Definition 3: the utility of itemset S in transaction t_q (The transaction-utility of itemset S), denoted as $l(S,t_q)$, is the sum of transaction-utility of item i_p contained in S, i.e.,

$$l(S,t_q)=\sum_{i_p\in S} l(i_p,t_q) \tag{3}$$

When S=t_q, we call it as the utility of transaction t_q (Transaction-utility of t_q) for short, recorded as $l(t_q,t_q)=\sum_{i_p\in t_q} l(i_p,t_q)$. Apparently, according to the definition, there is the formula (4) as blow:

$$l(S,t_q)\le l(t_q,t_q)\ \ (S\subseteq t_q) \tag{4}$$

Definition 4: the utility of itemset S, denoted as $u(S)$, is the sum of all the transaction-utility of itemset S, i.e.,

$$u(S)=\sum_{t_q\in T_S} l(S,t_q) \tag{5}$$

Definition 5: the motivation of itemset S, denoted as $m(S)$, is the product of the support and utility of the itemset, i.e.,

$$m(S)=s(S)*u(S) \tag{6}$$

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 transaction that contains itemset S, which is showed as below:

$$twu(S)=\sum_{tq\in T_S} l(t_q,t_q) \tag{7}$$

If the transaction weighted utilization of itemset S, i.e., twu(S), is not smaller than the threshold TWminutil 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)≥ u(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.,

$$twm(S)=twu(S)*s(S) \tag{8}$$

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) [3]. 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 $gu(S)=\lambda s(S)+(1-\lambda)u(S)$. "General Utility" does reflect the semantic characteristic and statistical characteristic of itemsets, but the weight value λ 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 [6] 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 $S^{k-1}\subset S^k$. 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_{S^k} is the collection of all the transactions which contain itemset S^k, and $T_{S^{k-1}}$ is the collection of all the transactions which contain itemset S^{k-1}. Because $S^{k-1}\subset S^k$, then $T_{S^{k-1}}$ is one of the superset of T_{S^k}. According to Definition 6 (Formula 7), there is:

$$twu(S^{k-1})=\sum_{t_q\in T_{S^{k-1}}} l(t_q,t_q)\ge \sum_{t_q\in T_{S^k}} l(t_q,t_q)$$

$$=twu(S^k)\ge TW\min util$$

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 $S^{k-1}\subset S^k$. 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 $twu(S^{k-1})\ge twu(S^k)$, and because of s(S^{k-1})≥s(S^k), there is :

$$twu(S^{k-1})*s(S^{k-1})\ge twu(S^k)*s(S) \tag{9}$$

if $twu(S^k)*s(S)\ge TW\min motivation$, then thers is $twu(S^{k-1})*s(S^{k-1})\ge TW\min motivation$.

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

TWminmotivation is equal to minmotivation, then there is $HM \subseteq HTWM$.

Proof: $\forall S \in HM$, If S is a high motivation itemset, then there is :

$$TW\min motivation = \min motivation \le s(S)*u(S)$$

$$= s(S)*\sum_{t_q \in T_S} l(S,t_q) \le s(S)*\sum_{t_q \in T_S} l(t_q,t_q)$$

$$= s(S)*twu(S) = twm(S)$$

In this way, through setting TWminmotivation=minmotivation, according to Theorem 3, we can use transaction-weighted motivation downward closure property to cut down the search space.

B. Algorithm

Based on the pruning strategy described above, we proposed a new algorithm which is similar to "Two-Phase", we call it HM-Two-Phase-Miner. HM-Two-Phase-Miner algorithm adopt the down-top searching strategy, repeatedly generates k-itemsets from (k-1)-itemsets, and calculate the motivation of each candidates. The description of algorithm is showed in Table 1:

TABLE 1. HM-TWO-PHASE-MINER ALGORITHM

```
Algorithm Name : HM-Two-Phase-Miner
Input : Database T, Threshold minmotivation
Output : set of high motivation itemsets HM
1.  {
2.  $C_k^{HTWM} = \phi$;   $C^{HTWM} = \phi$ ;   // $C_k^{HTWM}$ is the candidate set of high transaction-weighted motivation k-itemsets, k is the size of itemset; $C^{HTWM}$ is the candidate set of high transaction-weighted motivation itemsets. 1 is the largest size of high transaction-weighted motivation itemsets.
3.  $HM = \phi$;  // set of high motivation itemsets
4.  k =1;
5.  Scan Database T，get $C_1^{HTWM}$ ;
6.  While (| $C_k^{HTWM}$ | > 0)
7.    {
8.     k=k+1;
9.     $C_k^{HTWM}$ =Generate( $C_{k-1}^{HTWM}$ );
10.    $C^{HTWM} = C_k^{HTWM} \bigcup C^{HTWM}$ ;
11.    $C_k^{HTWM}$ =CalculateAndDiscoverHTWM( $C_k^{HTWM}$ , T, TWminmotivation);
12.   }
13. HM=HM ∪ CalculateAndDiscoverHM( $C^{HTWM}$ , T, minmotivation);
14. Return HM;
15. }
```

The first step to the 4th steps of the HM-Two-Phase-Miner algorithm is initialization. The 5th step scanning the database to get the high transaction-weighted motivation itemsets' candidate set C_1^{HTWM}. The 7th step to 12th step repeatedly scan the database to generate the candidate sets with different length. Among the process, in the 9th step, the function Generate () generates 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^{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^{HTWM}, we scan the database again in 13th step to calculate the real motivation of the itemsets in C^{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, and HM-Two-Phase-Miner use transaction-weighted motivation 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 Chao 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 appeared 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 unit utility of each item in the transaction, and use the mod 100 operation of 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. The larger the minmotivation, the less the itemsets satisfying the threshold, the shorter the running time.

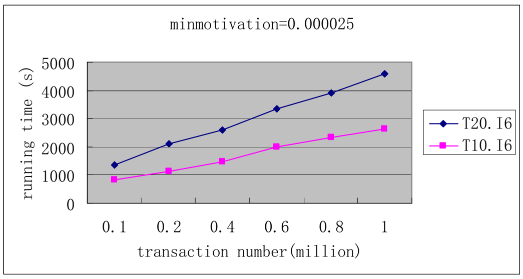

Figure 1 : The influence of the change of transaction number on algorithm’s performance

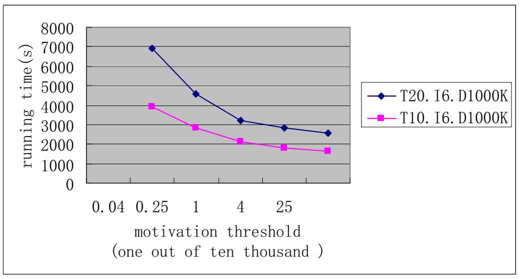

Figure 2 : The influence of the change of motivation on algorithm’s performance

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

[1] Rules [C]. In: Proc of 1994 Int’1Conf of Very Large Data Base. Santiago, Chili: VLDB Endowment, 1994, 487-499.

[2] Lu S.F., Hu H.P. and Li F. Mining weighted association rules [J]. Intelligent Data Analysis, 2001, 5: 211-225

[3] Shen Y. D., Zhang Z. and Yang Q. Objective-oriented utility-based association mining [C]. Proceedings of the 2002 IEEE International Conference on Data Mining, 2002, 426-433

[4] Roberto J. Bayardo, Jr , Rakesh Agrawal and Dimitrios Gunopulos, Constraint-Based Rule Mining in Large, Dense Databases [R], Data Mining and Knowledge Discovery, 2000, 4(2-3): 217-240

[5] Yao H. and Hamilton H.J. Mining itemset utilities from transaction databases [J]. Data & Knowledge Engineering, 2006, 59: 603 -626

[6] Liu Y., Liao W.K. and Choudhary. A fast high utility itemsets mining algorithm [C]. Proceedings of the First International Workshop on Utiliy-based Data Mining, 2005: 90-99

[7] Guangzhu Yu, Shihuang Shao, Bin Luo and Xianhui Zeng, A Hybrid Method for High-utility Itemsets Mining in Large High-dimensional Data, International Journal of Data Warehousing and Mining, 2009, 5(1): 57-73.

[8] Lqiang Geng and Howard J. Hamilton, Interestingness Measures for Data Mining: A Survey [J]. ACM Computing Surveys (CSUR), 2006, 38 (3): 61-93

[9] V. H. Vroom. Work and Motivation [M]. John Wiley, 1964

[10] Jing Wang, Ying Liu, Lin Zhou, Yong Shi and Xingquan Zhu, Pushing Frequency Constraint to Utility Mining Model [C], Lecture Notes in Computer Science, proceedings of international conference on computational science (ICCS), 2007, 685-692

[11] 2010/4/20. http://www.almaden.ibm.com/cs/projects/iis/hdb/Projects/data_mining/datasets/syndata.html