Web Usage Mining and User Behavior

Bolong Zhang
Computer Science
Southern Methodist University
United States
bolongz@smu.com

ABSTRACT

Today, the Internet has become an indispensable part of people's life, and the Internet can also be seen as a huge database in a sense, and involves various fields. So in this huge database, what is the use of data mining technology? Through a brief review of data mining on the Internet, this paper describes some trends and related technologies of data mining on the Internet, and focuses on the analysis of all kinds of the technologies related to the application direction of data mining on the Internet - Web usage log mining. Normally, web mining contains several steps, data cleaning, association rules and clustering, this article contains all cutting-edges techniques in every stages of web usage mining and aims to summarize current achievement of researchers and reveal the principle of web usage mining and users' behavior.

This paper also shows the implementation a recommender system base on the existing package of R language, this kind of recommender system is on the basis of User-Based Collaborative Filtering(UBCF) or Item-Based Collaborative Filtering(IBCF), which find the relationship between different items or different users. Combining the WUM principles, this paper use URLs and IPs to refer the websites and user, the result of this paper shows recommendations for users and recommendations for websites.

KEYWORDS

Web usage mining Recommender system Clustering Users' behavior R

Catalog & Content

1 Introduce

Nowadays, more and more people are using the World Wide Web. With the development of the World Wide Web, many new applications and technologies have emerged. Most of the Web structures are very complex and the amount of data is huge, so users are trying to browse. Sometimes, they are often missing, and are lost in a large amount of data, resulting in wrong query objectives, and even difficult to get clear and complete results. Based on this contradiction, **personalization** can effectively recommend the online content of the user to the user, thereby improving the efficiency of the website and expanding the business income. The personalization of the network refers to the information or service provided to meet the needs of a specific user or a group of users, the use of navigation recommendation and the personal interests of users to acquire knowledge, combined with the content and structure of the website, personalize the website as behavior. The goal of a network personalization system is to provide users with the information they want or need without expecting the user to specify their requirements. In the personalization system, the content of the website and even the structure are modified dynamically.

Network personalization mainly includes the following contents: 1. Preprocess network data and classify them. 2. Use some methods to identify the correlation between the data. 3. Based on the results of the first two steps, do relevant recommendations for user with interested content.

Therefore, network data plays a crucial role in network personalization. Generally speaking, network data contains the following main types: **Content data** is presented to users in need in a structured way. This type of data is generally the data that the user usually touches, such as text data, image data, or most of the data obtained from the search engine. This is the user. The most intuitively felt data. **Structural data** is generally expressed in the form of organization of content. Usually the user can't directly see it in the web page, and need to view the web page source code to get it. The **Usage data** indicates the usage of the website. For example, a visitor visits a specific URL at a certain time. It usually contains the IP address of the visitor, the access time and date, and the status code of the access, and some will contain the agent information, and the access is complete. The path (file or directory), the size of the data, and other properties that can be included in the web access log.

According to the data obtained above, we can mine the website data in order to get personalized recommendation content. In this project, this paper uses the collaborative filtering method to analyze the webpage usage data and the user's IP address, and get a rule. This rule can be used to recommend a website address to users with similar access records, or to make recommendations for potential users for certain websites.

2 Motivation & Background

The motivation of this project is based on the principle of web usage mining(WEBMINER[1] System) and collaborative filtering(Tapestry[2] System), this work want to combine web usage mining and collaborate filtering together to get the relation between the user and the website.

Normally, web usage mining can be divided into three main part[3], as shown in Fig. 1. The first part of WUB is preprocessing, the preprocessing part has five parts, data cleaning, user identification, session identification, path complete and formatting. Data cleansing is the task of deleting log entries that are not needed in the mining process. It can remove all data that is not related to the core content, and at the same time remove the dirty data with errors, and use some regex to remove data which does not satisfied the requirement, after data cleaning, the rationality of the data is enhanced. User identification is the process ofassociating page references, even those with the same IP address, with different users[3]. Session identification takes all of the page references for a given user in a log and breaks them up into user sessions[3]. The second part of WUB is pattern discovery[4], this step is to use the data produced by the server to identify frequent patterns, and the user can get the access to

many resources by clicking on the hyperlink. By identifying sequences of those clickstreams, patterns regarding user interests can be understood. If these patterns are within a certain time threshold, the requested session can be identified. By mining these sessions, user behavior and interests can be identified. These tasks exist in the pattern discovery process, which uses techniques in many fields, including statistics, data mining, machine learning, and pattern recognition. In this step, the main techniques involved are frequent item set mining(association rules), clustering, statistical analysis, classification and sequential analysis[4]. The third part of WUB is pattern analysis, which aim to filter and delete irrelevant rules or patterns[4].

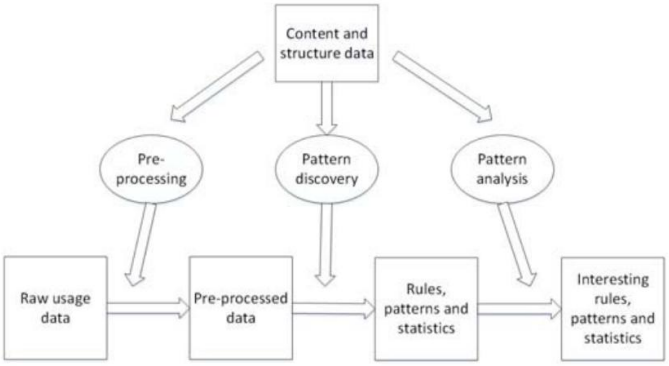


Figure 1: Phases of WUB

The first collaborative filtering system named Tapestry was designed by Xerox Company, in 1992[2]. It is mainly used for solving the problem of information overloading of Xerox's research center in Palo Alto. The research center's employees receive a lot of emails every day but there is no way to sort them, so the research center develops this experimental mail system to help employees solve this problem. Its operating mechanism is roughly as follows: (a) The user himself can determine the type of mail he is interested in, and classify the mail based on the user's personal interests; (b) The user randomly requests the system to request information related to itself, and waits for the system to respond. The system can respond to a considerable number of files as a response to the user request. (c) From the large amount of file data, at least three data that the system considers to be the most relevant to the user are selected. This data can be considered as the user most wants to see, and is most useful for the system. (d) Based on the data records obtained in these steps, the system can obtain a filter on the user information. This filter can filter the user's personal mail system. After that, every email received by the user will be filtered, and the filtered most relevant ones will be filtered. The file will be delivered to the mailbox first.

Therefore, for users, collaborative filtering has several advantages: (a) Can filter information that is difficult to analyze in the system. (b) Based on the experience sharing of many users, it can effectively avoid data incompleteness and inaccuracy. In multi-person systems, data filtering processed with some complex and abstract data. (c) Based on the historical record to recommend new information, the system can find data that is not similar in content. The user may not predict the recommended content in advance. The system can recommend relevant data for a specific user according to the history records of multiple users. The system finds interests and preferences that users potentially have but not aware of. (d) Personalized and automated, it can effectively use the history and feedback of other users with similar preferences.

3 Theoretical Principles & Algorithm

This part will introduce several theoretical principle & algorithm which will be used in the web usage mining

and recommender system, I utilize some of these principles and algorithm in my implementation. All these theoretical principles and algorithm can be divided into two distinct part, one is about the web usage mining, the other is about the recommender system.

3.1 Data Processing

In the field of web usage mining, the source data are log files. A log file is a set of data that is recorded in a back-end database and is used to record files when an operating system or program application is running, or it can be a logs of communication between different users. Typically, web log files are automatically created and maintained by the web server. The log file records the clickstream of each visiting user, including access to HTML, images, or other embedded objects on a website. The original web log file format is often a line of text, including the IP address, access time, access type, and agent information of the person accessing the site. Most servers have the log files format as

"**<ipaddress> <username> <password> <date/timestamp> <url> <version> <status-code> <bytes-sent> <request method> <referer>**"

In my implementation, because of only valid data can get valid result, thus filtering the data is crucial. Actually, just take the data format above as an example, in this line of data, we did not care the "password"(because it can be seen as random string for others except of the holder), "data/timestamp"(also random data for server end, you don't know when a visitor will click the specific URL), "version"(irrelevant data), "byte-sent"(irrelevant data). Base on this principle, we can utilize "ipaddress" and "url" to refer the "user" and "website"("referer" can also represent as the web), and "status-code" can reflect the status of the browsing history. Normally, in status-code field, as shown in Fig. 2, "1XX" represent Informational Responses, "2XX" is the status of Success, "3XX" represent Redirection, "4XX" means Client Error, "5XX" is Server Error. Thus, some data with invalid status code should be removed.

Code	Description	Code	Description
200	OK	400	Bad Request
201	Created	401	Unauthorized
202	Accepted	403	Forbidden
301	Moved Permanently	404	Not Found
303	See Other	410	Gone
304	Not Modified	500	Internal Server Error
307	Temporary Redirect	503	Service Unavailable

Figure 2: Common Status Code

Meanwhile, if a user clicked a hyperlink in the web site, which is an embedded object in this page, this part should also be removed because it can be a image(with the suffix .jpg or .jpeg, etc), a video(with the suffix .mp4 or .av) or any other objects, these field can not represent the website itself, mining these data is not web usage mining but web content mining, we don't care what user will like in this page but pay attention to what page the user will like. This data processing algorithm can be depicted as Fig. 3[5].

```
Input: log file
Output: cleaned log file
Step 1: Begin
      Read records in log file
Step 2:  For each record
      Read (status code, method)
Step 3:  If (status code= '40*' and method= '**')
                Then, remove that status field
                Get IP address and URL link
Step 4: If (suffix_URL_link= {.gif, .jpg, .css, .av}
                        && request= "implicit")
                Then, remove that URL_link
           Else
                Save IP address and URL_link
         End if
       End if
Step 5: If (status code! = '40*' and method= '**')
                Then, Get IP address and URL link
          If (suffix_URL_link= {.gif, .jpg, .css, .av}
                        && request= "implicit")
                Then, remove that URL_link
           Else
                Save IP address and URL_link
         End if
       End if
       Next record
    End for
End
```

Figure 3: Log Data Processing

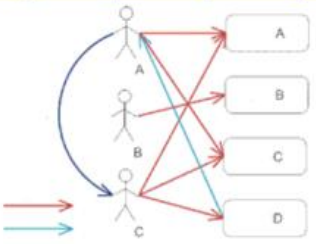
In my data, "302" "200" "304" are the most common data, which means these data can be use for mining. After this step, we can get a file only contains URL and IP information.

3.2 Collaborative Filtering

Collaborative filtering is simply to use a group with similar interest preferences and common or similar historical records to recommend information of interest to the user. Each individual gives a certain degree of response (such as rating or number of visits) to the information through some operation. Recorded by the system for filtering purposes, further helping others to filter information. There are several kinds of Collaborative Filtering methods, the typical two are User-Based Collaborative Filtering(UB-CF) and Item-Based Collaborative Filtering(IB-CF).

The basic principle of user-based CF is very simple. The neighbor users (users with similar preference) with the high relevance are found according to the user's preference for the item, and then the preferences of the neighbor users are recommended to the current user because the difference between them is low. Identifying the user's preferences for all items as a vector can be used to calculate similarities between users by calculating the degree of difference between vectors. After finding a specified number of neighbors, based on the similarity weights of the neighbors and the degree of preference for an item, it is possible to predict irrelevant items that the current user has no preference, and calculate a ranked list of items for recommendation. Fig. 4 left half part shows an example. For user A, according to the user's historical preference, only one neighbor user C is calculated here, because they all like item A and C, and then item D is recommended to user A because user C like it.

user/product	A	B	C	D
A	√		√	recommend
B		√		
C	√		√	√



User/product	A	B	C
A	√		√
B	√	√	√
C	√		recommend

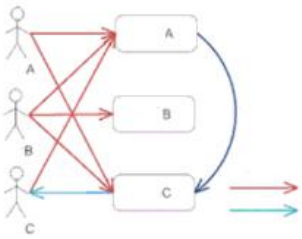


Figure 4: UB-CF & IB-CF

The similarity of UB-CF can use cosine similarity, as the formula (1)

$$similarity = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

In this formula, select two user A and B from the matrix if they are both have a value on an item, this item will be counted in, use the value of A multiple the value of B, then divided the product of root of the sum of all the square of values of A and root of the sum of all the square of values of B. For example, if every data in matrix of A equals every data in matrix of B, then this similarity is equals to 1, which means A has the very high similarity with B, when A prefer some thing, B also prefer that one with very high probability.

The principle of item-based CF is similar to user-based CF. It only considers the item itself when calculating neighbors, instead of using the user's choice as a reference, which means that all users' preferences are used to calculate similar items, and recommend this item to users with similar users. In the calculation part, all user preferences can be used as a vector to calculate similarities between items, resulting in similar items. The latest user's preferences are predicted based on their historical preferences, and an ordered list of recommendations is generated. The main difference between IB-CF and UB-CF is, IB-CF finds the similar items but UB-CF find the users with similar preference. Fig. 4 right part shows an example: for item A, users who like item A all like item C according to historical preferences of all users, conclude this phenomenon item A is similar to item C, and user C likes item A, so it can be inferred that user C may also like item C.

Just like the collaborative filtering principle, I use a IP-URL matrix to utilize the algorithm in it. Each IP can represent a single user or single organization, actually, IP address can be shared with multiple user(WIFI or NAT, here I only consider IPV4 but not IPV6), but if consider this situation, IP address in data can not represent a single user but a group of user, the preference get from the IP reflect a group preference, which is also valid. Suppose there are n IP address $\{IP_1, IP_2, IP_3, ..., IP_n\}$ and m URL $\{URL_1, URL_2, URL_3, ..., URL_m\}$, the value of this matrix can be record as $V_{ij}, i \le n, j \le m$, initially, every value in the matrix is 0, each time a user i brows a URL j, the value of V_{ij} will plus 1, it can depict as the frequency of browsing history, for example, if a specific user brows a given website frequently, it can illustrate that he has a strong connection with this website, a student at SMU will have a high-frequency browsing history on CANVAS or SMU official website, thus something relate to the SMU may get the concentration of this student. This value is just like the “rating” in recommendation system, the difference is, the “rating” is more concentrated, if someone like an item, he rates with 5 stars, if he strongly dislikes that one, rates 1 stars, but for website, if someone likes to use “Google” search engine, he may use it 20 times per day on average, but seldom to use “Bing”, maybe several times in a year, or he just click this URL by mistake but have no interest in this web. So maybe setting a threshold to this value is a good solution to the problem, if the value below to a threshold, reset the value as 0.

4 Reduction To Practice

In this section, the process of how to mining the log files and find the relevance between IPs and URLs will be presented, this part will output the IPs(users) for specific URLs(webs) and output the URLs(webs) for IPs(users) to illustrate that kind of IP/URL has the higher preference for this specific URL/IP on the basis of the browsing history of users’ behavior and the web records.

So the first step is to collect the data, there are many ways to get log files from your own PC(Appendix 1), here I got a public web log files from kaggle, which has 10.8K*4columns, as Fig. 5(right side). I downloaded

about 10 log data from different organization and compared the log files, some log files have 8 or more columns, the main difference between them and the data set I selected is they splits some fields into several fields, for example, a URL element “GET www.google.com HTTP/1.1” can be split to “GET”, “www.google.com” and “HTTP/1.1”, I think this is identical to my data, and some field “Hostname” and “Sent Bytes” are meaningless, because it’s easy to delete the invalid lines and these fields will not effects to the final result(If sent bytes = 0, then delete this row), finally, many columns were deleted besides of several columns. After get the data without irrelevant field, use R language to store the data in a variable named log_data, like log_data<-read.csv('webLog.csv',sep=',',header=TRUE), it still has some dirty data, base on the Status code shown in Fig. 2, the data with status code above 400 is useless, thus use the statement log_data[which(log_data$Staus == 200 | log_data$Staus == 302 | log_data$Staus == 304),] to remove that. The original data has 10788 rows, after removing, it has 10527 rows.

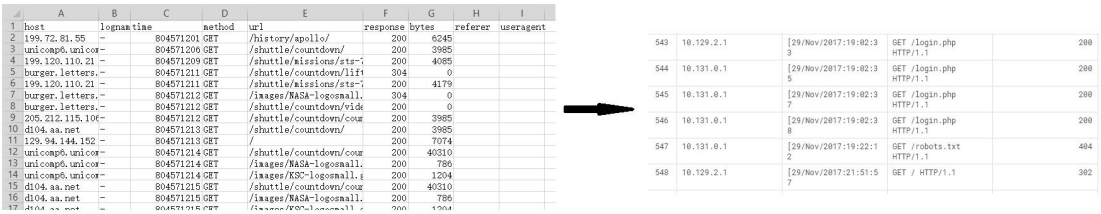


Figure 5: Web Log Files

The next step is to fix which URLs are similar, then the should be clustered together. I read some paper[6][7] about how to use clustering algorithm to cluster the IP address, and the paper said using the **longest IP prefix matching**, because IP address can be replaced as a 32-bit-long string, however, I found this method has a latent drawback, namely the IP address has no **practical meaning**, “10.1.10.5” and “10.1.11.5” have a common long prefix string, however, they may very far away from each other. However, if this method utilized in URL, it can make the practical meaning, because similar websites have the similar URL. For a instance, if a URL has the text like: 〝/google.com〞, the other URL has the text like: 〝/google.co.uk〞, then they should be consider as the 〝similar URL〞, or when you search something in search engine, the URL will return “/google.com/search?”, no matter what the suffix is, it means that those URLs with common prefix belongs to one search engine “google”, almost every search engine applied in this method. Base on this mechanism, I used lexicographical order to order the URL address with R statement “log_data[order(log_data$URL),]”, thus, all the URL with the similar prefixes are stored adjacently although some part of URLs has completely different adjacent neighbors. At this time, I got the data with 5 IP address and 260 URLs in ordered, then it can be used clustering algorithm to fix the specific range of the data. Here the clustering algorithm used are distance-based K-means and density-based DBSCAN, to compare different clustering algorithm effects on clustering the URL. Before clustering, use the R statement “data.frame(IP,URL)” to just store IP and URL columns. The clustering results can be shown as Fig. 6. All the IP

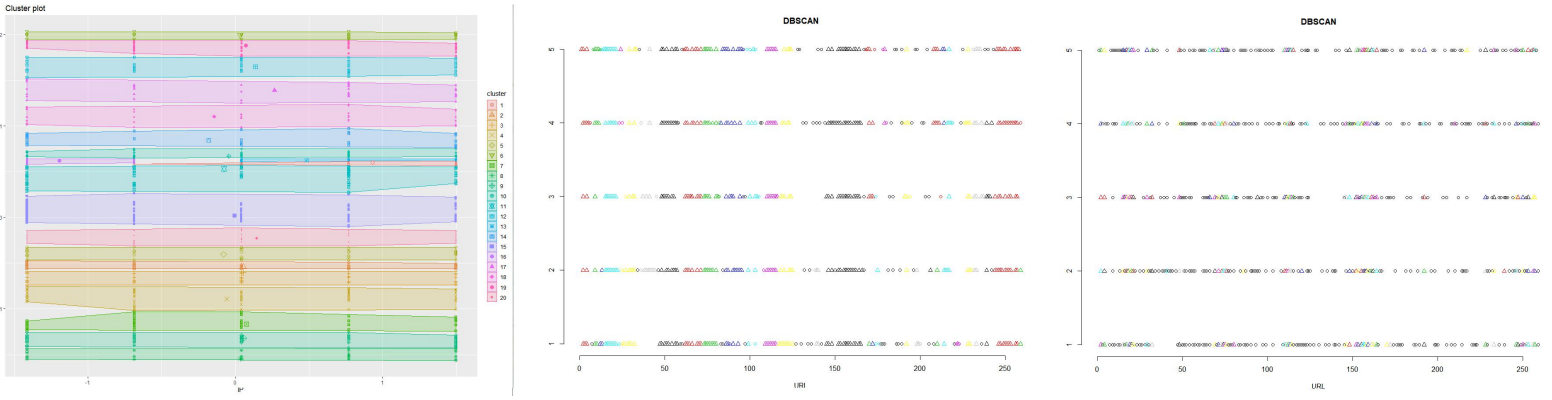


Figure 6: Cluster Plot Comparison

address and URLs are represented as natural number in the figures above because the clustering algorithm only accept numeric types in R. The figure on the left hand clustered with K-means and set k as 20, the middle figure and figure on the right clustered with DBSCAN, the only discrepancy is the middle figure has a eps > 1 (34 clusters), the left-hand figure has a eps < 1 (230 clusters), because the IP stored as sequential integers, their distance is 1, thus if you want to see the relation between specific URLs with a group of IPs, the radius must be bigger than 1. If a cluster has just one IP address, this cluster can be regarded as useless cluster, it can illustrate nothing, just like the right-hand figure shows, it is irregular.

After get the similar URLs basically, the next step is to create the IP-URL Matrix as the section 3.2 described, the R statement "as.data.frame(table(data.frame(IP,URL)))" can implement this function (Appendix [5]), each value in matrix means the frequency of the data, (ip-1 url-2 172) means user-1 has 172 browsing history about this url with number 2. The plot of URL and frequency shown as Fig. 7. The "bars" in the figure represent the range of the frequency, for example, some IPs brows a URL frequently, some does not, so the length of the bar is the largest frequency among all the IPs minus the smallest frequency among all the IPs, so it is the range. By calculating the frequency of the URLs, I see that there are too many URLs with just a few frequency, as in Fig. 7, almost about 200+ URLs have a frequency near 0, these data will effect the data, and should not be used, for example, when you click a web by mistake and close it immediately, you have a browsing history in this web, but maybe someone else also click this web by mistake, he will have the same frequency about this web with you, but this can not prove that you and he have the similar preference because the frequency is too low, so a threshold about frequency is necessary, the minimum threshold about this frequency is 2.

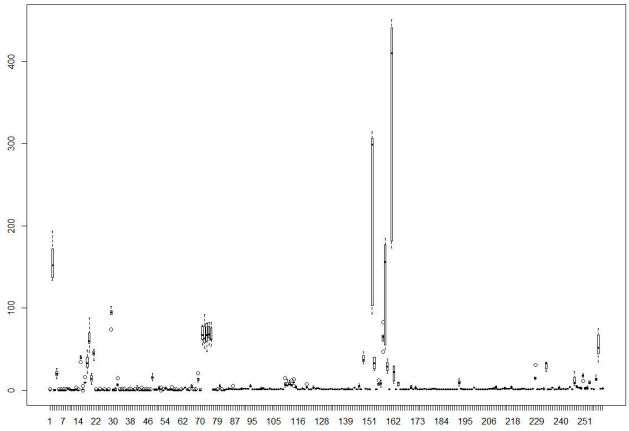


Figure 7: URL Frequency

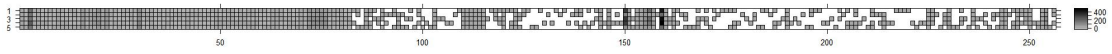


Figure 8: URL + IP Frequency

Fig. 8 shows the plot about IP and URL and Frequency. Vertical axis is IPs, horizontal axis is URLs, the frequency can be depicted with the darkness of the square, the darker square means higher frequency.

Next, the data is processed using the "reshape" package to generate a matrix of IP*URL, Freq as the values. Using R statement "data2 <- dcast(data,data$IP~data$URL,value="data$Freq")", At this time, data2 has two attribute values cast_df and data.frame. We need to change the data2 property to the data frame, where cast_df cannot be directly converted to matrix, so we need to remove this class property, leaving only data.frame. Using R "class(data2)<-"data.frame"", then we still have to process the data and convert it into the realRatingMatrix property that the recommenderlab package can handle. In the following, we first convert data2 into a matrix, and then use the as() function to perform a cast, which is the result we want, like: "data2 <-as.matrix(data2) data2<-as(data2,"realRatingMatrix")".

In the recommenderlab package, a total of six models are provided for the realRatingMatrix data type, I used Item-Based Collaborative Filtering (IBCF), random recommendation (RANDOM), User-Based Collaborative Filtering (UBCF). Collaborative filtering has two main steps: 1 Find a user group similar to the target user's browsing style based on the known browsing history of the target user. 2 Calculate the latent frequency of the user group for other URLs and use it as the predicted score of the target user. Using the R statement “Recommender(data4, method = "UBCF|IBCF|RANDOM")” and “predict(data5, data4, n = top_N value)” The result of the recommendation shown as Fig. 9, for each IP, it recommends the URL number with the order of descending probability. From left to right are UBCF, IBCF and Random. Just simply exchange the x-axis and y-axis of matrix, we can get the recommendation result for URL.

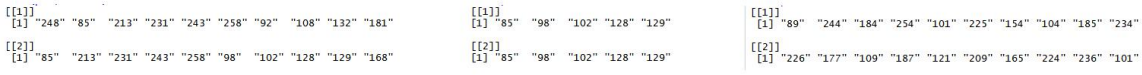


Figure 9: Result of Recommendation For IP

The values in the output represent the URL that this IP will interested, in Fig. 9. IP1 gets a result (248,85,213,243,258.....), that means 248 is the URL with the largest predicted value, 85 is the second largest, totally quantity of outputs will base on the top_N parameter. After get the result, use the RMSE/MSE to evaluate the result, as Fig. 10, RMSE is used to measure the deviation between the observed value and the true value. As shown in Fig. 10, the RMSE of UB-CF and IB-CF is about 15, it seem very large, but consider the range of the frequency is (0,451), this RMSE is acceptable.

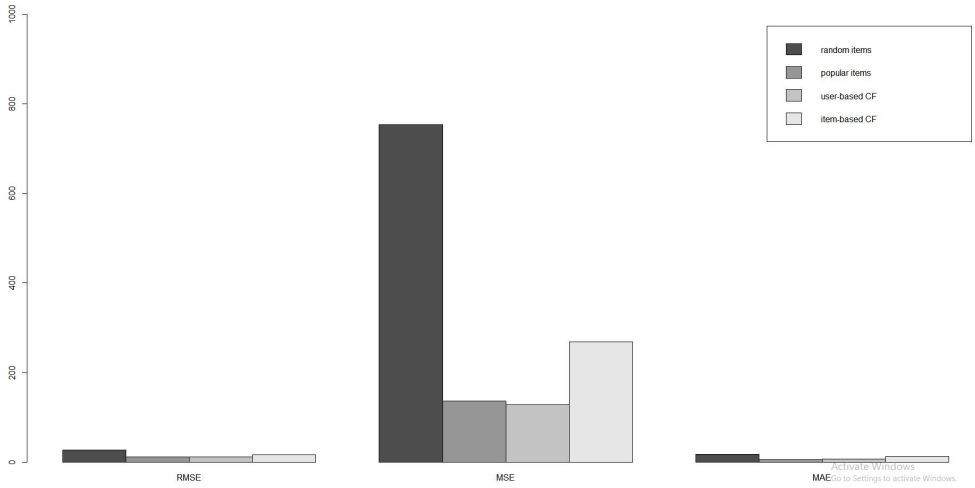


Figure 10: Evaluation

5 Conclusion & Future Work

This report proposes a method for mining web logs and user preferences. The inspiration comes from the recommender system based on collaborative filtering. It shows the principle of the method, data processing, clustering method, URL categorization, output and result evaluation. However, the amount of web log data is relatively small, resulting in insufficient training sets and verification sets, and the results obtained cannot have obvious features.

Analysis of the entire process of web usage mining, I found that there are some areas that can be improved,

there are several directions for research and application:

1. Use the tree structure to store URL information. Each domain name represented as a node of the tree, and mines the child nodes of the parent node, narrows the scope of the mining, and mines user preferences under given URLs.

2. Referer mining, referer refers to which website the user accesses this URL, or from which website the user click the URL, by mining the referer information, we are able to get the user's preference for the search engine more directly.

3. Taking time as a factor into account, for example, Users often visit the login page, but each visit time is very short, so the high frequency does not fully reflect user preferences, it must considered with long browsing time to show the preference of user, thus set the time threshold and filter the data whose browsing time much less than the average browsing time, the results could be more accurate.

4. Consider other data in the web log files, such as considering the relationship between the amount of bytes sent/received and the user's preference for the URL, in other words, whether the amount of information exchange is positively related to the frequency of access.

6 Reference

[1] R. Cooley,B. Mobasher,and J. Srivastava. "Web mining: Information and pattern discovery on the World Wide Web." *In International Conference on Tools with Artificial Intelligence*, pages 558–567,Newport Beach,CA,1997.

[2] David Goldberg, David Nichols, Brian M. Oki, Douglas Terry. "Using collaborative filtering to weave an information tapestry." *Communications of the ACM - Special issue on information filtering*: Volume 35 Issue 12, Dec. 1992. LINK

[3] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns." *Knowledge and Information Systems* Volume 1 Issue 1, Pages 5-32, February 1999

[4] Parth Suthar, Prof. Bhavesh Oza, "A Survey of Web Usage Mining Techniques." Parth Suthar et al, / *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 6 (6) , 2015, 5073-5076

[5] Priyanga P, Dr. Naveen N C. "User Identification, Classification and Recommendation in Web Usage Mining – An Approach for Personalized Web Mining." *IJISET - International Journal of Innovative Science, Engineering & Technology*, Vol. 2 Issue 4, April 2015. ISSN 2348 – 7968.

[6] Asim Karim, Syed Imran Jami, Irfan Ahmad, Mansoor Sarwar, and Zartash Uzmi, "Clustering IP Addresses Using Longest Prefix Matching and Nearest Neighbor Algorithms."

[7] Songjie Wei Jelena Mirkovic Ezra Kissel, "Profiling and Clustering Internet Hosts" *Proceedings of the 2006 International Conference on Data Mining, DMIN 2006*, Las Vegas, Nevada, USA, June 26-29, 2006. LINK

7 Appendix

[1] https://www.surveilstar.com/web-logs.html

[2] Https://www.r-statistics.com/tag/cast_df/

[3]https://medium.com/@wwwbbb8510/comparison-of-user-based-and-item-based-collaborative-filtering-f58a1c8a3f1d

[4] https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1

[5]

```
> as.data.frame(table(data.frame(IP,URL)))
    IP URL Freq
1    1   1    0
2    2   1    0
3    3   1    0
4    4   1    0
5    5   1    1
6    1   2  172
7    2   2  134
8    3   2  152
9    4   2  194
10   5   2  137
11   1   3    1
12   2   3    0
13   3   3    0
14   4   3    1
15   5   3    0
16   1   4   23
17   2   4   18
18   3   4   14
19   4   4   27
20   5   4   20
21   1   5    0
22   2   5    0
23   3   5    0
24   4   5    1
25   5   5    0
26   1   6    0
27   2   6    0
28   3   6    0
29   4   6    1
30   5   6    0
31   1   7    1
32   2   7    0
33   3   7    0
34   4   7    0
35   5   7    0
36   1   8    0
37   2   8    0
38   3   8    0
39   4   8    0
40   5   8    1
41   1   9    3
42   2   9    1
43   3   9    3
44   4   9    2
45   5   9    1
46   1  10    2
47   2  10    0
48   3  10    0
49   4  10    3
```