Study of High Motivation Itemsets Mining

Zhou Xian-shan1
College of computer science and technology Yangtze University
Jingzhou, China
E-mail:xszhou@yangtzeu.edu.cn

Wang Liang2
Information Center of Jingzhou Municipal Bureau of Environmental Protection Jingzhou, China
E-mail: wyh_0209@126.com

Yu Guang-zhu 3
College of Information Science and Technology Donghua University
Shanghai, China
E-mail:ygz@mail.dhu.edu.cn

Abstract—algorithms for support-based association rules mining can only discover frequent itemsets, can not discover the non-frequent itemsets with high utility values; utility-based association rules mining aims at discovering high utility itemsets, without considering the itemsets whose utility values are not high but the product of the support and utility of the same itemset is very large. To solve the problem, we propose a new measure, i.e., motivation, to measure the importance of an itemset and a down-top algorithm called HM-Two-Phase-Miner to discover high motivation itemsets. Motivation integrates the advantages of support and utility, and thus can reflect both the semantic significance and statistical significance of an itemset. In HM-Two-Phase-Miner algorithm, transaction-weighted motivation downward closure property is adopted to cut down the search space.

Key Words-high motivation itemset, association rule, support, utility-based, pruning strategy

I. INTRODUCTIONS

Support-based association rules mining algorithms[1][2][3][4] use support to measure the user's interest, it can not discover the non-frequent itemsets with high utility values, and thus it will result in the loss of some useful knowledge. For example, in the transaction database, some itemsets with a low support and a high utility can bring more business profits and arouse more business interests than those with a high support and a low utility. instead of support, Utility-based association rules mining(UMARM)[5][6][7] use utility to evaluate the importance of the itemsets, it can't discover the itemsets whose utility values are not high but the product of the support and utility of the same itemset is very large. Although the utility of these itemsets is not high, they may probably cause the user's interest. Compared to those itemsets with high-utility but low-support, the itemsets with high motivation always means a more safe and secure decision scheme: the possibility of success of the decision is very high, although the success may not bring many benefits. In real life, most people still prefer the safer scheme, while being cautious about those schemes (e.g. buy sports lottery) which will get a high profit but the success rate is very low.

The most important reason resulting in the loss of useful knowledge is that the two kinds of association rules mentioned above made too simple assumptions on the issues of measuring user's interest: the support-based association rule supposes that users only interest in the frequent itemsets while the utility-based association rule supposes that users only interest in the high-utility itemsets. In fact, the factors that determine people's interest exist in both subjective and objective aspects. Support, as an objective measurement, can not reflect the semantic characteristics of itemsets; Utility, as a subjective measurement, can not reflect the statistical properties of itemsets[8]. As a result, simple support-based or utility-based association rules can't express the user's interests accurately.

Expectancy Theory[9] said that the motivation is the process of evaluation and selection, the driving force(or motivation) people to take an action with depends on the value of their evaluation of the operation results(valence) and the estimation of the expected possibility of the achieving goals(expectancy). That means, the value of motivation depends on the product of valence and expectancy.

Motivation = Valence × Expectancy (1)

Therefore, the key points which determine people's interests in a specific itemset should at least including the support (correspond with expectancy) and utility (correspond with valence) of the itemset. According to formula (1), we use the product of support and utility which we define as motivation to reflect the statistical and semantic characteristics of itemset, and propose a high-motivation itemsets mining algorithm.

II. CONCEPTS AND DEFINITIONS

Assume that I={i1, i2, …, im} is a set of items, T={t1, t2, …, tn} is a transaction database, each transaction t ( t ∈ T ) is a subset of I, which means t ⊆ I. Suppose S is a subset of I, if every item in S is also contained in t, i.e., S ⊆ t, we say t supports or contains S. According to the concepts of references[5][6][7], the definitions are given out as bellow:

Definition 1: the set of transactions of itemset S (transaction-set of S), denoted as T S , is the set of all the transactions that contains S, i.e.,

\[ T_S = \{ t_t \mid S \subseteq t_t, t_t \in T \} \] (2)

Apparantly, if \( S_1 \supseteq S_2 \), then \( T_{S_1} \supseteq T_{S_2} \).

Definition 2: the utility of item i, in transaction t, (The transaction-utility of item i, denoted as \( I(t_t, i) \), is the utility brought on by item i when transaction t occurs. To facilitate the understanding, the utility mentioned within the paper all means the economic utility. In transaction database, the transaction-utility of the item means the product of the item’s unit profit and sales-amount in t.)
Definition 3: the utility of itemset S in transaction t_q (The transaction-utility of itemset S), denoted as \( I(S, t_q) \), is the sum of transaction-utility of item i_p contained in S, i.e.,
\[
I(S, t_q) = \sum_{i_p \in S} I(i_p, t_q)
\]  
(3)

When \( S = t_q \), we call it as the utility of transaction t_q (Transaction-utility of t_q) for short, recorded as \( I(t_q, t_q) \). Apparentlly, according to the definition, there is the formula (4) as blow:
\[
I(S, t_q) \leq I(t_q, t_q) \quad (S \subseteq t_q)
\]  
(4)

Definition 4: the utility of itemset S, denoted as \( u(S) \), is the sum of all the transaction-utility of itemset S, i.e.,
\[
u(S) = \sum_{t_i \in T} I(S, t_i)
\]  
(5)

Definition 5: the motivation of itemset S, denoted as \( m(S) \), is the product of the support and utility of the itemset, i.e.,
\[
m(S) = s(S) \cdot u(S)
\]  
(6)

If the motivation of an itemset is not smaller than the threshold (min-motivation) defined by users, we say that the itemset is a high-motivation itemset. Otherwise, we say this itemset is a low-motivation itemset. Our goal is to find all the high-motivation itemsets.

Definition 6: the transaction weighted utilization of itemset S, recorded as \( twu(S) \), is the sum of the utility of all the transactions that contains itemset S, which is showed as below:
\[
twu(S) = \sum_{t_i \in t_q} I(t_q, t_q)
\]  
(7)

If the transaction weighted utilization of itemset S, i.e., \( twu(S) \), is not smaller than the threshold Twminmotivation defined by users, then this itemset is a high transaction weighted utilization itemset, otherwise this itemset is a low transaction weighted utilization itemset. Apparently, \( twu(S) \geq \mu(S) \).

Definition 7: the transaction weighted motivation of itemset S, recorded as \( twm(S) \), is the product of the transaction weighted utilization and support of itemset S, i.e.,
\[
twm(S) = twu(S) \cdot s(S)
\]  
(8)

If \( twm(S) \) is not smaller than the threshold (TWminmotivation) defined by users, then this itemset is a high transaction weighted motivation itemset.

III. RELATED RESEARCH

Yi Dong Shen had proposed a Goal-oriented utility-based association rule mining model (OOA model) \[10\]. OOA model uses both support and utility to measure the importance of the specific itemset, and can discover the high utility frequent itemset. But the OOA model and related OOApriori algorithm has many differences with ours: (1) OOA model’s association rules do not require that the product of support and utility value is greater than or equal to a threshold; (2) In OOA Model, the support threshold minsup should be set to a higher value; otherwise will cause a lot of frequent itemsets. Therefore, OOA model will still lose some patterns with low support but high motivation.

Different from support-based association rule mining and utility-based association rule mining, which use support threshold or utility threshold to narrow search space, motivation-based association rule mining use motivation threshold (minmotivation) to prune off those unimportant rules. But in tests, we determine the value of minmotivation in terms of the value of minsup and minutil. Since the itemsets which meet both of the minsup and minutil constraint are very rare (as showd in Fig 1, 2), minsup and minutil can set to a relatively small value. Of course, the algorithm can also use the threshold minsup and minutil to filter out itemsets.

Reference \[10\] proposed to use “general utility” to measure the importance of itemsets. According to the definition, the general utility of itemset S is equal to the weighted-sum of its support and utility, denoted as \( gu(S) = I(S) + (1 - I(S))u(S) \). “General Utility” does reflect the semantic characteristic and statistical characteristic of itemsets, but the weight value \( \lambda \) is rather arbitrary, and its concept is not as intuitive as the motivation. Based on the probability theory and management science, the concept of Motivation is easier to understand.

Reference \[10\] proposed a utility-based association rules mining algorithm Two-phase. Just as the other utility-based mining algorithms, Two-phase will lose some high-motivation itemsets. However, the downward closure property of transaction weighted utility brought out by this algorithm gives the foundation of our research.

IV. ALGORITHM

A. Characteristics of Motivation

Related research shows that the utility constraint is neither monotone, anti-monotone, convertible, nor succinct. According to the definition of motivation, the motivation-constraint is neither monotone, anti-monotone, convertible, nor succinct.

Theorem 1 (transaction-weighted utility downward closure property): Assume that \( S^k \) is a k-itemset, \( S^{k-1} \) is a (k-1) itemset, and \( S^{k-1} \subseteq S^k \). If \( S^k \) is a high transaction-weighted utility itemset, then \( S^{k-1} \) is also a high transaction-weighted utility itemset.

Proof: Assume that \( T^k \) is the collection of all the transactions which contain itemset \( S^k \), and \( T^{k-1} \) is the collection of all the transactions which contain itemset \( S^{k-1} \). Because \( S^{k-1} \subseteq S^k \), then \( T^{k-1} \) is one of the superset of \( T^k \). According to Definition 6 (Formula 7), there is:
\[
twu(S^{k-1}) = \sum_{t_i \in T^{k-1}} I(t_q, t_q) \geq \sum_{t_i \in T^k} I(t_q, t_q) = twu(S^k)
\]

Theorem 2 (transaction-weighted motivation downward closure property): Assume that \( S^k \) is a k-itemset, \( S^{k-1} \) is a (k-1) itemset, and \( S^{k-1} \subseteq S^k \). If \( S^k \) is a high transaction-weighted motivation itemset, then \( S^{k-1} \) is also a high transaction-weighted motivation itemset.

Proof: According to Theorem 1, there is
\[
twu(S^{k-1}) \geq twu(S^k), \text{ and because of } s(S^{k-1}) \geq s(S^k), \text{ there is:}
\]
\[
twu(S^{k-1}) \cdot s(S^{k-1}) \geq twu(S^k) \cdot s(S^k)
\]

if \( twu(S^k) \cdot s(S) \geq TW \text{ min motivation} \cdot \text{ then there is}
\[
twu(S^{k-1}) \cdot s(S^{k-1}) \geq TW \text{ min motivation}.
\]

Theorem 3: Assume that HTWM is the collection of all the high transaction-weighted motivation itemsets in database T, HM is the collection of all the high motivation itemsets in database T. If
The candidate set of high transaction-weighted motivation k-itemsets by the concatenation operation of the (k-1)-itemsets in $C_{k-1}^{HTWM}$. The 10th step add all of candidate sets with different size into $C_k^{HTWM}$. In the 11th step, the function CalculateAndDiscoverHTWM() calculates the motivation of each candidate and discover the high transaction-weighted motivation k-itemsets in $C_k^{HTWM}$, forming the candidates which will be used to generate $C_{k+1}^{HTWM}$ in the next steps. The $C_k^{HTWM}$ generated in 9th step includes all the high motivation k-itemsets, and may also include those k-itemsets with lower transaction-weighted motivation. In order to cut down the search space, we should get rid of those k-itemsets with lower transaction-weighted motivation as soon as possible. So when we get the $C_k^{HTWM}$, we scan the database again in 13th step to calculate the real motivation of the itemsets in $C_k^{HTWM}$.

The structure of HM-Two-Phase-Miner and Two-Phase is similar with Apriori, but the differences are: (1) different pruning strategy. Apriori use the downward closure property of frequent itemsets to cut down search space, while Two-Phase use the transaction-weighted utility downward closure property to achieve the same goal; (2) During the process of generating the candidate-set of k-itemsets, Apriori generate $C_k$ from only large (k-1)-itemsets ($L_{k-1}$) (by concatenation operation), while HM-Two-Phase-Miner and Two-Phase generate a set of k-itemsets from (k-1)-itemsets which must be in a candidate-set, which means, to use older candidate-set to generate new candidate-set; (3) Compare to Apriori, HM-Two-Phase-Miner and Two-Phase need to scan the database once more to calculate the real motivation of each candidate, which will increase the computational complexity. But the experiments show that, due to the effective pruning strategy, the performance of the algorithm is good.

V. EXPERIMENT AND RESULT ANALYSIS

The experiment is running on the Lang Shao Xeon server, CPU’s frequency is 2.4G, Memory is 4G, running Windows 2003, the programs is written in Delphi 7. The experimental dataset is T10.I6.D1000K and T20.I6.D1000K, the number of items is 1K, produced by IBM Synthetic Data Generator[11]. The dataset only consists of 0 and 1, respectively representing whether the item appears in the transaction, without utility value. Therefore, in the experiment, we use delphi’s random function “RandG” to generate the random value (Gaussian distribution) to simulate the transaction's number (TID MOD 100) to simulate the sales amount. Thus, the utility of an item in a transaction is equivalent to the product of the sale amount and the unit utility of the item.

In the experiment, for ease of understanding the meaning and origin of the motivation threshold, we assume that minutil is equal to minsup. For example, minmotivation=0.0025 means minutil=minsup=0.05. In fact, the itemsets satisfying the requirement that both support and utility are greater than 0.05 are relatively seldom. So 0.0025 is a big threshold to minmotivation, and this is a great difference with the range of support and utility threshold. Figure 1 show the influence which the changes of transactions do to the algorithm performance. Because HM-Two-Phase-Miner needs to scan database multiple times, and may increase the number of candidates, when the number of transactions increase, HM-Two-Phase-Miner algorithm needs to run longer. In figure 2, minmotivation changes from 0.000004 to 0.0025.
larger the minmotivation, the less the itemsets satisfying the threshold, the shorter the running time.

VI. CONCLUSION

This paper analyzes the deficiencies of support and utility in measuring the importance of itemsets, and proposes a new interest measurement: Motivation. Motivation integrates the advantages of support and utility, and thus can reflect both the semantic significance and statistical significance of an itemset, which meet the people's decision-making habits. This paper also proves the existence of transaction-weighted motivation downward closure property, and uses this property in the new HM-Two-Phase-Miner algorithm to narrow search space. Experiments on synthetic data show that the HM-Two-Phase-Miner algorithm can get a good performance with the short-pattern datasets.

Reference