MINING FREQUENT itemsets using advanced partition APPROACH

Nay Chi Lynn, Khin Myat Myat Moe
University of Computer Studies, Magway
naychelynn@gmail.com

Abstract

Frequent itemsets mining plays an important part in many data mining tasks. This technique has been used in numerous practical applications, including market basket analysis. This paper presents mining frequent itemsets in large database of medical sales transaction by using the advanced partition approach. This advanced partition approach executes in two phases. In phase 1, the advanced partition approach logically divides the database into a number of non-overlapping partitions. These partitions are considered one at a time and all local frequent itemsets for those partitions are generated using the apriori method. In phase 2, the advanced partition approach finds the final set of frequent itemsets. The purpose of this paper is to extract the final sets of frequent itemsets from medical retail datasets and to support efficient information used to plan marketing or advertising strategies for medical stores and companies. Algorithms for finding frequent itemsets like Apriori, needs many database scans. But, this advanced partition approach needs to scan the entire database only one time. So, it reduces the time taken for the large database scan in mining frequent itemsets.

1. Introduction

Nowadays, retailing is becoming a high-performance sport. Like athletes, retailers are becoming competitive, seeking technology to gain, and trying to have more knowledge into customer buying behavior. Market basket analysis has emerged as a step of the retail merchandising, promotion, etc. Market basket analysis studies the buying habits of customers by finding frequent itemsets between the different items that customers purchase [3].

Mining frequent itemsets is important and interesting to the fundamental research in the mining of association rules. Frequent itemsets mining, one technique of the descriptive mining, is widely used for market basket analysis. It analyses customer buying habits by finding frequent itemsets between different items that occur frequently together in a given set of data. There are many algorithms for finding frequent itemsets. The Apriori is the basic first algorithm for finding frequent itemsets. In this system, the advanced partition algorithm is used for finding frequent itemsets with an entire single data pass. By the result of system testing, the user can know which algorithm is suitable for his medical company and store.

2. Related work

The works related to this system are presented in here. Frequent itemsets mining is well explored for various data types. The Apriori algorithm also called level-wise algorithm was proposed by Agrawal and Srikanth in 1994. The name of the algorithm is based on the fact that the algorithm uses the prior knowledge of frequent itemsets properties [1]. Nguyen and Orlowska show the data partition approach to further improve the performance of frequent itemsets computation. The methods focus on potential reduction of the size of the input data required for deployment of the partitioning based algorithms [4]. Kranthi and

Malreddy have proposed the advanced partition approach to generate the frequent itemsets in a single pass over the database [2]. In this paper the advanced partition approach has been used to find the frequent itemsets.

3. Theory background

Frequent itemsets mining is an important data mining task. It extracts interesting correlations, frequent itemsets among sets of items in the transaction databases.

3.1 Frequent Itemsets

A set of items is referred to as an itemset. An item set that contains *k* items is a *k*- itemset .The set {computer, finical-management software} is a 2-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset.

An itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to the product of min-sup and the total number of transactions in the database. The number of transactions required for the itemset to satisfy minimum support is therefore referred to as the minimum support count. If an itemset satisfies minimum support, then it is a frequent itemset [1]**.**

3.2 Advanced Partition Approach

The advanced partition approach is based on the premise that number of items in a transaction is quite less compared to total number of items in the transaction database. This advanced partition approach is efficient when the local support of each frequent item set in a partition is much higher than 1.

The advanced partition approach executes in two phases. In phase 1, the advanced partition approach divides the database into a number of non-overlapping partitions.

These partitions are considered one at a time and all frequent itemsets for that partition (L_i) are generated using the Apriori algorithm (presented in section 3.2.1). In addition, when taking each partition for calculating the frequent itemsets separately the local minimum support is set to 1.

At the end of phase I, the advanced partition approach merges all local frequent itemsets of each partition to generate the global candidate item sets ($C_k{}^{G}$).

Phase II: This phase just prune the item sets from the global candidate itemsets list whose combined support ($s(c)_{Tc}$) (total support of an item set in all the partitions) is less than the global minimum support. So here the advanced partition approach reads the entire database once during the Phase I. And also, partition sizes are chosen such that each partition can be accommodated in the main memory [2].

Table 1 Notations used for advanced partition approach

Notation	Meaning
L^i	Local Frequent Sets: Set of Local Frequent Itemsets of partition i.
C_k^G	Global Frequent Sets: Set of global candidate k-itemsets.
L_i^k	Local Frequent Sets: Set of local frequent k-Itemsets in partition i.
L^G	Global Frequent Sets: Set of global frequent Itemsets.
$s(c)_{Tc}$	Combined Support Total support of candidate set c in all partitions.

Below is the algorithm of advanced partition approach:

```
P = partition_database (T); N = Number of partitions;
// Phase I
for i= 1 to n do
begin
read_in_partition ( $T_i$ in P )
$L^i$ = generate all frequent itemsets of $T_i$ using Apriori algorithm in main memory.
end
```

```
// Merge Phase
for (k = 2; $L_i^{k} \neq \phi$, i=1,2,…,n; i++) do
begin
    $\mathbf{C_k^G = Y^{n}_{i=1} L_i^k}$
end
// Phase II
$L^{G} = \phi$ ;
for each c є $C^{G}$ do
begin
    if $s(c)_{TC} \geq \sigma$
        $L^{G} = L^{G} \cup \{s(c)\}$
end
Answer = $\mathbf{L^G}$
```

Figure 1: The Advanced partition approach for discovering frequent itemsets.

3.2.1 Apriori Algorithm

Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. Apriori uses prior knowledge of frequent item set properties. Apriori employs an iterative approach known as level-wise search, where k itemsets are used to explore (k+1)-item sets [1].

There are two-steps in Apriori Algorithm:

1. **The join step:** To find L_k, a set of **candidate** k- itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k. Let l_1 and l_2 be itemsets in L_{k-1}.
2. **The prune step:** C_k is a superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k. A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k.

Algorithm: Apriori. Find frequent itemsets using an interactive level-wise approach based on candidate generation.

Input: Database, *D*, of transactions; minimum support threshold, *min_sup.*

Output: *L*, frequent itemsets in *D*.

Method:

```
$L_1$= find_frequent_1-itemsets(D);
for ($k = 2$; $L_{k-1} \neq \phi$; $k$++) {
    $C_k$ = apriori_gen ($L_{k-1}$, $min\_sup$);
    for each transaction $t \in D$ { // scan $D$ for counts
        $C_t$ = subset ($C_k$, $t$); // get the subsets of $t$ that are candidates
        for each candidate $c \in C_t$
            c.count++;
    }
    $Lk = \{ c \in C_k \backslash c.count \geq min\_sup \}$
}
return $L = U_k L_k$;
```

Procedure apriori_gen (L_k-*1*: frequent (k – 1)-itemsets; min_sup: minimum support threshold)

```
for each itemset $l_1 \in L_k$-1
    for each itemset $l_2 \in L_k$-1
        if($l_1[1] = l_2[1]$) ^($l_1[2] = l_2[2]$) ^ ... ^ ($l_1[k - 2] = l_2[k - 2]$) ^ ($l_1[k - 1] < l_2[k - 1]$) then {
            $c = l_1 \infty l_2$; // join step: generate candidates
            if has_infrequent_subset($c$, $L_{k-1}$) then
                delete $c$; // prune step: remove unfruitful candidate
            else add $c$ to $C_k$;
        }
return $C_k$;
```

Procedure has_infrequent_subset (*c*: candidate *k*-itemset; L_{k-1}: frequent (*k* – 1)-itemsets);
// use prior knowledge

```
for each ($k$ – 1)-subset $s$ of $c$
    if $s$ ¢ $L_{k-1}$ then
        return TRUE;
return FALSE;[1]
```

Figure 2: The Apriori algorithm

4. System design and implementation

This paper finds frequent itemsets between medical products purchased together by customers. For the purpose of implementing, the retail datasets supplied by a medical Co, Ltd is used.

Using advanced partition approach, the system is efficient in computing frequent itemsets and can reduce the time spent in performing the I/O operations for large databases.

4.1 System Design

If the user wants to entry new medical products into his data set, the Entry new items process should be chosen. If the user wants to buy those above medical products, the Sales process should be chosen. Moreover, the user can also import the items from other datasets using import data menu. The system will find the global frequent itemsets by using the advanced partition approach. The user has to choose the global minimum support count either number or percentage.

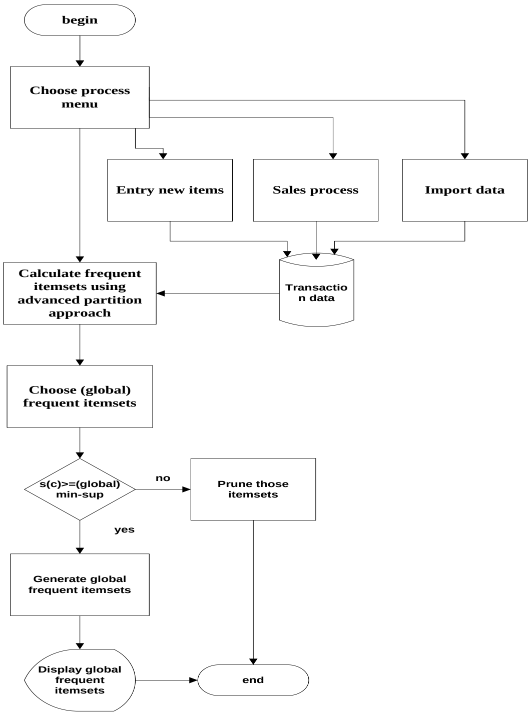
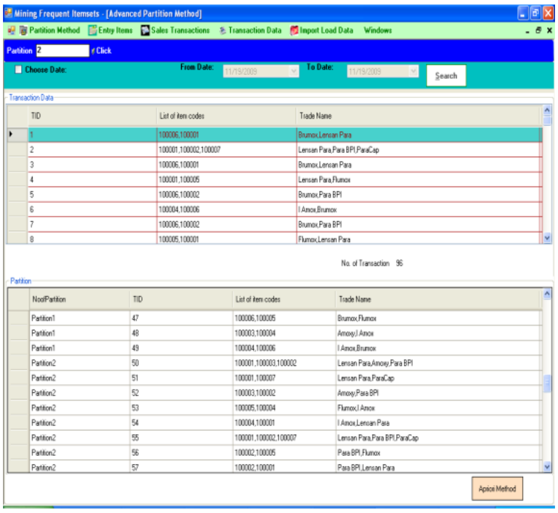

begin
Choose process menu
Entry new items
Sales process
Import data
Transaction data
Calculate frequent itemsets using advanced partition approach
Choose (global) frequent itemsets
s(c)>=(global) min-sup
no
yes
Prune those itemsets
Generate global frequent itemsets
Display global frequent itemsets
end

Figure 3: System flow diagram

4.2 Implementation of the advanced partition approach

Below is the example which shows the working of the system. The transaction dataset (T) consists of a set of transactions in the form (tid, items-purchased) where tid is transaction ID. This system includes medical sales items such as, Lensan Para, Para BPI, Amoxy, I Amox, Flumox, Brumox, Biogesic, etc.

The advanced partition approach executes in two phases. In this form, medical sales itemsets are partitioned according to user specified input partition number (N). If the user input partition number (N) is 2, then the advanced partition approach (phase 1) divides sales itemsets into two partitions such as partition 1 and partition 2 as shown in figure 4.



Partition 2

□ Choose Date: From Date: 11/19/2009 To Date: 11/19/2009

Transaction Data

TID	List of item codes	Trade Name
1	100006,100001	Brumox,Lensan Para
2	100001,100002,100007	Lensan Para,Para BPI,ParaCap
3	100006,100001	Brumox,Lensan Para
4	100001,100005	Lensan Para,Flumox
5	100006,100002	Brumox,Para BPI
6	100004,100006	I Amox,Brumox
7	100006,100002	Brumox,Para BPI
8	100005,100001	Flumox,Lensan Para

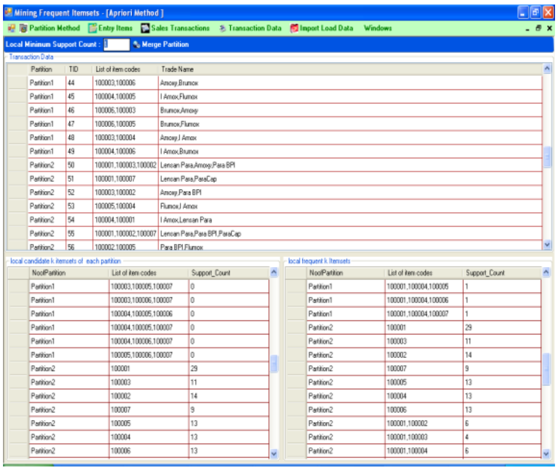
No. of Transaction 96

Partition

NoofPartition	TID	List of item codes	Trade Name
Partition1	47	100006,100005	Brumox,Flumox
Partition1	48	100003,100004	Amoxy,I Amox
Partition1	49	100004,100006	I Amox,Brumox
Partition2	50	100001,100003,100002	Lensan Para,Amoxy,Para BPI
Partition2	51	100001,100007	Lensan Para,ParaCap
Partition2	52	100003,100002	Amoxy,Para BPI
Partition2	53	100005,100004	Flumox,I Amox
Partition2	54	100004,100001	I Amox,Lensan Para
Partition2	55	100001,100002,100007	Lensan Para,Para BPI,ParaCap
Partition2	56	100002,100005	Para BPI,Flumox
Partition2	57	100002,100001	Para BPI,Lensan Para

Figure 4: Partition dataset form

Then, the advanced partition approach (phase 1) finds all local frequent itemsets of each partition using Apriori method in figure 5.



Transaction Data

	Partition	TID	List of item codes	Trade Name
	Partition1	44	100003,100006	Amoxy,Brumox
	Partition1	45	100004,100005	I Amox,Flumox
	Partition1	46	100006,100003	Brumox,Amoxy
	Partition1	47	100006,100005	Brumox,Flumox
	Partition1	48	100003,100004	Amoxy,I Amox
	Partition1	49	100004,100006	I Amox,Brumox
	Partition2	50	100001,100003,100002	Lensan Para,Amoxy,Para BPI
	Partition2	51	100001,100007	Lensan Para,ParaCap
	Partition2	52	100003,100002	Amoxy,Para BPI
	Partition2	53	100005,100004	Flumox,I Amox
	Partition2	54	100004,100001	I Amox,Lensan Para
	Partition2	55	100001,100002,100007	Lensan Para,Para BPI,ParaCap
	Partition2	56	100002,100005	Para BPI,Flumox

local candidate k itemsets of each partition

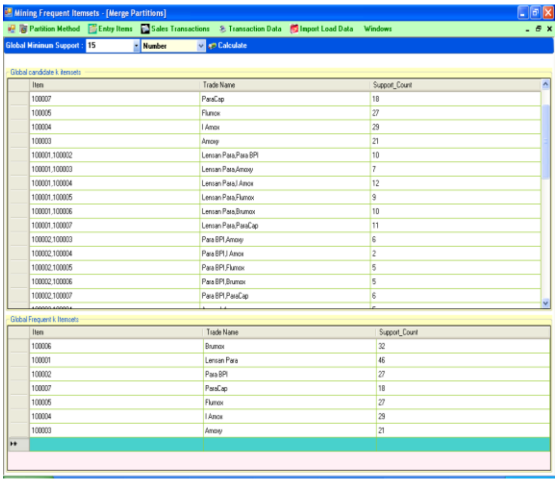
	NoofPartition	List of item codes	Support_Count
	Partition1	100003,100005,100007	0
	Partition1	100003,100006,100007	0
	Partition1	100004,100005,100006	0
	Partition1	100004,100005,100007	0
	Partition1	100004,100006,100007	0
	Partition1	100005,100006,100007	0
	Partition2	100001	29
	Partition2	100003	11
	Partition2	100002	14
	Partition2	100007	9
	Partition2	100005	13
	Partition2	100004	13
	Partition2	100006	13

local frequent k Itemsets

	NoofPartition	List of item codes	Support_Count
	Partition1	100001,100004,100005	1
	Partition1	100001,100004,100006	1
	Partition1	100001,100004,100007	1
	Partition2	100001	29
	Partition2	100003	11
	Partition2	100002	14
	Partition2	100007	9
	Partition2	100005	13
	Partition2	100004	13
	Partition2	100006	13
	Partition2	100001,100002	6
	Partition2	100001,100003	4
	Partition2	100001,100004	6

Figure 5: Local frequent itemsets form

The advanced partition approach (phase 2) finds global frequent itemsets which satisfy user-specified minimum support (if user specified minimum support count number is 10, it will show the itemsets above 10) in figure 6 and then gives final set of (global) frequent itemsets.



Global Minimun Support : 15 Number

Global candidate k itemsets

	Item	Trade Name	Support_Count
	100007	ParaCap	18
	100005	Flumox	27
	100004	I Amox	29
	100003	Amoxy	21
	100001,100002	Lensan Para,Para BPI	10
	100001,100003	Lensan Para,Amoxy	7
	100001,100004	Lensan Para,I Amox	12
	100001,100005	Lensan Para,Flumox	9
	100001,100006	Lensan Para,Brumox	10
	100001,100007	Lensan Para,ParaCap	11
	100002,100003	Para BPI,Amoxy	6
	100002,100004	Para BPI,I Amox	2
	100002,100005	Para BPI,Flumox	5
	100002,100006	Para BPI,Brumox	5
	100002,100007	Para BPI,ParaCap	6

Global Frequent k Itemsets

	Item	Trade Name	Support_Count
	100006	Brumox	32
	100001	Lensan Para	46
	100002	Para BPI	27
	100007	ParaCap	18
	100005	Flumox	27
	100004	I Amox	29
	100003	Amoxy	21
▸*			

Figure 6: Final set of frequent itemsets form

5. Conclusions

Mining frequent itemsets is important and is one of the primary sub-areas on the fields of data mining. Market basket data analysis has been well addressed in mining frequent itemsets for discovering the set of large items. This system is intended to implement frequent itemsets mining.

The discovery of frequent patterns and correlation relationships among huge amounts of data is useful in selective marketing, decision analysis, and business management. This system can help retailers, buyers, planners, merchandisers, and store managers to plan more profitable advertising and promotions, attract more customers, and increase the value of the market basket.

Moreover, one can use the results to plan marketing or advertising strategies, or in the design of a new catalog. For instance, it may help managers to design different store layouts. In one strategy, items that are frequently purchased together can be placed together in close proximity in order to further encourage the sales of such items together. This system can act as a consultant for medical stores by giving the information of frequent items.

References

[1] J.W.Han, M. Kamber, “Data Mining Concepts and Technique”, ISBN 1-55860-489-8, Morgan Kaufmann Publishers.

[2] Kranthi K. Malreddy, B.S, “Mining Frequent Itemsets Using Advanced Partition Approach”, Dean of the Graduate School, December, 2004.

[3] Lary Gordan, Partner, “Leading Practices In Market Basket Analysis”, the Face Point Group, 349 First Street, Los Altos, CA 94022 (650)559-2105, Gordan@Facepoint.Com

[4] Son N. Nguyen, Maria E. Orlowska, “A Further Study In The Data Partitioning Approach For Frequent Itemsets Mining”, School Of Information Technology And Electrical Engineering, the University Of Queensland, QLD 4072, Australia {Nnson, Maria} Itee.Uq.Edu.Au.