Asian Journal of Computer Science and Technology
ISSN 2249-0701 Vol. 2 No. 1, 2013, pp.19-21
© The Research Publication, www.trp.org.in

Sequential Pattern Mining Using Algorithm

Kirti S. Patil and Sandip S. Patil
S.S.B.T's COET, Bambhori Jalgaon - 425 201, Maharashtra, India
E-mail : kirti.1301@gmail.com, San_78004@yahoo.co.in

(Received on 25 January 2013 and accepted on 28 March 2013)

Abstract - **The concept of Sequential Pattern Mining was first introduced by Rakesh Agrawal and Ramakrishnan Srikant in the year 1995. Sequential Patterns are used to discover sequential sub-sequences among large amount of sequential data. In web usage mining, sequential patterns are exploited to find sequential navigation patterns that appear in users' sessions sequentially. The information obtained from sequential pattern mining can be used in marketing, medical records, sales analysis, and so on. In this paper, a new algorithm is proposed; it combines the Apriori algorithm and FP-tree structure which proposed in FP-growth algorithm. The advantage of proposed algorithm is that it dosen't need to generate conditional pattern bases and sub-conditional pattern tree recursively. And the results of the experiments show that it works faster than Apriori.**

Keywords: **Sequential patterns, Apriori algorithm, FP-tree algorithm, Sequential pattern mining**

I. Introduction

In Sequential Pattern Mining every single page access of a website can be recorded automatically in the web logs by the web server. In general, each line of web logs (one access record) includes the following key information: date-timestamp, client IP address, user ID, requested URL, and HTTP status code. In addition, other less important information such as server IP, OS and browser version is also stored in the web logs. We define web logs to be a collection of sequences comprising web access events from each user during their corresponding session in timestamp ascending order. Preprocessing must be performed on the web logs prior to applying sequential pattern mining. There are many sequential pattern mining algorithms [Agrawal and Srikant, 1995; Pei *et al.,* 2000; Zhou *et al.,* 2004a; Zhou *et al.,* 2004b] available. Generally, these algorithms will be able to generate the same sequential user access patterns based on the same support threshold.

Sequential pattern is a sequence of itemsets that sequentially occurred in a specific order, all items in the same itemsets are supposed to have the same transaction-time value or within a time gap. Usually all the transactions of a customer are together viewed as a sequence, usually called customer-sequence, where each transaction is represented as an itemsets in that sequence, all the communications are list in a certain order with regard to the transaction-time. The basic idea of sequential pattern mining was first introduced by Agrawal and Srikant in 1995 and can be briefly stated as follows. "We are given a set of sequences, called data-sequences, as the input data. Each data-sequence is a list of transactions, where each transaction contains a set of literals, called items. Given a user-specified minimum support threshold, sequential pattern mining finds all of the sequential subsequence's in the sequence database, i.e. the subsequence's whose ratios of appearance exceed the minimum support threshold."

In recent years, sequential pattern mining has largely useful to several application domains, such as market-basket data analysis, medicine, Web log analysis, and telecommunications etc. In the retailing business, sequential patterns can be mined from the transaction records of customers. The retailer can use such information to analyse the habits of the customers, to understand their interests, to satisfy their demands, and above all, to predict their needs. In the medical field, sequential patterns of symptoms and diseases exhibited by patients identify strong symptom/disease correlations that can be an invaluable source of information for medical diagnosis and preventive medicine. In Web log analysis, the exploring behaviour of a user can be extracted from member records or log files. The focus of this paper is to provide an approach for how to use sequential pattern mining techniques for discovering patterns in a Web log.

The organization of the paper is as follows. Section 2 introduces the problem definition. The algorithms used in the pattern discovery phase of the mining process are described briefly in Section 3. The proposed algorithm is described in Section 4. The results of the mining process can be found in Section 5.

II. Problem Definition

This section presents the formal definition of the problem of sequential pattern mining and its application to mining the web log.

Let I = {x_1, x_2, x_3 … x_n} be a set of items. An itemset X that is also called pattern is a subset of I, denoted by X⊆I. A transaction TX = (TID, X) is a pair, where X is a pattern and TID is its unique identifier. A transaction TX is said to contain TY if and only if Y⊆ X. A transaction database, named TDB, is a set of transactions. The number of transactions in DB that contain X is called the support of X. A pattern X is a sequential pattern, if and only if its support is larger than or equal to s, where s is a threshold called minimum support. Given a transaction database, TDB, and a minimum support threshold, s, the problem of finding the complete set of sequential itemsets is called the sequential itemsets mining problem.

III. Algorithms

A. Apriori Algorithm

Agrawal firstly proposed the Apriori algorithm, The Apriori algorithm is the most well known association rule algorithm and is used in most commercial products. The use of support for pruning candidate itemsets is guided by the following principles.

Property 1: If an itemset is sequential, then all of its subsets must also be sequential.

Property 2: If an itemset is insequential, then all of its supersets must also be insequential.

The algorithm initially scans the database to count the support of each item. Upon completion of this step, the set of all sequential 1-itemsets, F1, will be known. Next, the algorithm will iteratively generate new candidate k-itemsets using the sequential (k-1)-itemsets found in the previous iteration. Candidate generation is implemented using a function called Apriori-gen.

To count the support of the candidates, the algorithm needs to make an additional scan over the database. The subset function is used to determine all the candidate itemsets in Ck that are contained in each transaction t. After counting their supports, the algorithm eliminates all candidate itemsets whose support counts are less than minsup. The algorithm terminates when there are no new sequential itemsets generated .

B. FP-Tree Algorithm

Han *et al.* developed an efficient algorithm FP-tree. It mining sequential itemsets without generating candidates, this approach scans the database only twice [3]. The first scan is to find 1- sequential itemset, the second scan is to construct the FP-tree. The FP-tree has sufficient information to mine complete sequential patterns, it consists of a prefix-tree of sequential 1-itemset and a sequential-item header table in which the items are arranged in order of decreasing support value. Each node in the prefix-tree has three fields: itemname, count, and node-link. item-name is the name of the item. count is the number of transactions that consist of the sequential 1-items on the path from the root to this node. node-link is the link to the next same item-name node in the FP-tree.

Each entry in the sequential-item header table has two fields: item-name and head of node-link. item-name is the name of the item. head of node-link is the link to the first same itemname node in the prefix-tree.

IV. Proposed Algorithm

We consider using the apriori method to mining the sequential itemsets basing on the FP-tree, the divide-and-conquer strategy is still adopted by mining process. If there are n 1-sequential items Ii(i=1,2,….n), then the FP-tree can be divided into n conditional sub-tree FPTi (i=1,2,….n) , and FPTi is the conditional subtree associating with sequential item Ii .Then use the apriori algorithm to mine each conditional subtree, and gain all the sequential itemsets with the first prefix item Ii.

The proposed algorithm includes two steps, the first step is to construct the FPtree, the second step is to use of the apriori algorithm to mine the FP-tree. On the second step, it is needed to add an additional node Table, named NTable, each entry in the NTable has two fields: Item-name, and Item-support.

Item-name: the name of the node appears in the FPTi.

Item-support: the number of the node appear with Ii The pseudocode of the algorithm is described below.

ALGORITHM

```
Input: FP-tree, minimum support threshold $\xi$
Output: all sequential itemset L
L = L1;
for each item Ii in header table, in top down order
    LIi = Apriori-mining(Ii) ;
return $L = \{L \cup LI1 \cup LI2 \cup \ldots \cup LI n\}$;
```

pseudocode Apriori-mining(Ii)

```
Find item p in the header table which has the same name with Ii;
q = p.tablelink;
while q is not null
    for each node qi != root on the prefix path of q
        if NTable has a entry N such that N.Item-name= q i.item-name
            N.Item-support = N.Item-support + q.count;
        else
            add an entry N to the NTable;
            N.Item-name = q i. item-name;
            N.Item-support = q.count;
    q = q.tablelink;
k = 1;
$Fk = \{j \mid j \in NTable \wedge j.Item\text{-}support \geqq minsup\}$
repeat
    k = k + 1;
    Ck = apriori-gen(Fk-1) ;
    q = p.tablelink;
    while q is not null
        find prefix path t of q
        Ct = subset(Ck, t);
        for each $c \in C t$
            c.support = c.support + q.count;
        q = q.tablelink;
    $Fk = \{c \mid c \in Ck \in c.support \in minsup\}$
until $Fk = \varphi$
return $LI i = Ii \in F1 \in F2 \in \ldots \in Fk$ // Generate all sequential itemsets which with Ii as the prefix item.
```

V. Experimental Results

In order to verify the performance of the proposed algorithm, we compare it with Apriori algorithm. These algorithms are performed on a computer with a 2.00GHz processor and 512MB memory, running windows vista. The program is developed by Visual C# with ASP.net 2010. We present experimental results using the database. By observing the results we can say that proposed algorithm is more super than apriori ,because it dosen't need to generate 2-candidate itemsets and reduce the search space, and proposed algorithm dosen't need to much extra spaces on the mining process, so proposed algorithm has a better space scalability.

VI. Conclusion

In this paper, a new algorithm is proposed which combined Apriori algorithm and the FP-Tree structure. The experimental result shows that this new algorithm works much faster than Apriori. The future work is to optimize the technique for counting the support of the candidates and expand it for mining larger database.

References

[1] R. Agrawal, and R. Srikant, "Mining sequential patterns", In *Proceedings of 11th International Conference on Data Engineering* (ICDE). Taipei, Taiwan, pp.3-14. 1995.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In VLDBY94, pp. 487-499.

[3] J. Han, J. Pei, and Y. Yin. Mning Sequential Patterns without Candidate Generation (PDF), (Slides), Proc. 2000.

[4] ACM-SIGMOD Int. May 2000. Han J. and Fu Y. "Discovery of Multiple level association rules from large databases". *In Proceedings of the 21st International Conference on Very Large Databases,* Zurich, Switzerland, pp.1-12.1995.

[5] J. Pei, J. Han, and H. Lu. Hmine: Hyper-structure mining of frequent patterns in large databases. In ICDM, pp 441–448, 2001.

[6] Mohammad El - Hajj and Osmar R Zaïane. COFI Approach for Mining Frequent Item sets Revisited, 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD-04), Paris, France, June 2004.