Web Usage Mining Using D-Apriori And DFP Algorithm

Mrs.R.Kousalya
PhD. Scholar,
Manonmaniam sundaranar university, Tirunelveli
HOD/ Asst professor,
Dr. N.G.P. Arts and Science College,
Coimbatore-641 048, India
Mobile no. +91 9894656526
kousalyacbe@gmail.com

Ms.S.Pradeepa
M.phil.Scholar,
Department of Computer Science
Dr.N.G.P. Arts and Science College,
Coimbatore-641 048, India
Mobile no. +91 9489551185
Prathy.it@gmail.com

Dr.V.Saravanan
Professor & Director
Department of Computer Application
Sri Venkateswara College of Computer Application and Management
Coimbatore-641 105, India
Mobile no. +91 9894656526
Profsaran@hotmail.com

Abstract: Web Usage Mining is a branch of web mining. The data is assembled has result in awfully large information in web access and it represent in binary form. The data is grouped the neighborhood data by using divisive clustering method. The divisive analysis is one of the types of hierarchical method of clustering, the divisive analysis is used to separate each dataset from the clustered dataset. Here the new algorithm D-Apriori and DFP is proposed to find the frequently accessed webpage from web log database.

Index Terms: Apriori, Clustering, D-Apriori, DFP Algorithm, FP Algorithm and Web Usage Mining.

1 INTRODUCTION

DATA mining is the non-trivial extraction of implicit earlier unidentified and potentially useful information about data. Conventionally, the mined information is represented as a model of the semantic formation of the dataset [1].

Clustering is grouping the neighborhood object. The similar data are grouped together to form a node is called cluster [7]. Hierarchical clustering is a kind of clustering analysis. The hierarchical clustering is divided into two types: Agglomerative and Divisive clustering [8]. Divisive is a "top down" approach, all details combined in one cluster and splits into the many nodes [9]. Web mining is the application of data mining techniques to discover patterns from the Web. Web usage mining is the method of extracting valuable information from server logs. A Web Usage Mining process is divided into three phases: data preprocessing, patterns discovery and pattern analysis [3]. Web log databases are huge binary databases maintained by web servers. The databases are updated every time so it will be very complex for processing. At the end of the day, the pages visited or accessed need to be recognized. These web servers are used for website maintenance and report generation [2].

In this paper, the data is grouped the neighborhood datasets by using clustering method. In hierarchical method, the divisive analysis is used separate each grouped datasets from the cluster nodes. Here the new algorithm D-Apriori and DFP is proposed to find the frequently accessed webpage from web log database. Data is preprocessed by using D-Apriori. In Pattern Discovery the new algorithm DFP is used to mine the frequent accessed data from the cluster and data is analysis graphically by using pattern analysis.

2 HIERARCHICAL CLUSTERING

Hierarchical clustering is a process of cluster analysis which seeks to assemble a hierarchy of clusters [10] [6]. Strategies for hierarchical clustering generally fall into two types:

2.1 Agglomerative Analysis

This is a "bottom up" approach. Each observation starts in its own cluster and pairs of clusters are merged as one move up the hierarchy.

2.2 Divisive Analysis

This is a "top down" approach [9]. All explanation start in one cluster and splits are performed recursively as one move down the hierarchy. Here the datasets are clustered using divisive analysis, the clustered datasets are split into a single cluster.

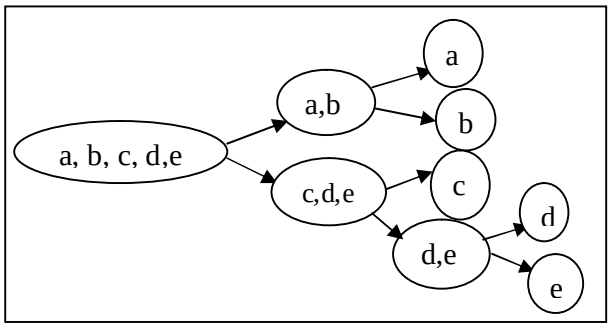

Figure 1: Divisive Analysis

3 WEBUSAGE MINING

Web mining is the application of data mining techniques to discover patterns from the Web. According to analysis targets, web mining can be divided into three different types, which are Web usage mining, Web content mining and Web structure mining. The process of Web Usage Mining consists of three main steps are Data Preprocessing, Pattern Discovery and Pattern Analysis [3].

3.1 DATA PREPROCESSING

In this phase, the D-Apriori applied in the processing to find the frequent accesses clustered dataset of the web log file. The data is preprocessed in data cleaning, user identification, session identification and path completion.

The data is preprocessed by using D-Apriori. The clustered datasets are separated by using divisive analysis, the splited cluster datasets are used in Apriori algorithm. Apriori algorithm is used for mining frequent item sets which are used for Boolean association rules generation. The wiener transformation is to convert the binary preprocesses data into real data. D-Apriori algorithm is to mine the frequent occurring nodes.

3.1.1 Apriori Algorithm

Apriori is designed to operate on databases containing transactions. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time and groups of candidates are tested against the data. Apriori algorithm for mining frequent item sets which are used for Boolean association rules generation. Apriori algorithm is a level-wise, breadth-first algorithm which counts transactions, which is explained in Algorithm 1[4]. Apriori uses an iterative approach known as a level-wise search, in which n-item sets are used to explore (n+1)-item sets. First, the set of frequent 1-itemsets is found. This set is denoted P1. P1 is used to find P2, the frequent 2-itemsets, which is used to find P3, and so on, until no more frequent n-item sets can be found. Finding of each Pn requires one full scan of the database.

Algorithm 1: Apriori algorithm for Frequent Item set Mining

Cdn: Candidate item set of size n
Pn: frequent item set of size n
P1 = {frequent items}; For (n=1; Ln! =∅; n++)
Do begin
Cdn+1 = candidates generated from Pn;
For each transaction T in database do Increment the count of all candidates in Cdn+1 that are contained in T
Pn+1= candidates in Cdn+1 with min_support
End
Return Un Pn

The main purpose of K-Apriori algorithm is scanning the clustered dataset for every process from the database, so the computational and scanning time is very high. So to overcome from this the D-Apriori is proposed to reduce the computation and scanning time.

3.1.2 WENIER TRANSFORMATION

The binary data is pre-processed by transforming into real data using the Wiener Transformation. The input for wiener transformation is fixed with known autocorrelation. It is a causal transformation. The Wiener transformation is best in terms of the mean square error. The syntax for Wiener filter is Y = wiener2 (X, [q, r], noise) for two-dimensional images, which is normally used for image restoration [4]. The same equation is used for data mining task of web log databases. Wiener method based on statistics, estimated from a local neighborhood of each element. Wiener estimates the local mean μ and variance σ^2 around each element of the matrix using the equations (1) and (2) given below.

(1) $\mu = 1/qr \Sigma n1, n2 E \acute{\eta} X(n1,n2)$ (1)

(2) $\sigma^2 = 1/qr \Sigma n1,n2 E \acute{\eta} X^2((n1,n2) - \mu)$ (2)

Where $\acute{\eta}$ is the local neighborhood of each element in the input matrix W. Wiener2 then creates a element-wise wiener transformation for each vector based on the neighborhood of the objects using equation (2) and (3) estimates in equation(3),

(3) $Y(n1, n2) = \mu + \sigma^2 + \lambda^2 / \sigma^2 (X((n1, n2) - \mu))$ (3)

where λ^2 is the average of all the local estimated variances. Since clustering finds similarity between objects, neighborhood property of the wiener transformation helps to find good clusters and makes it computationally efficient.

3.1.3 D-APRIORI ALGORITHM

In D-Apriori algorithm, the clustered datasets are separated by using Divisive algorithm. The clustered datasets is applied in Apriori algorithm to filter the frequent occurring web log files, the wiener transformation is to transform the binary

data into real domain. Large neighborhood dataset is grouped together and split into many nodes, so the database scanning time is reduce and the efficiency is increased. The D-Apriori algorithm is describes in Algorithm 2.

Algorithm2: D-Apriori Algorithm for Frequent Item set Mining

Input: Binary data matrix X of size p x q, D
Output: Frequent Item sets and Apriori
G = Call function wiener2 (Xi)
C1, C2 …Cn = Call function divisive (G,D)
For each cluster Ci; Cdn: Candidate item set of size n; Ln: frequent item set of size n
L1 = {frequent items}; For (n=1; Ln! = φ; n++)
Do begin
Cdn+1 = candidates generated from Ln;
For each transaction T in database do Increment the count of all candidates in Cdn+1 which are contained in T
Ln+1= candidates in Cdn+1 with min_support
End
∪nLn are the frequent item sets generated
End
End

Function wiener2 (Xi)

Input: Binary data vector Xi of size 1 X q
Output: Transformed data vector Yi of size 1 X q
Step 1: Calculate the mean μ for the input vector Xi around each element $\mu = 1/pq \sum n1, n2 \in \eta X(n1, n2)$ where η is the local neighborhood of each element.
Step 2: Calculate the variance σ^2 around each element for the vector $\sigma^2 = 1pq \sum n1, n2 \in \eta X^2((n1, n2) - \mu)$ 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 $Y(n1,n2) = \mu + \sigma^2 - \lambda^2 \sigma^2 (X(n1,,n2) - \mu)$ where λ^2 is the average of all the local estimated variances.

Function divisive (D, G)

Input: Wiener Transformed data matrix G and number of clusters D.
Output: D clusters nodes

1. Find the object, which has the highest average dissimilarity to all other objects. This object initiates a new cluster– a sort of a *splinter group.*
2. For each object *i* outside the *splinter group* compute.
3. D_i = [average $d(i,j)$ j $R_{splinter\ group}$] - [average $d(i,j)$ j $R_{splinter\ group}$]
4. Find an object *h* for which the difference D_h is the largest. If D_h is positive, then *h* is, on the average close to the splinter group.
5. Repeat *Steps* 2 and 3 until all differences D_h are negative. The data set is then split into two clusters.
6. Select the cluster with the largest diameter. The diameter of a cluster is the largest dissimilarity between any two of its objects. Then divide this cluster, following steps 1-4.
7. Repeat *Step* 5 until all clusters contain only a single object.

The confidence of K-Apriori is 67%. The calculated confidence result of the D-Apriori is 75%. Compare to K-Apriori D-Apriori confidence is high.

3.2 PATTERN DISCOVERY

In this phase, the DFP is applied on each clustered datasets to extract meaningful patterns after preprocessing the clustering nodes.
Pattern discovery is performed only after cleaning the data and after the identification of user transactions and sessions from the access logs [5]. The frequent accessed datasets are mined by using DFP in pattern discovery.

3.2.1 DFP ALGORITHM

The DFP is Divisive FP-growth. DFP is proposed to find the frequently accessed webpage from web log database. The frequent accessed datasets are mined by using DFP in pattern discovery.

Input: Web log File
Output: Frequent Pattern grouped datasets

A. Divisive Analysis

1. Find the object, which has the highest average dissimilarity to all other objects. This object initiates a new cluster– a sort of a *splinter group.*
2. For each object *i* outside the *splinter group* compute.
3. D_i = [average $d(i,j)$ j $R_{splinter\ group}$] - [average $d(i,j)$ j $R_{splinter\ group}$]
4. Find an object *h* for which the difference D_h is the largest. If D_h is positive, then *h* is, on the average close to the splinter group.
5. Repeat *Steps* 2 and 3 until all differences D_h are negative. The data set is then split into two clusters.
6. Select the cluster with the largest diameter. The diameter of a cluster is the largest dissimilarity between any two of its objects. Then divide this cluster, following steps 1-4.
7. Repeat *Step* 5 until all clusters contain only a single object.

B. Mining frequent pattern

1. FP-tree construction

The transaction database D is scan once. The root of an FP-tree is "null". Select the grouped dataset for transaction D. Let P be the element insert the tree. If E has the child node F such that F.item-name =p.item-name, then increment N's count 1,else create a new node N and let it count to be 1.Its parents link is linked to T and its node-link to the grouped dataset with the same item name. If P is nonempty, call insert tree (P, N) recursively.

2. Mining the DFP-tree

The FP-tree is mined by calling FP-growth (FP-tree, x)

FP-growth (Tree, x)

a) If Tree contains a single path P.
b) Then each combination is denotes as y, the grouped datasets in the path P do
c) Generate pattern y U x with support = minimum support of the grouped datasets in y;
d) Else for each a in the header of Tree do{
e) Generate pattern y=ai U x with support = ai.support;
f) Construct y's conditional pattern base and then y's conditional FP-tree Tree y;
g) If Tree Y=75
h) Then call FP-growth(Tree y, y)
i) The clustering groups the similar web access records from the web log files. The clustered datasets have been mined by using DFP –tree algorithm. The frequent accesses webpage are clustered by using DFP.

3.3 PATTERN ANALYSIS

In this phase, uninteresting patterns are removed from the patterns identified during pattern discovery phase and the data is analysis graphically.

In Pattern Analysis the uninteresting patterns are removed from the clusters and the pattern is identified during pattern discovery. There are two most common approaches for the pattern analysis: SQL query mechanism and constructing multi-dimensional data cube to perform OLAP operations [1].

4 RESULT

The result of the proposed algorithm is performing well in computation time. The log files and computation time is measured in this graph. In graph the D-Apriori and K-Apriori is compared, here the blue line denotes the D-Apriori and the red line denotes the K-Apriori, the result of D-Apriori is more efficient than the K-Apriori. The DFP and KFP is compared and the result is produced. The green line denotes DFP and the violet line denotes the KFP. The result of DFP is more efficient that the KFP.

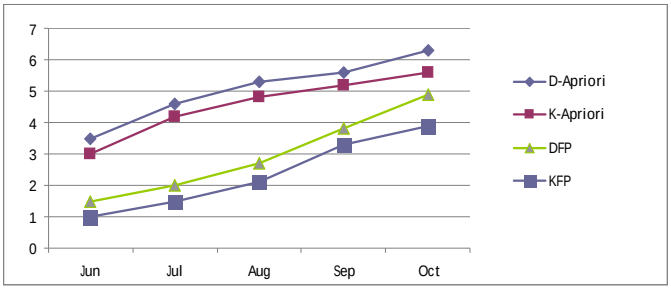

Figure 2: Comparing the D-Apriori and K-Apriori, DFP and KFP.

5 CONCLUSION

In this paper the D-Apriori and DFP is proposed to find the frequently accessed webpage from web log database. The D-Apriori and DFP takes less time for computation and more efficient compare to K-Apriori and KFP.

REFERENCES

[1] Jiawei Han and Michelin Kamber, "Data mining Concepts and Techniques", Elsevier publication, Edition 2006.
[2] Rajan Chattamvelli, "Data Mining Methods", Narosa publications, Edition 2009.
[3] Santhosh Kumar and Rukumani,"web usage mining", ijana publication, vol.1, pages 400-404, Edition 2010.
[4] Ashok Kumar D, Loraine Charlet Annie M.C, "Web log mining using K-Apriori algorithm", ijca publication, vol.41 Edition march 2012.
[5] Shyam Sundar Meena, "Efficient discovery of frequent pattern using KFP-Tree from web logs", ijca publication, vol.49, Edition July 2012.
[6] G.Sudamathy and C.Jothi venkateshwaran, "An efficient hierarchical frequent pattern analysis approach for web usage", ijca publication, vol.43, Edition 2012.
[7] Jianhan Zhu, Jun Hong and John G. Hughes, "Page clustering: Mining conceptual link hierarchical from web log files for adaptive websites navigation", ACM publication, vol.4, Edition 2004.
[8] Harish Kumar and Anil Kumar," Clustering Algorithm Employ in Web Usage Mining: An Overview", INDIA Com publication, Edition 2011.
[9] Idams, "Divisive analysis(DAINA) Algorithm", Uneseo publication, chapter 7.1.5, Edition 2005.
[10] Hussain T, "A hierarchical cluster based preprocessing methodology for web usage mining", IEEE publication, Edition 2010.