KI 2002/3, pp. 5-9, Special Issue on Adaptivity and User Modeling

Adaptivity through Unobstrusive Learning

Ingo Schwab, Alfred Kobsa

Abstract

In this paper, we present an approach for learning interest profiles implicitly from positive user observations only. This approach eliminates the need to prompt users for ratings, or to somewhat artificially infer negative evidences, which arises when traditional learning algorithms are used. We developed a methodology for learning explicit user profiles and recommending interesting objects. This highly dynamic process, which calculates the personalized recommendations in real-time, has been deployed in ELFI, a web-based system that provides information about research grants and is used by more than 1000 users in German research organizations who monitor and/or advise on extra-mural funding opportunities.

1. Introduction

In the last few years, many approaches and systems for recommending information, products and other kinds of objects have been developed (Kobsa, Koenemann, and Pohl 2001). Since the Internet is becoming extremely large, with vast amounts of information accessible to anyone with a computer, recommender systems are becoming increasingly important.

There are two main approaches for recommending objects to users. *Feature-based* filtering systems take individual preferences with respect to certain features of objects into account. Examples include actors and directors for movies, and resolution, price and maximum document size for printers[1]. *Clique-based* (aka "collaborative") filtering systems instead build on similarities between users with regard to the objects in which these users implicitly or explicitly express an interest.

Machine learning methods can be used to solve classification problems. A straightforward way of applying machine learning techniques to the acquisition of user interest profiles is to assume that the set of information objects can be divided into interest classes (e.g., one for "interesting" and one for "not interesting"). In many systems, users must give examples for both classes in an initial training phase, on the basis of which a classification algorithm is being inductively learned. Thereafter, this algorithm can decide whether new information objects belong to the "interesting" or to the "not interesting" class. Such explicit rating requires additional user effort and keeps users from performing their actual tasks, both of which is undesirable. As has been observed by Carroll and Rosson (1987), users are unlikely to engage in such additional efforts even when they know that they would profit in the long run. Additionally, motivating consumers to divulge personal data on the web is proving very difficult, mostly due to their lack of trust that web sites will respect their privacy. Users often withhold personal data or provide false data (Kobsa 2001). Conclusions about user interest should therefore not rely very much on user ratings, but rather take passive, unobtrusive observations about users into account as far as possible.

2. Previous Work

In the past, several systems have been developed that employ learning algorithms to identify individual users' interests with respect to information objects and their contents. They make use of such user interest profiles to generate personalized recommendations.

Lieberman (Lieberman 1995) developed the system Letizia, which assists users in web browsing by recommending links on the current web page in which the user is presumably interested. This prediction is made based on an interest profile of the user. The profile consists of a list of weighted keywords, which is obtained by aggregating the results of TFIDF analyses[2] of terms in previously visited pages and of search terms that the user had entered in search engines. The TFIDF algorithm needs positive and negative evidences for users' interest in these terms. Letizia employs heuristics to determine these evidences based on the user's browsing behavior. Viewing a page indicates interest in that page, bookmarking a page indicates even stronger interest, while

[1] In "content-based" information filtering, the content is described by a restricted number of characteristic features of the content, e.g. characteristic words.

[2] The TFIDF algorithm (for "term frequency inverse document frequency") assigns to each meaning-bearing term in a document a numeric weight that represents the representativeness of the term for the document. The weight is computed by dividing the frequency of the term in the document by the overall frequency of the term.

"passing over" links (i.e., selecting a link below and/or on the right of other links) is viewed as a disinterest in these links.

In Syskill&Webert (Pazzani and Billsus 1997), the user rates a number of pre-selected web documents from some content domain on a binary "hot" and "cold" scale. In this way, positive and negative learning examples become available to the system. Based on these ratings, it computes the probabilities of words being in “hot” or “cold” documents. A set of word probability triplets is formed for each user, which can be regarded as an interest profile that characterizes the average hot and cold documents of this user. Based on this profile, the Naive Bayes Classifier method is used to classify further documents as hot or cold, respectively.

The system Personalized WebWatcher (Mladenic 1996) also uses the Naive Bayes Classifier. This system watches individual users’ choices of links on web pages, in order to recommend links on other web pages that are visited later. The user is not required to provide explicit ratings. Instead, visited links are taken as positive examples and non-visited links as negative ones.

The Naive Bayes Classifier is again used in the system NewsDude (Billsus and Pazzani 1999), to recommend news articles similar to Syskill&Webert. In NewsDude, the inferred probabilities are taken to characterize a user's long-term interests. To avoid recommending too many similar documents, an additional short-term profile is built by memorizing recently read articles. New articles are then compared to the memorized ones; if they are too similar, they are not recommended even when they match the long-term interest profile. This procedure corresponds to the nearest-neighbor classification algorithm, which is well known in Machine Learning. Note that positive examples are only needed for the short-term profile (albeit to produce "negative" recommendations).

3. Learning User Profiles

In general, it can be assumed that dividing the objects in an information subspace into an “interesting” and a “non interesting” class is legitimate. People are normally interested or disinterested in given topics. However, obtaining an appropriate set of negative examples (i.e., examples of the “not interesting” class) is difficult. The central source of information about the user is his or her web navigation behavior, and especially the set of selected objects. Selections are made from the set of currently available information objects. As has been shown in Section 2, several systems use unselected objects as negative examples. However, we claim that users may generally still be interested in objects even when they did not select them. Sometimes pages are not visited at the moment but will so at a later point, and sometimes they are ignored forever even when the user is interested in them since it is too time consuming or simply not possible to follow every interesting link. Classifying the objects not visited as negative examples thus seems to be a dangerous assumption. It may happen that the machine learning algorithm learns wrong decision borders between the classification regions.

It therefore seems more effective to build a user profile based on positive examples of the “interesting” class only. Standard classification methods are then not applicable anymore. For learning interest profiles we thus had to invent new learning methods or revise existing ones.

.

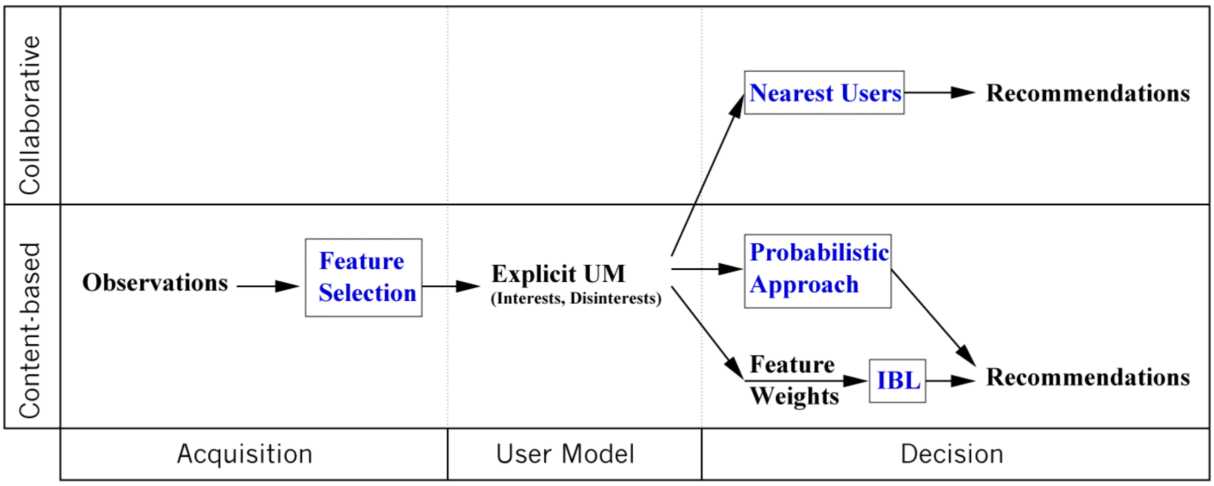

Figure 1. Learning mechanisms.

Figure 1 shows the structural design of our recommendation component, gives an overview of the learning methods used, and describes how they collaborate to recommend new or interesting objects to the user. There are three main stages in the recommendation process. In the *acquisition* stage, observations about a user are processed with a statistical significance analysis to extract those features of the objects that represent the user's interests viz. disinterests. In the second stage, the selected features are explicitly represented in the user model (UM). In a third stage, learning algorithms utilize the model for recommending new relevant objects to the user. This *decision* stage is split into two main parts. For content-based recommendations, we use a probabilistic and an instance-based learning (IBL) approach (namely k-Nearest Neighbor or Case Based Reasoning) (Mitchell 1997) to learn a general characterization of objects that are relevant to the user. Additionally, a collaborative approach compares the content-based user models and recommends objects that similar users frequently selected in the past.

We modified all approaches so that they can handle a single class only, by employing the notion of similarity or distance between objects. The main idea for the content-based recommendation algorithms is that the user is interested in objects which are similar to the ones she already selected. The selections represent some kind of "interest center". Then the similarity between the selections and the database of unseen objects can be

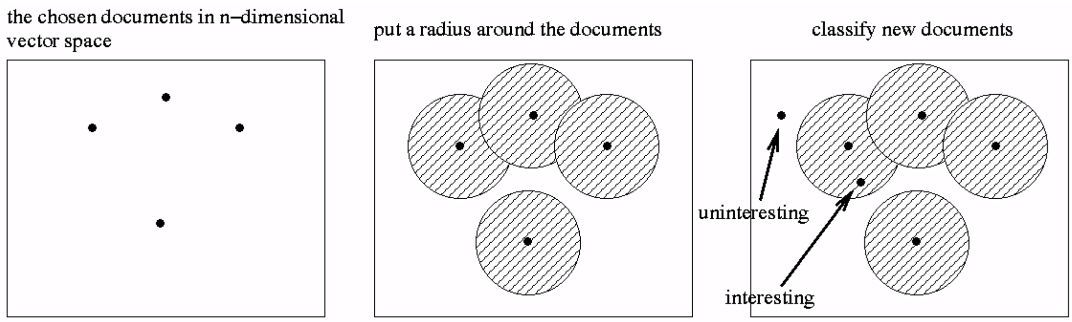

Figure 2. Instance-based learning approach.

immediately used to rank the unseen objects. The objects which are similar to at least one previously selected object receive higher values and should be shown to the user.

While the recommendation components try to assess objects as a whole when determining their interestingness, these learning results cannot be used very easily for characterizing individual users' preferences explicitly, which is a desirable feature of user modeling systems (Kobsa, Koenemann, and Pohl 2001). Therefore, we employed statistical methods to find the object features that are especially important to an individual user. This is more in line with traditional user modeling approaches where user models are knowledge bases with an explicit representation of user characteristics. Since such user profiles can be inspected and changed by the user, the system becomes much more transparent. Practice has shown that this is an extraordinary important feature for the acceptance of the adaptive system.

In the following subchapters, we will describe the recommendation components that are used at the decision stage. We can only give a brief summary of the different methods however. For a more detailed discussion of the used methods and of the other stages (namely the acquisition and the user model stage) see (Schwab, Pohl, and Koychev 2000).

3.1 Probabilistic Approach

We use Bayes' theorem to calculate the probability of user interest for a given document. That is, we apply a simple Bayes classifier to positive examples only. For each feature of the representation of a new object, a product is computed of the probabilities of this feature appearing in previously selected objects. The values of the features have to be equal to the values of the current object. Like the simple Bayes classifier, this approach assumes that the features are mutually independent.

This algorithm computes a single value for a given document. The idea is that interesting documents should receive higher values whereas uninteresting documents should receive lower values. When applying this algorithm to the available data for each user, unselected documents turned out to generally have lower values than selected ones. We therefore use this method in combination with an interest threshold. Documents with probability values greater than the selected threshold are assumed to belong to the class “interesting” and are recommended to the user. Alternatively, this method can also be used to rank the objects using the calculated probability value, and to propose the *n* best documents to the user.

3.2 Instance-Based Approach

One of the most popular machine learning algorithms is the k-Nearest Neighbor (kNN) approach (Mitchell, 1997). For this algorithm, learning means remembering previously (classified) experiences. Each experience (called instance) can be represented as a point in a predefined space (for example, a Euclidean space). Classification of a new instance then amounts to searching for its nearest neighbors, and to assign it to the class to which the most neighbors belong. Since the selected documents are considered as positive evidences and since we do not assume by default that the unselected ones are negative examples, we only have a single class. A standard kNN procedure would therefore classify each new document as positive. We therefore modified the Nearest Neighbor idea by examining a space of fixed size around each previously selected document. A new document is considered interesting if its distance to at least one previously selected document is less than this radius (see Figure 2).

To operationalize this idea, a suitable distance measure must be found, and the size of the examined space around new documents must be determined. We use a weighted distance measure for this purpose which is individually computed for each user. The underlying rationale is that a high weight for an attribute that is very crucial to a user will lead to larger distance values between objects that differ in this feature. We obtain such distance weights from the feature analysis mechanism that is used for learning explicitly represented user preferences at the acquisition stage.

3.3 Content-based Collaborative Recommendations

Forming user groups

User-adaptive systems typically learn about individual users by analyzing observations about their behavior. However, it may take a significant amount of time and observations to construct a reliable model of user interests, preferences and other characteristics. Additionally, it is useful to take advantage of the behavior of similar users. This problem is often solved by abandoning content-based user modeling in favor of the collaborative filtering approach (Konstan et al., 1997).

Earlier, the user modeling community provided a different answer, namely the stereotype approach (Rich 1979, 1989). During the development time of a system, user subgroups are identified and typical characteristics of members of these subgroups determined. During the runtime of the system, users are assigned to one or more of these predefined user groups and their characteristics attributed to these users. The need for an (empirically based) pre-definition of these stereotypes is an obvious disadvantage. As an alternative, Orwant (1995) and Paliouras (1999) used clustering mechanisms to find user groups dynamically, based on all available individual user models.

We pursue a similar approach to group modeling. Explicitly represented user models can be clustered and the descriptions of the clusters can be used like predefined stereotypes. In contrast to “real” stereotypes, clusters are acquired dynamically and can be revised whenever needed. Thus dynamic evolution of user groups can be accounted for. However, in some domains it can be difficult to find well-distinguished groups of users. Sometimes users’ interests are very idiosyncratic and sometimes users fit into more than one group. Users also can be grouped along different dimensions. Hence, the system should be able to manage many different groupings of users. Additionally, in some cases the current user’s interests will change frequently and rapidly. As a result, the dynamics of user group formation can be very fluid, which will require high computational costs to keep the users’ group memberships up to date, especially with large numbers of users. A possible solution is the Nearest Neighbor method for defining a small group of those users who are most similar to the current user. This approach can then be used for collaborative recommendations.

When grouping users, the key issue is to define an appropriate similarity measure. We employ Pearson correlation for measuring the similarity between users’ explicit profiles. The correlation between two users u_x and u_y is measured on the basis of their feature set, which is defined as the union of the selected features for each user: $F_{x,y} = F_x \cup F_y$. This avoids considering two users to be similar if they have a lot of irrelevant features in common. The values in the user vectors are the normalized weights calculated by the significance analysis at the acquisition stage. Therefore, the correlation between two users can be calculated as follows:

$$r(u_x, u_y) = \frac{\sum_{i \in F_{x,y}} (w_{x,i} - \overline{w_x})(w_{y,i} - \overline{w_y})}{\sqrt{\sum_{i \in F_{x,y}} (w_{x,i} - \overline{w_x})^2 \sum_{i \in F_{x,y}} (w_{y,i} - \overline{w_y})^2}} \quad \textbf{(1)}$$

where $w_{j,i}$ is the weight of the feature i for the user j, calculated by the feature selection test (see Figure 1); and $\overline{w_j}$ is the average weight for the user j.

In our domain, we mostly encounter short-time users and it is difficult to distinguish well-defined user groups. Hence, we opted for a Nearest Neighbor approach for collaborative recommendations. Let $D_{rec,x}$ be a set of documents that consists of a union of sets of documents that have been seen by the group of neighbor users but were not yet selected by the current user x:

$$D_{rec,x} = \bigcup_{u_i \in U_{sim}} D_{u_i} - D_x \quad \textbf{(2)}$$

In the next step, the documents from $D_{rec,x}$ are rated using weighted voting as follows:

$$R_{d_i \in D_{rec,x}} = \sum_{j=1}^{k} r(x, u_j) v_{j,i} \quad \textbf{(3)}$$

where, $v_{j,i} \in \{0,1\}$ and $v_{j,i} = 1$ iff $d_i \in D_{u_j}$. While the k-NN method may yield neighbors that are actually quite distant (in cases where the user has less than k nearby neighbors), weighted voting prevents such neighbor users from having a big influence on recommendations. Finally, the documents that received the highest ratings are recommended to the user x.

Collaborative recommendation can be given to users independently from content-based recommendation. This allows users to choose the information source for achieving their goals. Another approach is a weighted voting method to merge both sources of recommendation. The weight of each source can be dynamically adjusted by observing user's actions on the list of recommended documents.

Experiments

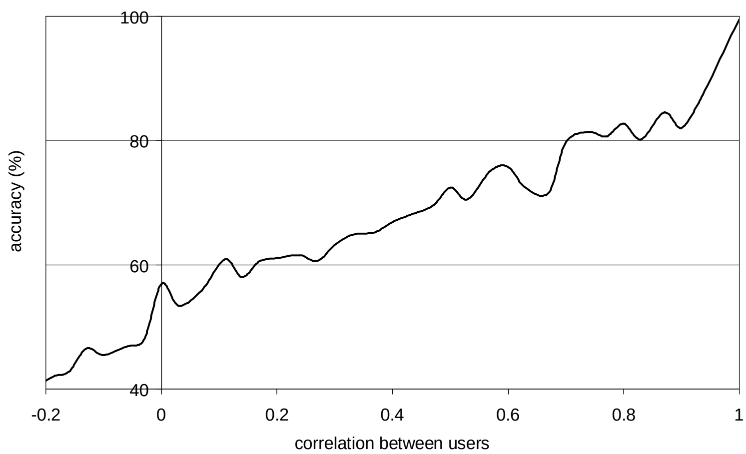

Figure 3. The relationship between the similarity between two users (based on the correlation between content-based profiles) and the quality of the collaborative recommendations**.**

To show the relevance of our algorithm we conducted several experiments. We calculated the correlation between each possible combination of two users. After that we tested how the correlation between two possible users is related to the prediction accuracy. We were able to show that there is a strong correlation (0.98) between the accuracy of collaborative recommendation as described above, and the correlation (i.e. similarity) between users (see Figure 3). The prediction quality of 50% represents a random recommendation strategy. If the correlation between two users is nearly 1, the average prediction quality is also very high (i.e. it approaches 100%). If the correlation is approximately 0, then the prediction quality is equal to that of a random recommender. We conclude that the ELFI domain is amenable to collaborative recommendation (i.e., it is possible to give accurate recommendations if users are similar).

This result can be used to select a lower threshold for the similarity between users that still preserves a desired level of expected recommendation accuracy. For example, a threshold of 0.95 for the correlation between users will guarantee that the average prediction accuracy of collaborative recommendation will be more than 90%. Additionally, our experiments suggest that collaborative recommendation based on similarity in content-based profiles achieves its high precision accuracy with considerably less computational efforts than traditional approaches, which compare items.

4. The Adaptive ELFI System

ELFI[3] is a WWW-based system that provides access to a database of funding programs and funding agencies. Users are grant officers in research institutions who monitor and advise on extra-mural funding opportunities, and also individual scientists. The information space that consists of these information objects is organized into hierarchies of, e.g., research topics (mathematics, computer science, etc.) and funding types (grant, fellowship,

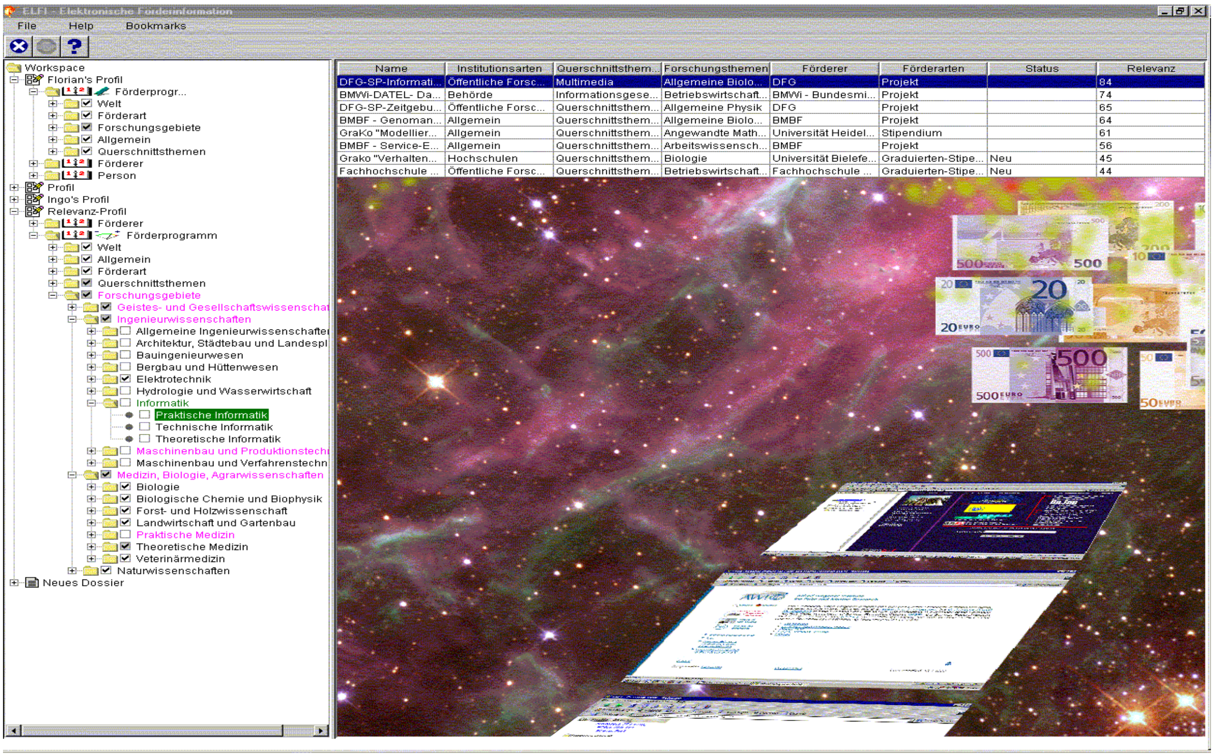

Figure 4. The user interface of the adaptive ELFI system.

etc.). At the user interface, these hierarchies are visualized as directory trees (see the left part of figure 4), which allow the user to navigate easily through the information space. In addition, the system permanently displays the contents of the current information subspace by listing links to the respective funding programs. For instance, when the user selects the research topic "computer science", links to all available fellowships in computer science are listed. The user is also allowed to check or uncheck special areas of the navigation tree. Unchecked topics are not further expanded. In Figure 4, the adaptive system has learned from the user's navigation behavior that the user is interested in "Applied Computer Science" (Praktische Informatik). It realizes that the area is unchecked and informs the user about this fact by highlighting it in green, thereby prompting the user to also check this area for the future. It also lists links to all available grants in applied computer science (see the top of Figure 1). Before being displayed, the research grant announcements of the selected information subspace are matched with the learned user model and ordered by relevance. The most promising research grants then appear at the top of the list. The user can select these links to obtain a more detailed view of the grants.

Conclusion

For developers of user-adaptive interactive systems, it remains a challenge to design interfaces that are able to acquire users' ratings of presented information in an unobtrusive manner (some special cases where this seems possible are discussed in (Kobsa, Koenemann, and Pohl 2001)). Whenever users have to select interesting

---

[3] See http://www.elfi.ruhr-uni-bochum.de/. A more detailed description of ELFI can be found in (Schwab, Pohl and Koychev, 2000).

objects from larger sets, like links on a web page or products from a web store, negative evidence (both explicit and implicit) will always be difficult to obtain. In such cases, the methods presented in this paper can be fruitfully employed.

References

Carroll, J. and Rosson, M. B. (1987). The Paradox of the Active User. In J. M. Carroll (Ed.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Cambridge, MA: MIT Press.

Billsus, D., and Pazzani, M. J. (1999). A Hybrid User Model for News Classification. In *Kay J. (ed.), UM99 User Modeling - Proceedings of the Seventh International Conference*, 99-108. Springer-Verlag, Wien, New York.

Kobsa, A., Koenemann, J. and Pohl, W. (2001). *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. The Knowledge Engineering Review* 16 (2), 111-155. http://www.ics.uci.edu/~kobsa/papers/2001-KER-kobsa.pdf

Kobsa, A. (2001). Tailoring Privacy to Users' Needs (Invited Keynote). In M. Bauer, P. J. Gmytrasiewicz and J. Vassileva, eds.: User Modeling 2001: 8th International Conference. Berlin - Heidelberg: Springer Verlag, 303-313. http://www.ics.uci.edu/~kobsa/papers/2001-UM01-kobsa.pdf

Konstan J., Miller B., Maltz D., Herlocker J., Gordon L., Riedl J. (1997). GroupLens: Applying Collaborative Filtering for Usenet News. In *Communications of the ACM*, March 1997/Vol.40, No. 3, 77-87.

Lieberman, H. (1995). Letizia: An Agent That Assists Web Browsing. *International Joint Conference on Artificial Intelligence*, Montréal, 924-929.

Mitchell T. (1997). Instance-Bases Learning. Chapter 8 of *Machine Learning*, McGraw-Hill.

Mladenic, D. (1996). Personal WebWatcher: Implementation and Design. Technical Report IJS-DP-7472, Department of Intelligent Systems, J. Stefan Institute, Slovenia.

Orwant J. (1995). Heterogeneous Learning in the Doppelgänger User Modeling System. *User Modeling and User-Adapted Interaction*, 4(2), 107-130.

Paliouras, G., Karkaletsis, V., Papatheodorou, C. and Spyropoulos, C. (1999). Exploiting Learning Techniques for the Acquisition of User Stereotypes and Communities. In J. Kay, ed.: *UM99 User Modeling: Proceedings of the Seventh International Conference*, Springer-Verlag, Wien, New York, 169-178.

Pazzani, M. J. and Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27, 313-331.

Rich, E. (1979). User Modeling via Stereotypes. *Cognitive Science* 3, 329-354.

Rich, E. (1989). Stereotypes and User Modeling. In: Kobsa, A. and W. Wahlster (eds.): *User Models in Dialog Systems*, Berlin, Heidelberg: Springer, 35-51.

Schwab, I., Pohl, W., and Koychev, I. (2000). *Learning to Recommend from Positive Evidence*, Proceedings of Intelligent User Interfaces 2000, ACM Press, 241-247.

Ingo Schwab

humanIT Human Information Technologies AG, Sankt Augustin, 53757, Germany.
Email: Ingo.Schwab@humanIT.de
After studying Computer Science at the University of Dortmund, Ingo Schwab joined GMD – German National Research Center for Information Technology. There he was a project manager in the research areas of User Modeling and Human Computer Interaction. Recently, he joined humanIT AG and is now responsible for several projects in the area of User Modeling and Personalization.

Alfred Kobsa

Dept. of Information and Computer Science, University of California, Irvine, CA 92697-3425, U.S.A.
Email: kobsa@uci.edu   http://www.ics.uci.edu/~kobsa/
Prof. Kobsa applies user modeling to information environments, multimedia educational software, expert finders, and user interfaces for the disabled and elderly. He is the editor of User Modeling and User-Adapted Interaction, editorial board member of several journals, and was the founding president of User Modeling Inc. He also co-founded the GI ABIS workshop series and the International User Modeling Conference series.