Int. J. of Advanced Networking and Applications
Volume:01, Issue:06, Pages: 400-404 (2010)

Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms

B.Santhosh Kumar
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: b.santhoshkumar@csice.edu.in

K.V.Rukmani
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: rukmani_arun@csice.edu.in

ABSTRACT

Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files. The comparison of memory usage and time usage is compared using Apriori algorithm and Frequent Pattern Growth algorithm.

Keywords : Apriori, Data cleaning, FP Growth, FP-tree, Web Usage Mining,

Date of Submission: 16, February 2010 Date of Acceptance: 08, A pril 2010

1. INTRODUCTION

The Web is a huge, explosive, diverse, dynamic and mostly unstructured data repository, which supplies incredible amount of information, and also raises the complexity of how to deal with the information from the different perspectives of view, users, web service providers, business analysts. The users want to have the effective search tools to find relevant information easily and precisely. The Web service providers want to find the way to predict the users' behaviors and personalize information to reduce the traffic load and design the Web site suited for the different group of users. The business analysts want to have tools to learn the user/consumers' needs. All of them are expecting tools or techniques to help them satisfy their demands and/or solve the problems encountered on the Web. Therefore, Web mining becomes a popular active area and is taken as the research topic for this investigation.

Web Usage Mining [4], [5] is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files.

1.1 APRIORI ALGORITHM

The present Research work is designed to operate on log files. The algorithm [7] attempts to find subsets which are common to at least a minimum number C (the cutoff, or confidence threshold) of the item sets. The system operates in the following three modules.

- Preprocessing module
- Apriori or FP Growth Algorithm Module
- Association Rule Generation
- Results

The preprocessing module converts the log file, which normally is in ASCII format, into a database like format, which can be processed by the Apriori algorithm.

The second module is performed in two steps.

- Frequent Item set generation
- Rules derivation

1.2 FP GROWTH ALGORITHM:

The FP Growth algorithm operates in the following four modules.

- Preprocessing module
- FP Tree an FP Growth Module
- Association Rule Generation
- Results

The preprocessing modules convert the log file, which normally is in ASCII format, into a database like format, which can be processed by the FP Growth algorithm.

The 2nd module is performed in two steps.

- FP Tree generation
- Applying FP Growth to generate association rules

FP tree is a compact data structure that stores important, crucial and quantitative information about frequent patterns.

The main components of FP tree are:

- It consists of one root labeled as "root", a set of item prefix sub-trees as the children of the root, and a frequent-item header table.
- Each node in the item prefix sub-tree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.
- Each entry in the frequent-item header table consists of two fields, (1) item-name and (2) head of node-link, which points to the first node in the FP-tree carrying the item-name.

Second, an FP-tree-based pattern-fragment growth mining method is developed, which starts from a frequent length-1 pattern (as an initial suffix pattern), examines only its conditional-pattern base (a "sub-database" which consists of the set of frequent items co-occurring with the suffix pattern), constructs its (conditional) FP-tree, and performs mining recursively with such a tree. The pattern growth is achieved via concatenation of the suffix pattern with the new ones generated from a conditional FP-tree. Since the frequent item set in any transaction is always encoded in the corresponding path of the frequent-pattern trees, pattern growth ensures the completeness of the result.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The Research work was initiated through a system study and analysis phase, where significant study was conducted to understand the existing system. Using Apriori algorithm for weblog mining is a novel technique. The explosive growth of the World Wide Web (WWW) in recent years has turned the web into the largest source of available online data.

- Situations like several unrelated topics in a single web page may lead to confusion and make harder to reach the information that the visitors are looking for.
- The design of the whole site (interface, content, structure, usability, etc.) is one of the most important aspects for any institution that wants to survive in the cyberspace.
- Understand the way user browses the site and find out which is the most frequent used link and pattern of using the features available in the site.

All these information is available online but are hidden for the users. Presently, there is no powerful that can analyze this hidden information and this Research work uses web usage mining (WUM) Apriori based approach for analyzing the visitor browsing behavior.

2.2 LIMITATIONS OF APRIORI ALGORITHM

Apriori algorithm, in spite of being simple, has some limitation. They are,

- It is costly to handle a huge number of candidate sets. For example, if there are 10^4 frequent 1-item sets, the Apriori algorithm will need to generate more than 10^7 length-2 candidates and accumulate and test their occurrence frequencies. Moreover, to discover a frequent pattern of size 100, such as $\{a_1, \ldots, a_{100}\}$, it must generate $2^{100} - 2 \sim 10^{30}$ candidates in total. This is the inherent cost of candidate generation, no matter what implementation technique is applied.
- It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns.

In order to overcome the drawback inherited in Apriori, an efficient FP-tree based mining method, FP-growth, which contains two phases, where the first phase constructs an FP tree, and the second phase recursively Researches the FP tree and outputs all frequent patterns.

2.3 PROPOSED SYSTEM

The main goal of the proposed system is to identify usage pattern from web log files of a website. Apriori [7] and FP Growth Algorithm is used for this purpose. The main goal of the proposed system is to identify usage pattern from web log files of a website. For this purpose, the usage of apriori and FP Growth algorithms are proposed. Both are influential algorithms for mining frequent item sets for boolean association rules [1], [9]. In computer science and data mining, Apriori is a classic algorithm for learning association rules [10]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or

details of a website frequentation).

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

The advantages of using apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

2.4 ADVANTAGES OF FP GROWTH ALGORITHM

The major advantages of FP-Growth algorithm is,
- Uses compact data structure
- Eliminates repeated database scan

FP-growth is an order of magnitude faster than other association mining algorithms and is also faster than tree-Researching. The algorithm reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP-tree. The FP-tree stores relevant information and allows for the efficient discovery of frequent item sets.

The algorithm consists of two steps:
- Compress a large database into a compact, *Frequent-Pattern tree* (FP-tree) structure
  - highly condensed, but complete for frequent pattern mining and avoid costly database scans
- Develop an efficient, FP-tree-based frequent pattern mining method (FP-growth)
  - A divide-and-conquer methodology: decompose mining tasks into smaller ones and avoid candidate generation: sub-database test only

2.5 ADVANTAGE OF FP-TREE STRUCTURE

The most significant advantage of the FP-tree is that the algorithm scans the tree only twice. Apart from this major advantage, the others include,

- Completeness:
  The FP-tree contains all the information related to mining frequent patterns (given the min_support threshold)

- Compactness:
  - The size of the tree is bounded by the occurrences of frequent items
  - The height of the tree is bounded by the maximum number of items in a transaction

Three major steps performed are starting the processing from the end of list L:

- Construct conditional pattern base for each item in the header table
- Construct conditional FP-tree from each conditional pattern base
- Recursively mine conditional FP-trees and grow frequent patterns obtained so far. If the conditional FP-tree contains a single path, simply enumerate all the patterns

3. SYSTEM DESIGN AND DEVELOPMENT

3.1 INPUT DESIGN

The input design is the process of converting user-oriented input to a computer-based format. The goal of the input design is to make the data entry easier, logical and error free. In the present Research work, the input is taken from the web log file. The web log file has the extension .log and contains ASCII characters. A log file is a text file in which every page request made to the web server is recorded. For each request the corresponding log file contains the following information:

IP address of the computer making the request;
- User ID, (this field is not used in most cases);
- date and time of the request;
- a status field indicating if the request was successful;
- size of the file transferred;
- Referring URL, that is, the URL of the page which contains the link that generated the request; name and version of the browser being used.

This information 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 an 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. Once the domain-dependent data transformation phase is completed, the resulting transaction data must be formatted to conform to the data model of the appropriate data mining task. For instance,

Volume:01, Issue:06, Pages: 400-404 (2010)

the format of the data for the [2],[12] discovery task may be different than the format necessary for mining sequential patterns. Finally, a query mechanism will allow the user (analyst) to provide more control over the discovery process by specifying various constraints.

2.5 OUTPUT DESIGN

The main objective of any system is the generation of reports. It has various uses. Some of them are,

- For the users, reports provide source of information required.
- They provide permanent hard copy of the results of transactions.

Careful consideration has been given in the designing of the reports as it helps in decision-making process. In the present work, the performance of the system is judged using two metrics. The first one is the amount of memory used and the second one is the time taken for the algorithm to create the association rules [3], [9]. It was found that the FP Growth algorithm is fast and uses only very small amount of memory when compared with Apriori algorithm.

4. CODING STANDARDS

4.1 APRIORI ALGORITHM

In computer science and data mining, Apriori is a classic algorithm for learning association rules [6],[11]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

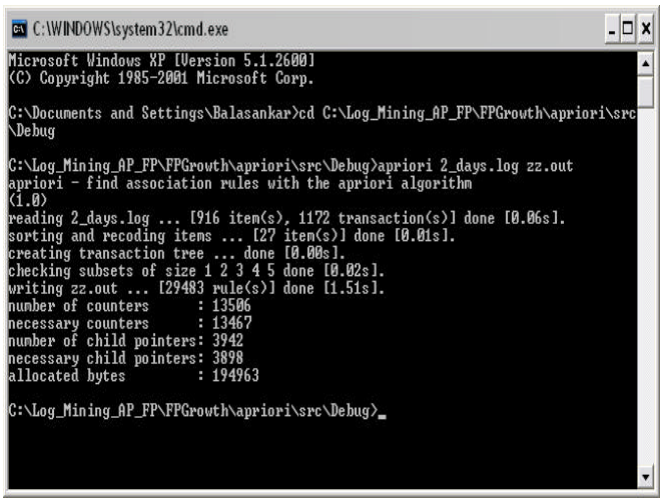
The advantages of using Apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

The Apriori algorithm is an efficient algorithm for finding all frequent item sets. It implements level-wise search using frequent item property and can be additionally optimised. The Apriori algorithm used is given below.

- L_k: Set of frequent item sets of size k (with min support)
- C_k: Set of candidate item set of size k (potentially frequent item sets)

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
return $\cup_k L_k$;



4.2 FP GROWTH ALGORITHM

Definition of FP-Tree

FP-tree is a frequent pattern tree consists of one root labeled as "null". It has a set of *item prefix sub trees* as the children of the root, and a *frequent-item header table*. Each node in the *item prefix sub trees* has three fields:

- item-name to register which item this node represents,
- count, the number of transactions represented by the portion of the path reaching this node, and
- Node-link that links to the next node in the FP-tree carrying the same item-name, or null if there is none.

Each entry in the *frequent-item header table* has two fields,

- item-name, and
- head of node-link that point to the first node in the FP-tree carrying the item-name.

5. CONCLUSION

Web usage mining is the application of data mining techniques to discover usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Web usage mining consists of three phases, namely preprocessing, pattern discovery, and pattern analysis. This Research work implements each of these phases. One of the algorithms which is very simple to use and easy to implement is the Apriori algorithm. This algorithm is used in the present Research work to generate association rules that associates the usage pattern of the clients for a particular website. The output of the system was in terms of memory usage and speed of producing association rules.

The main drawback of Apriori algorithm is that the candidate set generation is costly, especially if a large number of patterns and/or long patterns exist. The main drawback of FP-growth algorithm is the explosive quantity of lacks a good candidate generation method. Future research can combine FP-Tree with Apriori candidate generation method to solve the disadvantages of both apriori and FP-growth. In future the algorithm can be extended to web content mining, web structure mining, etc. The work can also be extended to extract information from image files.

6. ACKNOWLEDGEMENTS

"Implementation of web usage mining using Apriori and FP growth algorithms" research is supported by the management of ***C.S.I College of Engineering, Ketti***, The Nilgiris, whose support we are pleased to acknowledge. We are also grateful to our colleagues in CSICE for useful discussions, and thank to our beloved CSICE students.

REFERENCES

[1] Kotsiantis S, Kanellopoulos D., *Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering*, Vol.32 (1), 2006, pp. 71-82

[2] Agrawal R, Srikant R., "*Fast Algorithms for Mining Association Rules*", VLDB. Sep 12-15 1994, Chile, 487-99, pdf, ISBN 1-55860-153-8.

[3] Mannila H, Toivonen H, Verkamo A I., "*Efficient algorithms for discovering association rules." AAAI Workshop on Knowledge Discovery in Databases* (SIGKDD). July 1994, Seattle, 181-92.

[4] Tan, P. N., M. Steinbach, V. Kumar, "*Introduction to Data Mining"*, Addison-Wesley, 2005, 769pp.

[5] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques With Java Implementation*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2005.

[6] P. Becuzzi, M. Coppola, and M. Vanneschi, "*Mining of Association Rules in Very Large Databases: A Structured Parallel Approach,"* Proc. Europar-99, vol. 1685, pp. 1441-1450, Aug. 1999.

[7] R. Jin and G. Agrawal, *"An Efficient Implementation of Apriori Association Mining on Cluster of SMPs,"* Proc. Workshop High Performance Data Mining (IPDPS 2001), Apr. 2001.

[8] J. Han and M. Kamber, *"Data Mining: Concepts and Techniques"* .Morgan Kaufmann Publishers, 2000.

[9] E.-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* Proc. ACM SIGMOD 1997, May 1997.

[10] E-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* IEEE Trans. Data and Knowledge Eng.,vol. 12, no. 3, May/June 2000.

[11] H. Cokrowijoyo, D. Taniar, *A framework for mining association rules in Data Warehouses,* Proc. IDEAL 2004, Lecture Notes in Computer Science, vol. 3177, Springer, Berlin, 2004, pp. 159–165.

[12] L. Dehaspe, L. Raedt, *Mining association rules in multiple relations*, Proc. ILP'97, Lecture Notes in Computer Science, vol. 1297, Springer, Berlin, 1997, pp. 125–132.

Authors Biography

B. Santhosh Kumar received Bachelor degree in Computer Science & Engg from Bharathiar University and Master of Engineering in Computer Science & Engg from Anna University Trichy. He has published several papers in Data mining. Currently he is working as a lecturer- Computer Science & Engg in CSI College of Engg, Ketti.

K.V. Rukmani received Master of Computer Application from Mother Teresa Womens' University and Master of Engineering in Computer Science & Engg from Anna University Coimbatore. She has published several papers in Data mining. Currently she is working as an Asst Professor and HoD in charge- Computer Science & Engg in CSI College of Engg, Ketti.