IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

A Recommender System for Online Advertising

M. Sree Vani

Department of CSE, MGIT, Hyderabad

Abstract: Recommender systems are now evolving as essential marketing and decision supporting tools in online advertising environment, because many customers who deal with e-commerce and related domains are often overloaded with complex and raw data driven by dynamic workflow operations, processes and business guidelines. Recommender systems will analyze and filter complex data methodically to provide essential and useful information to customers (in e-commerce). In this paper, we have presented an efficient recommender algorithm based on user browsing behavior. This algorithm will study user's behavior, interests and other parameters to present them relevant ad(s) when they navigate through web pages. The Ad recommendation system works dynamically using input details about an advertisement and user past website visitation behavior. These inputs determine advertisement placement and presentation. The algorithm solves the problem of "recommending more related ad(s)" providing opportunity to generate higher user response, user satisfaction deriving more value for time and money. Our experiments and results prove that the dynamically recommended Ads by recommender system have got the relevance up to 92.47% out of 1900 instances.

Keywords: Recommender system, user behavior, online advertising.

1. INTRODUCTION

As the use of e-commerce span business across mobile and other ubiquitous environments, the importance of the recommender systems become critical to strategically increase advertising value. The recommendation system with the content-based (CB) approach analyzes statically ad(s) banner, content, placement to create a profile representing a user's interest in terms of ad items. A review will be done on that content of items by comparing them against created user's profile. It finally recommends results with the new items that are likely to satisfy the user's preference [2].

Recommender systems enhance E-commerce sales in three ways:

Converting Browsers into Buyers: Visitors to a Web site often look over the site without purchasing anything. Recommender systems can help consumers find products they wish to purchase.

Increasing Cross-sell: Recommender systems improve cross-sell by suggesting additional products for the customer to purchase. If the recommendations are good, then average order size should obviously increase. For instance, a site might recommend additional products in the checkout process, based on those products already in the shopping cart.

Building Loyalty: In a world where a site's competitors are only a click or two away, gaining consumer loyalties is an essential business strategy (Reichheld and Sesser, 1990; Reichheld, 1993). Recommender systems improve loyalty by creating a value-added relationship between the site and the customer. Sites invest in learning about their customers, use recommender systems to operationalize that learning to present custom interfaces that match consumer needs. Consumers repay these sites by returning to the ones that best match their needs. The more a customer uses the recommendation system – teaching it what he wants – the more loyal he is to the site. "Even if a competitor were to build the exact same capabilities, a customers would have to spend an inordinate amount of time and energy teaching the competitor what the company already knows" (Pine, et al., 1995). Creating relationships between consumers can also increase loyalty, for consumers will return to the site that recommends people with whom they will like to interact. Recommendation approaches typically rely upon implicitly or explicitly acquired behavioral data denoting users' interest. We developed a repeatable process for turning user subgroup response data into "business intelligence." The predictive models developed previously are the foundation for an optimization algorithm. The Ad recommendation system works dynamically using input details about an advertisement and user past website visitation behavior. These inputs determine advertisement placement and presentation. When clustering, we considered how to account for changes in user behavior and how to encompass this within cluster assignment.

2. METHODOLOGY

The response of the users to the published ad(s) depends on how best suitable and purposeful they are when user query or visit a page having a specific context and particular need in mind. In this paper, we used a very common content match scenario, where 1.

Web site owners (called publishers) provide the "ad-banner space" (i.e., a reserved portion of their page for placing ads, and 2. Ad server/network, an entirely different commercial entity, returns the ads that are most suitable for the page content.

2.1 The Proposed Recommender Algorithm

We present an efficient recommender algorithm based on user browsing behavior. A user can be presented with

DOI 10.17148/IJARCCE.2016.52137

particular ad(s) based on user's interests generating a more user response. The algorithm solves the problem of recommending more related Ads and provides an opportunity to generate a higher user response and satisfaction increasing business value. The steps in the algorithms are as follows:

Step 1: Data Preprocessing.
Step 2: Pattern Extraction.
Step 3: Ad(s) Recommendation (for the most related Ads based on users Behavior)

3. DATA PREPROCESSING

An important task of recommendations generation process is the creation of a preprocessed data set (obtained from available data sources). Since most of the data mining algorithms that can be applied in the next step of the recommendation process, namely the pattern discovery and analysis step, work on structured data, the aim of the preprocessing step is to form an appropriate, reliable and integrated data set that can be effectively used further used in the pattern extraction (discovery and analysis) step.
Usually, preprocessing is data-specific and several preprocessing methodology should be performed based on the type of data involved. The preprocessing approaches that can be applied on the content data differ from the ones that can be applied on the usage data. The preprocessing tasks applied in this step are grouped logically according to the type of data to be used in the recommendation model.

3.1 Web Usage Data Preprocessing

Web server logs are the primary sources for usage data in which the activities of web users are registered. These log files can be stored in various formats such as Common or Extended log formats. Basically, an entry in Common log format consists of the following fields as shown in the following Figure 1.

IP address: indicating the IP address from which the user is accessing Web server, in format of four three-digit number eliminated by dot.

Identity: indicating the identity defined by RFC 1413. The "hyphen" in the output indicates that the requested piece of information is not available;

User ID: indicating the User id of the person requesting the document as determined by HTTP authentication.

Time Stamp: indicating the time of request, in format of [dd/mm/yyyy:hh:mm:ss zone].

Request parameters: containing request method, session id, browser type, resource and protocol.

Access status: indicating request resulted in a successful response, a redirection, an error caused by the client, or an error in the server.

File size: indicating the size of file interacted from the request. This original format of log file needs be modified according to the following reasons as shown in Table 1.

```
141.243.1.172[29:23:53:25]"GET/Software.html HTTP/1.0" 200 1497
Quer2.lycos.cs.cmu.edu[29:23:53:36]"GET/Consumer.html HTTP/1.0" 200 1325
Tanuki.twics.com[29:23:53:53]"GET/News.html HTTP/1.0"200 1014
Wpbfl2-45.gate.net[29:23:54:15]"GET/HTTP/1.0"200 4889
Wpbfl2-45.gate.net[29:23:54:16]"GET/icons/circle_logo_small.gifHTTP/1.0"200 2624
Wpbfl2-45.gate.net[29:23:54:18]"GET/logos/small_gopher.gifHTTp/1.0"200 935
```

Figure 1: Web log file - sample template

Generally, there will be too many distinct values for some fields like IP address, day and time. For IP address, some values starting with same prefix can be interpreted requests from the same user or group of users. For the date part, we can certainly keep them as individual values of this field, as the whole data set is based on transactions in a one-month period.

In order to understand the user behavior the following information could be extracted from server logs.
Who is visiting the website: One of the major steps in Web usage mining is to identify unique users in order to obtain the path that each follows.
The path users take through the web pages: With the knowledge of each page that a user viewed and the order, one can identify how users navigate through the Web pages.
How much time users spend on each page: A pattern of lengthy viewing time on a page might lead one to deduce that the page is interesting?
Where visitors are leaving the web site: The last page a user viewed before leaving the website might be a logical place to end a server session.

In order to extract a particular user behavior from remaining logged users, each record in the log file should be written in such a way to uniquely identify users who performed it to study browsing behavior. A user here typically represents a person, computer, domain or company. It is an easy task if the log file records a person ID such as login user or computer name. However, it is a non trivial task in case of multiple users logging from a single computer especially when web sites do not require users to log in with a user name.

DOI 10.17148/IJARCCE.2016.52137

Table 1 Web log file Re-formatting

Field	Original format	Re-formatted
IP address	129.173.66.192	129.73
Date / Day	18/Oct/2009	{Mon, Tue, Wed, THU, Fri, Sat, Sun}
Time	13:04:56	{morning, afternoon, evening, night}
URL	/~user/index.html	/~user/index.html

Most web servers do not assist providing consistent user login identity to take help out. Thus, the information available according to the HTTP standard is not adequate to distinguish a user among all other users when browsing on same host and proxy. The most widespread remedy for this problem is the use of cookies and session variables. Another way to identify unique users is using a heuristic method in which unique IP address as a user will be identified with IP address when IP addresses resolve into domain names registered to a person, domain or company as it is possible to gather more specific information from Domain name servers. Once the users are identified, server log data passes through a session reconstruction step in which we process reconstructing the user's original sessions by using server log data. Reconstructing user sessions from server logs is a challenging task since the access logs protocol is stateless and connectionless. If neither the cookies nor the user-login information are available, the reconstruction of original session is based on two basic heuristics: 1. Time oriented (h1,h2) and 2. Navigation oriented (h3). Time oriented heuristic considers the browsing time patterns and past browsing analysis. Navigation oriented heuristic considers the site topology and trace route information. Accesses to cached pages are not recorded in the web log due to the browser or proxy cache does not send request to web servers. Therefore, references to cached pages will not be logged. However, the missing references in the log file can be found using a set of assumptions. The referrer field of the web log or the web site structure can be used to infer cached pages when requests are analyzed with other cookie and session information. If a requested web page P_i is not reachable from previously visited pages in a session, then a new session is constructed starting with page P_i. The irrelevant page requests which comprise of URLs of embedded objects with filename suffixes like .gif, .jpeg. png, .pdf etc., can be considered to remove from logging records unless they help to evaluate users behavior. Eventually, this step produces a set of user sessions $S=\{S_1,\ldots S_m\}$ with necessary parameters to uniquely identify a logged in user.

4. PATTERN EXTRACTION

Usually data mining methods are employed to automatically analyze usage patterns to generate recommendations. The main techniques used in "pattern discovery and analysis" step are clustering user sessions, generating associated rules, filtering collaboratively, generating sequential patterns and building Markov models. However, few hybrid methods have been developed to combine multiple techniques to improve recommendation accuracy. This section describes usage pattern discovery algorithms that have been applied in web ad recommendation. It is difficult to classify these algorithms based on the data mining techniques as they being used for user modeling. Most of the methods and techniques are combined together for discovering usage patterns. Some methods are proposed only for user modeling and discovered patterns are not used for recommendations. However, the discovered usage patterns in these methods are customized to be appropriate to integrate to use a part of recommender systems.

4.1 Data Models

A Web Ad Recommender System (WARS) accepts a set of web pages of a web site, user profiles and past existing user behavior or new user's interested Ads as input and generates a set of relevant Ads as output. In general the Web Ad Recommendation problem can be formulated as follows: Let $U=\{U_1,U_2,\ldots.U_k\}$ be the set of users of the web site obtained from registration data or server logs and $A=\{A_1,A_2,\ldots.A_n\}$ be the set Ads that can be recommended. Let $R(U_k, A_n)$ be a relevancy function that measures the gain or relevance of Ad to a user U_k ,i.e R : → UxA R where R is a totally ordered set. Then, for each user $U_k \in$ U, the aim of a Web Ad Recommender model is to choose a Ad $A^i \in$ A that maximizes the user's relevancy: Thus, the data model for representing the users is essential and crucial for Web Ad Recommender model. It is obvious that the data model depends highly on the available data sources and the algorithm to be used to generate the recommendation. Since data mining algorithms are usually employed on structured data, there is a tendency to convert the Web data into a structured data format such as matrix expression. In this expression, the rows of the matrix correspond to the objects and the columns correspond to the attributes. Each cell R(i, j) of the matrix is the value of the j^th^ attribute of the i^th^ object.

	A_1	…….	A_k
U_1			
. .		R(i, j)	
U_n			

Figure 2: Matrix expression of Web data model

Matrix expression is widely used in many web mining applications. The matrix expression shown in above Figure 2 can also be used to represent all the data collected for building a recommendation model. The Web user sessions can be modeled as sequence data using matrix expression.

In general a User-Page view matrix is used to describe the relationship between web pages and users who access these web pages. Let k be the number of Web pages and let m be the number of users as shown in Figure 3.

	P_1	P_2	…….	P_k
U_1				
U_2				
….				
U_m				

Figure3: User -Page view matrix

In the matrix, we view users as rows, Web pages as columns, and the frequency as the values of the elements of this matrix, that is F(i, j) as the count of user i accesses web page j during a defined period of time. The i^{th} row of matrix [i,] records the counts of the i^{th} user accesses of all the Web pages during the specified period of time, and the j^{th} column of matrix [, j] records the counts of all users who access the j^{th} web page during the same period of time.

4.2 Users based clustering on User Browsing Behavior

Web log files generated during a user's session is the rich source of information having clue about user's interest. These files will be analyzed to identify user's interest, access trends and potential information about browsed web pages. The K-Means clustering is used to analyze the web log files to form a group of similar user's and web pages. A user cluster is a group of users that seem to behave similarly when navigating through a web site specifically when they access conceptually related web pages of a web site during a given period of time.

We suppose that:

- Users with similar interests should have the similar browsing patterns.
- The general browsing patterns are not changeable during a given period of time for a given user although different user's browsing patterns might be different during the specified period of time.
- Associated Web pages should be browsed by the users with similar interests.

Based on the above assumptions, we can draw user clusters from web logs by analyzing user's browsing information during a period of time.

4.3 Clustering and matching advertisements with web pages

The typical Web Ad Recommender System approach for displaying relevant ads on web pages is outlined in Figure 4. On a single website, upon a request initiated by the user's browser (HTTP GET/POST request), the web proxy server logs the user information, validates the request based on the security policies before web server process the request. When web server received the requests, then it dynamically constructs webpage and sends back the requested page. Internally, web server consults WARS-Ads recommendation system to construct new and dynamically created web pages which cannot be processed ahead of time. Analyzing and constructing new web page at run time entails secure communication and latency costs. However, in a distributed Ad Server implementation, as the page is being returned by the web server will contain necessary client scripts (may be written in javascript, vbscript or ajax), user clicks will be send to the Ad server for dynamic processing of Ad information without page being refreshed or resubmitted to web server. Ad server will totally take care of ad management on the user client browser by getting necessary Meta data about the user behavior, web page information. When the page content is static (that is, the content associated to the given URL is not generated on-the-fly and changes infrequently), the ad server can invest computation resources in a one-time offline process that involves fetching the entire page and performing deep analysis of the page content to facilitate future ad matches.

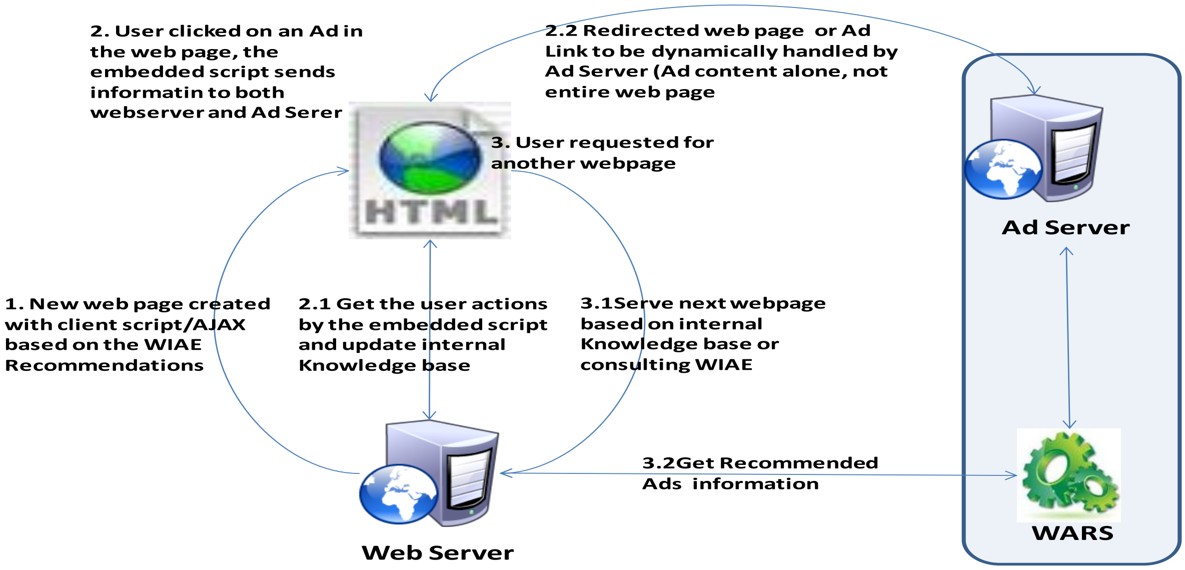
2. User clicked on an Ad in the web page, the embedded script sends informatin to both webserver and Ad Serer
2.2 Redirected web page or Ad Link to be dynamically handled by Ad Server (Ad content alone, not entire web page
HTML
3. User requested for another webpage
Ad Server
1. New web page created with client script/AJAX based on the WIAE Recommendations
2.1 Get the user actions by the embedded script and update internal Knowledge base
3.1Serve next webpage based on internal Knowledge base or consulting WIAE
3.2Get Recommended Ads information
WARS
Web Server

Figure 4: State Diagram of webpage with Ads

5. AD RECOMMENDER SYSTEM

The web page will have several words which can be extracted and the term frequencies could be analyzed to build tables. Similarly the advertisements will also have some text that describes the product. We believe that the terms in the webpage and ads are synonyms. Hence we implement a Naïve Bayes classifier which assumes that the each attribute chooses for classification is independent. The process works as described below in Figure 5, WARS encounter a new user when the user is served an impression. A user assigned website cluster corresponding to the visited website. The user is then assigned to a user cluster corresponding to the prior probabilities of the user visiting a history of website clusters.

Then user browsed web page will be directed to Naïve Bayes classifier. Then we classified each browsed page summary and each ad with respect to the taxonomy. Following [11], we constructed additional features based on these immediate classifications as well as their ancestors in the taxonomy (the weight of each ancestor feature was decreased with a damping factor of 0.5). Each page and ad was represented as a bag of words (BOW) and an additional vector of classification features. Finally, the ad retrieval function was formulated as a linear combination of similarity scores based on both BOW and classification features:

$$score\ (page\ , ad\ ) = \alpha \cdot simBOW\ (p, a) + \beta \cdot sim_{class}\ (p, a)$$

Where simBOW(p, a) and sim_{class}(p, a) are cosine similarity scores between page p and ad a using BOW and classification features, respectively. Then Ad recommendation system uses the above retrieve function to retrieve relevant ads from database scrolls the retrieved ads dynamically on user browsed web page. Thus, a user can be assigned a particular ad on a user interested webpage that generates a higher user response. By targeting particular campaigns to certain users on a given website, more advertising dollars will be generated. A snapshot of training sample is shown in Table 2.

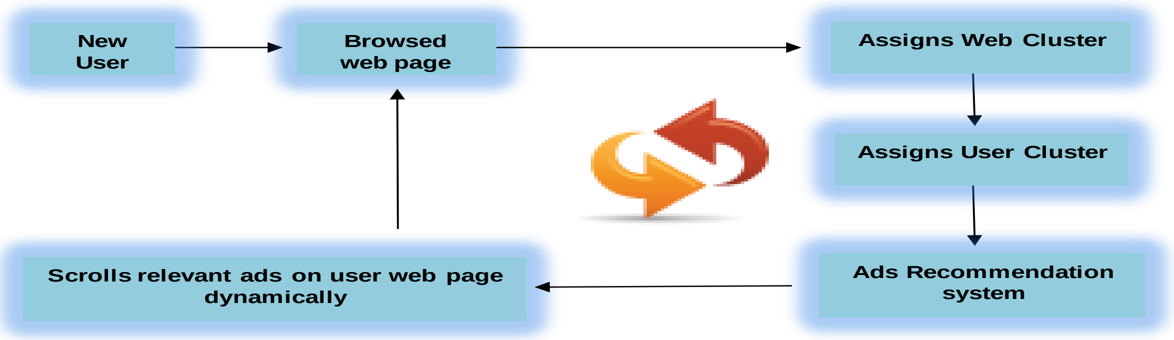
New User
Browsed web page
Assigns Web Cluster
Assigns User Cluster
Ads Recommendation system
Scrolls relevant ads on user web page dynamically

Figure 5: Progression of new user

Table 2: Training Sample Database

Adaccid	Webpage#	$Term_1$	$Term_2$	…….	$Term_n$	Cluster name
Ad_1	http://sports.yahooo.com/	23	11	…….	12	Sports
Ad_2	http://www.health.com/	22	55	……	15	Health
….	……	….	…..	……	….	
Ad_n	http://www.google.com/finance	12	11	……..	45	Finance

Table 3: Test Sample Database

Webpage#	$Term_1$	$Term_2$	…….	$Term_n$
http://sports.yahooo.com/	12	22	….	18
http://www.health.com/	13	10	……	12
….	….	….	….	……
http://www.google.com/finance	15	18	……	30

Table 4: Test Sample Database

Ad #	$Term_1$	$Term_2$	…….	$Term_n$
Sports	12	22	….	18
Health	13	10	……	12
….	….	….	….	……
Finance	15	18	……	30

DOI 10.17148/IJARCCE.2016.52137

The Naïve Bayes classifier is a lazy classifier, hence does not have immediate learning. On notification of new user visiting a page the probabilities are evaluated and based on the evaluation the relevant ads will be identified. A snapshot for sample test data is shown in Tables 3 and 4.

6. EXPERIMENTS AND RESULTS

Because of the high cost of processing time and processing power required by most of the classifiers, and because the experiment has to be repeated ten to hundred times to get the best out of the classifiers, the experiment was done stage by stage. The testing started with the data of five categories of Sports, Science, Finance, Movies and Health. As shown in the Table 5, the five categories have a total of 429 attributes including one attribute for the category label (i.e., they have 28 attributes in common between the three of them) and a total of 1900 instances.

Table 5: The five categories with their attributes

Category	Health	Science	Movies	Sports	Finance
Health	37	2	15	9	5
Science	2	55	10	13	22
Movies	15	10	176	22	36
Sports	9	13	22	100	10
Finance	5	22	36	10	192

The whole experiment dataset has two types of attributes: numeric attributes for the weight of the feature words and nominal attributes for the class labels. The classifier used in the experiment is the Naïve Bayes classifier. The testing on the dataset is done using 10-fold stratified cross-validation. Weka provides a number of options for measuring the performance of a classifier, out of which the summary statistics, detailed accuracy by class, and confusion matrix are shown. The confusion matrix is built by NB Classifier as shown in Table 6. Looking vertically at the value of a class in the confusion matrix, one can see the instances of a category as assigned by a classifier. For example, looking the above confusion matrix shows that the NB classifier has correctly classified 119 instances of the Health class (True Positives, TP) while 12 instances are False Positives (FP). Each row of the confusion matrix shows the actual number of instances in a category. For the five categories, the NBC correctly classified 92.47% of the 1900 instances.

Table 6: Confusion matrix

Health	Science	Movies	Sports	Finance	Class
119	1	2	4	5	**Health**
0	65	6	5	12	**Science**
3	5	489	11	34	**Movies**
1	5	8	462	18	**Sports**
7	4	15	18	601	**Finance**

Summary statistics:
1. Correctly classified instances 1,757 92.47%
2. Incorrectly classified instances 143 7.52%

Moreover, the confusion matrix can be taken as a data source for measures used for calculating the detail accuracy of each class, shown in Table 7.

Table 7: accuracy by class

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.915	0.005	0.994	0.915	0.952	**Health**
0.812	0.009	0.989	0.812	0.891	**Science**
0.940	0.016	0.983	0.940	0.961	**Movies**
0.924	0.020	0.978	0.924	0.950	**Sports**
0.897	0.038	0.959	0.897	0.926	**Finance**

Considering precision, recall, F-measure and observing the detail class statistics in Table 7, it can be seen that the NB classifier consistently gives better performance. For the five categories, the Ads Recommendation system dynamically recommended the relevant ads upto 92.47%.

7. CONCLUSION

With Web Ad Recommender System, its clients will ultimately benefit from the outcome of this research effort. They will be able to provide their consumers with more effective ad campaigns while generating more revenue for both parties. Our model use behavioral data about online consumers to publish ads more efficiently increasing user response to advertising through better placement, duration and format of advertisements to targeted audience. Web Ad Recommender System could reduce the total number of ads served for the same or better in time thus reducing costs. Using this framework we can increase Return On Investment (ROI) for Ad campaigns. We can also provide accurate and more related advertisements and information to users more purposefully. This analysis can also be used to further improve other aspects of Online Advertising business facilitating them to gain a competitive advantage in the growing and competitive market. The suite of algorithms that we have applied resulted in successful and acceptable quality predictions.

DOI 10.17148/IJARCCE.2016.52137