WEB USAGE MINING USING APRIORI AND FP GROWTH ALGORITHM

Girija Patil1, Priyanka Patkar2, Aanum Shaikh3, Aditi Thakkar4, Prof.Sumitra Sadhukhan5
aditithakkar_93@yahoo.co.in, aanumshaikh@ymail.com
Department of Computer Engineering
MCT’s Rajiv Gandhi Institute Of Technology
Andheri(W), Mumbai-53, India.

Abstract

In order to suffice the requirements of various web based applications that are growing at a bullet speed, Web Usage Mining has proved to be the most efficient escape route. It does so by discovering interesting and most frequent patterns based on users’ navigational behaviours. Usage data encapsulates the identity or origin of web users. Web server log files act as storage for frequent word sequences that are initially in textual format. The crucial information extracted is discovered with the application of association rules about users’ behaviours. This information collected comprises of the time that the user accessed the web, users’ IP address and pages referenced. The recent developments in the field of web usage mining are extraction of knowledge from the logging information produced by web servers. This study is accomplished by the two association rule algorithms namely, FP Growth and Apriori algorithm.

© 2014 The Authors. Published by Elsevier Ltd.
Selection and peer-review under responsibility of Organizing Committee of ICACIT 2014.

Keywords: Apriori; FP Growth; Web Usage Mining; Web server log files;

1. Introduction

The web is no more a term that needs to be looked into the dictionary for its meaning. It is a variant, vast and dynamic data repository that comprises of mostly raw data which is a source to the enormous supply of information. The information excavated from this repository is watched out by users, web service providers, business analysts, thus making it even complex to be dealt with. The web users hence, want to have the effective search tools to find relevant information easily and precisely. Data mining is the process of excavation for finding out knowledge from data.

Web mining is the process of excavating information and patterns from web. It is used to understand how the customer behaves, how effective is that particular website, and guides how the marketing of this website can be successful. The nature of users' navigational behaviour generates a pattern in the data. The tools required for looking into these patterns are web usage, content and structure mining. Web usage mining is applied to many real world problems to discover interesting user navigation patterns for improvement of web site design by making additional topic or recommendations observing user or customer behaviour.

This application of data mining technique i.e. web usage mining helps discover intriguing patterns from web database, for having best efforts put forth to satisfy the requirements of various web-based applications. Usage data encapsulates the identity or origin of web users. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web and application server data, application level data. Web server database comprises of users’ logs that are received at Web server. The crucial information extracted is discovered with the application of association rules about users’ behaviours. This information collected comprises of the time that the
user accessed the web, users’ IP address and pages referenced. This work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files.

Web usage mining is the application of data mining that apply data mining techniques to discover the behaviour pattern using web data. Web usage mining process is generally divided into three tasks: pre-processing, pattern analysis and discovery. Pre-processing includes the amalgamation and synchronism and episode recognition, user and session testimony, the assimilation of click stream data with order data sources such as content or semantic information. In the pattern analysis phase interesting knowledge is extracted from frequent patterns and these results are used for website modifications. In pattern discovery phase, frequent pattern discovery algorithms are applied on raw data. For finding out the information that is hidden in web logs, several data mining techniques are applied on web server logs.

This paper showcases the provisional research between the two association rule algorithms namely, FP Growth and Apriori algorithm.

1.1. Apriori algorithm

Apriori algorithm captures large data sets during its initial database passes and uses this result as the base for discovering other large datasets during subsequent passes. Large or frequent item sets are the one which have support level above minimum, while item sets having support level below minimum value are called small item sets. This algorithm mainly follows large item set property which states that: the subset of frequent item sets must be frequent and subset of large item set is large. The traditional algorithm for mining all frequent item sets and strong association rules was the AIS algorithm. After some days this algorithm was modified and named as apriori. Apriori algorithm is, the most supervised and important algorithm for mining frequent item sets.

To get the count of candidate item sets efficiently, apriori algorithm makes use of the breadth-first search and hash tree structure. From item sets of length k-1 candidate item sets of length k is generated. It then cuts off those candidate sets which are infrequent in sub pattern. As stated by downward closure lemma, the candidate set consists of all frequent k-length item sets. After performing this, it checks for the transaction database to determine frequent item sets among the candidates. Apriori is a supervised algorithm for mining frequent item sets for Boolean association rules. As the Algorithm uses previous knowledge of frequent item set it has been given the name Apriori. It is a repeated level wise search Algorithm, where k item sets are used to traverse (k+1)-item sets. First, the set of frequent item sets is generated. This set is denoted by L1. L1 is used to find L2, the set of frequent item sets, which is used to find L3 and it goes on, until no more frequent item sets can be found. In order to find each Lk scanning of full database is required.

General Process:
Association rule generation comprises of two separate steps:
1. First, minimum support is used to find all frequent item sets in a database of user.
2. Second, these frequent item sets and the minimum confidence constraint are used to form association rules.

Algorithm:
Procedure Apriori (T, min Support) {//T is the database and min Support is the minimum support
L1= {frequent items};
For (k= 2; Lk-1! =null; k++) {
Ck= candidates generated from Lk-1
For each transaction t in database do {
#increment the count of all candidates in Ck that are contained in t
Lk = candidates in Ck with min Support
}//end for each
}//end for
Return;
}
Apriori, which is traditionally significant, suffers from a number of difficulties or trade-offs, which have led to the generation of other algorithms. Candidate generation create large numbers of subsets (maximum candidate sets are loaded up which is the main goal of algorithm). Bottom-up subset observation finds maximum subset \( S \) only after all \( 2^{|S|} - 1 \) of its proper subsets.

**Advantages:**
- It is straightforward and simple algorithm.
- Implementing this algorithm is easy and elementary.

**Disadvantages:**
- Multiple scan over the database is done in order to generate candidate set.
- Memory consumption, space and time are more for candidate generation process.

1.2. **Fp Growth algorithm**

FP growth algorithm is used to produce frequent item sets. This is achieved with the help of FP-Tree by traversing in bottom up fashion. It allows frequent item set discovery without candidate item set generation. It is a two step approach:

Initially it builds a compact data structure called the FP-tree. It is built using 2 passes over the data-set. Later it extracts frequent item sets directly from FP-tree by traversing through FP-Tree.

**Algorithm:**
It takes as input a database \( D \), represented by FP-tree constructed and a minimum support threshold. It produces a complete set of frequent patterns.

Method: call FP-growth (FP-tree, null).

Procedure FP-growth (Tree, a) {
  if Tree contains a single prefix path then // Mining single prefix-path FP-tree {
    let X be the single prefix-path part of Tree;
    let Y be the multipath part with the top branching node replaced by a null root;
    for each combination (denoted as \( \beta \)) of the nodes in the path X do
      generate pattern \( \beta \cup a \) with support = minimum support of nodes in \( \beta \);
      let frequent pattern set(X) be the set of patterns so generated;
  }
  else let Y be Tree;
  for each item in Y do { // Mining multipath FP-tree
    generate pattern \( \beta = a_i \cup a \) with support = \( a_i \) support;
    Build \( \beta \)’s conditional pattern base and then \( \beta \)’s conditional FP-tree Tree \( \beta \);
    if Tree \( \beta \neq \emptyset \) then
      call FP-growth(Tree \( \beta \), \( \beta \));
    let frequent pattern set(Y) be the set of patterns so generated;
  }
  return(frequent pattern set(X) \cup frequent pattern set(Y) \cup (frequent pattern set(X) \times frequent pattern set(Y)))
}

When the FP-tree consists of a single prefix-path, the entire set of frequent patterns can be generated in three parts: the single prefix-path \( X \), the multipath \( Y \), and their combinations. The resulting patterns for a single prefix path are the enumerations of its sub paths that have the minimum support. Thereafter, the multipath \( Y \) is defined and the resulting patterns from it are processed. Finally, in line 14 the merged results are returned as the frequent patterns found.

**Advantages:**
- It uses Compact data structure.
• It eliminates repeated database scan.
• It is faster than Apriori algorithm.
• It reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP tree.

Disadvantages:
• It takes more time for recursive calls.
• It is good only when user access paths are common.
• It utilizes more memory

1.3. Web Server Log Files

The main goal is to convert user oriented input into a computer-based format i.e. to learn the user’s interests by converting the log records into beneficial outcomes. This is done using web log files. The log file entries produced in common log format will look something like this:

```
```

A log file is a text file in which every page request made to the web server is recorded. Log files are files that list the actions that have been occurred. These log files reside in the web servers, web proxy servers and client browsers. The web log file has the extension .log and contains ASCII characters. Log files contain the following information:

• **User name:** This identifies who had visited the web site. The identification of the user mostly would be the IP address that is assigned by the Internet Service provider (ISP). This may be a temporary address that has been assigned. Therefore here the unique identification of the user is lagging. In some web sites the user identification is made by getting the user profile and allows them to access the web site by using a user name and password. In this kind of access the user is being identified uniquely so that the revisit of the user can also be identified.

• **Visiting Path:** The path taken by the user while visiting the web site. This may be by using the URL directly or by clicking on a link or trough a search engine.

• **Path Traversed:** This identifies the path taken by the user within the web site using the various links.

• **Time stamp:** The time spent by the user in each web page while surfing through the web site. This is identified as the session.

• **Page last visited:** The page that was visited by the user before he or she leaves the web site.

• **Success rate:** The success rate of the web site can be determined by the number of downloads made and the number copying activity undergone by the user. If any purchase of things or software made, this would also add up the success rate.

• **User Agent:** This is nothing but the browser from where the user sends the request to the web server. It’s just a string describing the type and version of browser software being used.

• **URL:** The resource accessed by the user. It may be an HTML page, a CGI program, or a script.

• **Request type:** The method used for information transfer is noted. The methods like GET, POST. These are the contents present in the log file.

Web usage mining mines the highly utilized web sites. The utilisation would be the frequently visited web site or the web site being utilized for longer time duration. Therefore the quantitative usage of the web site can be analysed if the log file is well analysed.
1.3.1. Input
The web log files act as an input for interpretation of user behaviour. This information of web logs can be used to reconstruct the user navigation sessions within the site from which the log data originates. In an ideal scenario, each user is allocated a unique IP address whenever an access is made to a given web site. Moreover, it is expected that a user visits the site more than once, each time possibly with a different goal in mind.

1.3.2. Output
Learning the user’s expectation is a very tedious process. A single word may have different views by different users. Posing questions to the user every time he makes a search would be a tiring and uninteresting process for a user. Therefore the user’s interest can be analysed by the first attempt made to open a page. Then the next step done by the miner is to mine the web once again and provide the list of results meant only for the user’s area of interest. This may in turn minimize the list of options and make the searching process even more effective. Thus, the main objective of any system is the generation of reports that reflect user interests. These reports have several uses, some of which are as follows:

- Source of information required.
- Raw matter to perform personalization.
- Permanent hard copy of the results.

Careful consideration has been given in the designing of the reports as it helps in decision-making process. An example of report generated for the various browsers used is as shown in the figure below. Furthermore, with the knowledge that internet explorer is the most used browser, an even detailed report of its version used can be obtained as shown below:

<table>
<thead>
<tr>
<th>S.No</th>
<th>Browser</th>
<th>Hits</th>
<th>Visits</th>
<th>% of Total Visits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Internet Explorer</td>
<td>21,302</td>
<td>2,260</td>
<td>61.43%</td>
</tr>
<tr>
<td>2</td>
<td>Firefox</td>
<td>6,461</td>
<td>803</td>
<td>20.00%</td>
</tr>
<tr>
<td>3</td>
<td>Others</td>
<td>257</td>
<td>153</td>
<td>4.00%</td>
</tr>
<tr>
<td>4</td>
<td>Opera</td>
<td>40</td>
<td>100</td>
<td>2.00%</td>
</tr>
<tr>
<td>5</td>
<td>Google Desktop</td>
<td>60</td>
<td>50</td>
<td>1.40%</td>
</tr>
<tr>
<td>6</td>
<td>Mobile9.0 (compatible)</td>
<td>148</td>
<td>36</td>
<td>1.03%</td>
</tr>
<tr>
<td>7</td>
<td>ActiveRecode</td>
<td>19</td>
<td>19</td>
<td>0.52%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S.No</th>
<th>Browser</th>
<th>Hits</th>
<th>Visits</th>
<th>% of Total Visits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Internet Explorer 7.x</td>
<td>13,045</td>
<td>1,202</td>
<td>53.19%</td>
</tr>
<tr>
<td>2</td>
<td>Internet Explorer 6.x</td>
<td>8,012</td>
<td>950</td>
<td>42.04%</td>
</tr>
<tr>
<td>3</td>
<td>Internet Explorer 5.x</td>
<td>240</td>
<td>107</td>
<td>4.71%</td>
</tr>
<tr>
<td>4</td>
<td>Internet Explorer 2.x</td>
<td>5</td>
<td>1</td>
<td>0.04%</td>
</tr>
</tbody>
</table>

Total: 21,302, 2,260, 100.00%

Fig. 1. Sample of web log file
Fig. 2. Sample reports of most used browsers and versions
2. Conclusion

This paper analyzes the two association rule based algorithms namely, FP Growth and Apriori algorithm, which meet the needs of various web service providers and various viewers, users, business analysts, etc. It improves the techniques of Web Usage Mining by first discovering the log files of individual users at one place. This collective information consequently can be used to design business strategies to boom revenue, occasionally downstream costs, or both. The Apriori association algorithm is built upon pre-gauges recurrent item sets and it has to browse the entire transaction log/dataset or database which will become a conflict with huge item sets. With FP trees, there is no necessity for generation of candidate sets, as in the Apriori algorithm, and the recurrence of item sets are detected just by passing through the FP tree. This paper discusses the FP Tree and Apriori algorithm concept. We use this approach to determine association rules that occur in the dataset. Subsequently, we can authorize admissible rules and patterns in any set of records.

References


