# Revising User Profiles: The Search for Interesting Web Sites

**Daniel Billsus and Michael Pazzani**

Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717
{dbillsus, pazzani}@ics.uci.edu

**Abstract**

We describe Syskill & Webert, a software agent that learns to rate pages on the World Wide Web (WWW), deciding what pages might interest a user. The user rates explored pages on a three point scale, and Syskill & Webert learns a user profile by analyzing the information on each page. We focus on an extension to Syskill & Webert that lets a user provide the system with an initial profile of his interests in order to increase the classification accuracy without seeing many rated pages. We represent this user profile in a probabilistic way, which allows us to revise the profile as more training data is becoming available using "conjugate priors", a common technique from Bayesian statistics for probability revision. Unseen pages are classified using a simple Bayesian classifier that uses the revised probabilities. We compare our approach to learning algorithms that do not make use of such background knowledge, and find that a user defined profile can significantly increase the classification accuracy.

## Introduction

There is a vast amount of information on the World Wide Web (WWW) and more is becoming available daily. How can a user locate information that might be useful to that user? In previous work, we discussed Syskill & Webert, a software agent that learns a profile of a user's interest from an individual's ratings of the interestingness of Web pages, and uses this profile to identify other interesting web pages. Initial experiments with our system (Pazzani, Muramatsu, Billsus, 1996) have shown that although it is much more accurate than chance at predicting whether a user would be interested in a page, the accuracy does not increase substantially as the user rates more and more pages. In this paper, we address an approach to increasing predictive accuracy by obtaining an initial profile of the user indicating the user's interest. As more rated pages become available this profile is gradually revised to make it fit the actual observed rating. We represent this user profile in a probabilistic way, which allows us to revise the profile as more training data is becoming available using "conjugate priors", a technique from Bayesian statistics for probability revision.

Syskill & Webert uses a user profile to identify interesting web pages in two ways. First, it can annotate any HTML page with information on whether the user would be interested in visiting each page linked from that page. Second, Syskill & Webert can construct a LYCOS (Mauldin & Leavitt, 1994) query and retrieve pages that might match a user's interest, and then annotate this result of the LYCOS search. Figure 1 shows a Web page on independent rock bands that has been annotated by Syskill and Webert. In this case, the user has e.g. indicated strong interest in the band "The Breetles" (indicated by two thumbs up), a mild interest in "Dead Flower Bloom" (indicated by one thumb up and one thumb down) and no interest in "Middle Passage" (indicated by two thumbs down). The other annotations are the predictions made by Syskill & Webert about whether the user would be interested in each unexplored page. A smiley face indicates that the user hasn't visited the page and Syskill & Webert recommends the page to the user. The international symbol for "no" is used to indicate a page hasn't been visited and the learned user profile suggests the page should be avoided. Following any prediction is a number between 0 and 1 indicating the probability the user would like the page.

In this paper, we first describe how the Syskill & Webert interface is used and the functionality that it provides. Next, we describe the underlying technology for learning a user profile and how we addressed the issues involved in applying machine learning algorithms to classify HTML texts rather than classified attribute-value vectors. Based on results from the application of learning algorithms that do not make use of background knowledge, we suggest to provide the system with an initial profile of a user's interests. We show how this profile can be represented probabilistically and explain the underlying theory of conjugate priors that we use to gradually revise the model to reflect

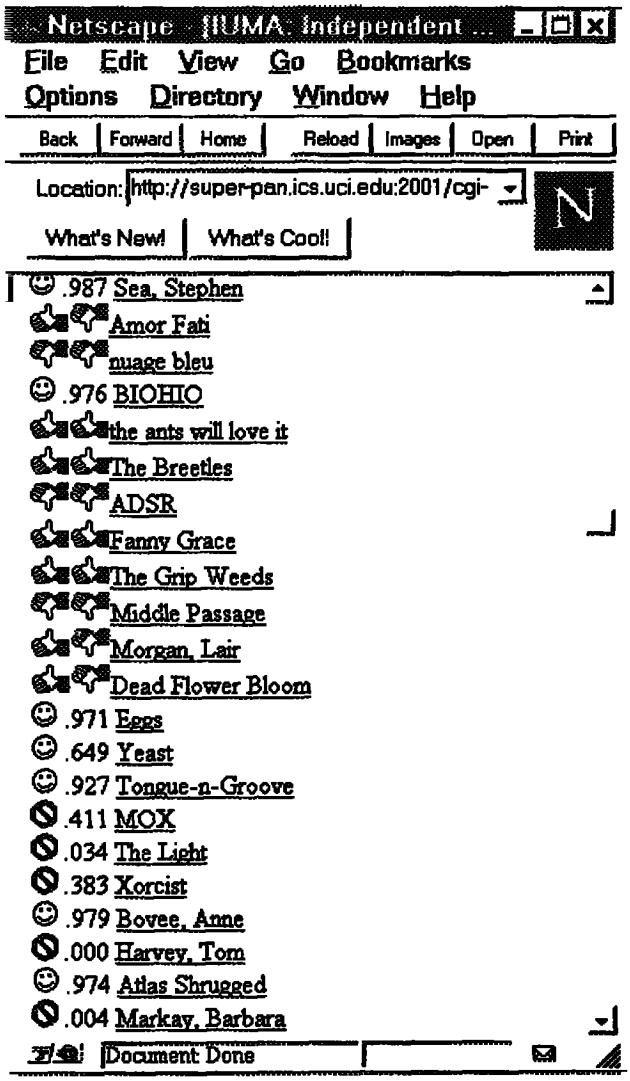


Figure 1: An example of an annotated page

available data. Finally, we describe experiments that compare the accuracy of our new approach with learning algorithms that do not make use of such background knowledge.

## Syskill & Webert

Syskill & Webert learns a separate profile for each topic of each user. We decided to learn a profile for user topics rather than users for two reasons. First, we believe that many users have multiple interests and it will be possible to learn a more accurate profile for each topic separately since the factors that make one topic interesting are unlikely to make another interesting. Second, associated with each topic is a URL that we call an index page. The index page is a manually constructed page that typically contains a hundred or more links to other information providers (see e.g. Figure 1). Syskill & Webert allows a user to explore the Web using the index page as a starting point. In one mode of using Syskill & Webert, it learns a profile from the user's ratings of pages and uses this profile to suggest other pages accessible from the index page. To collect ratings, the HTML source of web pages is intercepted, and an additional functionality is added to each page (see Figure 2). This functionality allows the user to rate a page as either hot (two thumbs up), lukewarm (one thumb up and one thumb down), or cold (two thumbs down). The user can return to the index page or switch topics. Furthermore, the user can instruct Syskill & Webert to learn a user-profile for the current topic, make suggestions or consult LYCOS to search the Web.

When a user rates a page, the HTML source of the page is copied to a local file and a summary of the rating is made. The summary contains the classification (hot, cold, or lukewarm), the URL and local file, the date the file was copied (to allow for the bookkeeping that would occur when a file changes), and the page's title (to allow for the production of a summary of the ratings).

Syskill & Webert adds functionality to the page (see Figure 2) for learning a user profile, using this user profile to suggest which links to explore from the index page, and forming LYCOS queries. The user profile is learned by analyzing all of the previous classifications of pages by the user on this topic. If a profile exists, a new profile is created by reanalyzing all previous pages together with any newly classified pages.

Once the user profile has been learned, it can be used to determine whether the user would be interested in another page. However, this decision is made by analyzing the HTML source of a page, and it requires the page to be retrieved first. To get around network delays, we allow the user to prefetch all pages accessible from the index page and store them locally. Once this has been done, Syskill & Webert can learn a new profile and make suggestions about pages to visit quickly. Once the HTML has been analyzed, Syskill & Webert annotates each link on the page with an icon indicating the user's rating or its prediction of the user's rating together with the estimated probability that a user would like the page. The default version of Syskill & Webert uses a simple Bayesian classifier (Duda & Hart, 1973) to determine this probability. Note that these ratings and predictions are specific to one user and do not reflect on how other users might rate the pages.

The extension to Syskill & Webert that we describe in this paper lets a user provide the system with words that are good indicators for a page to be interesting. In addition, the user can specify probability estimates that indicate how likely it is that a certain word would appear in an interesting page. In this paper we describe the way such a predefined user profile is revised as more rated pages are

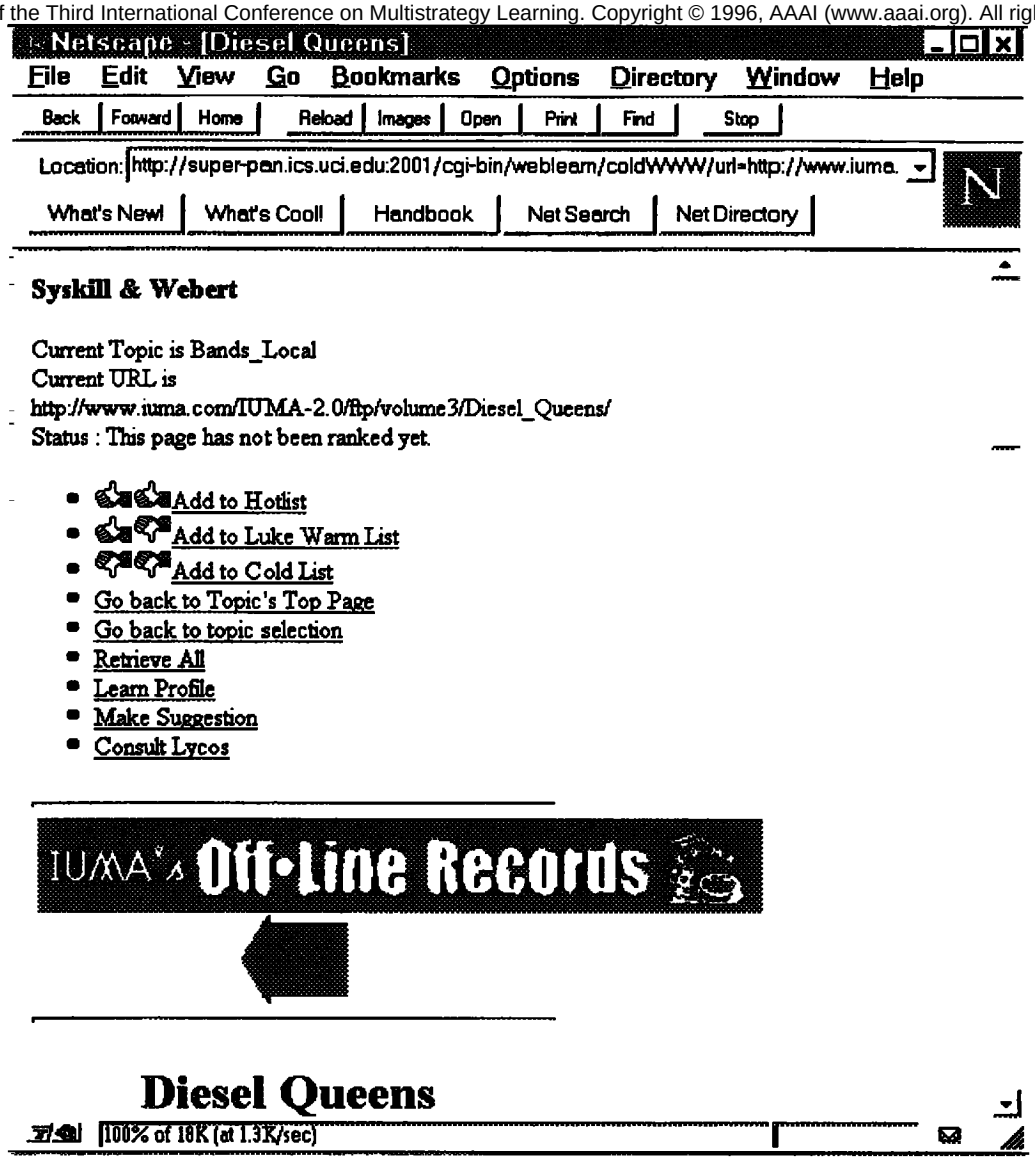


Figure 2: Syskill & Webert interface for rating pages

becoming available, and how we use the profile to classify unseen pages.

## Learning a User Profile

Learning algorithms require a set of positive examples of some concepts (such as web pages one is interested in) and negative examples (such as web pages one is not interested in). In this paper, we learn a concept that distinguishes pages rated as *hot* by the user from other pages (combining the two classes *lukewarm* and *cold*, since few pages are rated lukewarm, and we are primarily interested in finding pages a user would consider *hot*). Most learning programs require that the examples be represented as a set of feature vectors. Therefore, we have constructed a method of converting the HTML source of a web page into a Boolean feature vector. Each feature has a Boolean value that indicates whether a particular "word" is present (at least once) or absent in a particular web page. For the purposes of this paper, a word is a sequence of letters, delimited by nonletters. For example, the URL *<A HREF= http://golgi.harvard.edu/biopages/all.html>* contains nine "words" *a, href, http, golgi, harvard, edu, biopages, all,* and *html*. All words are converted to upper case.

Learning a user profile consists of two main steps. First, features are automatically extracted from rated pages, and all rated pages are converted to Boolean vectors containing only those extracted features. Second, learning algorithms are applied to the feature vectors.

## Feature Selection

Not all words that appear in an HTML document are used as features. We use an information-based approach, similar to that used by an early version of the NewsWeeder program (Lang, 1995) to determine which words to use as features. Intuitively, one would like words that occur frequently in pages on the hotlist, but infrequently on pages on the coldlist (or vice versa). This is accomplished by finding the expected information gain *(E(W,S))* (e.g., Quinlan, 1986) that the presence or absence of a word ($W$) gives toward the classification of elements of a set of pages ($S$):

$$E(W,S) = I(S) - [\,p(W = present)I(S_{W=present}) + p(W = absent)I(S_{W=absent})]$$

where

$$I(S) = \sum_{c \in \{hot,cold\}} - p(S_c)\log_2(p(S_c))$$

and $P(W = present)$ is the probability that $W$ is present on a page, and ($S_{W=present}$) is the set of pages that contain at least one occurrence of $W$ and $S_c$ are the pages that belong to class $c$.

Using this approach, we find the set of $k$ most informative words. In the experiments discussed in this paper, we use the 96 most informative words, because previous web page classification experiments with a simple Bayesian classifier resulted in the highest average accuracy over all tested domains for this value of $k$ (Pazzani, Muramatsu, Billsus, 1996). Table 1 shows some of the most informative words obtained from a collection of 140 HTML documents on independent rock bands as an example for features selected by expected information gain.

| | | |
|---|---|---|
| nirvana | suite | lo |
| pop | records | rockin |
| july | jams | songwriting |
| following | today | vocals |
| island | tribute | previous |
| favorite | airplay | noise |

Table 1: Some of the words used as features

Some of these selected words seem to be "music related" and might indeed discriminate well between interesting and uninteresting pages (e.g. *nirvana, pop, songwriting, vocals* etc.). In contrast, some words were selected because they were good discriminators on the training data, although it seems unlikely that they will be useful as part of a generally applicable profile to distinguish between interesting and uninteresting pages (e.g. *today, following, previous*). In addition, the user who rated these pages felt that most of the words he would have selected intuitively as good discriminators were not present in the set of automatically selected features. This problem can be addressed with an initial user defined profile.

## Learning Algorithms

Once the HTML source for a given topic has been converted to positive and negative examples represented as feature vectors, it is possible to run many learning algorithms on the data. In previous experiments (Pazzani, Muramatsu, Billsus, 1996) we have investigated the accuracy of 5 machine learning algorithms (simple Bayesian classifier, nearest neighbor, PEBLS, decision trees (ID3), and neural nets).

In summary, it appeared that decision tree learners as ID3 are not particularly suited to this problem, as one might imagine since they learn simple necessary and sufficient descriptions about category membership. Although one must be careful not to read too much into averaging accuracies across domains, the naive Bayesian classifier had the highest average accuracy (77.1%). In contrast, PEBLS was 75.2%, a neural net with backpropagation was 75.0%, nearest neighbor was 75.0% and ID3 was 70.6% accurate on average.

We have decided to use the naive Bayesian classifier as the default algorithm in Syskill & Webert for a variety of reasons. It is very fast for both learning and predicting. Its learning time is linear in the number of examples and its prediction time is independent of the number of examples. It is trivial to create an incremental version of the naive Bayesian classifier. It provides relatively fine-grained probability estimates that may be used to rank pages in addition to rating them. For example, on the biomedical domain (see experimental evaluation), we used a leave-one-out testing methodology to predict the probability that a user would be interested in a page. The ten pages with the highest probability were all correctly classified as interesting and the ten pages with the lowest probability were all correctly classified as uninteresting. There were 21 pages whose probability of being interesting was above 0.9 and 19 of these were rated as interesting by the user. There were 64 pages whose probability of being interesting was below 0.1 and only 1 was rated as interesting by the user.

## Using Predefined User Profiles

There are two drawbacks of our system that can be addressed by providing a learning algorithm with initial knowledge about a user's interests. First, looking at the features selected by expected information gain revealed that some of the features are irrelevant for discriminating between interesting and uninteresting pages. This occurs most frequently when there is only few training data available. Using lots of irrelevant features makes the learning

task much harder and leads most likely to a lower classification accuracy. Second, the classification accuracy tends to increase very slowly while more training data is becoming available. Since some users might be unwilling to rate lots of pages before the system can give reliable predictions, initial knowledge about the user's interests can be exploited to give accurate predictions even when there are only a few rated pages.

The only way to assess an initial user profile, especially when the training data is sparse, is to elicit it from the user. In Syskill & Webert we are asking the user to provide words that are good indicators for an interesting page and allow him to state as many or few words as he can come up with. In addition, we are asking for words that are good indicators for a page being uninteresting. However, the concept of an "uninteresting webpage" is hard to define, because an "uninteresting webpage" can be anything diverging from the user's interests. Therefore, it might be hard or somewhat unnatural to think of words that are indicators for uninteresting pages, and therefore the user does not have to state any words if he cannot think of any.

Since there might be different levels of how relevant a single word is to the classification of pages, we are also asking the user to provide an estimate of a probability for each word that indicates how likely it is that a page rated as hot (or cold respectively) would contain the word. If the user cannot assess this probability we are using a default value instead. In our current implementation we set this default probability to 0.7, because it is more likely that a feature given by the user correctly discriminates between interesting and uninteresting pages, and therefore a value greater than 0.5 should be used.

The initial user profile is only assessed once for each user and each topic and is then reused in all following sessions. However, a user can redefine his profile if his interests have changed or if he would like to add additional words to the profile.

The initial user profile used in Syskill & Webert consists of a set of probability tables. We associate one probability table with each word stated by the user, where each probability table contains the probabilities for the word appearing (or not appearing) in a page, given that the page was rated hot (or cold respectively). Such a probability table $p(word_i \mid class_j)$ contains 4 probabilities, namely $p(word_i \text{ present} \mid hot)$, $p(word_i \text{ absent} \mid hot)$, $p(word_i \text{ present} \mid cold)$, and $p(word_i \text{ absent} \mid cold)$. If the user provided probability estimates, they become the initial values for the corresponding probability $p(word_i \text{ present} \mid class_j)$ The probability $p(word_i \text{ absent} \mid class_j)$ is set to $1 - p(word_i \text{ present} \mid class_j)$. Since a user would usually not state the same word as being an indicator for hot pages and cold pages, only two probabilities out of the four probabilities are defined by the user. The remaining probabilities can be estimated from the training data.

After an initial user profile has been assessed, it has to be determined to which degree the system should "believe" in the user's estimates. This decision should clearly be correlated to the amount of available training data. While there are initially only a few rated pages available, the system should rely more on the profile given by the user than on hypotheses formed by looking at only a few available rated pages. This is due to the fact, that a few rated pages do not provide enough information to form concept descriptions that are general enough to accurately predict unseen pages. As more training data is becoming available, the system should gradually increase the belief in its own hypotheses and gradually decrease the belief in the initial user profile. The user profile should be gradually revised to form a user profile that fits the training data.

Since an initial user profile is represented as a set of probability tables, the process of revising the user profile consists of gradually updating the given probabilities in a way to make them reflect the seen training examples. A theoretically sound way of doing this is to express the uncertainty in a probability estimate with a conjugate probability distribution. While observing more training data this probability distribution can be updated to reflect both, the changed expected value of the probability, and the increased confidence in the accuracy of the probability estimate.

The probability tables that represent the current profile of the user's interests can be directly used in Syskill & Webert's default classification algorithm, the simple Bayesian classifier.

## Conjugate Priors

*Conjugate priors* are a traditional technique from Bayesian statistics to update probabilities from data (Heckerman, 1995). Using this approach, a probability density function $p(\theta)$ is associated with a random variable $\Theta$ whose value $\theta$ has to be learned. The probability density function $p(\theta)$ is gradually updated to reflect the observed data.

In our system $\Theta$ corresponds to the probability $p(word_i \text{ present} \mid class_j)$, and the observed data corresponds to the events $(word_i \text{ present} \,\&\, class_j)$ and $(word_i \text{ absent} \,\&\, class_j)$. We are uncertain about the value of $\Theta$, and as we observe $word_i$ appearing or not appearing in a document rated as being of $class_j$ we update our probability distribution for $\Theta$. Suppose we observe $(word_i \text{ present})$ in a document rated as $class_j$. Bayes theorem can now be applied to compute the posterior probability distribution $p(\theta \mid word_i \text{ present})$:

$$p(\theta \mid word_i \text{ present}) = c\, p(word_i \text{ present} \mid \theta)\, p(\theta)$$
$$= c\theta\, p(\theta)$$

where $c$ is a normalization constant.

Since $p(word_i\ absent\ |\theta)$ is simply $1 - p(word_i\ present\ |\ \theta)$ we can update $p(\theta\ |\ word_i\ absent)$ equally easily:

$$p(\theta|word_i absent) = c\,p(word_i absent|\theta)\,p(\theta)$$
$$= c(1-\theta)\,p(\theta)$$

These two equations can now be combined to compute the posterior probability distribution $p(\theta\ |\ \alpha, \beta)$ where $\alpha$ is the number of times $(word_i\ present\ \&\ class_j)$ is observed and $\beta$ is the number of times $(word_i\ absent\ \&\ class_j)$ is observed:

$$p(\theta|\alpha,\beta) = c\,\theta^{\alpha}\,(1-\theta)^{\beta}\,p(\theta)$$

A probability distribution given by the equation above is known as a *beta distribution* (since two parameters $\alpha$, and $\beta$ are a sufficient statistic to update the distribution). A beta distribution has the property to be *conjugate*, which means that applying Bayes law to a prior beta distribution results in a posterior distribution, which is also a beta distribution.

The effect of additional observed samples on a prior beta distribution is shown in Figure 3 and Figure 4. Figure 3 shows a beta distribution for the parameters $\alpha = 2$, and $\beta = 2$. Since both events were observed 2 times, both events are assumed to be equally likely and so the graph is centered at 0.5 (the expected value of $\theta$, written as $E(\theta)$, is 0.5). The distribution is only based on a small sample size $(\alpha + \beta = 4)$ and has therefore a high variance.

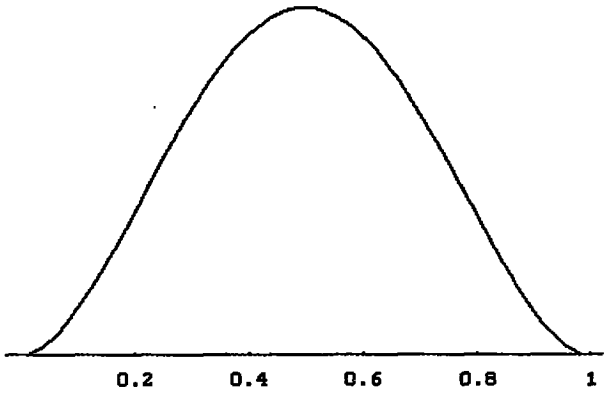


Figure 3: Beta distribution ($\alpha = 2$, $\beta = 2$)

Now, suppose that we observe the event corresponding to $\alpha$ 8 times, while we do not observe any events corresponding to $\beta$. Figure 4 shows the resulting posterior beta distribution. The posterior distribution is significantly shifted to the right and has a smaller variance (the graph is narrower; it has a lower spread of possible values). The graph gets narrower due to the increased sample size $(\alpha + \beta = 12)$. The sample size therefore directly reflects our confidence in the probability to be estimated.

Using beta distributions our current expectation of $\theta$ can be computed easily:

$$E(\theta) = \frac{\alpha}{\alpha+\beta}$$

These properties make the beta distribution a useful distribution for learning.

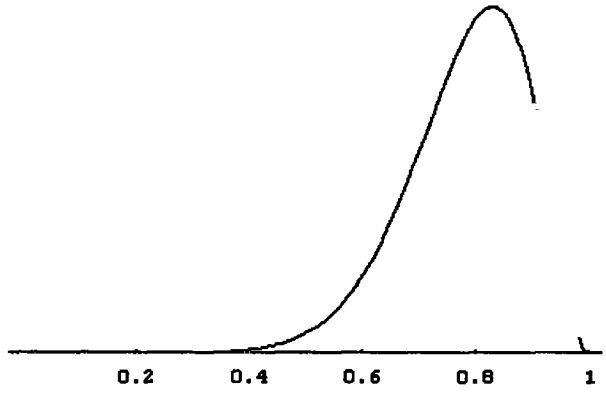


Figure 4: Beta distribution ($\alpha = 10$, $\beta = 2$)

In Syskill & Webert we are using the initial profile given by the user to compute prior beta distributions for the probabilities $p(word_i\ present\ |\ class_j)$. One way to assess a beta distribution is the equivalent-sample-size method (Winkler, 1967). The method is based on the idea to define the equivalent sample size as the number of observations we would have had to have seen starting from complete ignorance in order to have the same confidence in the values of $\theta$. In our current implementation we do not ask the user to provide levels of confidence for each probability estimate he provided. Instead, we are using a default sample size for all initial probability estimates. In the experiments described in this paper we set the equivalent sample size to 50, i.e. we define the "weight" of the user's estimate to be equivalent to observing 50 samples.

The parameters $\alpha$ and $\beta$ can be determined using the equations:

$$\alpha + \beta = \text{ equivalent sample size}$$
$$\frac{\alpha}{\alpha+\beta} = \text{ user's probability estimate}$$

The initial user profile can therefore be used to determine the values of $\alpha$ and $\beta$ for the prior probability distributions. As more examples are observed, $\alpha$ and $\beta$ are gradually increased, thus changing the expected value $E(\theta)$ and the confidence in the estimate.

## Using the Simple Bayesian Classifier

The probability tables that represent the current profile of the user's interests can be directly used in Syskill & Webert's default classification algorithm, the simple Bayesian classifier (SBC). The SBC (Duda & Hart, 1973) is a probabilistic method for classification. It can be used to deter-

mine the probability that an example $i$ belongs to class $C_j$ given feature values of the example. Applied to Syskill & Webert, this means that we are interested in the probability of a page being interesting or uninteresting, given that it contains or does not contain specific words:

$$P(class_j | word_1 \& word_2 \& \ldots \& word_n)$$

where $word_1$ to $word_n$ are Boolean variables that indicate whether a certain word appears or does not appear in the page. Applying Bayes rule reveals that this probability is proportional to:

$$P(class_j) P(word_1 \& word_2 \& \ldots \& word_n | class_j)$$

Under the assumption that words appearing or not appearing in a page are independent events given the class of the page, $P(word_1 \ \& \ word_2 \ \& \ \ldots \ \& \ word_n \mid class_j)$ can be expressed as:

$$\prod_i^n P(word_i | class_j)$$

Therefore, the probability of an example belonging to $class_j$ is proportional to:

$$P(class_j) \prod_i^n P(word_i | class_j)$$

To determine the most likely class of an example, the probability of each class is computed, and the example is assigned to the class with the highest probability. The probabilities used in this computation may be estimated from training data. In Syskill & Webert the class priors $P(class_j)$ are estimated from training data only. In contrast, if the initial user profile contains information about the user's estimate of a probability, the corresponding conditional probability $P(word_i \mid class_j)$ is determined using conjugate priors as described in the previous section and reflects both the initial profile provided by the user and the observed training data.

The assumption of attribute independence is clearly an unrealistic one in the context of text classification. However, the SBC performed well in our experiments, and it also performs well in many domains that contain clear attribute dependence. Domingos and Pazzani (1996) explore the conditions for the optimality of the SBC and conclude that the SBC can even be optimal if the estimated probabilities contain large errors.

## Algorithms to Evaluate Feature Selection

In the previous sections we showed how initial probabilities given by the user can be gradually revised to fit the observed training data. However, updating probabilities is only one part of the user profile revision process. Since it is very likely that the user will not mention all the words that are good discriminators between interesting and uninteresting pages, the question arises whether it is possible to increase the classification accuracy by using automatically selected features in addition to the features (words) provided by the user.

In order to assess the effect the feature selection has on the classification accuracy, we implemented three variations of the SBC.

**SBC-IFeatures.** This is the standard simple Bayesian classifier. It does not make use of features and probabilities provided by the user. All the features are automatically extracted using expected information gain. In the experiments described in this paper we are using 96 features.

**SBC-UFeatures.** This is the simple Bayesian classifier, operating only on the features that the user provided. The probabilities given by the user are not used. All the probabilities are estimated from data only.

**SBC-CFeatures.** In this approach, the simple Bayesian classifier is operating on both, the features provided by the user and the features extracted by expected information gain. The same number of features as in the SBC-IFeatures approach is used, where the $k$ features provided by the user replace the $k$ features that have the lowest information gain. Again, no probabilities provided by the user are used.

In addition, we have experimented with SBC variants that add different numbers of automatically extracted features to the user features (see experimental evaluation).

## Algorithms to Evaluate Conjugate Priors

In order to assess the effect of revising probabilities on the classification accuracy, we have implemented a set of SBC variants that are only using the features provided by the user.

**SBC-User.** This approach updates the probabilities given by the user using conjugate priors. Note, that a user does not have to define a complete profile: usually only the probability for a feature being an indicator for a hot page *or* for being an indicator for a cold page is defined, i.e. only half of the associated probability tables are given by the user, and the other half is estimated from data.

**SBC-Complete.** In order to evaluate the effect of estimating the missing probabilities in the SBC-User approach, we asked users to define a complete user profile. A complete user profile contains two probabilities for every feature: the probability of the feature being an indicator for a hot page, and the probability of a feature being an indicator for a cold page. All provided probabilities are revised using conjugate priors.

| User | Topic | URL of topic's index page | Pages |
|---|---|---|---|
| A | Bands (listening) | http://www.iuma.com/IUMA-2.0/olas/location/USA.html | 57 |
| B | Bands (reading) | http://www.iuma.com/IUMA-2.0/olas/location/USA.html | 154 |
| A | Biomedical | http://golgi.harvard.edu/biopages/medicine.html | 127 |
| A | LYCOS (Biomedical) | *not applicable* / results of a *LYCOS* search on biomedicine | 54 |
| A | Goats | *not applicable* / results of an *Inktomi* search on goats | 70 |

Table 2: Topics used in our experiments

**SBC-Fixed.** This approach is similar to SBC-Complete: a completely defined user profile is used, but probabilities are never changed. In fact, apart from estimating the class probabilities *p(hot)* and *p(cold)* there is no learning involved in this variant. However, it provides a basis to assess whether the SBC-User and SBC-Complete approaches do significantly better than using the user's estimates alone.

Furthermore, we have experimented with SBC variants that add different numbers of automatically extracted features to the user features. The user probabilities are revised using conjugate priors, all the remaining probabilities are estimated from data.

## Experimental Evaluation

To determine the effect of initial user defined profiles on the classification accuracy, we have had 2 users use the Syskill & Webert interface to rate pages and provide the system with initial user profiles. A total of five different profiles were collected (since one user rated pages on four different topics). The topics are summarized in Table 2 together with the total number of pages that have been rated by the user. Two users rated pages on independent rock bands. One (A) listened to an excerpt of songs, and indicated whether the song was liked. Of course, the machine learning algorithms only analyze the HTML source describing the bands and do not analyze associated sounds or pictures. Another user (B) read about the bands (due to the lack of sound output on the computer) and indicated whether he'd be interested in the band.

Syskill & Webert is intended to be used to find unseen pages the user would like. In order to evaluate the effectiveness of the learning algorithms and initial profiles, it is necessary to run experiments to see if Syskill & Webert's prediction agrees with the users' preferences. Therefore we use a subset of the rated pages for training the algorithm and evaluate the effectiveness on the remaining rated pages. For an individual trial of an experiment, we randomly selected $k$ pages to use as a training set, and reserved the remainder of the data as a test set. From the training set, we found the 96 most informative features, and then recoded the training set as feature vectors to be used by the learning algorithm. Next, the test data was converted to feature vectors using the same features used to convert the training set. Finally, the learned user preferences were used to determine whether pages in the test set would interest the user. For each trial, we recorded the accuracy of the learned preferences (i.e., the percent of test examples for which the learned preferences agreed with the user's interest). We ran 50 paired trials of each algorithm. The learning curves in this section show the average accuracy of each algorithm as a function of the number of training examples.

In our first set of experiments we compared the SBC-IFeatures, SBC-UFeatures, and SBC-CFeatures approaches in order to evaluate the effect of the features given by the user. Note, that no approach in this set of experiments makes use of the probabilities provided by the user. We also compare these results to the baseline accuracy (always guessing *cold* which is the most frequent class in all domains). Figures 5 to 9 show the learning curves for the five domains.

Tables 3 to 6 show the user profiles that were used for the five domains (only four profiles were defined, because the LYCOS and biomedical domains are using the same initial user profile). The two numbers next to each word are the probabilities *p(word present | hot)* and *p(word present | cold)*.

The results show the same pattern of performance for the two bands domains, the biomedicine and the goats domain. While SBC-IFeatures has the lowest classification accuracy, SBC-CFeatures performs consistently better than SBC-IFeatures, and SBC-UFeatures achieves the highest classification accuracy in all four domains. The features selected by the users seem to discriminate very well between interesting and uninteresting pages and outperform the approaches that are based on features selected by expected information gain. In addition, we ran experiments that gradually augmented the features given by the user with features extracted by expected information gain. Even when we added only two additional features (the two highest ranked) the classification accuracy was decreased. The

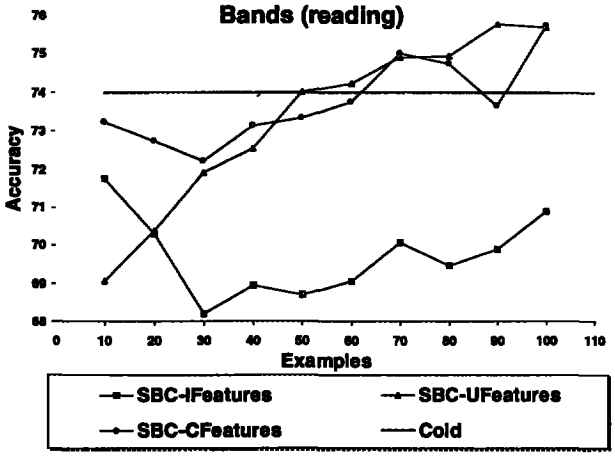


Figure 5: Test accuracy for the bands (reading) domain

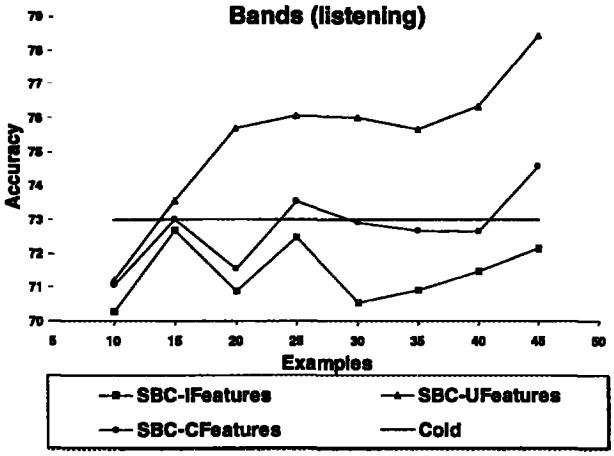


Figure 6: Test accuracy for the bands (listening) domain

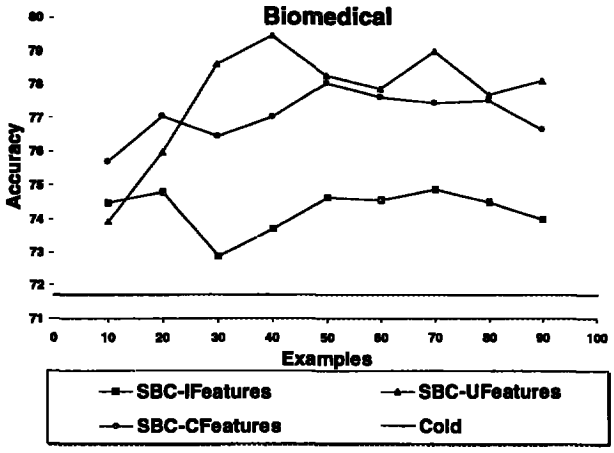


Figure 7: Test accuracy for the biomedical domain

classification accuracy decreased in an almost linear relation to the number of added features, i.e. the more features

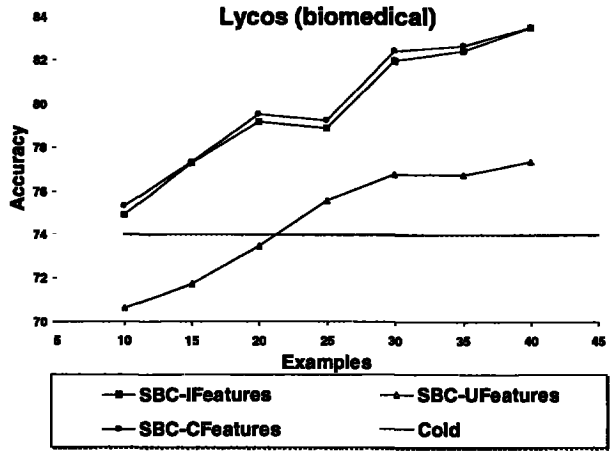


Figure 8: Test accuracy for the LYCOS domain

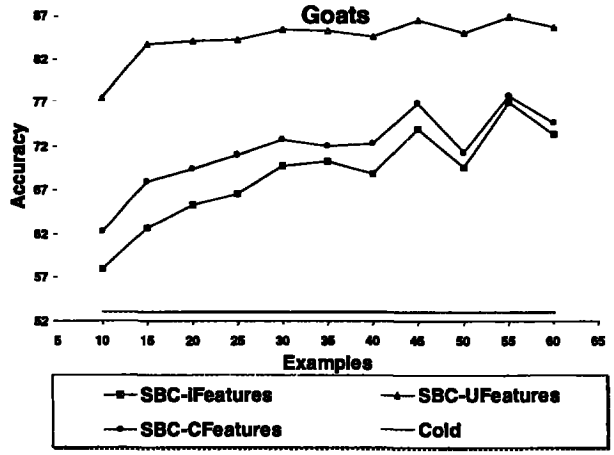


Figure 9: Test accuracy for the goats domain

we added, the lower the classification accuracy.

When we tested the algorithms on the LYCOS domain the results turned out to be very different. The LYCOS domain was generated by a LYCOS query that Syskill & Webert constructed using the profile learned from the biomedical domain. The idea is to let Syskill & Webert rate links returned by LYCOS using the same profile that the query was generated from. Therefore we used the same initial user profile as in the biomedical domain. The user ratings of only the pages retrieved by LYCOS were used to update the initial profile. In this experiment *SBC-UFeatures* performed worse than the other two approaches. The features selected by the user did not seem to be good discriminators for the pages returned by LYCOS. However, the *SBC-CFeatures* approach was not affected very much by the irrelevant user profile, and performed as good as *SBC-IFeatures*.

In summary, it appears that the *SBC-UFeatures* algorithm outperforms the other approaches in cases where the

initial user profile provides features that discriminate well between interesting and uninteresting pages. However, if the user profile is wrong or contains features that do not appear frequently in the pages to be rated, the *SBC-CFeatures* algorithm seems not to be affected negatively to a significant extent.

| | | |
|---|---|---|
| *guitar .9 .5* | *guitars .9 .5* | *acoustic .9 .5* |
| *independent .9 .1* | *nirvana .9 .1* | *pumpkins .9 .2* |
| *alternative .9 .1* | *college .9 .3* | *folk .9 .5* |
| *synthesizer .1 .6* | *keyboard .1 .6* | *dance .1 .6* |

Table 3: User profile for the bands (reading) domain

| | | |
|---|---|---|
| *acoustic .7 .1* | *flute .7 .1* | *she .8 .3* |
| *her .8 .3* | *punk .6 .2* | *jazz .7 .2* |
| *bass .7 .5* | *south .1 .1* | *cassette .3 .2* |
| *drugs .3 .1* | | |

Table 4: User profile for the bands (listening) domain

| | | |
|---|---|---|
| *grants .7 .2* | *database .8 .1* | *genome .6 .3* |
| *molecular .6 .1* | *protein .5 .2* | *prediction .9 .1* |
| *classification. 9 1* | *structure .6 .2* | *function .6 .1* |
| *webmaster .05 .1* | *com .1 .4* | |

Table 5: User profile for biomedical and LYCOS domains

| | | |
|---|---|---|
| *dairy .7 .3* | *pygmy .7 .3* | *angora .7 .3* |
| *cashmere .7 .3* | *milk .7 .3* | *doe .7 .3* |
| *farm .7 .3* | *buck .7 .3* | *wether .7 .3* |
| *sheep .7 .3* | *animals .7 .3* | *hay .7 .3* |
| *wine .3 .7* | *hill .3 .7* | *blows .3 .7* |

Table 6: User profile for the goats domain

We also noticed that the classification accuracy on the training set of all the algorithms that make use of the features provided by the user was significantly lower than the training accuracy of algorithms that only use features selected by expected information gain. We interpret this as an indicator that the features given by the user might be impossible to extract automatically from the training data.

In the following set of experiments we compared the SBC-User, SBC-Complete and SBC-Fixed algorithms to the SBC-UFeatures approach in order to evaluate the effect of the probabilities provided by the user. In addition, these experiments show the effect of probability revision using conjugate priors on the classification accuracy. Figures 10 to 14 show the learning curves for the five domains. Again, the results show the same pattern of performance for the two bands domains, the biomedicine and the goats domain. SBC-UFeatures (the best performing approach in the previous set of experiments) is the only algorithm in this set of experiments that does not make use of the probabilities provided by the user. Thus, on average its classification accuracy is significantly below the other approaches. As expected, the accuracy of the SBC-Fixed approach stays approximately constant, i.e. it is independent of the number of training examples. The SBC-User and SBC-Complete algorithms perform about equally well. Interestingly, both approaches exhibit the shape of an ascending learning curve that drifts away from the SBC-Fixed accuracy, which means that the provided probabilities can in fact be revised in a way that increases the classification accuracy. Although the SBC-User approach only uses the probabilities *p(word present | hot)* and estimates the probabilities *p(word present | cold)* from data, its performance is on average not worse than that of SBC-Complete. In some cases SBC-User even outperforms SBC-Complete (e. g. in the goats domain). This result suggests that it is sufficient to specify one probability for each feature. Note that all the approaches that make use of the user's probabilities perform significantly better than the SBC-UFeatures approach. SBC-UFeatures in turn performs better than all approaches that do not make use of the features provided by the user, such as SBC-IFeatures and all the standard machine learning algorithms that we tested in previous experiments (Pazzani, Muramatsu, Billsus, 1996). The difference between the accuracy of the algorithms that make use of the probabilities provided by the user and SBC-UFeatures is most significant when there are only few training examples. This result suggests, that initially defined probabilities are in fact a good way to increase the classification accuracy when the training data is sparse.

Similar to our experiments on the effect of feature extraction, we ran experiments that gradually augmented the features given by the user with features extracted by expected information gain. We used conjugate priors to re-

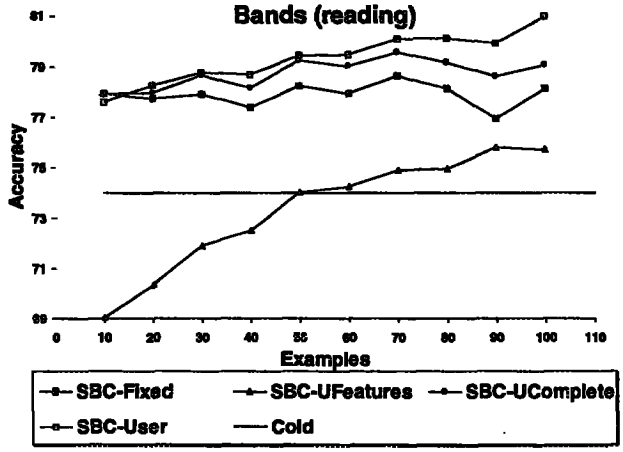


Figure 10: Test accuracy for the bands (reading) domain

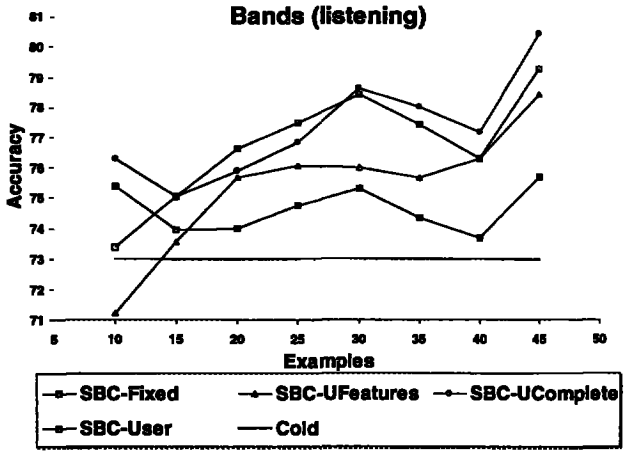


Figure 11: Test accuracy for the bands (listening) domain

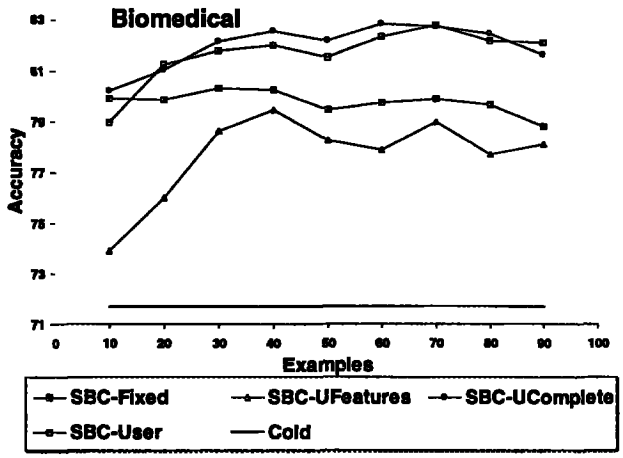


Figure 12: Test accuracy for the biomedical domain

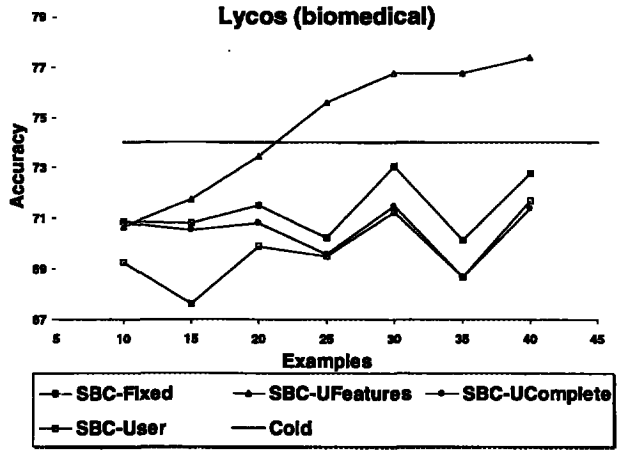


Figure 13: Test accuracy for the LYCOS domain

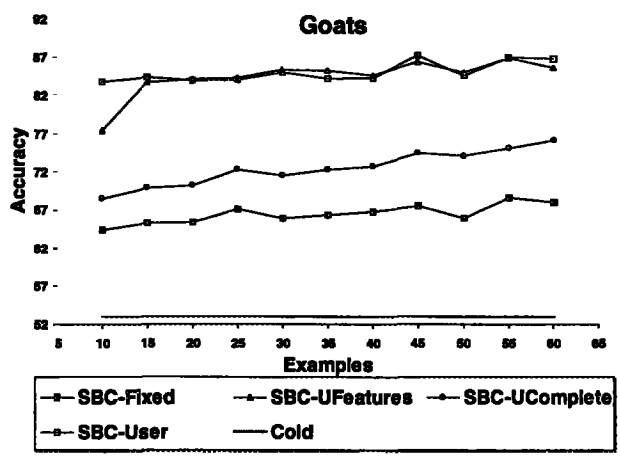


Figure 14: Test accuracy for the goats domain

vise the probabilities given by the user and estimated the remaining probabilities from data. Again, in no case did automatically extracted features added to the features provided by the user increase the classification accuracy. As in our feature extraction experiments, the results for the LYCOS domain differed significantly from the other four domains. Since the features selected by the user do not seem to discriminate well between pages returned by LYCOS, the classification accuracy cannot be improved by providing user defined probabilities.

All in all, it seems as if initially provided probabilities can significantly increase the classification accuracy in cases where the given features discriminate well between interesting and uninteresting pages. The given probabilities can be revised in order to reflect the available data, and the classification accuracy increases.

## Future Work

The experiments presented in this paper have shown that the selection of features is paramount to the achievable classification accuracy. We are planning to use linguistic and hierarchical knowledge in addition to expected information gain in order to extract features that, when added to the features given by the user, increase the classification accuracy.

Currently, words in the pages are used as features without any knowledge of the relationship between words such as "protein" and "proteins." Linguistic routines such as stemming (i.e., finding the root forms of words) may in addition be helpful, since a smaller number of more informative features would be extracted. Semantic knowledge, e.g. the relationship between "pigs" and "swine", may also prove useful. Similarly, knowing that "gif," "jpg" and "jpeg" are all extensions of graphic files would facilitate Syskill & Webert learning that a user has a preference

for (or against) pages with in-line graphics.

The results of the approaches described in this paper are highly dependent on the profiles provided by the users. Clearly, more data from different users has to be collected, to confirm that our results have general validity.

## Conclusions

We have introduced an agent that collects user evaluations of the interestingness of pages on the World Wide Web. We have shown that a user profile may be learned from this information and that this user profile can be used to determine what other pages might interest the user.

A predefined user profile can significantly increase the classification accuracy on previously unseen pages. The best results were achieved when only the features that were provided by the user were used. Using this approach, the classification accuracy can still be increased by revising the user defined probabilities using conjugate priors.

Since the classification accuracy on the training set significantly decreased for all tested algorithms that made use of a user defined profile, we think that the features given by the user cannot be extracted automatically from the training set by statistical means alone.

## Acknowledgments

The research reported here was supported in part by NSF grant IRI-9310413 and ARPA grant F49620-92-J-0430 monitored by AFOSR.

## References

Domingos, P., and Pazzani, M. 1996. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. To appear in the *Proceedings of the International Conference on Machine Learning*.

Duda, R. O., and Hart, P. E., 1973. *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons.

Heckerman, D. 1995. A Tutorial on Learning with Bayesian Networks, Technical Report, MSR-TR-95-06, Microsoft Corporation.

Lang, K. 1995. NewsWeeder: Learning to filter Netnews. In Proceedings of the Twelfth International Conference on Machine Learning, 331-339. Lake Tahoe, Calif.: Morgan Kaufmann.

Mauldin, M. L., and Leavitt, J. R. 1994. Web Agent Related Research at the Center for Machine Translation. Proceedings of the ACM Special Interest Group on Networked Information Discovery and Retrieval (SIGNIDR-94).

Pazzani, M., Muramatsu J., and Billsus, D. 1996. Syskill & Webert: Identifying interesting web sites. To appear in the *Proceedings of the National Conference on Artificial Intelligence*, Portland, OR.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning*, 1, 81-106.

Winkler, R. 1967. The assessment of prior distributions in Bayesian analysis. *American Statistical Association Journal*, 62:776-800.