k$. If I^k is a high general transaction-weighted utilization itemset, I^{k-1} must be a high general transaction-weighted utilization itemset.

Proof: Let T_{I^k} be the collection of the transactions containing I^k and $T_{I^{k-1}}$ be the collection containing I^{k-1}. Since $I^{k-1} \subset I^k$, $T_{I^{k-1}}$ is a superset of T_{I^k}. According to definition of *twu*(*X*) and *tu*(*X*), $twu(I^{k-1}) = \sum_{I^{k-1} \subseteq T_q \in D} tu(T_q) \geq \sum_{I^k \subseteq T_p \in D} tu(T_p) = twu(I^k)$. And the itemsets that contain I^k must contain I^{k-1}, so $sup(I^{k-1}) \geq sup(I^k)$. Thus we can get $tgu(I^{k-1}) = \lambda \frac{\sup(I^{k-1})}{S} + (1-\lambda) \frac{twu(I^{k-1})}{U} \geq \lambda \frac{\sup(I^k)}{S} + (1-\lambda) \frac{twu(I^k)}{U} = tgu(I^k) \geq \varepsilon'$.

The *General Transaction-weighted Downward Closure Property* indicates that only the combinations of high *general transaction-weighted* utilization (*k*-1)-itemsets could be added into the candidate set C_k at each level.

Theorem 2. Let *HGTWU* be the collection of all high *general transaction-weighted utilization itemsets* in a transaction database *D*, and *HGU* be the collection of high *general utility itemsets* in *D*. If $\varepsilon' = \varepsilon$, then $HGU \subseteq HGTWU$.

Proof: $\forall X \in HGU$, if *X* is a high *general utility itemset*, then

$$\varepsilon' = \varepsilon \leq gu(X) = \lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{u(X)}{U} = \lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{\sum_{X \subseteq T_q} u(X,T_q)}{U}$$

$$= \lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{\sum_{X \subseteq T_q}\sum_{i_p \in X} u(i_p,T_q)}{U} \leq \lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{\sum_{X \subseteq T_q}\sum_{i_p \in T_q} u(i_p,T_q)}{U}$$

$$= \lambda \frac{\sup(X)}{S} + (1-\lambda)\frac{twu(X)}{U} = tgu(X)$$

Thus, *X* is a high *general transaction-weighted utilization itemset* and $X \in HGTWU$.

4.2 Two-Phase Algorithm

According to the above two theorems, we can utilize the *General Transaction-weighted Downward Closure Property* in general transaction-weighted utilization mining in Phase I, assuming $\varepsilon' = \varepsilon$, and prune those overestimated itemsets in Phase II. (Note we use the new term *transaction-weighted utilization* to distinguish it from *utility*. The focus of this paper is not to propose this term, but to utilize the property of *transaction-weighted utilization* to help reduce the searching space in general utility mining.)

Phase I

Let's use the sample database in Table 1 to show how *general transaction-weighted utilization mining* model works. Assume the transaction-weighted utilization threshold

ε' =0.4 and λ=0.5. At level 1, $HGTWU_1$ ={{B}, {C}, {D}, {E}} and C_2 ={{B, C}, {B, D}, {B, E}, {C, D}, {C, E}, {D, E}}. After the second scan of database, *tgu* ({B, C}) =0.24<ε', *tgu* ({B, D}) =0.3275<ε', *tgu*({B, E})=0.4675, *tgu* ({C, D}) = =0.12125 < ε', *tgu* ({C, E}) = 0.4725 and *tgu* ({D, E}) = 0.51625. Thus, $HGTWU_2$ ={{B, E}, {C, E}, {D, E}}, and then C_3 = Φ. Candidate generation stops after the second database scan. The efficient candidate generation process results from the *General Transaction-weighted Downward Closure Property.*

Phase II

Based on Theorem 2, if we let ε'=ε, the complete set of high general utility itemsets is a subset of the high general transaction-weighted utilization itemsets discovered in phase I. In the above example, by scanning the database another time, we finally get *high general utility itemsets HGU* = {{B}, {B, E}}. The other five itemsets obtained in Phase I are pruned. Only three database scans are incurred in the whole process of Phase I and II.

4.3 Model Universality and Flexibility

Traditional association rules mining (ARM) [2] and utility mining [1, 3] can be viewed as special cases of our proposed general high utility mining model. Utility mining focuses on zone I and IV in Figure 1. ARM focuses on zone III and IV in Figure 1. Our general model focuses on the region above the straight line in Figure 1. The line is actually the visualization of formula 3.1.

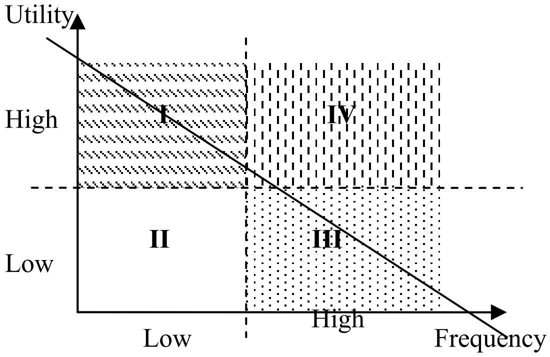

Fig. 1. High utility itemsets, frequent itemsets, high general utility itemsets

By adjusting λ, we can adjust the impact of frequency and utility in general utility. In Figure 1, when λ increases, frequency becomes more important in the application. When λ is 1, our model is the traditional association rules mining model. When λ decreases, utility becomes more important. When λ is 0, it is the utility mining model. By assigning different λ, users can mine different "useful" itemsets according to their own demands. For example, we set ε=0.4. When λ=0.2, *u*({B,E})=240, *sup*({B,E})=3, *gu*({B,E})=0.54>ε, {B,E} is a high general utility itemset; *u*({C,E})=85, *sup*({C,E})=5, *gu*({C,E})= 0.27<ε, {C,E} is not a high general utility itemset. However, if we set λ=0.8, *gu*({B,E})=0.36 < ε, it is not a high general utility itemset any more, but {C, E} becomes a high general utility itemset since *gu*({C,E})= 0.4425 > ε.

5 Experimental Evaluation

We evaluate our general utility mining model by using a real-world market data from a major grocery chain store in California, USA. It contains products of various categories, such as food, health care, gifts, and others. There are 1,112,949 transactions and 46,086 items in the database, and the total utility is 26,388,499.8 dollars. Each transaction consists of the products and the sales volume of each product purchased by a customer at a time point. The utility table describes the profit of each item. The size of this database is 73MByte. The average transaction length is 7.2. The subjective value table, which is the profit table, describes the profit of each product.

Table 2. Top 10 itemsets and corresponding support, utility and general utility when varying λ (ε =0.075%)

λ=0 (Utility Mining)			
Itemset	*Utility (%)*	*Support (%)*	*General Utility (%)*
39171, 39688	0.2435	0.3449	0.2435
39690, 39692	0.1942	0.028	0.1942
39182, 39206	0.1714	0.016	0.1714
39143, 39182	0.1631	0.0128	0.1631
5166, 16967	0.1606	0.227	0.1606
21283, 21308	0.1572	0.315	0.1572
16967, 16977	0.157	0.7009	0.157
21308, 22900	0.1528	0.2946	0.1528
10481, 16967	0.1296	0.1461	0.1296
16967, 21738	0.1083	0.3893	0.1083

λ=1 (ARM)			
Itemset	*Utility (%)*	*Support (%)*	*General Utility (%)*
16967, 16977	0.157	0.7009	0.7009
13743, 16967	0.0623	0.4967	0.4967
16967, 16975	0.0948	0.4033	0.4033
39430, 39432	0.031	0.399	0.399
16967, 21738	0.1083	0.3893	0.3893
3482, 3510	0.0763	0.382	0.382
16967, 16978	0.06	0.3784	0.3784
16967, 39684	0.0409	0.3733	0.3733
39171, 39688	0.2435	0.3449	0.3449
11780, 11783	0.0394	0.3404	0.3404

λ=0.1			
Itemset	*Utility (%)*	*Support (%)*	*General Utility (%)*
39171, 39688	0.2435	0.3449	0.2536
16967, 16977	0.157	0.7009	0.2114
39690, 39692	0.1942	0.028	0.1776
21283, 21308	0.1572	0.315	0.173
5166, 16967	0.1606	0.227	0.1672
21308, 22900	0.1528	0.2946	0.1669
39182, 39206	0.1714	0.016	0.1559
39143, 39182	0.1631	0.0128	0.1481
16967, 21738	0.1083	0.3893	0.1364
10481, 16967	0.1296	0.1461	0.1312

λ=0.5			
Itemset	*Utility (%)*	*Support (%)*	*General Utility (%)*
16967, 16977	0.157	0.7009	0.429
39171, 39688	0.2435	0.3449	0.2942
13743, 16967	0.0623	0.4967	0.2795
16967, 16975	0.0948	0.4033	0.249
16967, 21738	0.1083	0.3893	0.2488
21283, 21308	0.1572	0.315	0.2361
3482, 3510	0.0763	0.382	0.2292
21308, 22900	0.1528	0.2946	0.2237
16967, 16978	0.06	0.3784	0.2192
39430, 39432	0.031	0.399	0.215

We compare the itemsets discovered from general utility algorithm, Apriori and utility mining by varying the value of λ and the threshold ε. Table 2 shows the top 10 itemsets when ε=0.075%. (We only show itemsets longer than 1.) From Table 2, we can observe different top 10 high general utility itemsets when varying λ. For example, itemset {39690, 39692} is in the top 10 high general utility itemsets when λ=0.1 (assigning more weight to the utility factor), but left out when λ=0.5 (assigning equal weight to the utility and the frequency factor) due to its large utility 0.1942% but small support 0.028%. Itemset {39430, 39432} (utility = 0.031%, support = 0.399%) is just the opposite case. {21283, 21308} is in the top 10 itemsets when λ=0.5, but it is not in the top 10 frequent itemsets. It shows that the itemsets discovered by our proposed model are different with those by ARM or utility mining in many cases, more emphasizing the balance between frequency and utility.

6 Conclusions

General utility mining is a generalization of association rules mining (ARM) and utility mining. It balances the impact of frequency and utility by adjusting their weights, respectively. ARM and utility mining are two special cases of this model. General utility mining can overcome their weakness. It has a high universality and flexibility. We defined a term called *general utility* and *general transaction-weighted utilization* model which holds *Transaction-weighted Downward Closure Property*. We proposed a Two-Phase algorithm that can discover high general utility itemsets highly efficiently. A real data set from a chain grocery store was used to evaluate our proposed model and the experimental results showed that it could find itemsets that are missed by utility mining model and ARM. Our model can be applied in a broad range of applications, such as, business intelligence, web log mining, etc.

Acknowledgements

This research has been partially supported by a grant from National Natural Science Foundation of China (#70621001, #70531040, #70501030, #70472074, #60674109), 973 Project #2004CB720103, Ministry of Science and Technology, China, and BHP Billiton Co., Australia.

References

1. Ying Liu, Wei-keng Liao and Alok Choudhary: A Fast High Utility Itemsets Mining Algorithm. Utility-Based Data Mining Workshop with the 11th SIGKDD, 2005.
2. Agrawal and R. Srikant: Fast algorithms for mining association rules. 20th VLDB (1994)
3. Hong Yao, Howard J. Hamilton, and Cory J. Butz: A Foundational Approach to Mining Itemset Utilities from Databases. SDM (2004)
4. C.H. Cai, Ada W.C. Fu, C.H. Cheng, and W.W. Kwong: Mining Association Rules with Weighted Items. IDEAS (1998)
5. W. Wang, J. Yang, and P. Yu: Efficient Mining of Weighted Association Rules (WAR). 6th KDD (2000)
6. Feng Tao, Fionn Murtagh, and Mohsen Farid: Weighted Association Rule Mining using Weighted Support and Significance Framework. 9th KDD (2003)
7. S. Lu, H. Hu, and F. Li: Mining weighted association rules. Intelligent Data Analysis, 5(3) (2001), 211-225
8. B. Barber and H.J.Hamilton: Extracting share frequent itemsets with infrequent subsets. Data Mining and Knowledge Discovery, 7(2) (2003), 153-185
9. Raymond Chan, Qiang Yang, Yi-Dong Shen: Mining high utility Itemsets. ICDM (2003)
10. IBM data generator, http://www.almaden.ibm.com/software/quest/Resources/index.shtml