

# Creating Evolving Web User Behavior Models Automatically

Shivanandachary. P., *M.Tech Student, ASTRA, Bandlaguda, Hyderabad,*

Malathi. T. *Asst. Prof., ASTRA Bandlaguda, Hyderabad,*

**Abstract:** The available online information is increasing rapidly and it is becoming difficult for the users to locate the relevant web pages. Moreover the information is more valuable and its large volume is limiting its value. Recommendation system which aims at providing relevant information to users is very much important and desirable. Many researchers have indulged in constructing the user profiles based on the browsing history of the user. The user-profiling or user-modeling task involves inferring unobservable information about users from observable information about them, for example their actions or utterances. In order to create the user profile, the behavior of the user plays a vital role and it is difficult to build a full description of all possible behaviors of the users, since these behaviors evolve with time and they are not static. Hence, many researchers have given different techniques of building the user behavior profiles in static conditions and also in dynamic conditions. In this paper, we are going to have a study of some popular techniques/methods for collecting user related information, activities, interests and building user behavior profiles. We review how each of these profiles is constructed and give examples of projects that employ each of these technique.

**Keyword:** Web, User Modelling, User Classification.

## 1. Introduction

In this modern world the usage of communication networks, such as Internet, has been widely increased. The effort of finding information over the Web has been greatly facilitated over the recent years with significant improvements in the quality of the results returned by search engines. Internet provides a world of data in one single place. It's a valuable instrument. The users of internet, termed here as 'web users' are performing variety of tasks/activities ranging from accessing information about weather, sports related information, stock exchange information, news etc., even the web users use the internet to perform electronic commerce activities such as buying or selling goods/services online.

Several techniques have been developed to reduce the time that a user has to spend in accessing web pages or information of interests to the user. According to Susan Gauch [2], there are five basic approaches to identify the user: software agents, logins, enhanced proxy servers, cookies and session ids. Even there are some personalized systems which address the over burdening problems by building, managing and representing information customized for individual users.

The building of customization involves the identification of the irrelevant information and adding the information related to the interest of a web user.

In order to build a user profile we need to collect the individual user information. And there are two ways of collecting the information, the first one being Explicit Information collection, which is done through direct user interaction, such as questionnaires and the other being Implicit Information collection, which is done with the help of some agents, who monitor the activities of the user[7].

User profiles is based on heterogeneous information associated with individual user or a group of users who may have similar interests and similar navigational behavior. The other concept which comes into the picture of user profile is “The Behavior of the user”. Based on the behavior of the user we can build the user profiles. In the process of building user profiles the behavior plays a vital role, since the user may have a change in his interest or the time may force him to have a change in the behavior.

Hence, the profiles can be augmented as static and dynamic profiles. In static user behavior profile, the same information will be maintained over time. But whereas in dynamic user behavior profile, there will be changes in the interest, likes and dislikes etc. In section 2 we will discuss about the user profiling. In user profiling we will discuss about the collection, organizing and interpreting the user information. In section 3 we will discuss about the various approaches of different researchers in different environments.

Finally, we conclude the paper with the user-profiling task involves inferring unobservable information about users from observable information about them. User modelling is concerned with two main issues: acquisition and representation. The acquisition of user profile is related to the mechanism an

agent had to formulate assumptions about user.

## **2. USER Modelling:**

The process of gathering, organizing and interpreting the user information is called user profiling. In this section we will discuss about the collection of individual user information. The basic requirement for this is to identify the user as unique. And this is being discussed in section 2.1. The information collected may be explicitly input by the user or implicitly gathered by a software agent. It may be collected on the user’s client machine or gathered by the application server itself. Depending on how the information is collected, different data about the users may be extracted. And the methods of user information collection is discussed in 2.2.

### **2.1. USER IDENTIFICATION METHODS**

Accurate user identification is not a critical issue for systems that construct Profiles representing groups of users, it is a crucial ability for any system that constructs models that represent individual users. As we already had in the introduction that according to Susan Gauch [2], there are five basic approaches to user identification: software agents, logins, enhanced proxy servers, cookies, and session ids[6].

Software agents are small programs that reside on the user’s computer, collecting their information and sharing this with a server via some protocol. This approach is the most reliable because there is more control over the implementation of the application and the protocol used for identification.

The next most reliable method is based on logins. Because the users identify themselves during login, the identification is generally accurate, and the user can use the same profile from a variety of physical locations. On the other hand, the user must create an account via a registration process, and login and logout each time they visit the site, placing a burden on the user. Enhanced proxy servers can also provide reasonably accurate user identification. However, they have several drawbacks. They require that the user register their computer with a proxy server. Thus, they are generally able to identify users connecting from only one location, unless users bother to register all of the computers they use with the same proxy server. The final two techniques covered, cookies and session ids, are less invasive methods. The first time that a browser client connects to the system, a new userid is created. This id is stored in a cookie on the user's computer. When they revisit the same site from the same computer, the same userid is used. This places no burden on the user at all. However, if the user uses more than one computer, each location will have a separate cookie, and thus a separate user profile. Also, if the computer is used by more than one user, and all users share the same local user id, they will all share the same, inaccurate profile. Finally, if the user clears their cookies, they will lose their profile altogether, and if users have cookies turned off on their computer, identification and tracking is not possible. Session ids are similar, but there is no storage of the user id between visits – each user begins each session with a blank slate, but their activity during the visit is tracked. In this case, no permanent user profile can be built,

but adaptation is possible during the session. [5]

## 2.2. Methods of organizing user information

User profile construction techniques can be partitioned by the type of input used to build the profile. In this section we discuss about the explicit and implicit information collection. **Explicit User Information Collection.** Explicit user information collection methodologies, often called explicit user feedback, rely on personal information input by the users, typically via HTML forms. The data collected may contain demographic information such as birthday, marriage status, job, or personal interests. In addition to simple checkboxes and text fields, a common feedback technique is the one that allows users to express their opinions by selecting a value from a range. All these methodologies have the drawback that they cost the user's time and require the user's willingness to participate. If users do not voluntarily provide personal information, no profile can be built for them. Many sites collect user preferences in order to customize interfaces. This customization can be viewed as the first step to provide personalized services on the Web. The collection of preferences for each user can be seen as a user profile and the services provided by these applications adapt in order to improve information accessibility. For instance, MyYahoo! explicitly ask the user to provide personal information that is stored to create a profile. Users may not accurately report their own interests or demographic data, or, since the profile remains static whereas the user's interests may change over time, the profile may become increasingly

inaccurate over time. **Implicit User Information Collection.** User profiles are often constructed based on implicitly collected information, often called implicit user feedback. The main advantage of this technique is that it does not require any additional intervention by the user during the process of constructing profiles. Kelly and Teevan [7] give an overview of the most popular techniques used to collect implicit feedback, and the type of information about the user that can be inferred from the user's behavior. Because they only require a onetime setup, do not require new software to be developed and installed on the user's desktop, and only track browsing activity, proxy servers seem to be a good compromise between easily capturing information and yet not placing a large burden on the user. Browsing histories are a common source of information from which user interests are extracted. Letizia was one of the first systems to interactively collect and exploit implicit user feedback [8]. Based on previously visited pages and bookmarked pages, it suggests links on the current page that might be of interest. Other browsing assistants based on browsing agents are WebMate, Vistabar, and Personal WebWatcher. Some literature in this area distinguishes between browsing assistants and browsing agents. Vistabar is a prototypical browsing assistant, a tool that helps users track viewed urls, fill out forms or fetch pages without any specific agenda. In contrast, WebMate and Personal WebWatcher are examples of browsing agents that perform more critical tasks such as highlighting hyperlinks of likely interest to the user, recommending urls, or refining search keywords. The drawback to this approach is that this approach is that,

since it is resident on a personal computer, the user profile built would typically only be available when the user was using that particular computer. The above approaches all focus on collecting information about the users as they browse or perform other activities. Because they try to capture and share what the user is doing on their computer, they are essentially client-side approaches. All client-side approaches place some burden on the users in order to collect and/or share the log of their activities. Although they have access to less information than client-side approaches, they place no burden on the user at all, and can silently collect the information via cookies, logins, and/or session ids. The search Histories have been explored as a source of information for user profiling then can then be exploited to provide personalized search. Since implicit feedback places less burden on the user, and it automatically updates the user interacts with the system, it seems to be the preferable method of collecting information about users. One drawback to implicit feedback techniques is that they can typically only capture positive feedback. When a user clicks on an item or views a page, it seems reasonable to assume that this indicates some user interest in the item. However, it is not as clear, when a user fails to examine some data item, that this is an indication of disinterest. Thus, in general, implicit feedback techniques do not collect negative feedback.

### 3. The different approaches by different researchers

In this section we will discuss about the work done in the area of creating user behavior profile. Ramesh Subramonian, Ramana Venkata and others had others have invented a user profiling module (UPM) which executes on client computer and generates personalized user profile [1]. UPM builds user profiles by monitoring and collecting information based on the user's activities. The user profile creation process is performed in two stages, where in the first stage UPM monitors user activities and collects only that information which is permitted by the user. The activities monitored by UPM may include the user's interactions with browser, the user's interactions with other applications executing on client computer, activities performed by the user on external devices which are either coupled to client computer or which are capable of exchanging information with client computer, and other like activities. UPM collects both content information and context information for the monitored user activities as shown in Fig. 2. This information is used by UPM to generate user profiles.

The interactions monitored by UPM as shown in Fig. 1 may include user's web surfing activities, monitoring electronic commerce transactions, web searches, financial transactions, interactive activities such as participation in chat rooms and games and the like. After collecting information associated with the various user activities, the profile creation process takes place in the second stage. According to this technique the user profile is constructed by UPM which

executes on client computer. Therefore, the created profile is stored on client computer and not on some remote server. Hence, the profile is not exposed to the outside world without explicit permission for the user. Therefore the by building the user profile on the client computer and by storing the user profile on the client computer, the user profile is built without distribution of the user profile to a computer other than the client computer.

There exists several definitions for user profile [1]. It can be defined as the description of the user interests, characteristics, behaviors, and preferences. According to DANIELA GODOY and ANALIA AMANDI [3]. The user-profiling or user-modeling task involves inferring unobservable information about users from observable information about them. User modeling is concerned with two main issues: acquisition and representation. The acquisition of user profiles is related to the mechanisms an agent has to formulate some assumptions about users. In this regard, users provide information about themselves during interaction with system. The most usual approach to profile acquisition, however, is the application of learning mechanisms. Learning of user profiles based on the observation of user behavior leads to explicit representations of user interests that enable agents to make decisions about future actions.

A user profile allows agents to make decisions about actions to be carried out with individual, previously unseen pieces of information. If user profiles are acquired using a learning algorithm, decision making is directly supported by the learning method. In other cases,



several methods can be used in order to compare user interests and information items. Besides profile-item matching, agents acting collaboratively with other agents toward a common goal also need to be endowed with a method to match user profiles in order to find users with similar interests [5]. The adaptation of user profiles is also an important factor in user profiling. However, since the interaction may extend over a long period of time, the user interests cannot be assumed to remain constant during such a time.

Three main approaches have been developed to provide agents with this knowledge: the user-programming, the knowledge-engineering and the machine-learning approaches.

### 3.1 Observation of user behavior:

An explicit user profile is elicited from a series of questions designed to acquire user interests and preferences precisely. The main advantage of this method is the transparency of agent behavior as decisions can be easily deduced from the data provided. However, it requires a great deal of effort from users and, additionally, users are not always able to express their interests because they are sometimes still unknown.

Implicit knowledge acquisition is often the preferred mechanism since it has little or no impact on the user regular activities. Unobtrusive monitoring of users allows agents to discover behavioral patterns that can be used to infer user interests, preferences and habits. In order to achieve this goal, a number of heuristics are commonly employed to infer facts from existing data.

Some sources of information left by a user after browsing include:

- 1) The history of the user requests for current and past browsing sessions that is maintained by most browsers;
- 2) Bookmarks giving a quick means for accessing a set of documents exemplifying user interests;
- 3) Access logs where entries correspond to HTTP requests typically containing the client IP address, time-stamp, access method, URL, protocol, status and file size;
- 4) Personal homepages and material as well as their outgoing links.

A personal information agent typically learns about individual users by observing their behavior over time. However, it may take a significant amount of time and observations to construct a reliable model of user interests, preferences and other characteristics. To reduce this time, agents can take advantage of the behavior of similar users accessible through the knowledge of other agents.

### 3.2 User-profile learning and representation:

In order to adapt their assistance to individual users, agents have to learn about user preferences and attitudes and model them into user profiles. According to A. Alaniz Macedo, K.N. Truong, J.A. Camacho Guerrero, and M. Graca Pimentel [4], the effort of finding information over the Web has been greatly facilitated over the recent years with significant improvements in the quality of the results returned by search engines. Two issues arise because search engines depend on a set of words given by users. First, the number of keywords provided by users is often suggested to

be small. Second, users must formulate an appropriate query for search engines to return completely satisfactory results. Existing history mechanisms do not provide appropriate searching services on the recorded data.

Recommendation systems are another category of application aimed at supporting users when searching for information. They are based on the idea that users often face the problem of having to make choices without sufficient experience and can use other people's recommendations. Recommendation systems leverage the notion that people are better at recognizing information needed that they see than at handling keywords over search engines. In this paper, WebMemex, a system geared towards making recommendations based on those pages the users themselves have previously seen. This is achieved by continuously capturing users' Web surfing activity. The WebMemex prototyped assists the users through a set of different infrastructures and applications supported by an open architecture.

This system:

- a) Captures navigation using an extensible capture and access infrastructure;
- b) Identifies semantic relationships between Web pages browsed by users using a linking server that manipulates semantic as similarity of terms according to Latent Semantic Indexing theory;
- c) Stores the associations identified in an open linkbase;
- d) Handles the groups of people each user wants to share

These characteristics make WebMemex an example of applying open hyper media technology on the Web in this case, especially to create a recommender system. The WebMemex application captures and recommends Web pages for groups of users. The WebMemex service is supported through an augmented Web proxy server. When information is requested, the proxy server retrieves the information and immediately delivers it back to the requesting client. If users enable capture, then the retrieved document is also passed to the capture component. The proxy only logs information returned to the Web browser when the content type is text/html. When users want to visit the related pages using WebMemex, the access component will retrieve the links from the storage component. [8]

Usually the information is gathered without the user's permission by processes resident on web servers which are typically remote from the client computer used by the user. The user typically has no control either on the contents of the collected information or on when the information is collected. This lack of control rises in security concerns for the user.

#### 4. The Proposed Approach

The proposed approach introduces EvWUBM (Evolving Web User Behaviour Model) which can perform automatic clustering, classifier design, and classification of the behaviour models of users. The user behaviour classifier is based on Evolving Fuzzy Systems and it takes into account the fact that the behaviour of any user is not fixed, but is rather changing. It starts to be filled in "from scratch" by assigning temporarily to the library the

first observed user as a prototype. The model evolve according to the changing user behaviours observed in the environment.

This EvWUBM (Evolving Web User Behaviour Model) algorithm has two main steps as followed.

### EvWUBM:

This approach involves two steps:

#### 1) Creating and Evolving the Classifier

This action involves in itself two sub actions.

a) Creating the user behavior models: This sub action analyses the sequence of actions performed by different web users and creates corresponding models.

b) Evolving the classifier: This sub action update of the classifier, including the potential of each behavior to be a prototype, stored in EMLib.

#### 2) User Classification

The user models created in the previous action are associated with one of the prototypes from the EMLib.

These steps are detailed in the following:

### 4.1. Creating and Evolving the Classifier

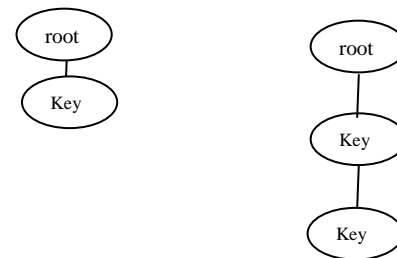
This step includes two steps

#### i) Creating User Behaviour Model

#### ii) Evolving the Classifier

### 4.1.1 Creating User Behaviour Model

- In this step, first the user behaviour on web will be captured.
- Then the user Behaviour model will be created according to the captured information and stored.
- Changes in the behaviour of a user will be updated into the model.
- The following diagram gives an example for user behaviour.



### 4.1.2 Evolving the Classifier

- a) Calculating the potential of a data sample: A prototype is a data sample (a behaviour represented by subsequence of actions).

The classifier is first initialised with the first data sample, which is stored in EMLib. Then, each data sample is classified to one of the prototypes defined in the classifier.

Finally based on the similarity of the new data sample, to become a prototype, it could form a new prototype.

Therefore EvWUBM uses cosine distance method to calculate the similarity between two data samples, as it described below,



cosineSimilarity=

$$\frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

cosDist(Ai,Bi)=

$$1 - \frac{A \cdot B}{\|A\| \|B\|}$$

$$= 1 - \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Where Ai, Bi represents the two samples to measure its distance and n represents the number of different attributes in the both the samples.

Cosine distance has an advantage that it tolerates different samples to have different number of attributes.

b) Creating new prototypes: After calculating the distance between two data sample, according to the result, a new prototype is formed.

If the distance (similarity) is low, then a new prototype will be created and then all the users with similar data sample are assigned to the prototype.

Consider two user with behaviours:

User1: Key1, Key3, Key4

User2: Key1, Key2, Key3

Key1: 1 1

Key2: 0 1

Key3: 1 1

Key4: 0 1

The two vectors are:

U1: [1, 0, 1, 0]

U2: [1, 1, 1, 1]

The similarity between these two vectors is 0.6667.

The cosine distance = 1-0.6667 =0.3333

Hence we can say that the difference between these two samples is 0.333.

## 4.2. User Classification

- First, the user will be classified with to the existing prototypes, based on the basic information given.
- Then, according to the user behaviour captured on web, data sample collected and compared with all the prototypes stored in EMLib using cosine distance method.
- Based on Cosine distance values the users will be classified (clustered) accordingly.

The smallest distance determines the closest similarity.

## 5. Conclusion:

Hence in this paper, we propose an evolving method to create and update user behaviour models for a web user as considering the behaviour of a user will change according to time. This model is useful to give appropriate recommendations to the user according to his behaviour model.

## References:

1. Ramesh Subramonian, Ramana Venkata, Pangal P. Nayak, Joy A. Thomas, "Methods for creating user profiles," 2000.
2. Susan Gauch, Micro Speretta, Araving Chandramouli and Alessandro Micarelli, "User Profiling for Personalized Information Access," The Adaptive Web, Lncs 4321, pp. 54-89, 2007.
3. D. Godoy and A. Amandi, "User Profiling in Personal Information Agents: A Survey," Knowledge

- Eng. Rev., vol. 20, no. 4, pp. 329-361, 2005.
4. A. Alaniz Macedo, K.N. Truong, J.A. Camacho Guerrero, and M. Graca Pimentel, "Automatically Sharing Web Experiences through a Hyperdocument Recommender System," Proc. ACM Conf. Hypertext and Hypermedia (HYPERTEXT '03), pp. 48-56, 2003.
  5. D. Godoy and A. Amandi, "User Profiling for Web Page Filtering," IEEE Internet Computing, vol. 9, no. 4, pp. 56-64, July/Aug. 2005.
  6. Kelly, D., Teevan, J.: Implicit feedback for inferring user preferences: a bibliography. ACM SIGIR Forum 37(2) (2003) 18-28.
  7. Lieberman, H.: Letizia: An Agent That Assist Web Browsing In: Proceedings of the 14<sup>th</sup> International Joint Conference On Artificial Intelligence, Montreal, Canada, August (1995) 924-929.

IJERT