Implementation of Pharmacy Sale System Using Frequent Pattern Growth Algorithm

Aye Mya Mya Win, Win Mar
Computer University, Mubin
pheeale@gmail.com

Abstract

Frequent Pattern mining is played an essential role in data mining. In this system proposes to implement Pharmacy Sale System (PSS) by using Frequent Pattern Growth (FP-growth) Algorithm. This system can manipulate how many frequent patterns will be mined with a specified the minimum support. Besides, can manipulate data and make decision effectively and quickly and may have a series of judgment that predicts sales of items rental on month and year. The development of a Pharmacy sale System will be provided management level to make decisions effectively and quickly.

Keywords: Data Mining, Frequent Patterns Growth, Association Rule

1. Introduction

Nowadays, most of the business organizations are widely used in the data mining system. Data mining is a knowledge discovery in databases, which is the automated or convenient extraction of patterns representing knowledge implicitly stored in large databases, data warehouses, and other massive information repositories.

It is a well accepted verity that the process of data mining produces numerous patterns form the given data. The most significant tasks in data mining are the process of discovering frequent itemsets and association rules. Numerous efficient algorithms are available in the literature for mining frequent itemsets and association rules.

Incorporating utility considerations in data mining tasks is being gained popularity in recent years. Certain association rules enhance the business value and the data mining community has acknowledged the mining of these rules of interest since a long time. Several business applications have been found to benefit from the discovery of frequent itemsets and association rules from transaction databases. And the system apply FP-growth algorithm (Mining frequent itemsets without candidate generation) to discover frequent patterns from sequential database.

Mining sequential patterns from a large database is important and interesting to the fundamental research in the data mining community. It is also useful for a variety of applications such as marketing data analysis and stock trend prediction. Searching for frequent patterns in transactional databases is considered one of the most important data mining problems.

Mining association rules from databases has attracted great interest because of its potentially very practical applications. Given a database, the problem of interest is how to mine association rules (which could describe patterns of consumers' behaviors) in an efficient and effective way. The databases involved in today's business environments can be very large. Thus, fast and effective algorithms are needed to mine association rules out of large databases. Previous approaches may be caused exponential computing resource consumption. A combinatorial explosion occurs because existing approaches exhaustively mine all the rules.

In this paper, system using FP-growth algorithm is proposed. The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 describes concepts of data mining. Section 4 reviews implementation of Pharmacy Sale System. Finally, conclusion and limitation and further extension this paper in section 5.

2. Related Work

Today, organizations and enterprises are gathered and stored large amount of data in their daily activity and operations continuously. Understanding or learning about the implicit or hidden information in the data is important for strategic decision support or technical operations. Discovery of interesting association relationships among huge amounts of data will be helped marketing, decision making and business management. Therefore, mining association rules from large data sets has been a focused topic in recent research into knowledge discovery in databases [2].

The growth in the number of available databases far outstrips the growth of corresponding knowledgebase. This creates both a need and an opportunity for extracting knowledge from databases [1].

Association rules are used to identify relationships among a set of item in database. These relationships are not based on inherent properties of the data themselves, but rather based on co-occurrence of the data items. We are given a large database of customer transactions, where each transaction consists of customer-id, transaction time, and the items bought in the transaction.

Association rule mining is found interesting association or correlation relationships among a large set of data items. The mining task can be mapped into the problem of discovering frequent itemsets which appear in a sufficient number of transactions. The problem of discovering frequent itemsets can be solved by constructing the candidate set, those itemsets which meet the supports requirement [7]

Several algorithms have been proposed in the literature to address the problem of mining association rules. FP-growth was recently proposed by Han et al. [11]. This algorithm creates a relatively compact tree-structure that alleviates the multi-scan problem and improves the candidate itemset generation. The algorithm requires only two full I/O scans for the dataset. Our approach presented in this paper is based on this idea. In spite of the significance of the association rule mining and in particular the generation of frequent itemsets, few advances have been made on parallelizing association rule mining algorithms [12].

3. Basic Concepts of Data Mining

Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques.

The term data mining is often used to apply to the two separate processed of knowledge discovery and prediction. Knowledge discovery provides explicit information that has a readable form and can be understood by a user (e.g., association rules mining).Forecasting or predictive modeling is provided predictions of future events and may be transparent and readable in some approaches (e.g., rule-based systems) and opaque in others such as neural networks.

Data mining is relied on the use of real world data. These data are extremely vulnerable to co linearity precisely because data from the real world may have unknown interrelations. An unavoidable weakness of data mining is that the critical data that may expose any relationship might have never been observed.

3.1 Frequent Pattern Mining in SQL Based on FP-Growth

Let $I = \{i_1, i_2, \ldots . i_m\}$ be a set of items and D be a transaction database $\{T_1, T_2, \ldots \ldots . T_n\}$ where each transaction Ti D is a subset of I. The frequency of a pattern X $\{x_1, x_2, \ldots ., x_p\}$ is the number of transactions containing the pattern in the transaction database. The problem of frequent pattern mining is to find the complete set patterns satisfying a minimum support in the transaction database [3]. In [11], frequent pattern mining consists of two steps:

1. Construct a compact data structure, frequent pattern tree (FP-tree), which can store more information in less space.

2. Develop an FP-tree based pattern growth (FP-growth) method to uncover all frequent patterns recursively.

Frequent pattern growth (FP-growth) is a method of mining frequent item sets with candidate generation, which adopts a divide-and-conquer strategy. It constructs a highly compact data structure (an FP-tree) to compress the original transaction database.

3.2. Association Rule Mining

With massive amount of data continuously begin collected and stored, many industries are becoming interested in mining association rules from database. Association rule mining is searched for interesting or correlation relationships among items in a given data set. [10]The mining of association rules from large databases is a two-steps process:

1. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count.

2. Generate strong association rules from the frequent itemsets: By definition, these rules must satisfy minimum support and minimum confidence.

4. Implementation of PSS by using FP-growth Algorithm

In this paper, we implemented Pharmacy Sale system based on data mining aspect. There are several methods in data mining such as cluster analysis, factor analysis, decision tree, data visualization, neural networks, rule induction and association rule. Among of them association rule mining can be used for interesting relationships among items in a given data set. The mining of association rules from large databases has two methods, finding frequent itemsets using candidate generation and mining frequent itemsets without candidate generation. Mining frequent itemsets without candidate generation is used in this system.

In this paper we applied frequent pattern growth algorithm to extract frequent items which are commonly sold by customers. When we used computerized Pharmacy Sale system we can get too many advantages such as this system can provide convenience for manager. Manager to know the habits of customers and which item groups or set (medicine groups) are more purchased together in a single transaction.

Information from transaction databases is essential for mining frequent patterns. Therefore, if we can be extracted the concise information for frequent pattern mining and store it into a compact structure, then it may facilitate frequent pattern mining. Motivated by this thinking, in this section, we develop a compact data structure, called *FP-tree*, to store complete but no redundant information for frequent pattern mining. [5]

The FP-growth method is transformed the problem of finding long frequent patterns to looking for shorter ones recursively and then concatenating the suffix. It focuses on frequent pattern growth which avoids costly candidate generation, resulting in greater efficiency.

FP-growth is constructed a highly compact data structure (an FP-tree) to compress the original transaction database. It focuses on frequent pattern growth which avoids costly candidate generation, resulting in greater efficiency.

It is used the least frequent items as a suffix, offering good selectivity. The method substantially reduce the search costs. An interesting alternative is to first partition the database into a set of projected databases. Construct an FP-tree and mine it in each projected database. Such a process can be recursively applied to any projected database if its FP-tree still cannot fit in main memory. [6]

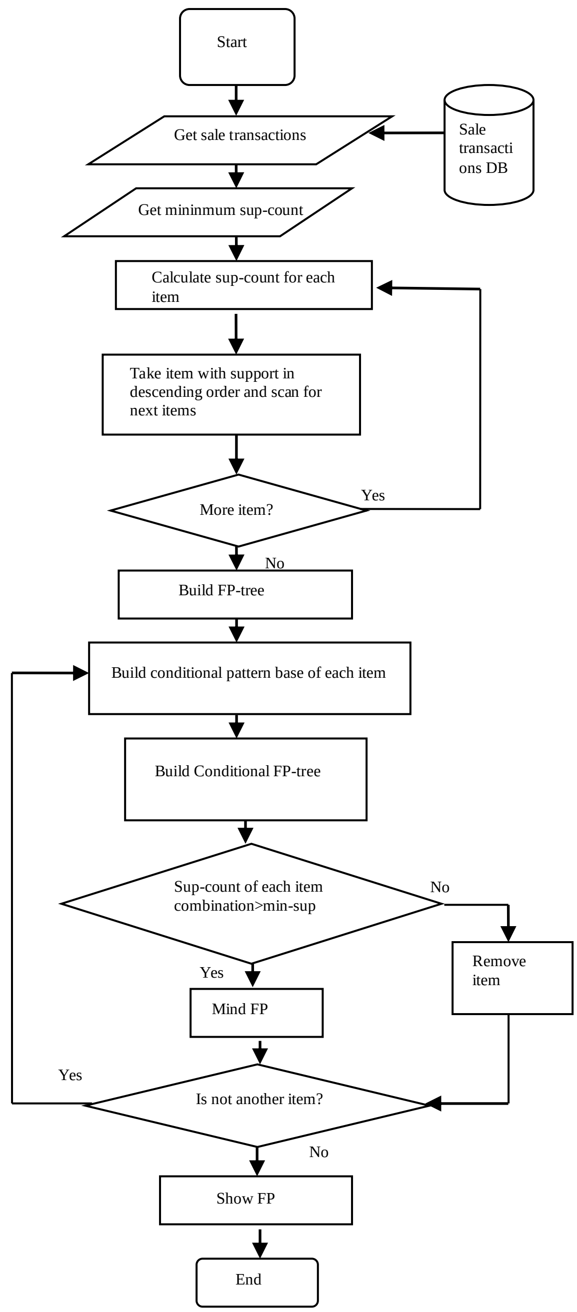

Figure 1 System Flow of Proposed System

4.1 Frequent Pattern Growth Algorithm

Algorithm: FP-growth. Mine frequent patterns using an FP-tree by pattern fragment growth.
Input: A transaction database, *D*; minimum support threshold, *min-sup*.
Output: The complete set of frequent patterns.
Method:
(1)The FP-tree is constructed in the following steps. (a) Scan the transaction database *D* once. Collect *t* set of frequent items *F* and their supports. Sort *F* in support descending order as *L*, the list of frequent items.
(b) Create the root of an FP-tree, and label it as "null". For each transaction *Trans* in *D* do the following.
Select and sort the frequent items in *Trans* according to the order of *L*. Let the sorted frequent item list in *Trans* be [*p*/*P*], where *p* is the first element and *P* is the remaining list. Call **insert_tree** ([*p*/*P*], *T*), which is performed as follows. If *T* has a child *N* such that *N.item-name* = *p.item-name*, then increment *N*'s count by 1; else create a new node *N*, and let its count be 1, its parent link be linked to *T*, and its node-link to the nodes with the same item-name via the node-link structure. If *P* is nonempty call insert-tree (*P*, *N*) recursively.
(2)Mining of an FP-tree is performed by calling **FP-growth** (*FP-tree*, *null*), which is implemented as follows.

Procedure FP-growth (*Tree*, **α**)
(1) if Tree contains a single path *P* then
(2) for each combination (denoted as **β**) of the nodes in the path *P*
(3) Generate pattern **β∪α** with support = minimum support of modes in **β**;
(4) else for each **α**, in the header of Tree{
(5) generate pattern $\beta = \alpha_i \cup \alpha$ with support = α_i.support;
(6) Construct **β**'s conditional pattern based and then **β**'s conditional FP-tree $Tree_\beta$;
(7) If $Tree_\beta \neq \emptyset$ then
(8) Call FP-growth($Tree_\beta$,β);}

Figure 2 The FP-growth algorithm for discovering frequent itemsets without candidate generation.

4.2 Detail of the FP-Growth Approach

When we start the system, we examine the mining of transaction database, *D*, of Figure 3 using the frequent-pattern growth approach.

TID	List of item_IDs
001	Citrizine, Biogesic,Digene
002	Biogesic, Flemex
003	Biogesic, Voltran
004	Citrizine, Biogesic, Flemex
005	Citrizine, Voltran
006	Biogesic, Voltran
007	Citrizine, Voltran
008	Citrizine, Biogesic, Voltran, Digene
009	Citrizine, Biogesic, Voltran

Figure3 Example of transactional data for a medician branch.

The first scan of the database, which derives the set of frequent items (1-itemsets) and their support counts (frequencies). Let the minimum support count be 2. The set of frequent items is sorted in the order of descending support count. This resulting set or list is denoted L. Thus, we have L= {Biogesic:7, Citrizine:6, Voltran:6, Flemex:2, Digene:2}.

An FP-tree is then constructed as follows. First, create the root of the tree labeled with "null". Scan database D a second time. The items in each transaction are processed in L order and a branch is created for each transaction. For example, the scan of the first transaction, "001: Citrizine, Biogesic, Digene", which contains three items (Biogesic, Citrizine, Digene) in L order, leads to the construction of the first branch of the tree with three nodes:{(Biogesic: 1), (Citrizine: 1), (Digene: 1)}, where Biogesic is linked as a child of the root, Citrizine is linked to Biogesic and Digene is linked to Biogesic. The second transaction, 002, contains the items Biocesic and Flemex in L order, which would result in a branch, where Biogesic is linked to the root and Flemex is linked to Biogesic.However, this branch would share a common prefix, (Biogesic), with the existing path for 001. Therefore, we instead increment the count of the Biogesic node by 1, and create a new node, (Flemex: 1), which is linked as a child of (Biogesic: 2). In general, when considering the branch to be added for a transaction, the count of each node along a common prefix is incremented by 1, and nodes for the items following the prefix are created and linked accordingly.

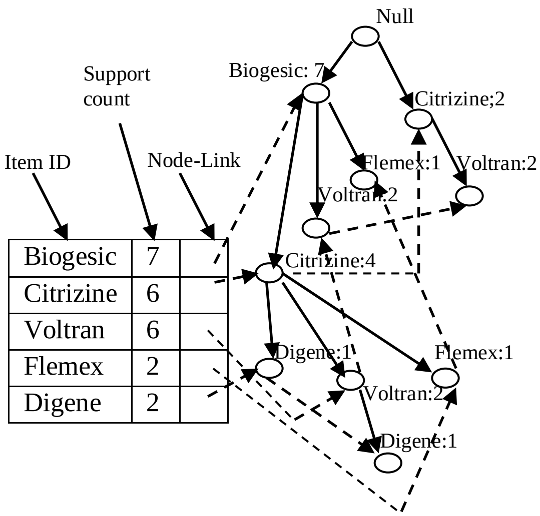

Figure 3.1n FP-tree that registers compressed, frequent pattern information.

Item	Conditional pattern base	Conditional FP-tree	Frequent patterns generated
Digene	{(Biogesic Citrizine:1) , (Biogesic Citrizine Voltran:1)}	(Biogesic:2, Citrizine:2)	Biogesic Digene:1, Citrizine Digene:2, Biogesic Citrizine Digene:2
Flemex	{(Biogesic Cirizine:1), (Biogesic:1)}	(Biogesic:2)	Biogesic Flemex:2
Voltan	{(Biogesic1 1:2), (Biogesic:2), (Citrizine:2)}	(Biogesic:4, Citrizine:2), (Citrizine:2)	Biogesic Voltran:4, Citrizine Voltran:2, Biogesic Citrizine Voltan:2
Citrizine	{(Biogesic: 4)}	(Biogesic:4)	Biogesic Citrizine:4

Table 3.1 Mining the FP-tree by creating conditional pattern base.

To facilitate tree traversal, an item header table is build so that each item points to its occurrences in the tree via a chain of node-links. The tree obtained after scanning all of the transactions is shown in Figur3.1with the associated node-links. Therefore, the problem of mining frequent patterns in databases is transformed to that of mining the FP-tree.

The mining of the FP-tree proceeds as follows. Start from each frequent length 1 pattern (as an initial suffix pattern), construct its conditional pattern base (a “subdatabase” which consists of the set of prefix paths in the FP-tree co-occurring with the suffix pattern), then construct its (conditional) FP-tree, and perform mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

Mining of the FP-tree is summarized in Table 3.1. Let first considering Digene which is the last item in L, rather than the first. The reasoning behind this will become apparent as we explain the FP-tree mining process. Digene occur in two branches of the FP-tree of Figure 3.1. The paths formed by these branches are (*Biogesic* Citrizine Digene: 1) and (Biogesic Citrizine Voltran Digene: 1). Therefore, considering Digene as a suffix, its corresponding two prefix paths are (Biogesic Citrizine: 1) and (Biogesic Citrizine Voltran: 1), which form its conditional pattern base. Its conditional FP-tree contains only a single path, (Biogesic: 2, Citrizine: 2); Voltran are not included because its support count of 1 is less than the minimum support count. The single path generates all the combinations of frequent patterns: Biogesic Digene: 2, Citrizine Digene: 2, Biogesic Citrizine Digene: 2. And then, continues the remain items.

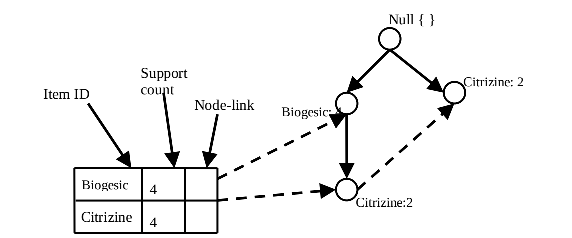

Figure 3.3 The conditional FP-tree associated with the conditional node Voltran.

Citrizine Voltran:2}. Finally, Citrizine’s conditional pattern base is {(Biogesic: 4)}, whose FP-tree contains only one node (Biogesic: 4), which generates one frequent pattern, Biogesic Citrizine: 4. this mining process is summarized in Figure 3.3.

5. Conclusion

This system is implemented for Pharmacy Sale System which used FP-growth Algorithm. The algorithm has so many advantages such as can reduce search costs, and so on.

This system is to be discovered buying patterns such as two or more items that are bought together often from market basket data or the pharmacy data, where the base information consists of register transactions of drug stores.

Using FP-growth based data mining system will play an increasingly important role in Pharmacy Sale System and may improve the quality of the system effectively and efficiently than the manual system.

References

[1] Jawed Han, Cai Yandong
Data-driven discovery of quantitative rules in relational databases
IEEE Trans. Knowledge and Data Engineering

[2] Jiawei Han, Micheline Kamber
Data Mining Concepts and Techniques
Morgan Kaufmann Publishers

[3] Jiawei Han, Pei, J., Yin, Y.,
Mining frequent patterns without candidate generation: a frequent pattern tree approach
Data Mining and Knowledge Discovery, Vol.8, pp. 53.

[4] Klemettinen, Mika; Mannila, Heikki; Ronkainen, Pirjo; Toivonen, Hannu; Verkamo, A.Inkeri
Finding interesting rules from large sets of discovered association rules
Proc. 3rd Int'l Conf. on Information and Knowledge Management
pp. 401-408, Gaithersburg, Maryland, November, 1994

[5] M.H.Margahny,A.A.Mitwaly
"Fast Algorithm for Mining Association Rules"

[6] M.Zaki, S.Parthasarathy, M.Ogihava, and W.Li.
"New Algorithm for fast Discovery of Association Rules"

[7] Park, Jong Soo; Chen, Ming-Syan; Yu, Philip S.
An effective hash-based algorithm for mining association rules
Proc. 1995 ACM-SIGMOD Int Conf. Management of Data, San Jose, CA, May 1995

[8] R. Agrawal and R. Srikant.
" Mining sequential patterns."
In *Proc*. *1995* Int.Conf. Data Engineering (ICDE’95), pages 3–14, Taipei, Taiwan, Mar. 1995.

[9] R. S rikant and R. Agrawal. "Mining sequential patterns:Generalizations"

[10] R. S rikant and R. Agrawal. T. Imielinski, A. Swami. “Mining association rules between sets of items in large database” In Proc. Of the ACM SIGMOD Conference on Management of Data, pages 207, Washington, D.C., May 1993.

[11] Y. Yin. and J. Han, J. Pei, Mining frequent patterns without candidate generation.
In *ACM- SIGMOD*, Dallas, 2000.

[12] Y. Fu. , D. Cheung, J. Han, and V. Ng, A. Fu,
A fast distributed algorithm for mining Association rules. In Proc.1996 Int. Conf. Parallel and Distributed Information Systems,
pages 31–44, Miami Beach, Florida, Dec. 1996.