A Parallel and Distributed Method to mine Erasable Itemsets from High utility patterns

Ms. Ruchi Patel
Assistant Professor, Department of Information Technology
Gyan Ganga Institute of Technology and Sciences, Jabalpur

Abstract

High utility pattern mining becomes a very important research issue in data mining by considering the non-binary frequency values of items in transactions and different profit values for each item. These profit values can be computed efficiently in order to determine the gain of an itemset which in turn will help in production planning of any company. This gain value is needed to prune some of the irrelevant items from the high utility patterns at the time of economic crisis through erasable itemset mining. But all of the existing erasable itemset mining algorithms are based on centralized database and today’s internet era databases are large and inherently distributed. This inherent distribution source of data and the voluminous in size emerges to develop scalable parallel and distributed approach for erasable itemset mining. This paper proposes a parallel method in distributed environment through which the erasable itemsets can be mined from the high utility patterns without which the loss of the profit is no more than the given threshold. This algorithm is designed in such a way so that it can efficiently compute the gain of an itemset as well as prune the irrelevant data or items from the high utility patterns and decrease the execution time.

Keywords: Data mining, Distributed Database, Erasable itemsets, High utility patterns.

1. Introduction

Mining frequent itemsets from a transaction database is a fundamental task for knowledge discovery such as association rules, sequential patterns and classification. In the past, numerous methods were proposed to discover frequent itemsets. Among them, the most two famous kinds were level-wise algorithms and pattern-growth methods. These approaches, however, only considered whether an item was bought in a transaction or not. Thus, frequent itemsets just reveal the frequency of occurrence of the itemsets, but do not reflect any other factors, such as price or profit. Thus, frequent pattern mining has following 2 limitations:

1. First it treats all items with the same importance/weight/price.
2. Second, in one transaction each item appears in a binary (0/1) form, i.e. either present or absent.

In the real world, however, each item in the supermarket has a different importance price and one customer can buy multiple copies of an item. This gives motivation to develop a mining model to discover itemsets, which contributes to business organization with high profit. Recently, a Utility Mining Model (UMM) was defined to solve limitations of frequent pattern mining. This model allows users to express their preference or expectations regarding each item in terms of weight or utility values, and find patterns above the user specified minimum utility threshold.

In some situations, frequent itemsets may only contribute a small portion to the overall profit, while non-frequent ones may contribute a large portion to the profit. For example, the sale of diamonds may occur less frequently than that of clothing in department store, but the former gives a much higher profit per unit sold than the latter. Only frequency is thus not sufficient to identify the items which are highly profitable or have other potential effects.

However, high profit items are always purchased rarely. If we just consider the purchased frequencies of itemsets, then high profit itemsets may not be discovered. For example, the profit of television is much higher than milk, but the purchased frequency of television is much less than milk.

Nevertheless, the profits for items should be related to the purchased quantities of the items. If purchased quantity for a low profit item is large, then the total profit for the item will increase. Hence, both profits and purchased quantities for items should be considered.

Up till now the high utility patterns or itemsets have been mined from transactional database to extract more useful and semantically significant knowledge. These high utility patterns have high utility/gain value that helps to know the profit of
any company. But at time when financial crisis is coming, the Company should carefully plan their production because it has not enough money to purchase all needed components or items as usual. For the sake of commercial interests, the loss of the company’s profit caused by stopping manufacturing some products should be controllable. Hence, the key to the problem is how to efficiently find these components, without which the loss of the profit is no more than the given threshold. These components are also called as erasable itemsets.

Erasable itemsets are those itemsets that without these items, the loss of profits does not exceed ξ percents of the original profits. For example, let ξ be 10%. {i6, i7} is an erasable. If a manufactory does not purchase i6 and i7 as raw materials because of economic crisis, the manufactory cannot manufacture products that are type of P1, P2 or P5. However, the lost profit is no more than 10% of the original profit. Erasable itemsets are especially useful for manufacturers to decide how to purchase raw materials and plan the process of manufacturing products in the case of economic crisis.

In this paper, first mined the frequent itemsets using A-Priori algorithm then its corresponding gain value is computed, based on which the high utility itemsets are mined according to the user specified threshold. Now from the high utility itemsets, erasable itemsets are mined from the user specified gain value inorder to deal with the financial crisis. Thus, the new algorithm will mine both high utility as well as erasable itemsets from the product database.

All of the existing erasable itemset mining algorithms are based on the centralized database but today’s internet era databases are inherently distributed. Most of the organizations operate business in global markets require to perform data mining on distributed data sources to turn them into realistic and meaningful knowledge for their future use and the volume of data available for usage is very high. This inherent distribution source of data and the voluminous in size emerges to develop large-scale parallel and distributed approach for erasable itemset mining.

In this paper, a parallel and distributed approach is proposed which contains one master node and some slave nodes according to requirement. Very large database is distributed to number of slave nodes. Each node scan its local database and generates the frequent itemsets using A-Priori algorithm then its corresponding gain value is computed. Based on this gain value, the high utility itemsets are mined according to the user specified threshold and then from the high utility itemsets, erasable itemsets are mined inorder to deal with the financial crisis. After this every slave node send these high utility and erasable itemsets to master node. Finally, all itemsets are cached by the Master node. Thus, the new method can mine both high utility as well as erasable itemsets from the large distributed product database.

The rest of the paper is organized as follows. Section 2 presents the review of some related research works. Section 3 describes Problem Statement. Section 4 terms and definitions and Section 5 presents the proposed framework and algorithm in details. Finally Section 6 concludes the paper.

2. Related Work

Literature reviews about high utility mining and erasable itemset mining are given in this section.

2.1 High Utility Pattern Mining

High utility pattern mining finds all itemsets in a transaction database with utility value greater or equal to the user specified minimum utility threshold. It also discovers the semantic significance among items in the mining process. Yao et al. [10] proposed a framework for high utility itemset mining and theoretical model called Mining with Expected Utility (MEU). This model cannot maintain downward closure property, and heuristic technique was used to determine candidate set. The same author [11] proposed two utility mining algorithms UMining and Umining_H based on efficient pruning strategies using upper bound. These algorithms overestimate too many patterns in the beginning and also suffer from excessive candidate generations. The pruning strategy used in Umining_H may miss some of high utility itemset [11]. Liu et al. [9] proposed Transaction weighted utility model which uses Two-Phase algorithm to efficiently prune down number of candidate itemsets using intermediate antimonotone property to reduce search space. Transaction weighted utility model is efficient in terms of (1) Fewer candidates set (2) Accuracy and (3) Less arithmetic complexity compared to UMining and Umining_H. This algorithm suffers from the same problem of level-wise candidate generation-and-test methodology. Erwin et al. proposed [4] CTU-mine algorithm for mining high utility itemsets using pattern growth approach.

2.2 Erasable Itemset Mining

The problem of mining erasable itemsets originates from production planning. Deng et al. (2009) first introduced the problem of erasable itemsets mining and proposed an algorithm called META to deal with this problem[7]. Although META is capable of finding all erasable itemsets in reasonable time, it has two important weaknesses. The first weakness is that the time efficiency of META is
poor because it scans database repeatedly. The second weakness is that META can not automatically prune irrelevant data. VME algorithm can overcome the weaknesses of META, to mine erasable itemsets efficiently. NC_set-based algorithm called MERIT is also used to overcome the weakness of VME and META for mining erasable itemsets efficiently[8].

3. Problem Statement

The problems addressed are as follows:
1. How to plan the manufacture of production due to financial crisis.
2. How to efficiently find the components, without which the loss of the profit is no more than the given threshold.

In order to deal with the above stated problems the new algorithm is proposed in such a way that can deal with the economic crisis along with mining the high utility itemsets in distributed environment. The algorithm will efficiently remove those items for which the loss of profits does not exceed the user specified threshold percent of the original profits. It is designed so that it can automatically prune the irrelevant data or items from the transactional or product distributed database.

4. Formulae Used

4.1 For Mining High Utility Patterns

The basic terms and formal definition of high utility itemset mining based on [11][12] and related concepts are described below. Let I = {i1, i2, i3… im} be a set of items. An itemset X is nonempty subset of I. TDB = {T1, T2, T3….Tn} is a transactional database. Each transaction Ti is a set of items and subset of I. The local quantity of an item ip in a transaction Tq is denoted by l (ip, Tq), is defined as sales quantities stored in the transaction. The external utility e(ip) is the profit value per unit of item ip in the profit table. The utility mining problem is to discover all itemsets in a transaction database D with utility values higher than the minimum utility threshold.

<table>
<thead>
<tr>
<th>TID</th>
<th>Transactions</th>
<th>Transaction Utility(tu)</th>
<th>Assigned Slave Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>B(3), C(2), D(3)</td>
<td>59</td>
<td>P0</td>
</tr>
<tr>
<td>T2</td>
<td>A(3), D(2), E(2)</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>B(3), E(2)</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td>A(1), B(1), C(1)</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td>A(2), B(3), D(5)</td>
<td>77</td>
<td>P1</td>
</tr>
<tr>
<td>T6</td>
<td>A(3), B(4)</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td>E(1)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>T8</td>
<td>B(2), D(2)</td>
<td>34</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Transaction Table

<table>
<thead>
<tr>
<th>Item</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2. External Utility

Definition 1: The utility of item ip in transaction Tq, is the quantity measure denoted by U (ip, Tq). Where

\[
U(ip, Tq ) = l(ip, Tq ) \times e(ip) \tag{1}
\]

Definition 2: The utility value of an itemset X in the database U(X), is given as

\[
U(X) = \sum_{ip \in X} \sum_{Tq \in D} U(ip, Tq) \tag{2}
\]

Definition 3: The transaction utility of transaction Tq, denoted as tu (Tq), is the sum of the total profit of all items in Tq and it is defined by,

\[
tu(Tq) = \sum_{ip \in Tq} U(ip, Tq) \tag{3}
\]

The last column of Table 1(a) gives the transaction utility of each transaction.

Definition 4: The minimum utility threshold is the user preferred percentile of total transaction utility value of the given database.

\[
\text{min_util} = \delta \times \sum_{Tq \in D} \text{tu}(Tq) \tag{4}
\]

where \(\delta\) is the user preferred percentage.

Definition 5: Local transaction utility utilization of an itemset X, denoted by Itwu(X), is the sum of the transaction utilities of all transactions containing X in particular node is defined by,

\[
\text{Itwu}(X) = \sum_{X \subseteq Tq \in D} \text{tu}(Tq) \tag{5}
\]

Where X \(\subseteq\) Tq means X is subset of Tq.

Definition 6: Global transaction utility utilization of an itemset X, denoted by gtwu(X), is the sum of the transaction utilities of all transactions of all the nodes that containing X and defined by,

\[
\text{gtwu}(X) = \sum_{i = 1}^{p} \sum_{X \subseteq Tq \in D} \text{tu}(Tq) \tag{6}
\]

Where X \(\subseteq\) Tq means X is subset of Tq.
4.2 For Mining Erasable Itemsets

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items, which are the abstract representation of components, and a product database $DB = \{P_1, P_2, \ldots, P_n\}$, where $P_i (i \in [1, \ldots, n])$ is a product and is presented in the form of $[\text{PID}, \text{Items}, \text{Val}]$. PID is the identifier of $P_i$. Items are all items that constitute $P_i$. Val is the profit that a factory obtains by selling $P_i$. A is called a itemset if and only if A is a set of items.

**Definition 7 (Gain):** Let $A ( \subseteq I)$ be a set of items, the gain of $A$ is defined by

$$\text{Gain}(A) = \Sigma P(k) \text{Val} \quad (7)$$

That is, the gain of itemset $A$ is the sum of values of the products that include at least one item in $A$ as their components.

Consider an example of a product database shown in Table 3. There are seven different items and six products. Each product consists of these fields: PID, Items and Val. Based on Table 3, we know item $i_1$ is a component of product 3 ($\{i_1, i_2, i_3, i_5\}$) and product 4 ($\{i_1, i_2, i_4\}$). Therefore, the gain of itemset $\{i_1\}$, $\text{Gain}(\{i_1\})$, is 8500 (500+8000) in terms of Definition 7. In the same way, we know the gain of itemset $\{i_6, i_7\}$, $\text{Gain}(\{i_6, i_7\})$, is 1000 (500+200+300). Let the threshold $k$ to be 5, Table 4 shows all high-value 5 erasable itemsets.

### Table 3. Product database

<table>
<thead>
<tr>
<th>PID</th>
<th>Items</th>
<th>Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>${i_2, i_3, i_4, i_6}$</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>${i_2, i_6, i_7}$</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>${i_2, i_3, i_5}$</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>${i_3, i_2, i_4}$</td>
<td>8000</td>
</tr>
<tr>
<td>5</td>
<td>${i_6, i_7}$</td>
<td>300</td>
</tr>
<tr>
<td>6</td>
<td>${i_3, i_4}$</td>
<td>500</td>
</tr>
</tbody>
</table>

### Table 4. All high-value 5 erasable itemsets

<table>
<thead>
<tr>
<th>Rank</th>
<th>Erasable Itemsets</th>
<th>Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>${i_7}$</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>${i_6}$</td>
<td>700</td>
</tr>
<tr>
<td>3</td>
<td>${i_7}$</td>
<td>800</td>
</tr>
<tr>
<td>4</td>
<td>${i_6, i_7}$</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>${i_3, i_6, i_7}$</td>
<td>1500</td>
</tr>
</tbody>
</table>

5. Proposed Framework and Algorithm

5.1 Framework

---

**Figure 1. Framework of Master Node**

All Rights Reserved © 2012 IJARCSEE
The proposed algorithm works in the following steps:

Input: A Product Database
Output: All High Utility and Erasable Itemsets

Method:
1. Distribute large database from master node to all the Slave nodes.
2. Each slave node scan local database.
3. Start selecting items from the dataset.
   a. Select multiple items
   b. Add them to the transaction list
   c. Maintain a transaction dictionary of the transaction id and the items chosen for that transaction.
   d. Loop
4. Enter the minimum local support and confidence.
5. The frequent itemsets are generated using the A-priori Algorithm.
6. Compute the gain value of each itemset.
7. Calculate the local transaction weighted utilization (ltwu) of each node.
8. Send (ltwu) to the Master node.
10. Broadcast the gtwu to all slave nodes.
11. Each local node builds their global transaction table using global transaction weighted utilization and prunes the items that do not satisfy the given threshold min_util.
12. At each slave node for each computed gain value
   if gain(x) >= minimum utility threshold
   x is high utility itemset
   end if
   end for
13. At each slave for each high utility value
   if value < user specified value
   display erasable itemsets
   end if
14. Each slave node send these local potential high utility and erasable itemsets to Master node.
15. At last, global high utility and erasable itemsets are cached by the Master node.

6. Conclusion

In this paper, a parallel method is proposed to generate complete set of high utility itemsets with erasable itemsets from large distributed databases. This method can efficiently remove those items for which the loss of profits does not exceed the user specified threshold percents of the original profits. This approach creates distributed environment with one master node and some slave nodes. Large database is distributed to all salve nodes. At first level, each slave node generates frequent itemsets from its local databases through A-Priori. At second level, every node calculate local weighted utility, mine high utility and erasable itemsets and send it to Master node. Then master node calculate global weighted utility and find global high utility and erasable itemsets by accumulating local patterns. According to the market needs it can mine the irrelevant patterns leaving the most important patterns that will result in company’s profit at the time of economic crisis. So, the proposed method can provide the high scalability and performance gain and require minimum communication among the nodes. It also can decrease the execution time by parallelizing pattern mining.

7. References

[2]. Hua-Fu Li · Hsin-Yun Huang · Suh-Yin Lee “Fast and memory efficient mining of high-utility itemsets from data streams” © Springer-Verlag London Limited 2010.


