Improving the Efficiency of Web Usage Mining Using K-Apriori and FP-Growth Algorithm

Mrs.R.Kousalya, Ms.K.Suguna, Dr.V. Saravanan

Abstract- Web mining is used to discover the information from the World Wide Web and their usage patterns. Web usage mining is the one of the type of web mining which allows for the collection of Web access information for Web pages. The binary data is transformed into real data by using wiener transformation algorithm. Using k-means algorithm the transformed data are clustered based on the similarities. Then the K-Apriori algorithm is used for data preprocessing to find the frequent patterns. In this paper we proposed the FP-Growth Algorithm on web log files to extract the most frequent pattern.

Keywords - World Wide Web, Web Mining, Web Usage Mining, Apriori, K-Apriori, FP Growth, Clustering.

1 INTRODUCTION

THE World Wide Web enabled the spread of information over the Internet through an easy-to-use and flexible format. The Web is a system of interlinked hypertext documents accessed through the Internet. The web mining is used to extract the useful information from the World Wide Web by using data mining techniques. Web mining can be classified into three categories: (i) web content mining, (ii) web structure mining and (iii) web usage mining. The web content mining is used to search the web pages via content. The web structure mining is used discover the web page structure and hierarchy of hyper links in the web site.

The web usage mining is the task of applying data mining techniques to discover usage patterns from Web data in order to understand and better serve the needs of users navigating on the Web. It is used to discover the navigation patterns from web data, predicts the user behavior while the user interacts with the web and also it helps to improve large collection of resources.

Web usage mining consists of three main steps: (i) preprocessing, (ii) pattern discovery and (iii) pattern analysis. The data extracted from the web log files are preprocessed to remove the noisy data. The data collection should be done before the preprocessing phase.

In data preprocessing phase the data are cleaned and the frequent patterns are mined. The preprocessed file consists of information such as who accessed the web site, what pages where accessed and how long the user accessed that page. In pattern discovery phase the activities of the users on the web are discovered. The frequent patterns discovery phase needs only the Web pages visited by a given user. In this stage the sequences of the pages are irrelevant. Also the duplicates of the same pages are omitted, and the pages are ordered in a predefined order. In pattern analysis phase the patterns extracted from the pattern discovery phase are processed to get most frequent pattern.

2 WEB USAGE MINING

Web usage mining is the process of finding the navigation patterns and their use of web resources. The data collection, data preprocessing, pattern discovery and pattern analysis are the major task of the web usage mining. [2]

2.1 *Data Collection*

The data collection phase contains the raw data from the data source. In this paper the binary database is the data source. The binary database should be transformed into real data by using wiener transformation algorithm to make the data to ready for the pre processing.

2.2. Data preprocessing

a) Data cleaning: The web log file may consist of some irrelevant data so it is necessary to remove that unwanted data. In this phase that unwanted data are removed from the given dataset.

b) Path completion: In this path completion phase the user access paths are identified and the missing paths are added.

c) Session Identification: The timeout method is used to identify the individual user session, if the page request time exceeds a certain limit that implies user is started a new session.

d) User Identification: Users are identified, who contact web server, requesting for some resource on the web.

2.3. K-Apriori

a) Apriori: Apriori Algorithm is an influential algorithm for mining frequent item sets for boolean association rules. The Apriori algorithm is used in data mining process for mining frequent patterns from the given data set. This algorithm uses an iterative approach called level-wise search, in which n-item sets are used to explore n+1 item

sets. The set of frequent 1_itemsets, frequent 2_itemsets and frequent 3_itemsets are found until no more frequent n_itemsets can be found.

Some of the issues of Apriori algorithm are: Database scanning of the whole dataset for each iteration, the computational efficiency is very less because the whole database scans is needed every time, the cost of generating large number of candidate sets and scanning the database repeatedly. The repeated scan of the database is very costly.

To overcome these issues a new frequent item sets mining method K_Apriori is introduced.

b) K-Apriori: In K-Apriori algorithm, the binary data is transformed into real domain using linear wiener transformation, based on its neighborhood property. The Wiener transformed data is partitioned into K clusters using the multi-pass K-means algorithm. Apriori procedure is used for the K similar groups of data from which, frequent item sets can be generated and association rules are derived. Large datasets are partitioned so that the candidate item sets generated will be very less and database scanning will also be done for adequate data which increases the efficiency. [1]The K-Apriori algorithm is described in Algorithm1.

Step3: Compute the new cluster centroids $Z_1^*, Z_2^*, \ldots, Z_K^*$ as

$$Z_i^{\square} = \frac{1}{l_j} \sum X_j \in C_j \, X_i$$

Algorithm 1 (K-Apriori Algorithm for Frequent Item set Mining)

Input: Binary data matrix X of size p x q, K
Output: Frequent Item sets and Association rules
V = Call function wiener2 (X_i)
$C_1, C_2 \ldots CK$ = Call function kmeans (V, K)
For each cluster C_i
Cd_n: Candidate item set of size n
L_n: frequent item set of size n
L_1 = {frequent items};
For (n=1; L_n! = ϕ; n++)
Do begin
Cd_{n+1} = candidates generated from L_n;
For each transaction T in database do
Increment the count of all candidates in Cd_{n+1}
which are contained in T
L_{n+1}= candidates in Cd_{n+1} with min_support
End
µnLn are the frequent item sets generated End
End

Function wiener2 (X_i)

Input : Binary data vector X_i of size 1 X q
Output : Transformed data vector Y_i of size 1 X q

Step 1: Calculate the mean µ for the input vector X_i around each element

$$\mu = \frac{1}{pq} \sum_{n1,n2 \in \eta} X(n_1, n_2)$$

Where η is the local neighborhood of each element

Step 2: Calculate the variance σ^2 around each element for the vector

$$\sigma^2 = \frac{1}{pq} \sum_{n1,n2 \in \eta} X^2((n_1, n_2) - \mu)$$

Where η is the local neighborhood of each element

Step 3: Perform wiener transformation for each element in the vector using equation Y based on its neighborhood

$$Y(n_1, n_2) = \mu + \frac{\sigma^2 - \lambda^2}{\sigma} (X(n_1, n_2) - \mu)$$

Where λ^2 is the average of all the local estimated variances.

Function kmeans (V, K)

Input: Wiener Transformed data matrix V and number of clusters K.
Output: K clusters

Step 1: Choose initial cluster centroids $Z_1, Z_2, \ldots, Z_K$ randomly from the N points; $X_1, X_2, \ldots X_p$, $X_i \in R^q$
Where q is the number of features/attributes

Step 2: Assign point X_i, i = 1, 2, …, p to cluster C_j, where j = 1,2,…,K, if and only if

$|| X_i - Z_j || < || X_i - Z_t ||$, t = 1, 2,…,K. and j ≠ t.

Ties are resolved arbitrarily.
Where i = 1, 2,…,K, and l_j =Number of points in C_j.

Step 4: If $Z_i^* = Z_i$, i = 1, 2,…, K then terminate.
Otherwise $Z_i \in Z_i^*$ and go to step 2.

EXAMPLE 1:
Support 25%
CLUSTER 1:

A, H, M, O, R: 1-Itemset
OR, MR, MO, HO, HM, AM: 2-Itemsets
HMO, MOR: 3-Itemsets

A		AH, AM, AO, AR
H		HM, HO, HR
M	⟶	MO, MR
O		OR
R		OR
MR		
MO	⟶	HMO, MOR
HO		
HM		
AM		

100%Confidence:

A->M H->A HO->M

Figure 1: Cluster 1 with 25% Support

CLUSTER 2:

H, M: 1-Itemset
HM: 2-Itemsets
H
M ⟶ HM
90 % CONFIDENCE:
H -> M

Figure 2: Cluster 2 with 25% Support

2.4 Pattern discovery

(a) FP Growth: The proposed system is frequent pattern tree structure, which is used to develop an efficient FP Tree based mining method. FP Growth algorithm generates frequent item sets from FP-Tree by traversing in bottom up fashion. [4]

The major steps of FP-Growth algorithm:

Step 1: Construct the conditional pattern base for each item set in the header table. Then start at the bottom of frequent item header table in the FP-Tree.

Step 2: For each pattern base.

- Accumulate the count for each item in the base
- Construct the conditional FP-Tree for the frequent item sets of the pattern base

Step 3: Recursively mine the conditional FP-Tree.

Algorithm 2 (FP-growth: Mining frequent patterns with FP-tree by pattern fragment growth).

Input: A database DB, represented by FP-tree constructed according to Apriori, and
A minimum support threshold ξ.
Output: The complete set of frequent patterns.
Method: call FP-growth (FP-tree, null).
Procedure FP-growth (Tree, α)
{
If Tree contains a single prefix path
Then
{
Let P be the single prefix-path part of Tree;
Let Q be the multipath part with the top branching node replaced by a null root;
For each combination (denoted as β) of the nodes in the path P do
Generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in β;
Let freq pattern set (P) be the set of patterns so generated;
}
Else let Q is Tree;
For each item ai in Q does
{
Generate pattern β = ai $\cup \alpha$ with support = ai .support;
Construct ß's conditional pattern-base and then ß's conditional FP-tree Tree β;
If Tree$\beta \neq \emptyset$
Then call FP-growth (Treeβ, β);
Let freq pattern set (Q) be the set of patterns so generated;
}
Return (freq pattern set (P) $\cup$ freq pattern set (Q) $\cup$ (freq pattern set (P)
×freq pattern set (Q)))
}

EXAMPLE 2:

TID	ITEMSET
T100	K O N M E Y
T200	K O N D E Y
T300	M A K E
T400	M I C K Y
T500	C O O K I E

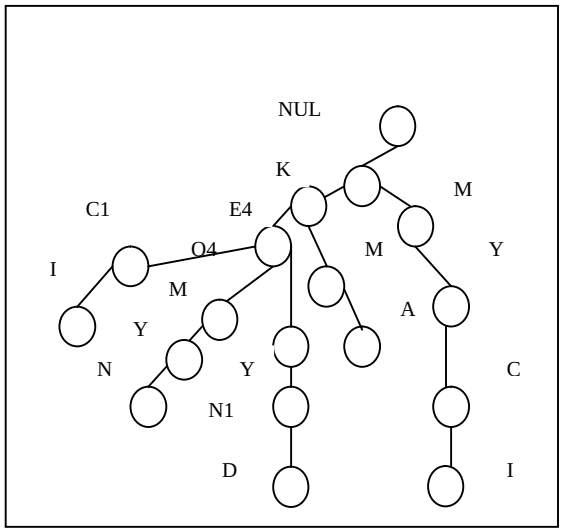
STEP 1: Find the support count of each item

STEP 2: items in descending order

ITEMSET	SUPPORT
K	5
E	4
O	4
M	3
Y	3
C	2
N	2
I	2
A	1
D	1

(FP tree is generated from step 3 to step 7)

STEP 7:
T500 : C O O K I E
Lorder: K E O O C I



2.5 Pattern Analysis

The pattern analysis is used to find frequent pattern in web usage mining, in this phase the most frequent pattern are mined.

3 RESULTS

ITEMSET	SUPPORT
A	1
C	2
D	1
E	4
K	5
I	2
M	3
N	2
O	4
Y	3

In this paper the performance of K-Apriori and FP Growth Algorithm are compared. The Graph represents the performance analysis of K-Apriori and FP Growth algorithm. The result of this graph is that FP Growth Algorithm is more efficient because the most frequent item sets are mined compared to K-Apriori Algorithm.

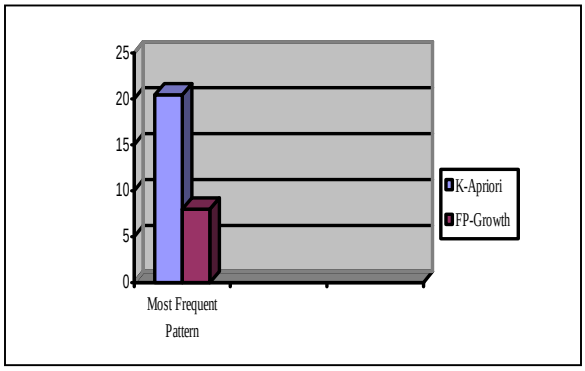


Figure 3: Performance Analysis of K-Apriori and FP Growth

4 CONCLUSIONS

The data from the web log files are preprocessed and the preprocessed data are stored in the database. The frequent pattern mining algorithm is applied to that data to get a most frequent pattern from the web log files.

REFERENCES

[1] Ashok Kumar D.Loraine Charlet Annie M.C., "web log mining using K-Apriori Algorithm", volume 41, March -2012,
[2] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Hua Zhu, "Mining Access Pattern Efficient from Web Logs"
[3] B.Santhosh Kumar, K.V. Rukmani," Implementation of Web Usage Mining Using Apriori and FP-Growth Algorithms", volume: 01, Issue: 06, Pages: 400-404(2010)
[4] J.Han and Kamber,"Data Mining: Concepts and Techniques", Morgan Kaufman Publishers, 2000
[5] Jiawei Han, Ian Pei, Yiwen Tin, Runying Mao, "Mining Frequent Pattern without Candidate Generation: A Frequent Pattern Tree Approach", Volume-8
[6] Harish Kumar and Anil Kumar," Clustering Algorithm Employ in Web Usage Mining: An Overview", INDIA Com publication, Edition 2011.
[7] Rajan Chattamvelli, "Data Mining Methods",
[8] Narosa publications, Edition 2009.