Learning to Recommend from Positive Evidence

Ingo Schwab, Wolfgang Pohl, and Ivan Koychev
GMD-FIT.MMK,
D-53754 Sankt Augustin, Germany
+49 2241 14-2856
{Ingo.Schwab,Wolfgang.Pohl,Ivan.Koychev}@gmd.de

ABSTRACT

In recent years, many systems and approaches for recommending information, products or other objects have been developed. In these systems, often machine learning methods that need training input to acquire a user interest profile are used. Such methods typically need positive and negative evidence of the user's interests. To obtain both kinds of evidence, many systems make users rate relevant objects explicitly. Others merely observe the user's behavior, which fairly obviously yields positive evidence; in order to be able to apply the standard learning methods, these systems mostly use heuristics that attempt to find also negative evidence in observed behavior.

In this paper, we present several approaches to learning interest profiles from positive evidence only, as it is contained in observed user behavior. Thus, both the problem of interrupting the user for ratings and the problem of somewhat artificially determining negative evidence are avoided.

The learning approaches were developed and tested in the context of the Web-based ELFI information system. It is in real use by more than 1000 people. We give a brief sketch of ELFI and describe the experiments we made based on ELFI usage logs to evaluate the different proposed methods.

Keywords

Adaptive recommendation interfaces, user modeling, machine learning, evaluation of methods

1. INTRODUCTION

In recent years, many systems have been developed which try to help users to find pieces of information or other objects that are in accordance with their personal interest. There have been mainly two different approaches: On the one hand, content-based (information) filtering systems (see [5] for some examples) mostly take individual preferences with respect to object content into account.

On the other hand, recommender systems [11] typically build on similarities between users with respect to the objects they interact with.

In many systems, users must provide explicit ratings to express their attitudes. This requires additional user effort and keeps users from performing their real task. Both are undesirable. Alternatively, conclusions about user interest should be drawn from merely observing user interactions.

Our goal is to develop a content-based recommendation component for the ELFI information system. In order to be unobtrusive, it shall learn individual interest profiles from observation only. In ELFI (and this is typical for other systems, too), observation provides information about what the user prefers, but not about what the user does *not* prefer.

With respect to the application of learning methods this means that mainly positive evidence (i.e., training examples) is available. In the next section, we briefly describe the ELFI system. Then, we present the learning methods we used and (partially) developed to cope with this problem. In order to evaluate these methods, we performed experiments using observation logs of real ELFI users; these experiments and their results are described afterwards. Finally, we discuss the results of our work.

2. RELATED WORK

In the past, several systems have been developed employing learning procedures to identify individual user's interests with respect to information objects and their contents and make use of this interest profile to make personalized recommendations.

Lieberman [4] developed the system Letizia, which assists a user in Web browsing. It tries to anticipate interesting items on the Web that are related to the user's current navigation context (i.e., the current Web page, a search query, etc.). For a set of links it computes a preference ordering, based on a user profile. This profile is a list of weighted keywords, which is obtained by aggregating the results of TFIDF analyses of pages [5]. Letizia uses heuristics to determine positive and negative evidence of the user's information interest. Viewing a page indicates interest in that page, bookmarking a page indicates even stronger interest, while "passing over" links (i.e., selecting a link below and/or on the right of other links) indicates disinterest in these links.

A classification approach is taken by the system called Syskill&Webert [9]. The user rates a number of Web documents from some content domain on a binary "hot" and "cold" scale. Thus, positive and negative learning examples become available to the system. Based on the 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, resp.

Figure 1 Screenshot of ELFI. Recommended funding programs are shown on the bottom of the screen.

Also the system Personalized WebWatcher [7] uses the Naive Bayes Classifier. This system watches individual users' choices of links on Web pages 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, non-visited links as negative ones. For ELFI, we could have taken a similar approach. However, we think that non-selection does not necessarily mean disinterest. Such a procedure may lead to many misclassified negative examples and, hence, to too much noise in the training set.

The Naive Bayes Classifier is again used in the system NewsDude [2], similarly to Syskill&Webert, to recommend news articles to users. In NewsDude, the probabilities are taken to characterize the long-term interests of a user. To avoid recommending too many similar documents to a user, an additional short-term profile is built by memorizing currently read articles. New articles are then compared to the memorized ones; if they are too similar, they are not recommended although they will typically match the long-term interest profile. This procedure corresponds to the nearest neighbor classification algorithm well known in Machine Learning. Note, that for the short-term profile only positive examples are needed (however, to produce "negative" recommendation). Thus, we can use a similar nearest neighbor procedure also for ELFI (see the section "Learning about Interesting Documents").

3. THE ELFI INFORMATION SYSTEM

ELFI[1] (ELectronic Funding Information) is a WWW-based system that provides information about research funding. ELFI is described in detail in [8]. Here, we will give only a brief impression of the system.

Essentially, ELFI provides access to a database of funding programs and funding agencies. The information space that consists of these information objects is organized into hierarchies of, e.g., research topics (mathematics, computer science) or funding types (grant, fellowship). At the user interface, these hierarchies are visualized as directory trees, which allow the user to navigate through the information space. In addition, the system permanently displays the contents of the current information subspace by listing links to so-called detailed views (DVs) of relevant funding programs. For instance, when the user selects the research topic "mathematics" and the funding type "fellowship", links to all available DVs of fellowships in mathematics are listed. The user can select such a link to visit a DV. A DV displays the available data about the program like an abstract of the program, research topic(s) covered, etc. in a structured way (see Figure 2).

Since ELFI users are not supposed to rate the usefulness of information objects (i.e., DVs), adaptivity in ELFI must be based on an analysis of observed user behavior. ELFI records all user interactions with the system. In particular, the log contains information about the DVs selected by ELFI users. In our current work, this information is central for determining interest profiles.

In the next section, we describe methods that we developed for learning interest profiles in ELFI. With such a learning method incorporated, ELFI could be extended to make personalized suggestions: When a user logs into ELFI, newly available DVs are matched against her individual interest profile. Particularly relevant DVs are presented to her in an unobtrusive way: Suggestions are listed below the familiar display of the current information subspace. In Figure 1, this recommendation facility is demonstrated; it suggests the three funding programs that match the current user profile best.

4. LEARNING FROM POSITIVE EVIDENCE

Machine learning methods can be used to solve classification problems. Hence, a straightforward way of using machine learning for acquiring interest profiles is to assume that the set of information objects can be divided into classes (e.g., "interesting" and "not interesting"). Then, content-based filtering can be achieved by letting the user classify given documents, thus

[1] http://www.elfi.ruhr-uni-bochum.de/elfi/

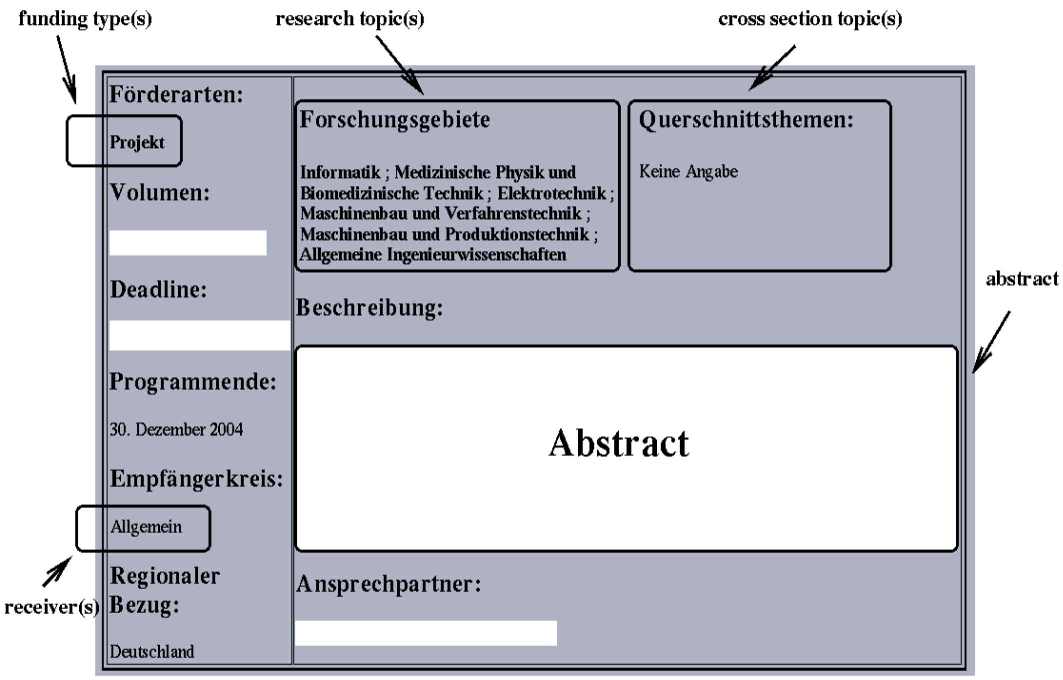

Figure 2 ELFI detailed view

providing examples for both classes (see, e.g., [9]), and apply an inductive classification algorithm to these examples. For further documents, the classification algorithm can determine whether they belong to the "interesting" or to the "not interesting" class.

Also in ELFI, it can be assumed that documents (the DVs) can be divided into such two classes. However, supplying an appropriate set of negative examples (i.e., examples of the "not interesting" class) is problematic. The central source of information about the user is the sequence of selected DVs. Selections are made from the current information subspace, that is a set of available DVs with common properties. We have already mentioned that there are systems, which in similar situations (i.e., selection from a set of objects), use unselected objects as negative examples. However, for ELFI we claim that unselected DVs may exist, which are interesting to the user (they may just have been overlooked by the user or will perhaps be visited later). Classifying them negative is a dangerous assumption, since many of these classifications may be wrong and too much noise in the training data may result. It is more suitable to take only selected DVs as examples for the "interesting" class. However, in this case standard classification methods are not applicable. Thus, for learning interest profiles in ELFI we had to invent new learning methods or modify existing ones. This section presents the different methods after describing the available input data more closely.

A probabilistic approach and an instance-based approach were developed that can be applied to learn a general characterization of documents being relevant to the user. Both approaches deal with the positive examples problem by employing a notion of similarity or distance. However, it is difficult to use these learning results to characterize the individual user's preferences explicitly, which is a desirable feature of user modeling systems [10]. Therefore, we developed a third mechanism that aims at selecting those features that are extraordinarily important to the user for identifying relevant DVs. It turned out that this feature selection method additionally helps to improve the distance measure for instance-based learning. Moreover, feature selection can be combined with both probabilistic and instance-based learning to focus the learning task.

4.1 Input for Learning

Like in many approaches to learning interest profiles, representations of the (information) objects the user is dealing with are needed as input for learning. For this representation, crucial features need to be identified and appropriately coded. ELFI DVs consists of several features (mentioned in Figure 2), of which we chose the five most important ones: The suitability of the selected features was tested on a log file of several months of system usage. The mean frequencies of occurrence of feature values were calculated for selected and unselected DVs. If a user has a selection strategy and is not reading a random sample of DVs, the mean for the selected DVs should be higher for interesting and lower for uninteresting features. We found that some features are better indicators of interest than others. For example, the feature "funding types" characterizes the set of selected DVs well.

The chosen features of the DVs are represented by one vector. All features are set-valued (the text of the program abstract can be considered a set of words). We use the natural representation for sets as Boolean vectors (one bit for every element of the base set; a bit is positive iff the respective element is in the current value set); i.e., each bit corresponds to one possible value of one DV feature. The vectors for the selected features are concatenated to produce the DV representation. In order to avoid getting an unusably large DV vector, we reduced the base set of the abstract (all possible words) to the 189 most discriminating words.[2]

4.2 Learning about Interesting Documents

4.2.1 Probabilistic Approach

We used Bayes' theorem to calculate the probability of user interest for a given DV. That is, we applied a simple Bayes classifier to only positive examples. Thus, for the vector representation of a new DV a product is computed of the probabilities for each bit that in previously selected DVs the value

[2] A TFIDF measure was applied to determine these words. For sake of brevity, we will not go into details here.

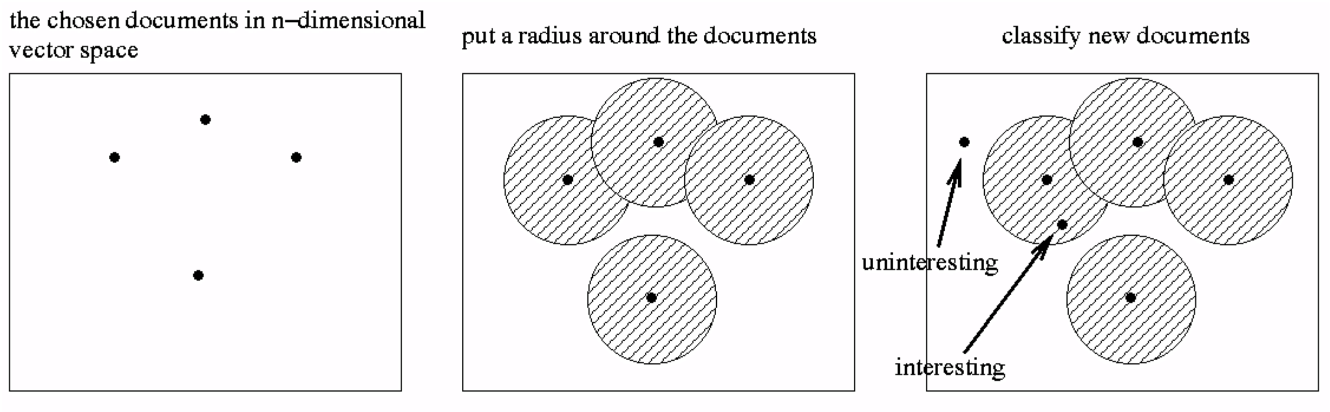

Figure 3 Instance-Based Learning Approach

of this bit is equal to its value in the current vector. Like the simple Bayes classifier, this approach assumes that the bits are mutually independent.

This algorithm computes a single value for a given DV. The idea is that interesting DVs should receive higher values whereas uninteresting DVs should receive lower values. Experiments were quite encouraging. Among the available data, in general unselected DVs have lower values than the selected ones. This suggests that the approach can be used for interest prediction, if a sensible threshold value is chosen. Then, new funding programs with values greater than the threshold can be assumed to belong to the "interesting" class and can be recommended to the user. One alternative to using a threshold for classification would be to use the resulting values immediately for ranking. The best n DVs could be proposed to the user.

4.2.2 Instance-Based Approach

One of the most popular Machine Learning algorithm [6] is the k-Nearest Neighbor (kNN) approach. For this algorithm, learning means remembering previous (classified) experiences. Each experience (or instance) can be represented as a point in an Euclidean space. Then classification of a new instance means searching the nearest $k, k \geq 1$ neighbors of the new instance. The class of the majority of these neighbors determines the result of classification. Since selected DVs are positive examples only, a standard kNN procedure would always classify a new DV positive. We modify the kNN idea by examining a space of fixed size around each previously selected DV (see Figure 3). A new DV is considered interesting if its distance to at least one previously selected DV is less than this radius

When implementing this idea, two questions arise. First, what is a good distance measure? In a first approach, we used a Hamming distance, which is the number of different bits of two compared vectors. Second, how large should the examined space around new DVs be? Like with the probabilistic approach, a good threshold is needed.

Experiments showed that the quality of the Hamming distance is insufficient. Since every bit in the representation vector is assumed to be equally important to every user, it does not take individual user interests into account. Therefore, a weighted distance measure is needed which is individually computed for each user. The idea is that a large weight for an attribute being very crucial to a user will lead to larger distance values between documents that differ in this feature. We obtained such distance weights from the feature selection mechanism used for learning explicit preference information. See next section for a description of the exact procedure.

4.3 Feature Selection: Learning Explicit Information about User Preferences

In this section a statistical approach is described, which can be used to generate explicit assumptions about a user and can do so from positive examples only. It uses a univariate significance analysis to determine if a user is interested in specific values of the DV features. It is based on the idea that attribute values in random samples are normally distributed. If the value appears in the selected DVs significantly more frequently than in a random sample, the user is interested in it. On the other hand, if the selection frequency is lower, the user is not interested in that value.

To explain this idea, we take a typical example from ELFI. We want to determine if a user is interested in the funding type "project" (i.e., the value "project" of the DV attribute "funding type"). First we calculate the probability of this funding type in all DVs. Let us assume that in ELFI there are 815 DVs available (which is a typical number); and 316 DVs contain this feature. Thus, the probability to randomly select a DV with this feature is $p = \frac{316}{815} = 0.39$. For random DV selections from the overall set, however, there will be a mean error, so that a confidence interval around the actual p needs to be determined. If the actual frequency lies outside this interval, it can be assumed with a certain confidence that the user has not made a random choice and that there is a kind of strategy involved in the user's selection. The confidence interval $[c_1, c_2]$ is given by the following formula:

$$c_{1,2} = \mu \pm z * \sqrt{p * (1-p) * n}$$

μ is the mean of the distribution and equal to the above overall probability p multiplied by the number of selections, while z is the critical value. It determines the area under the standard normal curve; for a confidence rate of 95% the value is 1.96. This means that 95% of random samples fall within this interval and

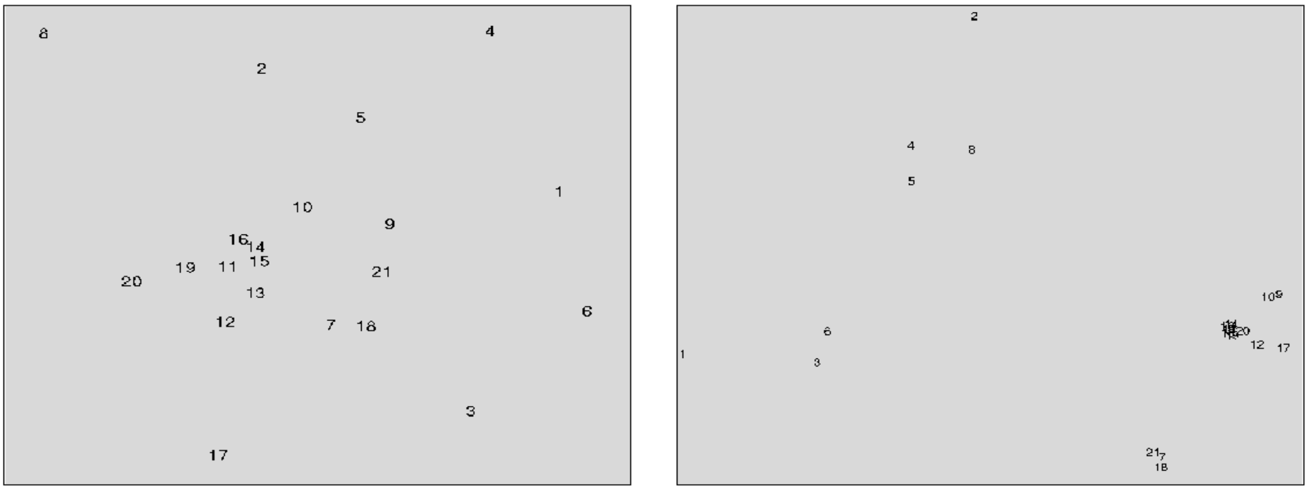

Figure 4 Selected DVs, displayed using an unweighted (left) and a weighted (right) distance measure.

5% are outside. Thus, there is a chance of 5% of misclassifying a user. For a greater confidence also z is greater; e.g., for 99% confidence, z is 2.576.

Let us now assume that a user selects 30 DVs. Then the 95% confidence interval for the bit that corresponds to the "project" funding type can be calculated: $c_1 = 6.4$ and $c_2 = 16.86$. These numbers yield the following procedure for acquiring explicit assumptions with respect to the value "project" of the attribute "funding type": If the value appears in 6 or less of the selected DVs then the user is not interested in documents with this value. If 17 or more DVs contain the value, the user can be regarded as interested in documents with this value. If the number of selected documents with this value is between 6 and 17, then this value is not significant and will be used neither as a positive nor as a negative indicator of interest. Using this procedure an explicit user profile can be constructed. For every feature such a univariate significance analysis can be done and explicit information about users can be derived.

Figure 5 shows an example output of this procedure. The attributes, which are important to the user, are listed. The value in the bracket is the normalized value of interest. It lies between -1 (totally uninteresting) and 1 (totally interesting). In this example 15 attributes are of (positive) importance to the user.

4.3.1 Obtaining Weights for Distance Measuring

As mentioned before the results of the univariate significance analysis can be used to obtain feature weights for the distance measure, which is needed for the instance-based learning approach.

The effect of the weighted distance measuring can be seen in Figure 4. It shows two visualizations of user selected DVs. This visualization uses a technique called multi-dimensional scaling [3]. It allows us to show the relationships between selected DVs in two dimensions. Here the selected DVs are numbered from 1 to 21 in a chronological (according to the selection time) order.

In the left picture a simple Hamming distance is used. Here, the user's behavior and the resulting preferences do not become visible. The right picture visualizes the same DV selection using a weighted distance measure; weights are obtained from feature selection. Here, user behavior is clearly visible. In the beginning (DVs from 1 to 8) the user tries to find the interesting DVs. Perhaps she is just playing or experimenting and tries to figure out which kind of information or which interaction features ELFI offers. But after this training period she has found the information she was looking for. In the rest of her ELFI usage the selected DVs are very similar (and therefore form a cluster in the

User is interested in:
- *Research Topics:*
  - Mathematik (0.85)
  - Luft-undRaumfahrttechnik (0.88)
  - Regelungstechnik (0.88)
  - Verkehrsforschung (0.52)
- *Funding Type:*
  - Druckkostenzuschuss (0.56)
- *Receivers:*
  - Welt (0.38)
  - Entwicklungslaender (0.46)
- *Abstract:*
  - BILDVERARBEITUNG (0.39)
  - FINANZIERUNG (0.52)
  - FREQUENZEN (0.56)
  - LEBEN (0.64)
  - LUFTFAHRTFORSCHUNG (0.99)
  - MULTIMEDIA (0.66)
  - WISSENSCHAFTLICHES (0.39)

User is *not* interested in:
- *Research topics:*
  - Phytomedizin (-0.84)
  - PhysischeGeographie (-0.76)
  - TheoretischeMedizin (-0.79)
- *Funding Type:*
  - Stipendium(-0.55)
- *Abstract:*
  - TEILZEITARBEIT (-0.49)
  - DISSERTATIONEN (-0.41)

Figure 5 An explicit user profile.

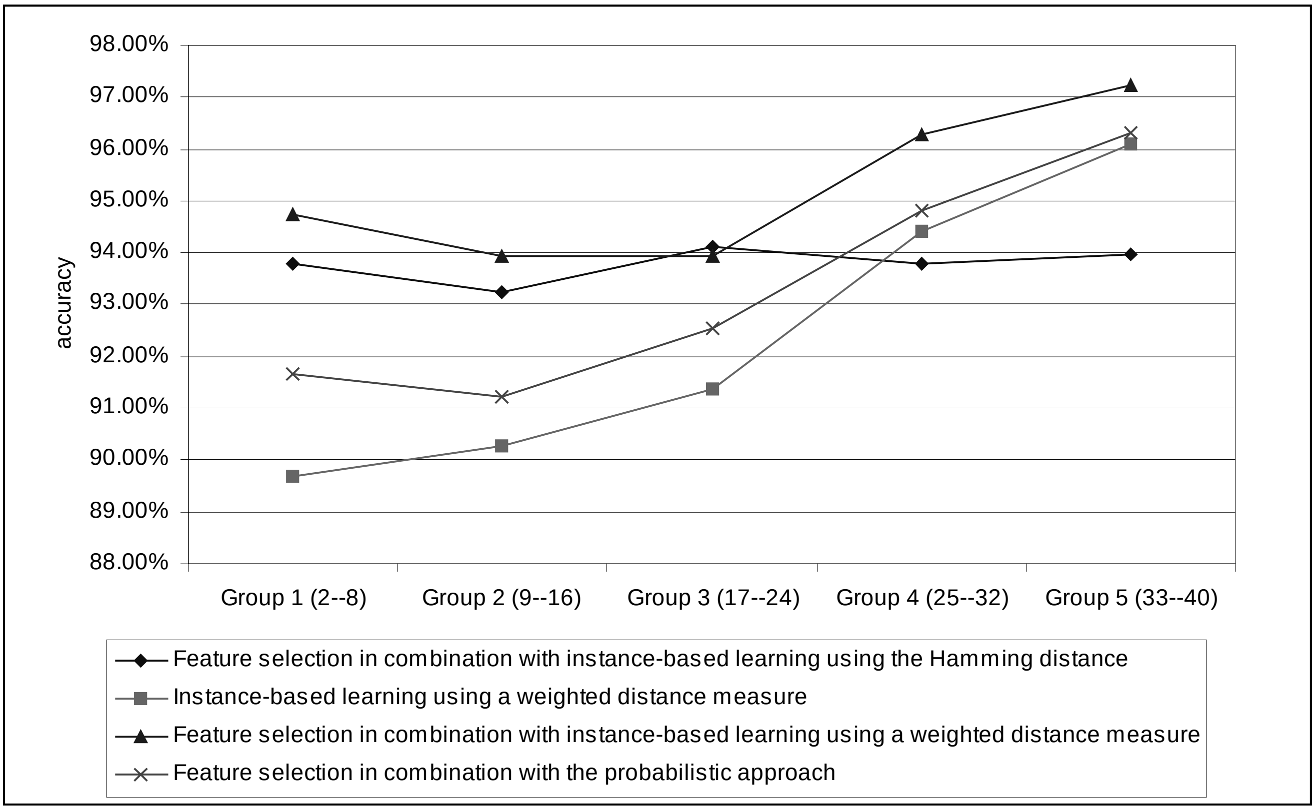

Figure 6 Prediction accuracy of compared approaches according to the number of selected documents.

right picture). New or overseen DVs similar to the DVs of this cluster could be recommended to the user.

4.4 Combining Learning with Feature Selection

A problem with ELFI observation data is that the dimensionality of the DV-describing vectors is quite large. We have many (420) features, while the amount of training data is very limited. This typical problem often arises in adaptive interactive systems that use learning methods. Learning under these conditions is not practical, because the amount of data needed to approximate a concept in *d* dimensions grows exponentially with *d*, a phenomenon commonly referred to as the curse of dimensionality [1]. Hence, there is a need for dimensionality reduction.

Normally a-priori information can help with the curse of dimensionality. Careful feature selection and scaling of the inputs fundamentally affect the severity of the problem, as well as the selection of the learning algorithm. We believe that our feature vector contains no unnecessary features. All features are important for one kind of research or research interest. Every feature can help to model user profiles. Furthermore we believe that every user has different interest and therefore also different features that are important to her. A feature selection should be individualized and processed for each user. The univariate significance analysis (see the previous section) is able to execute this task. It considerably reduces the dimension of the learning task and the significantly uninteresting and interesting features for each user still remain. In the example shown in Figure 5 the algorithm is able to reduce the considered features from 420 to 20. Thus, the feature selection can be simply combined with our existing learning algorithms. In a first step the individualized features are determined. With this subset of features the learning task is performed. This combined learning approach is much more noise resistant, learns much faster and the performance of our learning algorithms can be significantly improved, as we will show in the next section.

5. EVALUATION OF LEARNING METHODS

The observations about users are sequences of user actions. Usually these sequences are considered as time series. In the ELFI environment the relationships between selections are not causal. That means, the next selections do not depend on previous ones. The user selections are primarily goal driven. The user aims to select documents that are ”interesting” for her. Therefore, in our study we regarded user selections as relatively independent tries to find interesting documents (DVs), which allows us to use standard cross-validation techniques for determining the prediction accuracy of the used learning algorithms.

Furthermore, for every ELFI user a relatively small set of document selections can be observed. We decided to use the leave-one-out cross-validation technique, working as follows: For each selected DV pick it out and take the remaining selections as training set for the learning mechanism. Then each DV of the ELFI database is ranked by the learning mechanism. After that it is determined at which position the selected DV would be proposed to the user. At the end the results are averaged over all selected DVs.. This result expresses an average performance of the considered learning algorithm. The vertical axis in Figure 6 presents these values

We evaluated the different methods with a set of 220 users who selected at least two DVs. The users were divided regarding the number of document selections. The users in the first group selected 2 to 8 DVs, in the second group 9 to 18 DVs, and so on (altogether they selected 1886 DVs that is a mean of 8.5 DVs for every user). The horizontal axis in Figure 6 presents these groups.

Our experiment shows that the combination of instance-based learning with weighted distance measure and feature selection performs best. In average only three percent of the remaining DVs are ranked better and probably they are really interesting to the user.

Ideally, we would get a perfect fit for every user and every selected document. That means every selected DV would also be the favorite one for the learning algorithm. But achieving this perfect fit is nearly impossible with real user data. During the experiments we discovered that there are some "random users". They have no real selection strategy and are indeed unpredictable. Maybe, they are not familiar with a computer system and normally tend to "play" with it. The users with a random strategy are still in our data sets. But even with them the results are quite good. Since all learning algorithms significantly outperform a randomly generated advice, which would be rated at 50 percent. We should also keep in mind that the user's interest is not permanent. It can shift during the time between different logins and it can be interdisciplinary.

The conclusions of our experiments are twofold. First, our experiments show that the use of feature selection significantly improves the performance of the learning algorithms. Instance-based learning plus feature selection works well for small training sets even with a simple Hamming distance. However, with growing training set it becomes apparent that weighted distance measure learns much faster. This is an additional improvement. Second, the instance-based learning approach performs better than the probabilistic approach.

6. DISCUSSION

We described several ideas and approaches to learning information interest from positive evidence. While standard classification methods are not appropriate, we use both a probabilistic and an instance-based approach to characterize the set of information objects a user observed to interact with in a positive way.

Since we are not only interested in a single adaptivity task (i.e., predicting user-specific degrees of object relevance) but also in determining explicit information about user interest and/or preferences, we employed statistical methods to find the object features that are especially important to an individual user. While the former methods try to develop an overall idea of how interesting objects look like, this latter approach results in interest degrees for selected features that characterize the user instead of the objects. This is more in line with traditional user modeling approaches where user models are knowledge bases with explicit representation of user characteristics [10]. Moreover, the feature selection mechanisms turned out to be very beneficial in combination with the other methods and helped to strongly improve ELFI's recommendation capabilities.

While the basic mechanisms of ELFI have been designed with adaptivity in mind (e.g., logging of user actions into files was early available in ELFI), the interface has not. It may be argued that the need to rely on positive evidence only is forced upon us by ELFI's interface restrictions. Indeed, our experience with ELFI shows that it can be difficult to implement adaptivity into a system as an add-on feature. For developers of intelligent interactive systems, it remains a challenge to design interfaces that can acquire user feedback in an unobtrusive way to make negative evidence more easily available. However, we think in cases where users have to select interesting objects from larger sets, negative evidence will always be hard to obtain. Then the methods presented in this paper can be used beneficially.

7. REFERENCES

[1] Bellman R. Adaptive Control Processes: A Guided Tour. Princeton University Press, 1961.

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

[3] Kruskal J. Multidimensional scaling by optimising goodness of fit to a non-metrical hypothesis. *Psychometrika* 1, 1-27, 1964

[4] Lieberman H. Letizia: An Agent That Assists Web Browsing. *International Joint Conference on Artificial Intelligence*, Montréal, 1995.

[5] Lieberman H. Information Agents at MIT. *KI*, 12, 3, 17-23, 1998.

[6] Mitchell T. Instance-Based Learning. Chapter 8 of *Machine Learning*. McGraw-Hill, 1997

[7] Mladenic D. Personal WebWatcher: Implementation and design. Technical Report, IJS, October 1996.

[8] Nick A. and Koenemann J. and Schal E. ELFI : information brokering for the domain of research funding. *Computer Networks and ISDN Systems*, 30, 1491-1500, 1998.

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

[10] Pohl W. and Nick A. Machine Learning and Knowledge-Based User Modeling in the LaboUr Approach. In *Kay J. (ed.), UM99 User Modeling - Proceedings of the Seventh International Conference*, pp. 179-188. Springer-Verlag, Wien, New York, 1999.

[11] Resnick P. and Varian H. R. Recommender Systems. *Communications of the ACM*, 40, 3, 56-58, 1997.