Information Retrieval, 6, 199–223, 2003

Simulation Studies of Different Dimensions of Users' Interests and their Impact on User Modeling and Information Filtering

JAVED MOSTAFA jm@indiana.edu
Laboratory of Applied Informatics Research, Indiana University, 030A, 1320 E 10th Ave., Bloomington, IN 47405-3907, USA

SNEHASIS MUKHOPADHYAY AND MATHEW PALAKAL
Department of Computer & Information Science, Indiana University Purdue University, 723 W. Michigan St., Indianapolis, IN 46202, USA

Received June 3, 2002; Revised December 11, 2002; Accepted February 24, 2003

Abstract. Modeling users in information filtering systems is a difficult challenge due to dimensions such as nature, scope, and variability of interests. Numerous machine-learning approaches have been proposed for user modeling in filtering systems. The focus has been primarily on techniques for user model capture and representation, with relatively simple assumptions made about the type of users' interests. Although many studies claim to deal with "adaptive" techniques and thus they pay heed to the fact that different types of interests must be modeled or even changes in interests have to be captured, few studies have actually focused on the dynamic nature and the variability of user-interests and their impact on the modeling process. A simulation based information filtering environment called SIMSFITER was developed to overcome some of the barriers associated with conducting studies on user-oriented factors that can impact interests. SIMSIFTER implemented a user modeling approach known as reinforcement learning that has proven to be effective in previous filtering studies involving humans. This paper reports on several studies conducted using SIMSIFTER that examined the impact of key dimensions such as type of interests, rate of change of interests and level of user-involvement on modeling accuracy and ultimately on filtering effectiveness.

Keywords: information filtering, user modeling, reinforcement learning, user-oriented factors

1. Introduction

The goal in information filtering (IF) is to organize, sort, or prune content so that users can consume information in an effective and efficient manner. IF systems are especially useful in dynamic information environments with many users where the amount of new information production and distribution is usually high. To organize, sort, or prune content in a manner that "matches" the demands of individual users, IF systems capture and maintain representation of users' interests called user models. User models are usually maintained internally by the IF system, and they support a mapping function that associates documents with the degree of user-interest. The outcome of the mapping function is treated as a set of predictions by the IF system about the relative importance of incoming documents to users.

Users are influenced by their environment, their situations, their knowledge, and by numerous other factors. Many of these factors continuously change, thus they have varying levels of influence on users over time. How such factors impact dimensions of users' interests and indirectly user models is not well understood. The difficulty of studying the relationship between such factors and user models is due to several key barriers: (1) it is expensive to conduct long-term studies with human subjects, (2) tracking changes in users' environment, situations, knowledge, etc. over a long period of time is a highly complex task, and, finally, (3) systematically manipulating the user-oriented factors is excessively intrusive, impractical, and may even violate common norms or ethics. The challenges these barriers pose are immense, and this may explain why there are so few researchers that actually attempted to study the relationship between the user-oriented factors and their impact on user models.

A simulation environment allows a researcher to replicate some of the "real-world" conditions of the actual environment. It also permits the study of specific phenomenon in an in-depth manner while overcoming some of the barriers listed above. For example, simulation environments are regularly used in economic forecasting that allow manipulation of a wide variety of parameters without the costs associated with disrupting the flow of critical goods or intruding on human lives. To study the influence of a limited number of user-oriented factors on user models and ultimately on information filtering a simulation environment called SIMSIFTER was developed. In the rest of the sections, this paper presents: the motivation behind the selection of the user-oriented factors and the problems studied, the design of the simulation environment, related literature, the research methodology, the experimental results, and finally the major conclusions and future work.

2. Motivation and problem definitions

User modeling approaches that are highly dependent on knowledge bases, rules, or any type of pre-built knowledge structures (e.g., stereotypes (Rich 1983)) are generally too brittle to be effective across diverse user-interests and heterogeneous content. Supervised machine learning approaches are alternative means for establishing functional mapping between content and relevance assessments. But they are dependent on a priori training and large volume of training data, and they cannot be easily tuned in response to changes in the system or user environments. Therefore, supervised learning suffers from similar disadvantages associated with the lack of flexibility in adaptation. Online, learn-while-perform, approaches that capture interest information in an incremental fashion are better suited for information filtering because such approaches offer more flexibility in adapting to different and changing user-interests. In this research, the modeling approach chosen is called reinforcement learning (Mitchell 1997, Narendara and Thathachar 1989) that acquires and updates a user model for individual users based on content rating information collected directly from the user in an ongoing fashion. The reinforcement learning approach has been implemented in two different interactive filtering systems, one for filtering medical records (MedSIFTER) and the other for filtering music (TuneSIFTER), and both systems have proven to be effective in experiments conducted with humans (Quiroga and Mostafa 2001, Zhang and Mostafa 2002).

The studies conducted using MedSIFTER and TuneSIFTER emphasized mainly usability aspects. Important factors that demanded direct intervention and manipulation of users' interests, for example, changes in interest or rate of interest change, were considered excessively intrusive and not manipulated in interactive experiments. The problem area of this paper, therefore, consists of critical factors of user-interests deserving closer scrutiny in terms of their impact on filtering performance and that were not manipulated in interactive experiments. Below more details are presented on the problem area and the specific factors addressed in this paper.

In IF systems that employ reinforcement learning user model adaptability and ultimately filtering performance are directly dependent on quality of rating information collected from users. Quality of ratings is influenced by a number of factors. A key factor is degree of interest in topics. For example, how sure the user is about his/her interest in particular topics may directly impact the frequency and consistency of ratings. Hence, a problem requiring investigation is:

Problem 1. Relationship between the concreteness of user-interests and filtering performance.

Depending on previous experience and the level of exposure to topics covered by the filtering system, the scope of user-interests—the number of topics considered relevant by the user—may vary. The number of ratings provided and the number of ratings needed by the system are dependent on the number of topics considered relevant by the user. As the number of ratings collected can directly impact the user model, another problem of investigation is:

Problem 2. Scope of user-interests in terms of coverage of topics and its impact on filtering performance.

Over time, with increasing exposure to topics covered by the system or due to changes in external factors in the world, users may experience changes in interest. Such changes may take different forms. For example, a novice user may start with a relatively shallow interest in certain topics and over time may experience growing and more intensifying change in those topics. Users with mid- or expert-level experience may begin with strong interest in certain topics but may lose interest in those topics in a gradual or rapid manner. There may also be mixed cases where a user's interest in certain topics grows while interest in other topics degrades. Hence, another interesting problem for investigation is:

Problem 3. Different types of interest change and their influence on filtering performance.

In a typical modality of system use, during the first few execution cycles there is little or no interest information available to the system. Hence, there is heavy dependence on ratings collected from the user which may demand significant effort and time in the initial few sessions ("latency period"). It is possible to directly collect interest-specific information from the user in the first session based on an online form or a brief question/answer session and update the user model accordingly. This direct modality would reduce the user-effort in the initial period of use.[1] Collecting interest-specific information in such a manner would certainly require less time. This task, however, may pose some challenges to novice users. In the first session of use, the user may not be sufficiently knowledgeable about all the topics covered by the system, and, hence, may not be prepared to judge the relevance of topics. In contrast, as the rating data is always collected in relation to specific content (as covered in documents), the modality based on documents may be cognitively less burdensome to the user and may produce more accurate information related to interests. Hence, another interesting problem for investigation is:

Problem 4. Influence of modalities of collecting interest-specific information on system latency and filtering performance.

Finally, there are some system-oriented aspects of reinforcement learning as applied in IF systems that deserve closer examination. User's interests as modeled in the system are based on accumulation of rating information for a set of topics. The rating is converted to a numerical value for each topic representing the degree of interest in the topic (see next section on the learning algorithm). In our approach to filtering, incoming documents are *not pruned*, rather their core topics are identified (i.e., they are classified according to topics), and the documents are then presented to users in a sorted order based on the degree of interest associated with their topics as indicated by the user model (see figure 1).

Rating plays a critical role in capturing the user model and the frequency and type of rating are influenced by what the user sees. Hence, the order of incoming documents and their content can have an indirect and significant impact on the user model. Imagine the distribution of topics in the initial few sessions is skewed toward certain topics while other topics receive scant coverage. The user model will reflect this topic distribution—certain topics will receive rating information while others will have no interest-specific information. This may lead to a circulatory phenomenon that may produce "information blind spots"

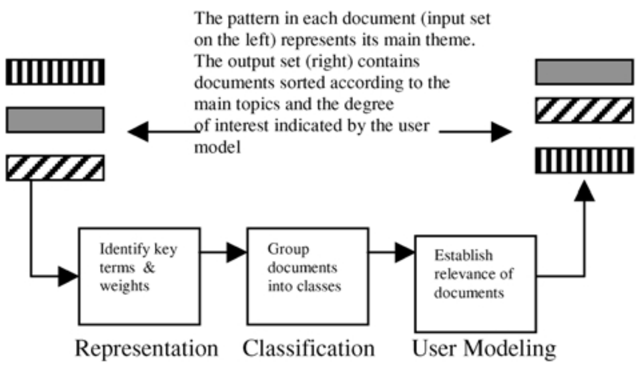

Figure 1. Major document filtering steps in SIMSIFTER.

in filtering systems. Documents on topics covered in the early iterations will continue to appear at or near the top because the model will have interest-specific information on those. While less frequently occurring documents on other topics will appear toward the bottom decreasing the likelihood that they will receive rating and, thus, ensuring that they will continue to appear in the bottom.

To avoid information blind spots resulting from skewed topical distribution in specific document streams, an IF system's rate of learning (i.e., accumulation of interest-specific information) can be modulated. One type of modulation may involve lowering the weight of individual rating and lengthening the period to reach convergence (see next section for details), which can reduce the impact a single or a few iterations of use has on learning. In other words appropriate modulation can achieve a delay in learning. This would increase the likelihood that the user would get to see documents from more varied topics and the user model thus captured would be more representative of user's actual interests. Reducing the emphasis on iteration of use (or sessions), however, may imply collection of additional ratings, over more sessions, before satisfactory performance is attained. In sum, this is a "trade-off" situation between accuracy in the user model and the amount of rating. This problem deserves further investigation and it can be described as following:

Problem 5. Impact of the number of rating and the learning rate on filtering performance.

3. Filtering simulations using SIMSIFTER

In this section we describe the architecture and the underlying algorithms of the SIMSIFTER system. SIMSIFTER models an information filtering environment that offers a continuous service. It is assumed, users of the filtering system would execute the filtering system once, and the system would continuously bring new information to the attention of the user as it becomes available. Due to the continuous service assumption, SIMSIFTER performs all its key steps in an online fashion, i.e., in real-time. It processes new documents as soon as they are retrieved, identifies the significant terms in documents, groups documents by classifying them into a set of core topical areas of interest to the user, and finally establishes for each new document group a specific relevance value (determined according to interest values in the user model). These key steps, representation, classification, and relevance assessment, are shown in figure 1.

The representation step involves conversion of documents to numeric representations. For each document a set of weights are generated (referred to as a vector) of the form: $w_1, w_2, \ldots, w_n$, whereby each weight represents the degree of discriminatory power of a specific term in the document. The terms in each vector are chosen from a thesaurus that consists of list of terms with synonyms and related terms. Any list of terms can be used, of any length, and thus SIMSIFTER offers a way to filter information on domains of any scope and depth. The popular tf.idf approach (Salton and McGill 1983) is used to establish the discriminatory power of each term or its weight value per document. The tf.idf approach takes into consideration frequency of each term in the document and its relative frequency in the collection as a whole (usually selected to represent a particular domain).

After document representation, the next major step involves identification of the core theme of each document—a step we call classification. The classifier module, responsible for this step, maintains a pre-determined set of core topics or classes.[2] Each class contains a small group of key terms (e.g., breast cancer) that are selected from the thesaurus. A class is actually represented as a vector with suitable weights chosen for each of its key terms. The class vectors and the document vectors are based on the same thesaurus and they have the same length, i.e., they are isomorphic in structure. The vectors only vary in terms of their weight values. The representation module is responsible for producing the weight values in the document vectors. SIMSIFTER allows the researcher running the simulation to provide the weight values for terms in class vectors, thus permitting additional degree of control to fix the topical coverage of classes. To identify a particular class for a specific document, i.e., establish its main theme, the cosine similarity measure (Salton and McGill 1983) is used to compute the similarity between the document vector and the class vectors. The class that produces the highest similarity value is chosen as the primary class and the document is assigned to it.

Both the representation and classification steps can be fully automated in SIMSIFTER. That is terms for the thesaurus can be automatically extracted, and the class vectors and term weights in the class vectors can be automatically determined. The automated approach eliminates the possibility of the researcher introducing conceptual bias. However, automatically extracting terms representative of a domain and automatically identifying topical categories so that they represent key areas in a domain are computationally intensive processes. If terms and classes are selected from independent and authoritative sources, such as thesauri or classification schemes produced by government agencies (e.g., NIH), database vendors (e.g., Lexis/Nexis), or established professional organizations (e.g., ACM) the researcher's bias can be minimized. In this study we used a subset of topics used by the creators of the HealthSourcesPlus database. In two previous studies we compared the performance achieved through fully automated representation and classification with manually produced thesaurus and classes. We found similar filtering performance or better can be achieved using manual means for producing the thesaurus and classes. In the past studies we examined different automated techniques (supervised and unsupervised), parameters to control scope and granularity of concepts, and the impact of the representation and classification modules on the profile capture module in an in-depth manner (Mostafa et al. 1999, Mostafa and Lam 2000). As the focus of this paper is on user-oriented factors and user modeling, we refer the reader to the previous studies for further information on issues concerning representation and classification in filtering.

The user modeling module is responsible for establishing the relevance of incoming documents. The relevance of a document is determined according to the class in which it has been classified, and, not directly based on the document. The model is a probabilistic one, and at its core is a set of values, $u_1, u_2, \ldots, u_n$, each representing the probability that the user would find the corresponding class relevant (assuming there are n classes tracked by the system).

To avoid information blind spots, a second level assessment is conducted by the user modeling module to establish and fix the top class. The top class determination is also handled probabilistically. For this, a separate vector t is maintained, $t_1, t_2, \ldots, t_n$, where

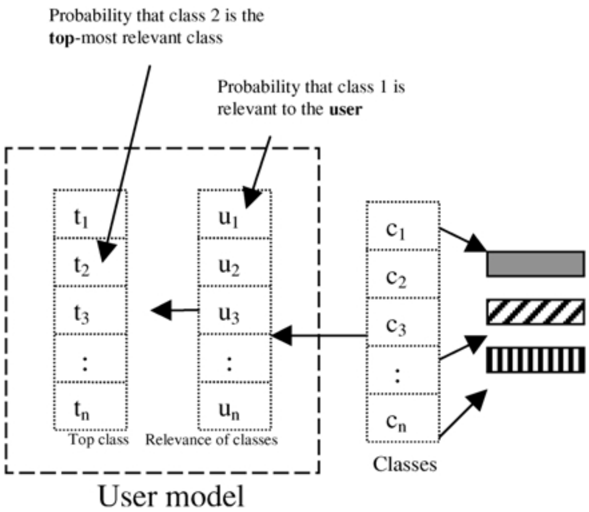

Figure 2. User model and its relation to classes.

t_i represents the probability that the corresponding class i is the most relevant class. The figure 2 shows the relationship among the t vector, the u vector, and the classes.

In the first session of use, all the elements in the u vector are initialized to be zero. In each session, the relevance ratings collected over documents presented to the user are used to update the u vector (see update algorithm below). After a few sessions of use, based on a relatively small number of relevance ratings, the values in the u vector are actually sufficient to sort all the documents for presentation purposes. However, to avoid presenting documents exclusively based on ratings collected from the user in the initial few sessions (as the sequence of documents may bias the information collected in u) the t vector is also used.

The t vector introduces a "delay" or latency in the selection of the top class, thus, it increases the likelihood that documents from a wider group of classes would be seen by the user. To appreciate how the delay is introduced, the algorithm for updating the u and t vectors must be understood. Below, we describe the procedure.

3.1. Algorithm for updating the user model

(1) In the first session, initialize all elements in the u vector to zero. Initialize all elements of the t vector to be equal to each other, i.e., each t_i $(i = 1, \ldots, n)$ is initialized to be $1/n$.

(2) Display documents and collect ratings on documents. Each rating is assumed to be a binary value (0, 1) given to each document. Assuming a rating value r is collected for a document that belongs to the class c, the corresponding value u_c is updated as the running average of rating values collected for that class. As following:

$$\frac{\sum_{i=1}^{n-1} u_{ci} + r}{n}$$

where n is the total number of sessions.

(3) Assuming the current class with the maximum value in the u vector is the mth class in that vector (i.e., the mth vector element), then the elements in the t vector, which is of the same length as the u vector, are updated as following:

$$\begin{aligned} t_i &= t_i - \eta(t_i), \quad \text{if } i \neq m \\ &= t_i + \eta(1 - t_i), \quad \text{otherwise} \end{aligned}$$

where η is a suitably chosen learning rate for the t vector to converge to the top-most relevant class (i.e., this is the latency or delay factor).

(4) Sort new documents according to the t and the u vectors and go to step 2.

To sort documents, the class with the highest value in the t vector is selected as the top-most class and any documents belonging to that class is given the ranking of 1 (highest). The order for the rest of the classes is determined based on their corresponding values in the u vector, and ranking of documents belonging to these classes are assigned according to the relative order in the u vector.

In addition to the updating and sorting of documents, in SIMSIFTER there are two other key components responsible for simulating document retrieval and document presentation. For simulating document retrieval, a database of documents is created covering topics of interest to the researcher. The database may contain either full-text or bibliographic records (title, abstract, keywords, etc.) in ASCII text format, and the records are usually maintained sorted according to date of publication. The retrieval component's behavior is controlled based on two parameters: retrieval set size and the number of sessions. These two parameters determine the number of documents that are retrieved in each session and the number of times they are retrieved. At the start of simulation, the retrieval component reads in the specified number of documents from the database starting with the first record, and in subsequent sessions it reads additional records starting with the last record read $+1$ record number. This way, in every session only "new" documents are processed by SIMSIFTER.

Typically our judgment about a document depends on its content and our interest at a particular time. In most situations if there is an overlap between our interest and the content then the document is likely to be judged highly. But there may be occasions when a document is not viewed favorably despite a strong overlap with our interest. We may ignore a document (or information described in a document) because we may feel overexposed to certain information or we may feel the need for a change. In an inverse situation due to the influence of certain recent events we may judge a document to have high worth despite the fact that its content is usually of low interest to us. In other words, the judgment we make about a document is a probabilistic event dependent on the document-content and our interest at a specific time period. Hence, it is simulated as such in SIMSIFTER.

The presentation component's behavior is controlled by two parameters: the number of documents viewed and user's interest. Beyond fixing the number of new documents retrieved, say n, the researcher can choose a value $v \leq n$ to simulate the number of documents the user actually "views" in each session. To simulate a "user" with a set of interests, before running any session, the researcher enters into SIMSIFTER a set of interest values (between 0–1) for the classes that are potentially of interest to the simulated user. These interest values are used to determine user's particular interest in relation to a document. These interest values are in turn used by SIMSIFTER to probabilistically determine relevance (i.e., judgment) for the documents and ultimately acquire the user model. Hence, the interest values entered by the researcher represent the *potential interests* and the user model represents the *system-acquired interests*.

Two steps are followed to simulate user's interest and ultimately relevance assessment of a document. In the first step the document's class membership is determined which is accomplished using the classifier. In the second step the user's interest in the document is derived in relation to the class it belongs to. Recall that the researcher has entered a potential interest value for each class (a value between 0–1) and that actual interest for the particular document must be probabilistic. To simulate probability based decision-making a pseudo-random number (PSN) between 0–1 is generated and compared to the interest value that the researcher has originally entered. If we assume that the computer has a good PSN generator then the production of PSNs would generally cover the range of all the numbers between 0–1 with approximately equal probability. Hence, if the interest value were to be close to 1, say 0.99, it would be more likely that the PSN would be less than the interest value. In contrast, if the interest value were to be very low and close to 0, say 0.0001, it would be more likely that the PSN would be higher than the interest value. If the interest value were to be higher than the PSN than the document would be selected as relevant (i.e., of possible interest), otherwise it would be marked as not relevant (i.e., of no interest). This procedure essentially ensures that the interest values originally entered by the researcher, that is the potential interests, "drive" the assessment of relevance but not in a deterministic or absolute manner. It would be possible to conduct this more directly by examining the interest value and using a threshold, say greater than 0.5, to establish interest for each new document. But we know from our experience that relevance assessment is not a deterministic event.

User's interest is only determined in relation to the top v documents presented. In each session for each document appearing in the top v set, after user's simulated interest is established, the relevance assessment is generated as a feedback rating for each document. If it is determined that the user has prospective interest then the rating value of 1 is assigned to the document. Otherwise, a rating of 0 is assigned to the document. It should be noted that the interest values entered by the researcher as the initial set of interests is typically maintained as unchanging during execution of SIMSIFTER sessions. To simulate changing interest in relation to topics, however, SIMSIFTER also allows the researcher to modify the interest values entered in the original set in an ongoing and systematic way.

A final level of control offered by SIMSIFTER is the learning rate as discussed in the update algorithm presented earlier. The value of η can be set by the researcher (between 0–1) to control the rate at which the top element in the t vector reaches 1 and the rest of the elements reach 0 (i.e., the rate at which the t vector approaches a unit vector). A low value, close to 0, of η changes the probability values in the t vector at a slow rate, thus it increases the degree of "exploration" the system is permitted in selecting a class to be the top class. This in turn allows a wider variety of classes to be selected as the top class and a more accurate model to be captured.

4. Related literature

Various approaches have been applied for user modeling in information filtering systems. In recent research machine learning techniques are favored over other techniques, and these can be grouped into several classical machine learning approaches, including rule induction, statistical classification, neural networks, genetic algorithms, and reinforcement learning. Here we review a few studies that applied different techniques.

Rule-based techniques are commonly used for email and news filtering. Marx and Schmandt (1996) described a system that is capable of automatically prioritizing voice and text messages managed by a computer-based system. The system called CLUES generates a rule-based user model from sources such as calendar appointments, logs of outgoing telephone, email messages, and personal rolodex. In typical use of the system CLUES achieved a success rate of 88%. However, a significant drawback pointed out by the authors is the fact that the user model cannot be adapted based on user's estimation of the relative performance of the system. That is the system cannot take advantage of feedback from the user to adjust its rule-base.

Assuming a sufficient amount of training data can be acquired, neural network based approaches can be used for capturing and modeling user-interests. To reduce the burden of acquiring labeled training examples, Goecks and Shavlik (2000) proposed capturing typical user actions such as clicking on a hyperlink and using such actions as "surrogate" indicators of interest. They demonstrated that by using a relatively small training set of 200 web pages, and a small number of related actions it is possible to train a three-layered neural network through back propagation to predict several surrogate interest indicators. The system predicted hyperlink clicks, scrolling, and mouse activity with fairly high accuracy. However, it was assumed that the training would be conducted offline during a period that the user doesn't use the system.

Menczer and Belew (1998) described a more complex scheme for modeling user's interest in the form of "agents" that can explore relevant hyperlinks on web pages to autonomously and iteratively bring to the user's attention relevant documents. In their system, named ARCHNID, the agents consist of genotypes of relevant query words collected from the user and represented with suitable weights as a feed-forward neural network. The agents start off with a given set of relevant URLs, for example, a book-mark file. The next page is retrieved based on the best link prediction, then the individual agents' "fitness" is assessed using an energy function whereby an agent is rewarded if the new page is established to be relevant and it is correspondingly penalized for using up network resources (if it is a remote access). Each agent's prediction function (i.e., the neural net) is also updated accordingly by comparing the estimate of the best link with the relevance assessment made for the new page retrieved using a connectionist version of Q-learning. By mutating the genotype and other key parameters based on a genetic algorithm each agent is replaced with its offspring during every iteration to introduce exploratory behavior in agents. Preliminary results reported using search length, a criterion combining recall and precision, showed that the system with the adaptive means could significantly accelerate discovery of relevant documents and shorten search length by a factor of 10.

4.1. Reinforcement learning

According to Mitchell (1997, p. 367) reinforcement learning addresses a particular branch of machine learning whereby an "autonomous agent" attempts to reach a state where it generates optimal responses by learning to predict those actions based on feedback it receives from the "trainer". Reinforcement learning (RL) has been applied in such diverse domains as control of mobile robot, optimizing factory operations, and playing board games. It has been studied extensively in both the Artificial Intelligence and Mathematical Psychology communities (Narendra and Thathachar 1989) and a wide variety of techniques have been proposed to implement it.

In the simplest form, RL is modeled as a learning agent that performs a finite set of actions operating in a feedback configuration with its environment (trainer). The agent chooses its action based on a probability distribution over the set of actions, which is equal for all the actions in the initial execution cycle of the system. Each action is reciprocated with a reinforcement signal from the environment that can be a binary value (e.g., reward or punishment) or may be a discrete or continuous value in a range. The environment is modeled as a conditional probability distribution over the reinforcement signals given an action chosen by the agent. During the initial cycles of use the environment is usually assumed to be unknown to the agent, and the only information it receives about the environment is through the reinforcements (i.e., samples from the environmental reinforcement probability distributions). The goal of the learning agent in this model is to update its action probabilities (based on the reinforcement signals) in a manner that improves its potential of receiving rewards.

The above simple formulation of RL, similar to the type known as Q learning, fits very well the problem domain of an intelligent information filtering agent involved in selection of the most relevant documents in an environment where it must learn interest information in a "learn-while-perform" fashion (i.e., no prior training data is available). The basic operation is as follows: the agent is responsible for sorting incoming documents based on the interest information of a particular user, it receives rewards or punishment accordingly (i.e., user feedback), and it adjusts the interest information so that it can receive consistently high and positive feedback from the user. We chose this simplified RL approach for the user modeling module in this study. In our implementation the t vector corresponds to the action probabilities and the u vector corresponds to the reinforcement probabilities. User's interest is primarily reflected in the reinforcement probabilities that represent estimates of the expected relevance or rating over the set of classes of documents. The ratings generated upon presentation of documents are used to update the u vector, which in turn is used to update the t vector (for details on this procedure see Section 3).

Our group is not alone in applying RL in the domain of information filtering. In fact, the study described earlier by Menczer and Belew (1998) had applied an RL algorithm to generate relevance assessments for web pages (using a variant of the Q learning approach). Mitchell and his colleagues developed a web information filtering system called WebWatcher where they applied RL for user modeling (Joachaims et al. 1997). This system maintains a user model containing key-terms or phrases and uses it to rank links according to their potential of leading to desired information. An RL algorithm similar to the

Q learning approach was used to approximate a $Q_{\text{term}}(s, a)$ function, for each term in the model, whereby s is the current page and a is a link on that page. Then the summation of all the Q values for every term in the model, for a given link, was used to arrive at a weight for the link. As a reward or reinforcement value for $Q_{\text{term}}(s, a)$ function, an appropriately discounted tf.idf (Salton and McGill 1983) value of the corresponding term in the pages was used. This RL approach for ranking links was compared with three other approaches: Random ranking, MATCH (similarity distance calculated between terms in the user model and link text), ANNOTATE (link text annotated with terms from user models of all users who followed the link and then applying MATCH), and POPULARITY (links suggested based on frequency of use). The RL approach outperformed all the others and achieved an accuracy rate of 44.6%. When compared to human suggested links (based on observation of terms in the profiles and knowledge of the web pages used in the experiments) the difference in accuracy was found to be about 3% less for the RL approach.

In the web agents for information retrieval (WAIR) system individual user's interest is modeled using an RL approach that uses implicit feedback instead of explicit rating (Seo and Zhang 2000), motivated by reasons that were also discussed by Goecks and Shavlik (2000). In WAIR, as in WebWatcher, the user model contains a set of terms, and weight values of terms in the model are adjusted in order to maximize future reinforcements or rewards. As indicators of implicit feedback the system uses reading time (or document open time), book-marking, and following up of links on presented documents. A specific weight value is associated with each indicator and the weight values are approximated based on offline training sessions captured from actual usage data. The authors compared the performance of WAIR with two traditional relevance-feedback methods for document retrieval, and found that over time, with increasing number of documents acquired, WAIR outperforms the traditional methods in terms of average values of explicit relevance feedback obtained (an indicator of user's estimate of the relevance of documents retrieved).

To conclude, the survey of literature showed that a wide variety of machine learning techniques have been proposed and evaluated for modeling user-interests, and among them reinforcement learning takes a prominent place. It's popularity lies in the fact that it is a flexible technique, it makes relatively low computational demands, and it offers the advantage of online incremental learning. However, little evidence was found on how user-centric factors such as characteristics of users' interests, rate and nature of interest change, and different means of interest information acquisition may impact document filtering conducted using reinforcement learning. In this paper, we attempt to further clarify the relationship of several key user-centric dimensions to user modeling based on reinforcement learning in the context of document filtering.

5. Methodology and research results

A set of approximately 1400 document titles with abstracts dealing with general health issues were downloaded from the HealthSourcesPlus database. The areas covered included anxiety, allergy, heart, cholesterol, depression, diet, environment, exercise, eye, headache, lung, medicine, teeth, men-health, and women-health. Among downloaded documents each class was represented on average by 90 documents and the total set was randomized. The randomized document set was used to create the test database. Documents collected from the HealthSourcesPlus database were pre-labeled—the appropriate terms representing the topical category or class were included as an additional field along with title and abstract in each document. These terms were used to manually produce the thesaurus and the classes. The classifier module used the embedded labels along with other terms appearing in each document for determining the most appropriate class for individual documents. Possible errors associated with choice of terms, representation, and classification and their impact on filtering were not studied, as they and other related issues were part of two previous in-depth studies described in Mostafa et al. (1999) and Mostafa and Lam (2000).

Six key parameters of the SIMSIFTER system, namely, the number of newly arriving or retrieved documents (document stream size), the number of documents actually viewed, interest values, interest rate change, learning rate, and the number of sessions, were manipulated as independent variables to conduct several simulation experiments and study the problems specified in the motivation and problem sections earlier.

The dependent variable was calculated based on the position or ranking of *relevant* documents in the list of sorted documents produced by SIMSIFTER. Before providing details on how the dependent variable was actually calculated, we must explain the difference between sorting the retrieved documents in each session and determining the relevance of individual documents. As described in Section 3, SIMSIFTER orders all retrieved documents by using the system-acquired interest information represented internally in the t and u vectors. The ranking is based on class membership of documents. The class with the highest value in the t vector is selected as the class in which the user has the highest interest and therefore is assigned the highest ranking. The ranking of the rest of the classes are established according to their corresponding interest values in the u vector.

While all retrieved documents are sorted by using the user model, the determination of relevance of documents is dependent on the interest values originally entered by the researcher, i.e., the potential interests in classes. This relevance assessment step is conducted as part of the same process applied in producing ratings. As described in Section 3, the process involves probabilistically establishing relevance—the interest value corresponding to the document's class is compared to a pseudo random number. If the interest value is higher then the document is considered relevant (rating of 1), otherwise it is not considered relevant (rating of 0).

For the purpose of evaluation only the relevance of documents in the top 10 were assessed and their rankings noted. The overall ranking of all the relevant documents in top 10 thus identified was converted to a normalized precision score using the formula below which was treated as the dependent variable in each session. Salton (1968) had proposed the normalized precision formula as a composite measure to calculate retrieval effectiveness based exclusively on ranking which does not distinguish between retrieved versus non-retrieved documents. Normalized precision is computed as a difference between the area under the precision curve of the actual performance and the precision curve for an ideal system where the relevant documents appear sequentially at the head of the list. In the formula below, N is the total number of documents in the stream, REL is the total number of relevant documents identified, and $Rank_i$ is the ranking of the relevant document i in the output. In our case the rankings of relevant documents are those in $Rank_i$ (actual ranking) and they are compared with 1, 2, ..., *REL* (ranking that would be produced in an ideal system).

$$\text{Precision}_{\text{norm}} = 1 \frac{\sum_{i=1}^{REL} \log(Rank_i) - \sum_{i=1}^{REL} \log(i)}{\log(N!/(N - REL)!REL)}$$

5.1. Problem 1

The goal of studying this problem was to establish how specificity level or concreteness of user's interest could impact the acquisition of a user model and ultimately the filtering performance. Five topical areas, namely, anxiety (topic1), cholesterol (topic2), diet (topic3), teeth (topic4), and men-health (topic5), were chosen as the scope of interest for an imaginary user. By manipulating different levels of interest values, i.e., probability values for interest in each topical area, three different interest profiles were created. The following values were chosen for the five topical areas.

- For the concrete profile: all values were set to 1.0.
- For the middle profile: all values were set to 0.6.
- For the mild-to-low profile: interest values in the five topics ranged between 0.2–0.4.

Each of the three profiles was entered into the system as a particular type of simulated user, and a fixed number of sessions were executed per user. In addition, a "no-learning" level simulation was also conducted with the middle level user profile and the no-learning parameter turned on. Internally, this meant that the user model was not updated and documents were presented in the same sequence they were accessed.

The retrieval parameter was set at 30 and duration of use was set at 45 sessions. The view parameter was set at 10. The values of these parameters ensured that the upper limit of document stream size of newly arriving or accessed documents would be 30 per session, documents that would be viewed or scanned by the user would fall in top 10, and length of use would be 45 sessions.

It was found that the simulated user with the concrete interest profile produced the best overall performance. The average normalized precision achieved at the concrete level was 0.89, at the middle level was 0.72, at the mild-low level was 0.52, and finally at the no-learning level was 0.4. As can be seen in figure 3 (result was smoothed by averaging consecutive sessions), filtering performance for the concrete user started off higher than the other users, went through a rapid rise until Session 7, beyond which the performance remained approximately at the same level for the rest of the sessions. Whereas the mild-low case and the no-learning case showed much more unpredictable behavior and the performance remained low compared to the other levels.

A Person Product Moment correlation showed that the normalized precision at the concrete level had a relatively high correlation with the middle level (0.6), the correlation of the concrete level with the mild-low level was lower (0.4) and the concrete level had much lower correlation with the no-learning level (0.05). These correlation values provide a longitudinal picture of the performance—clarifying the behavior of the system *over a period of time*. Two observations can be made: (1) concreteness of interest does impact

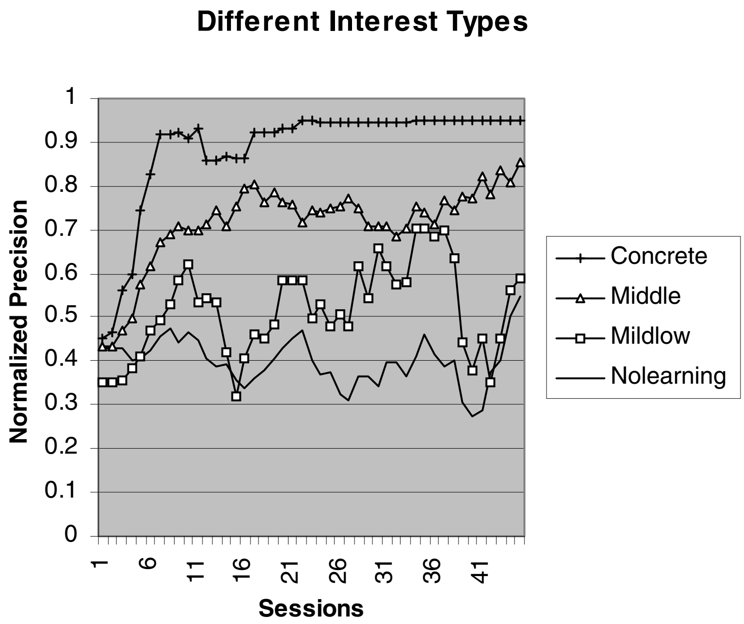

Figure 3. Impact of concreteness of user-interests on filtering.

filtering performance differently over time and (2) with increasing concreteness of interest filtering performance is likely to be higher and more consistent (i.e., likely to match that of the most concrete interest profile), while with decreasing concreteness of interest filtering performance may degrade.

We counted the total number of relevant documents identified that appeared in top 10 over all sessions. We also calculated the total number of documents that would appear in top 10 if sorting were based on selection of all documents classified into classes in which the simulated user had at least *nonzero* level (>0) of interest. The former number is the *actual* number of documents identified as relevant by SIMSIFTER, while the latter number represents the total number of *potentially relevant documents*. We will report the total number of potentially relevant documents as the *baseline* number along with the actual number wherever appropriate.

In the concrete case a total of 365 documents were identified as relevant, in the middle level 216 documents were relevant, in the mild-low case 83 documents were flagged as relevant, and finally in the no-learning case 97 documents were determined as relevant (baseline in all cases = 365). The range in this data was remarkable—showing approximately 4:1 ratio between the best and the lowest performing level. Another interesting result was that the no-learning case produced better performance in terms of the number of relevant documents identified compared to the mild-low case. This result shows that uncertainty about interests may have a cost—in the form of producing incorrect or faulty user models. Use of a faulty user model to sort incoming documents may end up causing more harm than good. The results in the mild-low case as compared to the no-learning case shows that if the user

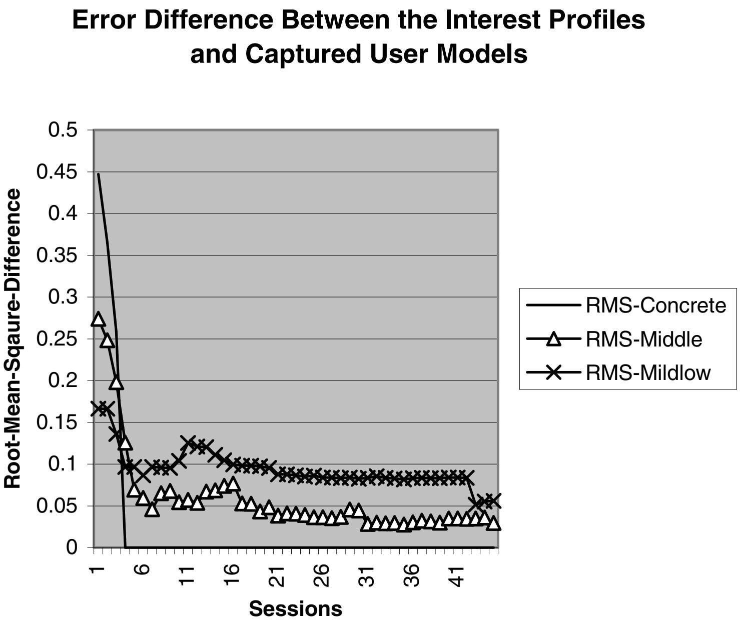

Figure 4. Comparison of performance on user model acquisition for interest profiles with different degrees of concreteness.

model in mild-low level was not used to sort documents and documents were presented in the order they arrived the user would see more relevant documents.

To assess the impact on the user model captured more directly, the root-mean-square difference was computed per session between the entered profile and the u vector (internal representation of interest) across all sessions. It was found that within the first 4–5 sessions the user models were nearly completely captured for all three types of profiles (figure 4). The concreteness of interest impacted the magnitude of error difference in the early 4–5 sessions, but did not strongly influence the number of sessions needed to reach minimal difference. The data revealed that uncertainty in interest, as represented in the middle and mild-low profiles, can lead to errors in user model acquisition in the 5–12% range beyond the first 4–5 sessions. Even though the user models acquired after the first 4–5 sessions were nearly complete (with little inaccuracies) in these two levels, the filtering performances achieved were not as good as the performance achieved in the concrete level. It appears that in cases where user-interests are inherently low, mild, or reflect uncertainty it is likely that performances would be low and inconsistent, and that improving the quality of the user models captured may have little or no impact.

5.2. Problem 2

This problem was concerned with the scope of user's interest and studying its impact on filtering performance. The number of topical areas associated with interest values was

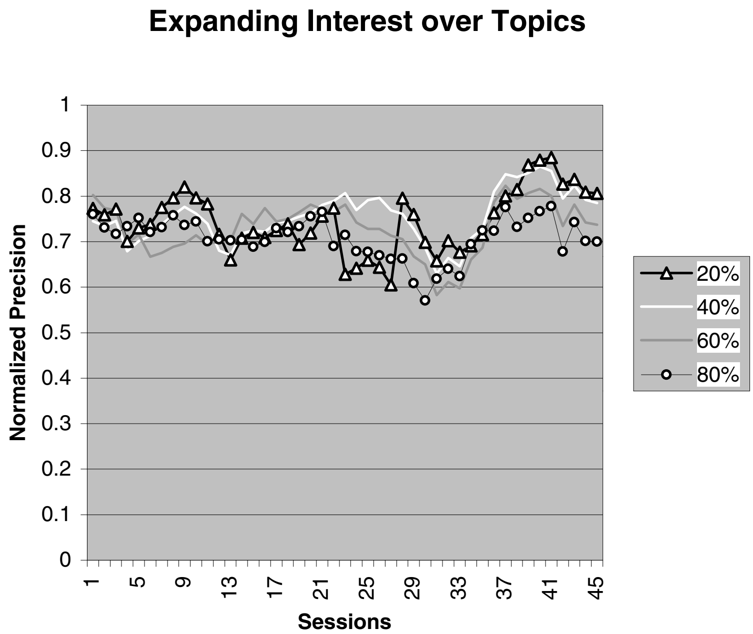

Figure 5. Impact of expanding interest scope on filtering performance.

manipulated to produce four interest profiles, named 20pct, 40pct, 60pct, and 80pct. In each profile the exact proportion of topical areas, 3, 6, 9, and 12 respectively, were set to 0.8 to represent high level of interest in the corresponding areas. In addition, the retrieval parameter was set at 30 and the view parameter was set at 10. The profiles were entered into SIMSIFTER one at a time and 45 sessions were executed for each profile.

It was found that expanding interest scope had relatively small impact on performance. The average normalized precision for the 20pct level was 0.75, for the 40pct level was 0.75, for the 60pct level 0.73, and for the 80pct level was 0.71. As can be seen in figure 5, the performance of all the levels "tracked" each other quite well, demonstrating strong correlation. The number of documents identified as relevant in the top 10 across all levels were approximately similar when compared to baseline results (note: the baseline for each level was different as the number of topics at each level was different). In the 80pct level 346 documents were identified as relevant. In this level the baseline was 440, hence, 78% of potentially relevant documents were identified. In the 60pct level 324 documents were found as relevant. Here, the baseline was 423. Hence, 76% of documents were detected as relevant. In the 40pct level 305 documents were established as relevant. The baseline was 393 at this level, which meant 77% of the relevant documents were identified. Finally, in the 20pct level 210 documents were flagged as relevant. At this level the baseline was 260, hence, 80% of the relevant documents were detected. This performance appeared to indicate that increasing the breadth or scope of interest does not translate to increase in proportion of relevant documents identified by the system (assuming view window size does not change to accommodate changing scope of interest).

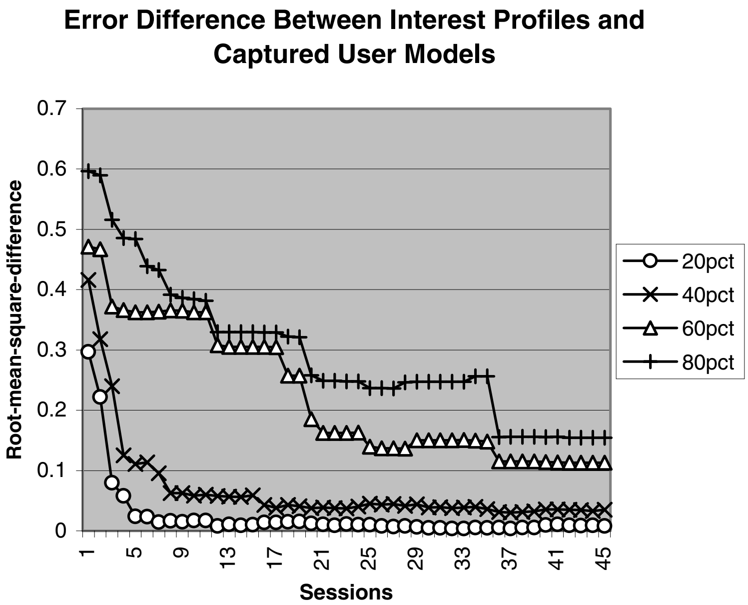

Figure 6. Comparison of performance on user model acquisition for interest profiles with different interest scopes.

When root-mean-square analysis was conducted it was discovered that with expanding scope of interest the error increased in user model acquisition. The 80pct level showed the highest error range between 60–15%, whereas the 20pct level showed the lowest error range between 30–1%. When considered in the context of normalized precision performance, the error associated with the profiles of the expanded scope, i.e., 40pct, 60pct, and 80pct, seemed to have minor impact as only a slight decrease in average normalized precision was observed (see figure 6).

It is possible that certain relevant documents were "not seen" or "missed" due to a relatively small and fixed window used for identifying relevant documents (top 10). This phenomenon is likely to have occurred at the two broadest scope levels, 60pct and 80pct, where a relatively large proportion of incoming documents could belong to classes of interest to the user. This may explain the larger error in the user model captured at the 60pct and 80pct levels.

5.3. Problem 3

This problem has to do with how interest may develop or degrade over time. Instead of assuming a user possesses static interest—interest that does not change over time—the interest profile is manipulated during system use to induce change over time and the impact on user modeling and system performance is recorded. Different types of changes are possible. Changes may occur in an incremental fashion; that is change starting with the initial session and continuing throughout duration of usage. Or change may occur in an abrupt manner—suddenly affecting a few interest areas and then the influence subsiding after a period.

The following patterns of changes were examined, divided into two levels: incremental and abrupt.

Incremental

- Starts with low interest in the topics anxiety, cholesterol, diet, teeth, and men-health and then interest increases in an increment of 0.02 (low-to-hi)
- Starts with high interest in the above five topics and then loses interest in an increment of 0.02 (hi-to-low)
- Develops interest in cholesterol and diet in an increment of 0.02 and simultaneously loses interest in anxiety, teeth, and men-health in an increment of 0.02 (hybrid-change)

Abrupt

- Develops sudden and incremental interest in anxiety and cholesterol between the sessions 5–15 only (called sudden-dev)
- Develops sudden and incremental interest in anxiety and cholesterol between the sessions 5–15, and then loses interest in anxiety and cholesterol in an incremental manner in the sessions 16–45 (called sudden-dev-loss)
- Develops sudden and incremental interest in anxiety and cholesterol between sessions 5–15, loses interest in anxiety and cholesterol between the sessions 16–45, while develops new incremental interest in the sessions 20–45 in medicine, teeth, and men-health (sudden-dev-loss-dev)

For all sessions executed, the retrieval parameter was set at 30 and the view parameter was set at 10.

The results produced for the three profiles manipulated for the incremental change level showed that the hybrid-change profile generated slightly higher normalized precision performance than the low-to-hi profile: the average normalized precision of the hybrid-change profile was 0.85, while the average normalized precision for the low-to-hi profile was 0.82. It was found, as expected, that the performance of hi-to-low gradually dropped (see figure 7), producing an average normalized precision of 0.24.

When the total number of relevant documents in top 10 was considered, an interesting hidden facet of the performance was revealed. It was found that even though the hybrid-change level produced slightly better performance than the low-to-hi level in terms of normalized precision, 51 more documents were identified as relevant in the low-to-hi level compared to the hybrid-change level (baseline = 365). In total, low-to-hi level generated 213 relevant documents, whereas the hybrid-change level generated 162. In the low-to-hi level the interest profile and its evolution were relatively simple: interest in all topics started off at a low level and over time interest in topics gradually increased at an equal rate. In the hybrid-change level the initial interest pattern and the evolution of interest over time were more complex. The complexity of interest and the system's capacity to capture it were responsible for the lower number of relevant documents identified in the hybrid-change level.

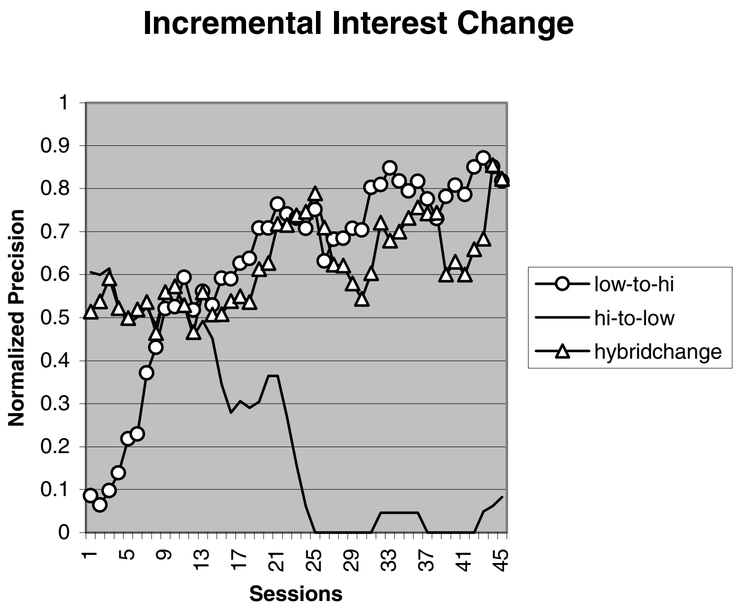

Figure 7. Impact of incremental interest change on filtering performance.

In the abrupt interest change level, it was found that the most simple positive interest change involving intensifying interest between 5–15 sessions and never degrading, named sudden-dev, produced the best overall performance (see figure 8). The average normalized precision for the sudden-dev was 0.45, for the sudden-dev-loss-dev level it was 0.31, and for the sudden-dev-loss it was 0.17. Hence, loss in interest without corresponding development of new interest in any topic had the most negative impact on ranking. The sudden-dev level generated a total of 58 relevant documents (baseline = 187; 31% of total relevant documents identified), sudden-dev-loss level identified 18 documents as relevant (baseline = 187; 9% of relevant documents identified), and finally the sudden-dev-loss-dev level produced 38 relevant documents (baseline = 358; 10% of relevant documents identified). Note that the baseline values were different because the profile content varied across levels. These results indicate that the least complex interest level, i.e., sudden-dev, outperformed the other two levels in a significant way (by about 20%).

It must be noted here that we have not examined *why* interest profiles change over time. A variety of factors may be responsible, but two factors may be especially influential. The topical interests users bring to the system and topics they actively seek out is one such factor. Interest evolution may also be influenced by the characteristics of the document stream—the content of the incoming documents, novelty in their content, and significant content changes in the documents. In previous studies we examined granularity of topics and its influence on filtering (Mostafa et al. 1999, Mostafa and Quiroga 2002), and we also examined factors associated with document stream such as volume of incoming documents (Mostafa et al. 1997). However other key factors associated with document stream (e.g.,

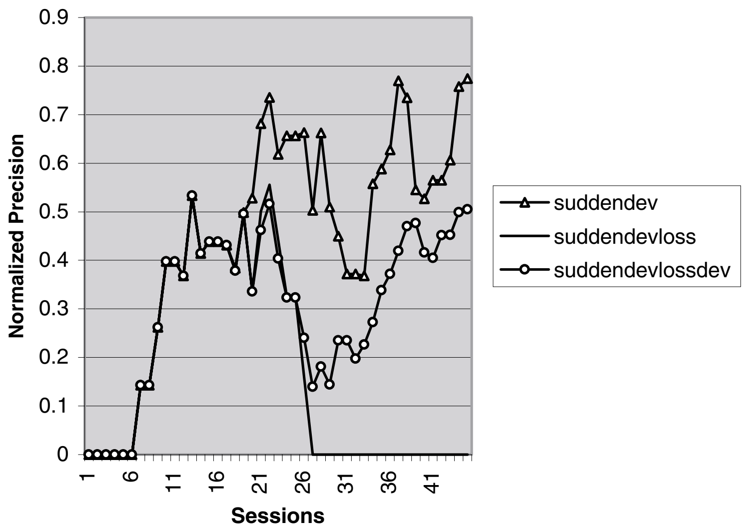

Figure 8. Impact of abrupt interest change on filtering.

novelty) and topic selection, and their impact on filtering deserve closer attention. This is outside the scope of the present study and should be considered in a future project.

5.4. Problem 4

This problem investigated the influence of different modalities of collecting rating information from the user on user modeling and ultimately filtering performance. In one modality, the user model was acquired by generating only rating information based on the entered interest profile (called rating). In another modality, the user model was manually initialized in the first session based on the interest information, and it was continuously updated using rating generated from the entered interest profile (called init-plus-rating). In both cases, the interest profile used was: anxiety, cholesterol, diet, and teeth (all values set to high, i.e., 1.0). The retrieval parameter was set at 30 and the view parameter was set at 10. The profiles were entered into SIMSIFTER and 45 sessions were executed for each modality.

It was found that the init-plus-rating level outperformed the rating level in terms of normalized precision. As can be seen in figure 9, the performance for the init-plus-rating level started off at a high level, i.e., no latency was present, and it continued approximately at the same level throughout all the sessions. Whereas, the normalized precision performance of the rating level started off at a relatively low level (below 0.6), it gradually climbed to a high level of 0.95 by session 20, and ranged between this high level and 0.72 subsequently.

In terms of total relevant documents identified in the top 10, the init-plus-rating level also demonstrated superior performance compared to the rating level. It was found that the init-plus-rating level duplicated the baseline results (330 documents) but the rating only level produced 29 less relevant documents. It was not surprising that the init-plus-rating

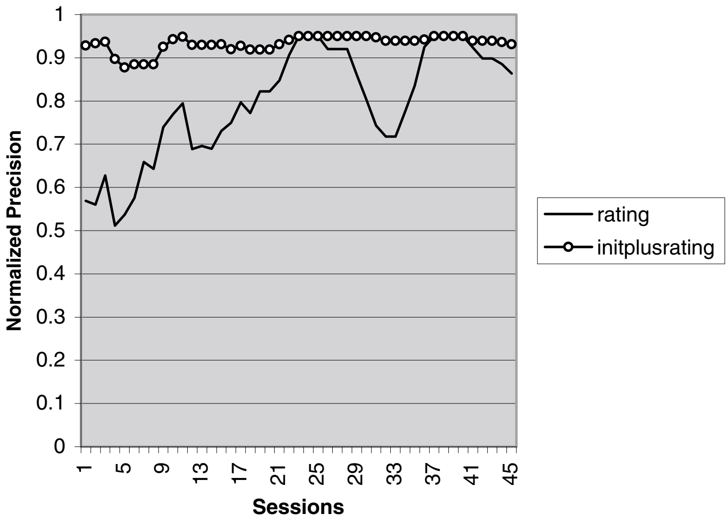

Figure 9. Impact of different modalities of collecting interest information on filtering performance.

generated better performance than the rating only level, in that the former had the advantage of being provided more interest-specific information. It is important to note, however, that in the initial session, the users may not always be fully clear about their interest and they may not have sufficient experience to assign interest information correctly to the various topics. Inaccuracies or "noise" in interest information introduced in the initial session, would certainly lead to degradation of performance in a system using the same modality as init-plus-rating.

5.5. Problem 5

This problem concentrated on the influence of amount of rating and the learning rate on filtering performance. The system is dependent on rating information to acquire the user model. The system is also influenced by its learning rate, i.e., η. Experiments were designed that examined the influence of these variables together. For six different values of η, 0.9, 0.6, 0.3, 0.09, 0.06, and 0.03, results were produced at two levels: one where amount of rating was fixed at 3 (level1), and another where the amount of rating was fixed at 7 (level 2). By combining the values of the two variables, η and amount of rating, the following 12 experimental conditions were created:

Level1

- Condition 1: $\eta = 0.9$, Amount of rating = 3
- Condition 2: $\eta = 0.6$, Amount of rating = 3
- Condition 3: $\eta = 0.3$, Amount of rating = 3
- Condition 4: $\eta = 0.09$, Amount of rating = 3

- Condition 5: $\eta = 0.06$, Amount of rating = 3
- Condition 6: $\eta = 0.03$, Amount of rating = 3

Level2

- Condition 1: $\eta = 0.9$, Amount of rating = 7
- Condition 2: $\eta = 0.6$, Amount of rating = 7
- Condition 3: $\eta = 0.3$, Amount of rating = 7
- Condition 4: $\eta = 0.09$, Amount of rating = 7
- Condition 5: $\eta = 0.06$, Amount of rating = 7
- Condition 6: $\eta = 0.03$, Amount of rating = 7

In both levels, the interest profile was of middle type: anxiety (0.7), cholesterol (0.7), diet (0.7), teeth (0.6), men-health (0.6), and women-health (0.6). The retrieval parameter was set at 30. To align the experiments with the two levels, the view parameter was set at 3 for level1 and 7 for level2. The profile was entered into SIMSIFTER and 45 sessions were executed per experimental condition for each level (i.e., 12 sets of results were generated).

Level2 produced slightly higher normalized precision results than the level1. At level2 the average normalized precision ranged between 0.59–0.75, while at level1 the average normalized precision ranged between 0.60–0.73.

The two variables influenced the actual user model captured in subtle and interesting ways (see figure 10). The root-mean-square analysis revealed that in level1, the η value had to be set to 0.09 (Condition 4) for the error difference between the captured model and the interest profile to reach the minimum. In comparison the approximately same minimum

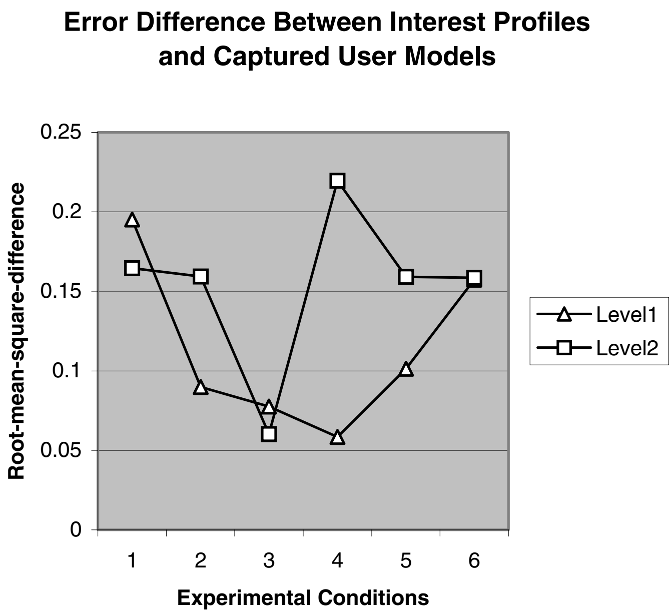

Figure 10. Impact on user model acquisition due to learning rate and amount of rating.

difference was reached in level2 with the η value of 0.3 (Condition 3). This indicated that in level1, due to small number of rating data collected in each session, the system required more "exploration" or longer convergence period to attain the best user model. In level2, more rating data was available in each session, as the "user" viewed and provided rating to 7 documents in each session. This made it possible to attain the best user model with less exploration or shorter convergence period.

It was also observed that in both level1 and level2 initially the error decreases with corresponding decrease in η, but after a period the error begins to rise again (see Condition 5 for level1, and Condition 4 for level2). It appears that the exploration period or the delay in convergence should be delimited carefully. Depending on the number of documents being viewed, too much exploration may result in the system not bringing the right documents to the user's attention (i.e., in the view range: top 3 or 7 here), or inversely bringing the wrong documents to the user's attention and thus failing to acquire an accurate user model.

6. Conclusions and future work

In this paper, several key dimensions associated with users' interests and their impacts on user modeling were studied. The primary means of the study was simulation analysis conducted using the SIMSIFTER system which employs reinforcement learning for user modeling. The major findings include the following: (1) uncertainty in interest is a dominating factor in leading to degraded performance in terms of both sorting of relevant documents and identifying them, (2) widening scope of interest, in terms of the number of topics, may impact accuracy of the user model captured, especially if the number of documents viewed does not increase in proportionate manner, (3) the user modeling approach can handle different types of interest change, both incremental and abrupt, however, the more complex the interest change pattern the larger would be the cost in filtering performance, (4) initialization or seeding the user model in the first session may eliminate latency and produce a more steady and consistent performance, and finally (5) rate of learning and amount of rating interact in an interesting way, whereby a specific combination of values of these two variables can ensure the most accurate user model, i.e., there is a minimum error point and values of the variables preceding the point or subsequent to the point may lead to loss in accuracy. In addition, a broad finding was that reinforcement learning is a highly robust technique for capturing diverse and changing user-interests.

The goal is to extend this research to examine the impact of characteristics associated with document/information streams. Factors such as size of the stream, heterogeneity of topics present, biases in terms of scope and distribution of topical coverage, and topic-drift in the stream may have direct impact on the representation and classification components, and ultimately on the user modeling effectiveness. These issues will be considered in a future study.

Acknowledgment

Several key people in the SIFTER research group contributed to the design of the SIMSIFTER system. They include W. Lam, A. Hudli, L. Xue, and T. Blekher. We are grateful for their contributions in helping us refine the design of the system.

Notes

1. It should noted that to track changing user-interests and to tune the user model in an ongoing fashion, an IF system using reinforcement learning would require occasional rating of content beyond the initial period of use.
2. The terms topic/topics and class/classes are used interchangeably.

References

Goecks J and Shavlik J (2000) Learning users' interests by unobtrusively observing their normal behavior. In: Proceedings of the 5th International Conference on Intelligent User Interfaces, New Orleans, Louisiana, January 2000, pp. 129–132.
Jochaims T, Freitag D and Mitchell TM (1997) WebWatcher: A tour guide for the World Wide Web. In: Proceedings of the 5th International Joint Conference on Artificial Intelligence, Ngoya, Aichi, Japan, August 1997, pp. 770–777.
Marx M and Schmandt C (1996) CLUES: Dynamic personalized message filtering. In: Proceedings of the Computer Supported Cooperative Work, Boston, MA, November 1996, pp. 113–121.
Menczer F and Belew RK (1998) Adaptive information agents in distributed textual environments. In: Proceedings of the Autonomous Agents, Minneapolis, MN, May 1998, pp. 157–164.
Mitchell TM (1997) Machine Learning, McGraw-Hill, Boston, MA.
Mostafa J and Lam W (2000) Automatic classification using supervised learning in a medical document filtering application. Information Processing & Management, 36(3):415–444.
Mostafa J, Mukhopadhyay S, Lam W and Palakal M (1997) A multilevel approach for intelligent information filtering: Model, system and evaluation. ACM Transactions on Information Systems, 15(4):368–399.
Mostafa J, Quiroga LM and Palakal M (1999) Filtering medical documents using automated and human classification methods. Journal of the American Society for Information Science, 49(4):1304–1318.
Narendra KS and Thathachar MAL (1989) Learning Automata—an Introduction. Prentice-Hall, Englewood Cliffs, NJ.
Quiroga LM and Mostafa J (2000) Empirical evaluation of explicit versus implicit acquisition of user profiles in information filtering systems. In: Proceedings of the American Society for Information Science & Technology Conference, Chicago, IL, November 2000, pp. 1–10.
Rich E (1983) User are individuals: Individualizing user models. International Journal of Man-Machine Studies, 18:199–214.
Salton G (1968) Automatic Information Organization and Retrieval. McGraw-Hill, New York, NY.
Salton G and McGill MJ (1983) Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY.
Seo Y-W and Zhang B-T (2000) A reinforcement learning agent for personalized information filtering. In: Proceedings of the 5th International Conference on Intelligent User Interfaces, New Orleans, LA, January 2000, pp. 348–351.
Zhang J and Mostafa J (2002) Examining the relationships among user involvement, profile quality, and information filtering performance. In: Proceedings of the American Society for Information Science & Technology Conference, Philadelphia, PA, Nov. 2002, pp. 298–306.