Content-Based Recommendation for Web Personalization

R. Kousalya\(^1\), K. Saranya\(^2\), Dr. V. Saravanan\(^3\)

\(^1\)PhD Scholar, Manonmaniam Sundaranar University, Tirunelveli
HOD, Department of Computer Applications, Dr. NGP Arts and Science College, Coimbatore, India
e-mail: kousalyacbe@gmail.com

\(^2\)M.Phil Scholar, Department of Computer Science, Dr. NGP Arts and Science College, Coimbatore, India
e-mail: tksaranya@gmail.com

\(^3\)Director, Department of Computer Applications, Sri Venkateswara College of Computer Applications and Management, Ettimadai, Coimbatore, India
e-mail: profsaran@hotmail.com

Abstract - The amount of information available online is increasing exponentially. The existence of such abundance of information, in combination with the dynamic and heterogeneous nature of the web, makes web site exploration a difficult process for the average end user. One approach to satisfy the requirements of the user is to personalize the information available on the Web, called Web Personalization. Recommender systems have the effect of guiding users in a personalized way to interesting objects in a large space of possible options. In this paper, we aim at describing architecture of Content-based recommendation systems which try to recommend items similar to those a given user has liked in the past. Indeed, the basic process performed by a content-based recommender consists in matching up the attributes of a user profile in which preferences and interests are stored, with the attributes of a content object (item), in order to recommend to the user new interesting items.

Keywords – Web Personalization, Recommender Systems, Content-based Recommendation.

I. INTRODUCTION

The personalization process can result in the dynamic generation of suggestions, the creation of pages according to the needs of the user, highlighting of existing hyperlinks that are exactly required by the users. Most of the earlier research efforts in Web Personalization deal with Web Usage Mining [1].

Pure usage-based personalization, however, presents certain shortcomings, such as when there is insufficient use of data available in order to extract patterns, or when the web site’s content changes and new pages are added but are not yet included in the web logs. The users’ visits usually aim at finding information concerning a particular subject, thus the underlying content semantics should be a dominant factor in the process of web personalization. There have been a number of research studies that integrate the web site’s content in order to enhance Web Personalization process [2].

Content-based filtering originates from information retrieval and case-based reasoning research (Hammond et al., 1996). The success of the content-based method relies on an ability to accurately represent recommendable items in terms of a suitable set of content features, and to represent user profile information in terms of a similar feature set. The relevance of a given content item to a specific target user is proportional to the similarity of this item to the user’s profile; content-based filtering methods select content items that have a high degree of similarity to the user’s profile. Content-based methods build models that link information about the content of items a user manipulates to the user’s preferences concerning those items. In summary, content-based approaches typically offer us a means for describing items of
user interest and a means for comparing item descriptions to locate close matches.

II. RECOMMENDATION SYSTEMS FOR WEB PERSONALIZATION

When users browse through a web site they are usually looking for items they find interesting. Interest items can consist of a number of things. For example, textual information can be considered as interest items or an index on a certain topic could be the item a user is looking for. Another example, applicable for a web vendor, is to consider purchased products as interest items. Whatever the items consist of, a web site can be seen as a collection of these interest items.

Dynamically adding hyperlinks is often used for personalization and is the only approach that will be considered here. Recommender systems can present their recommendations in other ways however. Amazon.com for example, also delivers recommendations through email. Another approach is to display the average rating of an item from people who are correlated with the user.

In general, the better the web site is organized, the harder it will be to personalize the site. Ironically enough, many information filtering techniques can be used to improve the structure of a web site. Examples are recommendations that Amazon.com makes when visitors select certain books. A related book is suggested if other visitors have purchased it along with the selected book. These recommendations are not personalized but are the same for every visitor.

Search engines exist that uses document clustering techniques to organize a directory of web sites. This directory is formed by partitioning web pages into domains via clustering. Most web sites, especially larger ones, however can never be perfectly optimized for all users. Users have different interests and personalizing a web site could help them find information faster as they otherwise would have or wouldn’t have found at all.

III. ARCHITECTURE OF CONTENT-BASED RECOMMENDER SYSTEMS:

Content-based Information Filtering (CBIF) systems need proper techniques for representing the items and producing the user profile, and some strategies for comparing the user profile with the item representation. The high level architecture of a content-based recommender system is depicted in Figure 3.1. The recommendation process is performed in three steps, each of which is handled by a separate component:

- **Content Analyzer** – When information has no structure like text, some kind of pre-processing step is needed to extract structured relevant information. The main responsibility of the component is to represent the content of items like documents, Web pages, news, product descriptions, etc. coming from information sources in a form suitable for the next processing steps. Data items are analyzed by feature extraction techniques in order to shift item representation from the original information space to the target one such as Web pages represented as keyword vectors. This representation is the input to the profile learner and filtering component;
- **Profile Learner** – This module collects data representative of the user preferences and tries to generalize this data, in
order to construct the user profile. Usually, the generalization strategy is realized through machine learning techniques [3], which are able to infer a model of user interests starting from items liked or disliked in the past. For instance, the profile learner of a Web page recommender can implement a relevance feedback method [4] in which the learning technique combines vectors of positive and negative examples into a prototype vector representing the user profile. Training examples are Web pages on which a positive or negative feedback has been provided by the user;

• **Filtering Component** – This module exploits the user profile to suggest relevant items by matching the profile representation against that of items to be recommended. The result is a binary or continuous relevance judgment (computed using some similarity metrics [5]), the latter case resulting in a ranked list of potentially interesting items. In the above mentioned example, the matching is realized by computing the cosine similarity between the prototype vector and the item vectors.

The first step of the recommendation process is the one performed by the content analyzer, that usually borrows techniques from Information Retrieval systems [6,7]. Item descriptions coming from Information Source are processed by the content analyzer that extracts features from unstructured text to produce a structured item representation, stored in the repository Represented Items. In order to construct and update the profile of the active user (for which recommendations must be provided) her reactions to items are collected in some way and recorded in the repository Feedback. These reactions, called annotations[8] or feedback, together with the related item descriptions, are exploited during the process of learning a model useful to predict the actual relevance of newly presented items. Users can also explicitly define their areas of interest as an initial profile without providing any feedback.

Typically, it is possible to distinguish between two kinds of relevance feedback: positive information (inferring features liked by the user) and negative information (i.e., inferring features the user is not interested in [9]). Two different techniques can be adopted for recording user’s feedback. When a system requires the user to explicitly evaluate items, this technique is usually referred to as “explicit feedback”; the other technique, called “implicit feedback”, does not require any active user involvement, in the sense that feedback is derived from monitoring and analyzing user’s activities. Explicit evaluations indicate how relevant or interesting an item is to the user [10]. There are three main approaches to get explicit relevance feedback:

• **Like/dislike** – items are classified as “relevant” or “not relevant” by adopting a simple binary rating scale, such as in [11];
• **Ratings** – a discrete numeric scale is usually adapted to judge items, such as in [12]. Alternatively, symbolic ratings are mapped to a numeric scale, such as in Sysskill & Webert [13], where users have the possibility of rating a Web page as hot, lukewarm, or cold;
• **Text comments** – Comments about a single item are collected and presented to the users as a means of facilitating the decision-making process, such as in [14]. For instance, customer’s feedback at Amazon.com or eBay.com might help users in deciding whether an item has been appreciated by the community. Textual comments are helpful, but they can overload the active user because she must read and interpret each comment to decide if it is positive or negative, and to what degree. The literature proposes advanced techniques from the affective computing research area to make content-based recommenders able to automatically perform this kind of analysis. Explicit feedback has the advantage of simplicity, albeit the adoption of numeric/symbolic scales increases the cognitive load on the user, and may not be adequate for catching user’s feeling about items.

### IV. MANAGEMENT OF USER PROFILES

Content-based filtering also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user.
Several architectures are used for personalized services on the Web and they differ mainly in the locations of the management and storage functions.

The most common architecture is the server-based architecture, in which the user profiles are both stored and managed at the server as in figure 4.1A. Since the profiles of all users are centralized, the server needs to identify the user in order to extract the right user profile. This is done by using an authentication mechanism.

![Server-based architecture](image)

**Fig 4.1 : Architectures for Users’ Profile Management**

This architecture is efficient in that the user profiles do not transit through the network. The centralization of all the user profiles enables the use of both content-based but prevents user profiles from being shared between applications on different servers. With this architecture, the service provider has to supply both hardware and software for the management and storage of the users’ profiles. For a worldwide service, those profiles may represent a large amount of data.

The second architecture stores the user profiles on the client side and manages them on server side as given in the figure 4.1B. This architecture enables the use of content-based filtering and profile sharing. The browser must provide a mechanism for permanently storing data on the user's computer, and this is a sensitive issue because most browsers, for security purpose do not allow a Web application (for example a Java applet) to permanently store any information on the terminal. The "Cookie" mechanism introduced by Netscape is an exception to this rule. By setting a cookie, an application can get data permanently stored by the browser and automatically sent back when the user accesses the application again. The main advantage of this second architecture is the distributed nature of the storage, which frees the service provider from supplying software and disk space for the database, but the transmission of the user profile between its storage location (client) and the management location (server) increases the response delay. The third architecture manages and stores the user profiles on the client side as represented in figure 4.1C. In this, the personalization is done by the browser, and the architecture is therefore not a client-server architecture anymore (at least with respect to the personalization). This architecture enables the use of content-based filtering and user profile sharing.

Furthermore, no standard such as the **Common Gateway Interface** (CGI) has been defined for the management and storage of the user profiles on the server side. Each personalized Web application that uses the server-based architecture has to interface individually with the database that contains the user profiles.

**V. EXPLORATION STRATEGIES**

The concept applied to content-based filtering try to find the most relevant documents based on the user’s behavior in the past. Such approach however restricts the user to documents similar to those already seen. This is known as the over-specialization problem.

The interests of a user are rarely static but change over time. Instead of adapting to the user’s interests after the system has received feedback one could try to predict a user’s interests in the future and recommend documents that contain information that is entirely new to the user. A search engine has to decide between two types of information delivery when providing the user with recommendations:
• Exploitation: The system chooses documents similar to those for which the user has already expressed a preference.

• Exploration: The system chooses documents where the user profile does not provide evidence to predict the user’s reaction.

VI. CONCLUSION

In this paper, we analyzed the main content recommender systems, by highlighting the reasons for which a more complex “semantic analysis” of content is needed in order to go beyond the syntactic evidence of user interests provided by keywords. Moreover, we also observed that our method can provide more suitable content or user preferences, even when the number of recommended results is small.

VII. REFERENCES


