Acquisition of User's Interests

Ramin Yasdi[1]

Abstract A significant problem in many user modeling systems is that the user is forced to define his interests explicitly. This task is unacceptable for most users. Approaches which automatically generates interest profiles suffer from the disadvantage that the profile is very complex. In this work we discuss the problems related to this topic.

1 Introduction

One of the most difficult problem of applying user modeling to computer system is the acquisition of the user models. Information about user must gathered in order to build up the user models. Even after constructing an initial user model, the acquisition process often continues throughout the entire life of the user modeling system. Whilst many user models can function perfectly with behavioral impression of the user, the provision of user profile requires a richer understanding and incorporating information about the user's interests. This raised a number of important and difficult questions: How can we know what the user interests are, and how can we know that we know enough about user's interest? We argue that user's interest can be modeled, up to a certain point, but to ask whether or not we can know what the user's interest are is a misunderstanding question.

Can a user's actions be sufficient input for a system to adapt in order to accommodate the particular needs of a particular user or user type? Even if we include a user's linguistic interaction with a system - in the form of text for inquiring the user, then providing that a model can anticipate the form and content of this inquiry, still are misunderstanding of behavior. In some domains (and perhaps in all domains at some times), it is impossible to anticipate the form and content of a user's interest. In an interactive system feedback from user given explicitly or implicitly (e.g. the time of displaying a message can be used). The evaluation of the feedback indicated by implication is problematic because the user's behavior must be interpreted correctly[2]. An above average long displayed text can have different reasons:

- The user is interested in the text and therefore reads it exactly. The longer time of display in this case indicates a positive feedback.
- The user needs a lot of time to notice that the text is not interesting because the text has a high similarity with an interesting text. The interpretation as a positive feedback is not completely wrong.
- The user was interested in the message but, his interest was satisfied by the presented information. Therefore, a negative feedback would be carried out.

[1] Author address: GMD FIT, German National Research Center for Information Technology, Human-Computer Interaction Research Institute, 53754 Sankt Augustin, Germany, email: ramin.yasdi@gmd.de

- The user was interrupted while reading the text. For this reason the reading seems to take a lot of time. This problem can be cleared only by an extensive observation of the user which certainly notices what the user is actually doing.
- In a network like the Internet it is possible that the reading process is delayed by a short network failure.

As a consequence of this problems implicit feedback is not very useful. Explicit feedback can be obtained as a rating value. In a graphic user interface users are asked to give their preference in various environmental conditions. User interests are acquired by using a set of sample documents, a few from each domain category. Each document is attached with a rating (default value = indifferent). The user may give the rating value to *interesting* or *not interesting.* Each documents can be given a weight, according to its importance. Strotypical knowledge is the least importance source, while explicit statement is the most important one. The final document rating are used to initialize the rating and the confidence factor of each category. A not interesting category can become indifferent if this confidence factor is (more) less a certain threshold. The user optionally can also answers some questions about himself and his answers classify him under one or more stereotypes. The stereotypes that apply to him provide only interesting domain categories and therefore increases the confidence factor for them. One of the problem with these techniques is that they take up the user's time, so busy users may unwilling to invest the effort to build user models, especially when they have not seen how adding a user model might impact their future interaction with the system. Although the user must specify which pages are interesting and which pages are not interesting, the user does not have direct, explicit, involvement in the construction of the profile.

Based on these a user model is created which describes the knowledge and interest of a user in a specific subject domain. It is used for designing appropriate actions that are tailored to him. Many intelligent user modeling systems are based on the hypothesis that the user's interest does not change over long time period, accordingly, the user model is built without considering the different movements at which data about the user have been collected. This hypothesis makes it possible to simplify the modeling process greatly, but unfortunately it seems to far removed from a realistic view. As a result, user initialized techniques are best when the user is expected to be using the system for a long time.

Because of these difficulties in acquiring user models, researcher in many field are skeptical of the worth of user modeling. For example, some practitioners in HCI (Human-Computer Interface) believe that the cost-benefit ratio for user modeling is much too high for current generations of user interfaces. Similarly, a number of researcher in ITS (Intelligent Tutoring Systems) argue that it is more cost effective to devote resources to incorporate better pedagogic techniques than it is to model users. These experts in two disparate fields avoid user modeling in their applications, largely because of difficulties in user model acquisition.

Despite the many difficulties in acquiring user model, a large variety of techniques have been developed to address this problem. We believe that the goal of creating self-improving user model is similar task to requirements in different area of AI for example self-improving Web-sites. Our challenges then is this: How can we build a user model which improves itself over time in response to user interactions with the site? This challenge poses a number of different, but not impossible questions.

- What kind of generalizations can we draw from user access patterns and what kind of changes could we make?

- How we do design an adaptive user model? We might specifically design part of the model to be changeable. For example, we might present our user with a "tour guide" and adapt the model occurring to user interests. Or we may provide semantic information about the model and allow an agent to reason about the relationship between concepts.

- How do we effectively collaborate with the user to suggest and justify potential adaptation? For example, our system might accumulate observations and suggested changes and then explains its observations and justifying the change it recommends.

- How do we move beyond one-shot learning algorithms that continually improves with the experience? Over time, our adaptive model will accumulate a great deal of data about its users and should be able to use its rich history to continually evolve and improve.

1.1 Related researches

Many of currently approaches in existence require some knowledge about the user in order to operate effectively. When a user marks a page as interesting (or not interesting), or the user visit a particular page, such system typically perform some analysis of that page in order to determine what feature(s) caused the user to be interested (or disinterested) in it. Keyword extraction is a common technique for the identifying the words within the text. Each word is extracted, and a count made of the frequency of occurrence in that document (the term frequency, *tf*). This may be compared with the frequency of the same word in a more general language corpus (inverse document frequency, *idf*) in order to determine the information content of the word in that document(the *tf-idf* technique [5]). The rationale for this is that words which are rare in the the language, but more common in a document are more informative (i.e. have a larger value) than those words which are common in the language. Each document can then be represented as a vector of key words which can be stored as an exemplar for later comparison, or it become a training instance (positive or negative) for a learning algorithm to induce a set of rules (for example) describing what does and does not interest this user. Such a profile is inherently data-driven: little, if any user input is provided to guide the construction of the profile. Although the user must specify which pages are interesting and which pages are not interesting, the user does not have direct, explicit, involvement in the construction of the profile accurately.

Letizia is similar to Web Watcher in the sense that the system accompanies the user while browsing [3]. It is an agent oriented system that uses a machine Learning approach. It observes the user's behavior, and tries to infer interest automatically. It records every choice the user makes in the browser, and reads every page the user reads. It takes the act of viewing a pages as user evaluation of the page without asking explicit user evaluation of the page, although it could perhaps be improved by some sort of of evaluate feedback from the user. It applies a modified *tfidf* analysis to each page, and adds the result to a user profile, which is essentially a list of weighted keywords. Leitizia is located on a single user's machine and learns only his current interest. By doing lookahead search Leitizia can recommend nearby pages.

An alternative to *tfidf* has been use of Boolean feature vectors [4], where each feature encodes the present or absence of a particular word in the document. Some processing is required to derive the most informative words in the document (Typically along the lines of the word information content weighting), but the advantage of using boolean feature is that many existing machine learning algorithms can be readily applied to them. Indeed, Pazanni & Bilsus provide a comprehensive and informative comparison of several existing machine learning algorithms when applied to the problem of learning user interests. Their comparison includes ID3, nearest, neighbor, back propagation and Bayesian classifier algorithms, with the latter producing the best performance. The work is an important review of many algorithms within the context of learning user interest profiles.

Recommender system like MORSE [1] generally requires the user to explicitly state their rating for a particular film before they can be provided with ratings for films they have not seen. Consequently the user has to expend much effort in order to provide an accurate profile of their preferences for this domain, since without an accurate profile they will not be able to receive acceptably good recommendations. However, until subjective judgments can be made on behalf of a user by their agent (and more importantly people are comfortable with this scenario), rating-based system such as Morse will have to rely on user-specified rating. Morse operates by comparing the rating given to films by the user with the ratings given by users to the same film. Once the comparison has been made the algorithm can produce a suggested rating for new films based on the ratings of the other users. MORSE, like other social filtering agents derive a user profile which is specific to a certain topic of the area, but nonetheless accurately reflects the user's preferences. All such profiles are generally numerically-based, since correlation between users are necessary of the social filtering algorithms.

References

[1] D. Fisk. An application of social filtering to movie recommendation on the web. In H. Nwana and N. Azarmi, editors, *Software Agents and Soft Computing*. Springer-Verlag, 1997.

[2] E. Haneke. Learning based filtering of text information using simple interest profiles. In P. Kandzia and M. Klusch, editors, *Cooperative Information Agents, First International Workshop, CIA'97*. Springer-Verlag, Lecture Notes in AI 1202, 1997.

[3] H. Lieberman. Leitzia: An agent that assists web browsing. In *International Joint Conference of Artificial Inteligence*. Morgan Kaufmann, Montreal, 1995.

[4] M.J. Pazzani and D. Bilsus. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27(3), 1997.

[5] G. Salton. *Automatic Text Processing. The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.