International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-3 Issue-12 September-2013

Extract Knowledge and Association Rule from Free Log Data using an Apriori Algorithm

Hemant N. Randhir[1], Ravindra Gupta [2],G.R. Selokar[3]

Abstract

This paper aims to present technique to make private log information public and apply Apriori algorithm on collected log file to extract knowledge from public and free log files with Web Usages Mining Technique.

Keywords

Apriori Algorithm, Association rule, Free log, WUM (web usages mining)

1. Introduction

Web Usage Mining is the application of data mining technique to discover information from the web server log file data in turn to understand and serve the needs of Web based applications. Web server log file collects the characteristics of web users along with their browsing behavior at a Web site while surfing on internet [1]. In turn the problem is web service provider do not share their private log files information with other peoples. If all the people get this log information, all people get opportunities to extract knowledge from that log data and accordingly they can make changes in their web site to better serve the people. On internet some web site have heavy traffic and some similar service provider web site have scarcity in traffic, but if they get the information of another web site who have huge demand they gets opportunities to make changes accordingly to their web site. Many researcher wants a log file for their research work but they does not get these log files for their research work because log files of different web site are private they are not public. This paper aims to present technique to make private log information public and apply Apriori algorithm on collected log file to extract knowledge from public and free log files.

Hemant N. Randhir, Dept. of Information Technology, SSSIST, Sehore (MP) India.
Ravindra Gupta, Dept. of Information Technology, SSSIST, Sehore (MP) India.
G.R. Selokar, Dept. of Information Technology, SSSIST, Sehore (MP) India.

This paper aims to present technique to make private log information public and apply Apriori algorithm on collected log file to extract knowledge from public and free log files. This paper has been organized as follows: Section 2 present techniques for free public log file, Section 3 discussed Apriori Algorithm, Section 4 Introduces association rule mining, Section 4 describing methodology, Section 5 provides illustrative example.

2. Techniques For Free Log

2.1 Generating Public Log file

Open access Public log information gives opportunity to the people to develop a web site by analyzing log information of other web sites. It can be useful for researchers if they got all different web servers log file information for purpose of web personalization, web recommendation by analyzing this log files.

To generate free public log file, we have developed Google Chrome extension.

- Google Chrome Extension:

Google Chrome supports two ways of installing external extensions.

i. Using preferences .json file
ii. Using the windows registry .crx file

We have implemented a tool using .json file. Example is as shown below.

```
{
 "name": "WEB LOG GENERATOR",
 "description": "Generates User Web Activity Log.",
 "version": "1.1",
 "background_page": "background.html",
 "permissions": [
          "webRequest",
          "webRequestBlocking",
          "tabs",
          "http://*/*",
          "https://*/*"
 ],
 "browser_action": {
   "default_title": "WEB LOG GENERATOR",
   "default_icon": "off_16x16.png"
 }
}
```

Example 1 .json file

As shown in above example application logic implemented in html page "background.html" embedded with Java Script. Java Script is the technology used to process the information, manage the events, and communicate with the servers and services.

The most significant way to gather users' behavior information when the users' browsing the web pages. A tool implemented that is Google Chrome Extension. The performance parameter used to at client side that is Fetch Response and Fetch Transmission time.

- Fetch Transmission Time: The time between the requests is sent and the first byte of the response is received.
- Fetch Response Time: The time between requests is sent and last byte of the response is received.

Sometime user is resist to share his private behavior information about interface with web site, also the web service provider not violet government laws of privacy. So the user has privileges to decide way of sharing his behavior information. Our google chrome extension provides the functionality as play and pause. If web user is clicked on the pause button of the chrome extension, the chrome extension does not share users' web usage behavior information with the public.

Google Chrome extension forwards the information like URL, IP address, time, request ID and type to the local database. From local database, it can be uploaded to public server.

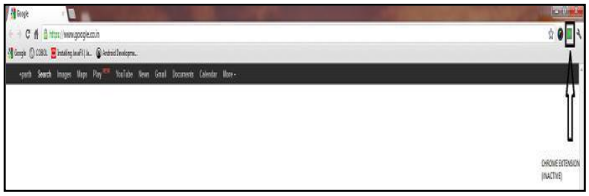

Figure 1. Google Chrome Extension when inactive

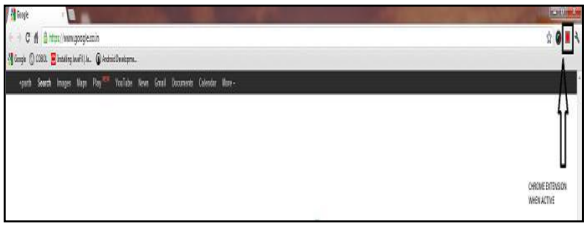

Figure 2. Google Chrome Extension when active

2.2 Downloading Log

There are two ways to stored private information to make public. The one way is to downloading log module. Downloading log module send request to web mining service which is created on the server. Web mining service is a server side program to handle client request.

As shown in figure 3 the information downloaded on Downloading log module and at the same time user could do the analysis on them. For conceptual implementation, we have designed few graphs which provide information in graph form. Hit graph shows total number of time a web page visited, Time graph shows a time required to download a webpage.

It provides lots of knowledge to the researcher about the web users navigation behavior on the Internet. This information will be helpful for customizing, modifying, enhancing their own web site or help to implement their own user interface.

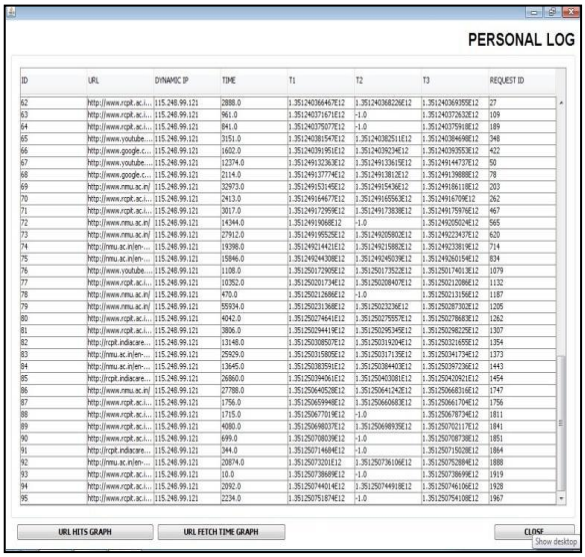

Figure 3. Download log module

3. Apriori Algorithm

In data mining, Apriori is a standard algorithm for learning association rules. Apriori is intended to operate on databases included transactions (for example, set of items bought by customers, or details of a website frequentation). Other algorithms are intended for identifying association rules in data having no transactions, or having no timestamps [2, 3].

The proposed algorithm in this paper was implemented based on apriori algorithm proposed by Agrawal and Srikant, and contains the first support, the second support and Rel-confidence and has been extensively used for the web mining recently[4].

K-candidate item sets, where k is the number of data items in its each itemset, is represented as C_k. I_k represents the candidate itemset from C_k. I_k.support represents the support for candidate itemset I_k.

The Apriori algorithm is an effective algorithm for idenfifying all frequent item sets. Using frequent item property apriori algorithm implements level wise search. The Apriori algorithm is describe given below [4,5]

Algorithm1 (function for generating candidate itemsets,
candi-gen(L_k-1))
if (k=2) then insert into C_k;
select p.iteml; q.item1 from L1
else
insert into C_k
select p.iteml; p.item2;. . . ; p.itemk-1; q.itemk-1
where p.item1=q.item1, . . . , p.itemk-2=q.itemk-2,
p.itemk-1 < q.itemk-1Algorithm2:
D: Database (a set of transactions)
$I = i_1, i_2, \ldots, i_n$: a set of data items accessed by all transactions in D
$C_1 = \{ i_1, i_2, \ldots, i_k \}$;
for (ik ∈ I)
if (ik.support=1st support) then ik∈L1
else remove ik from C1
end
for (k=2; L_k-1 != ∅; k++) do
C_k = Candi-gen(L_k-1);
T = { };
for all transactions t in D do
$C_t = \{ \}$; /* $C_t = \{I_k | I_k \in C_k$ and all data items in I_k are included by transaction t}*/
for all $I_k \in C_k$ do
if (all data items in Ik are included by transaction t)
then Ct = $C_t \cup I_k$
end
for all candidates $I_k \in C_k$ do
I_k.count++
end
T = T∪ C_t
end
for all candidates I_k in T do
if (I_k.count=1st support) then $L_k = \{ I_k | I_k \in$ T and I_k.count=1st support}
else
Rel-confidence ($i_1, i_2, \ldots, i_k$) = max(sup($i_1, i_2, \ldots i_k$)/sup(i_1), sup($i_1, i_2, \ldots, i_k$)/sup(i_2), . . . , sup($i_1, i_2, \ldots, i_k$)/sup(i_k))
if (I_k.count=2nd support and I_k. max-conf=min-conf)
then $L_k = \{ I_k | I_k \in$ T and I_k. max-conf=min-conf }
else remove I_k from C_k
end
answer =∪ $_k$ L_k
end

4. Generating Association Rules

Once the frequent item sets from transactions in a database have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence) [6, 7]. This can be done using Equation (1) for confidence, where the conditional probability is expressed in terms of item set support:

$$\text{Confidence } (A \rightarrow B) = \text{Prob } (B/A) = \frac{\text{Support(AUB)}}{\text{Support(A)}}$$

Where support (A B) is the number of transactions containing the itemsets A B, and support (A) is the number of transactions containing the itemset A. From this equation, association rules can be as follows.

- For each frequent itemset, l, generate all nonempty subsets of l.
- For every non-empty subset s, of l, output the rule

"s → (l - s)"

$$\frac{\text{Support(t)}}{\text{Support(s)}} >=$$

If minimum confidence, where minimum confidence is the minimum confidence threshold [8].

5. Experimental Analysis

The improved Apriori algorithm applied on the collected log as shown in figure 3. Here we have designed Apriori algorithm for three different relations like User-URL, User-Time and URL-Time to extract knowledge.

Here we have extract knowledge, like how much time website take to upload on a browser. From the figure Apriori Mining result we might see the number of web site takes time in-between 0 to 500 ms, similarly others 500 to 1000, 1500 to 2000. There is only one

web page "timesofindia.indiatimes.com" which takes time in-between 1000 to 1500 ms to upload on browser.

Figure 4. Apriori Mining Result

This figure calculate the record and seen the result of Apriori Algorithm where Minimum Support Count is 4 and Minimum Confident is 60 %.Below example is Finding Frequency of page with cut-off with Min Support Count.

Creating Initial Item List!
ITEM (0) : 1, %: 4 SELECTED!
ITEM (1) : 1, %: 4 SELECTED!
ITEM (2) : 2, %: 8 SELECTED!
ITEM (3) : 1, %: 4 SELECTED!
ITEM (4) : 3, %: 12 SELECTED!
ITEM (5) : 0, %: 0 DROPEED!
ITEM (6) : 0, %: 0 DROPEED!
ITEM (7) : 0, %: 0 DROPEED!
ITEM (8) : 1, %: 4 SELECTED!
ITEM (9) : 1, %: 4 SELECTED!
ITEM (10) : 2, %: 8 SELECTED!
ITEM (11) : 7, %: 29 SELECTED!
ITEM (12) : 0, %: 0 DROPEED!
ITEM (13) : 0, %: 0 DROPEED!
ITEM (14) : 1, %: 4 SELECTED!
ITEM (15) : 0, %: 0 DROPEED!
ITEM (16) : 0, %: 0 DROPEED!
ITEM (17) : 0, %: 0 DROPEED!
ITEM (18) : 0, %: 0 DROPEED!
ITEM (19) : 2, %: 8 SELECTED!
ITEM (20) : 0, %: 0 DROPEED!
ITEM (21) : 0, %: 0 DROPEED!
ITEM (22) : 1, %: 4 SELECTED!
ITEM (23) : 1, %: 4 SELECTED!
ITEM (24) : 11, %: 45 SELECTED!
ITEM (25) : 5, %: 20 SELECTED!
ITEM (26) : 3, %: 12 SELECTED!
ITEM (27) : 4, %: 16 SELECTED!
ITEM (28) : 1, %: 4 SELECTED!
AFTER INITIAL PRUNING:

Creating New C
Old set size: 1
CANDIDATE ITEMS: (0) (1) (2) (3) (4) (8) (9) (10) (11) (14) (19) (22) (23) (24) (25) (26) (27) (28)

NEW SET ENTRIES: 16

NEW CREATION:
ITEM (0) (28) : 24
ITEM (1) (24) : 24
ITEM (2) (25) : 24
ITEM (3) (24) : 24
ITEM (4) (24) : 24
ITEM (8) (27) : 24
ITEM (9) (27) : 24
ITEM (10) (27) : 24
ITEM (11) (24) : 24
ITEM (11) (25) : 24
ITEM (11) (26) : 24
ITEM (14) (24) : 24
ITEM (19) (25) : 24
ITEM (19) (26) : 24
ITEM (22) (24) : 24
ITEM (23) (24) : 24

Creating New C
Old set size: 2
CANDIDATE ITEMS: (0) (1) (2) (3) (4) (8) (9) (10) (11) (14) (19) (22) (23) (24) (25) (26) (27) (28)

NEW SET ENTRIES: 0

NEW CREATION:

NULL SET FOUND!
Final Set:
(0) (28) :1, (0) :1
(0) : (28) , CONF: 100ACCEPTED

(0) (28) :1, (28) :1
(28) : (0) , CONF: 100ACCEPTED

(1) (24) :1, (1) :1
(1) : (24) , CONF: 100ACCEPTED

(1) (24) :1, (24) :11
(24) : (1) , CONF: 9REJECTED

(2) (25) :2, (2) :2
(2) : (25) , CONF: 100ACCEPTED

(2) (25) :2, (25) :5
(25) : (2) , CONF: 40REJECTED

(3) (24) :1, (3) :1
(3) : (24) , CONF: 100ACCEPTED

(3) (24) :1, (24) :11
(24) : (3) , CONF: 9REJECTED

(4) (24) :3, (4) :3
(4) : (24) , CONF: 100ACCEPTED

(4) (24) :3, (24) :11
(24) : (4) , CONF: 27REJECTED

(8) (27) :1, (8) :1
(8) : (27) , CONF: 100ACCEPTED

(8) (27) :1, (27) :4
(27) : (8) , CONF: 25REJECTED

(9) (27) :1, (9) :1
(9) : (27) , CONF: 100ACCEPTED

(9) (27) :1, (27) :4
(27) : (9) , CONF: 25REJECTED

(10) (27) :2, (10) :2
(10) : (27) , CONF: 100ACCEPTED

(10) (27) :2, (27) :4
(27) : (10) , CONF: 50REJECTED

(11) (24) :3, (11) :7
(11) : (24) , CONF: 42REJECTED

(11) (24) :3, (24) :11
(24) : (11) , CONF: 27REJECTED

(11) (25) :2, (11) :7
(11) : (25) , CONF: 28REJECTED

(11) (25) :2, (25) :5
(25) : (11) , CONF: 40REJECTED

(11) (26) :2, (11) :7
(11) : (26) , CONF: 28REJECTED

(11) (26) :2, (26) :3
(26) : (11) , CONF: 66ACCEPTED

(14) (24) :1, (14) :1
(14) : (24) , CONF: 100ACCEPTED

(14) (24) :1, (24) :11
(24) : (14) , CONF: 9REJECTED

(19) (25) :1, (19) :2
(19) : (25) , CONF: 50REJECTED

(19) (25) :1, (25) :5
(25) : (19) , CONF: 20REJECTED

(19) (26) :1, (19) :2
(19) : (26) , CONF: 50REJECTED

(19) (26) :1, (26) :3
(26) : (19) , CONF: 33REJECTED

(22) (24) :1, (22) :1
(22) : (24) , CONF: 100ACCEPTED

(22) (24) :1, (24) :11
(24) : (22) , CONF: 9REJECTED

(23) (24) :1, (23) :1
(23) : (24) , CONF: 100ACCEPTED

(23) (24) :1, (24) :11
(24) : (23) , CONF: 9REJECTED

AR SIZE: 13
(0) >> (28)
(28) >> (0)
(1) >> (24)
(2) >> (25)
(3) >> (24)
(4) >> (24)
(8) >> (27)
(9) >> (27)
(10) >> (27)
(26) >> (11)
(14) >> (24)
(22) >> (24)
(23) >> (24)

Finally we get the 13 result the information is extracted from the database. Vectors return the result in the form of time and URL, as shown in figure 4.

6. Conclusion

In this paper we have extended previous work, we have to implement a tool which share user log file with central database. The proposed algorithm in this paper was built based on Apriori algorithm proposed by Agrawal and Srikant, and contains the first support, the second support and Rel confidence, and has been extensively used for the web usage mining recently. This algorithm is very useful to extract the users behavior information from the public log file. Similarly we can apply association rule to extract knowledge from the free central database log file. Through this research work have to show that this mining API and Association rule can be useful to business analysts, web service provider and targeted marketing.

References

[1] Rajni Pamnani, Pramila Chawan, "Web Usage Mining: A Research Area in Web Mining," in ISCET 2010, Punjab, India, 2010, pp.73-77.

[2] Jiawei, Han, and Micheline Kamber. "Data mining: concepts and techniques." San Francisco, CA, itd: Morgan Kaufmann 5 (2001).

[3] B.Santhosh Kumar, K. V. Rukmani, "Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms," Int. J. of Advanced Networking and Applications Volume:01, Issue:06, Pages: 400-404 (2010).

[4] Rakesh Agrawal, Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules," 20th International Conference on Very Large Data Bases, Chile, Sept. 1994.

[5] Liu Jian, and Wang Yan-Qing, "Web Log Data Mining Based on Association Rule," Eighth International Conference on Fuzzy Systems and Knowledge Discovery, Volume 3, Pages: 1855-1859(2011).

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

[7] S.Veeramalai, N.Jaisankar and A.Kannan," Efficient Web Log Mining Using Enhanced Apriori Algorithm with Hash Tree and Fuzzy," International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, August 2010.

[8] V.Chitraa and Dr. Antony Selvdoss Davamani," A Survey on Preprocessing Methods for Web Usage Data" (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3, 2010.