Learning Probabilistic User Models

Daniel Billsus and Michael Pazzani
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425, USA
{dbillsus, pazzani}@ics.uci.edu

Abstract
We describe two applications that use rated text documents to induce a model of the user's interests. Based on our experiments with these applications we propose the use of a probabilistic learning algorithm, the Simple Bayesian Classifier (SBC), for user modeling tasks. We discuss the advantages and disadvantages of the SBC and present a novel extension to this algorithm that is specifically geared towards improving predictive accuracy for datasets typically encountered in user modeling and information filtering tasks. Results from an empirical study demonstrate the effectiveness of our approach.

1. Introduction

The acquisition of user models for interactive computer systems has been addressed in many different ways. Most approaches discussed in the user modeling literature (e.g. work reported in Kobsa, Wahlster 1989) are based on predefined knowledge about users or groups of users, e.g. in form of stereotypes or inference rules. For tasks that require a detailed model about the user's preferences, likes and dislikes, these acquisition techniques are often not flexible enough. In this paper we propose the use of a probabilistic machine learning algorithm for the induction of models of users' interests and preferences.

Learning models of users' interests with respect to arbitrary topics appears to be a research area of steadily increasing importance in the dawning information age, as it allows intelligent computer systems to adapt to users' information needs in a personalized way. The advent of the Internet and the World Wide Web, as well as increased interest in intelligent agents that aim to fulfill tasks for users based on their interests and information needs, has led to a surge of work in machine learning applied to Information Retrieval (IR) tasks. We see this work as being directly related to the automated induction of user models. Specifically, we are looking at the following learning task: given a set of text documents rated by the user with respect to her interests, we induce a model of the user's interests, which can then be used by a software agent to adapt to the user's information needs, and search for, filter or suggest content the user might be interested in. In this paper we describe two applications currently in use at UCI that induce user models. We motivate the use of a probabilistic learning algorithm, the Simple Bayesian Classifier (SBC), and discuss its advantages and disadvantages. We present a novel extension to the SBC that is specifically geared towards improving predictive accuracy for datasets typically encountered in user modeling and information filtering tasks. Results from an empirical study demonstrate the effectiveness of our approach.

2. Example Applications

Learning a model of a user's interests from rated text documents can be seen as a general knowledge acquisition technique that can be applied to different kinds of information systems. The following two sections briefly discuss two applications, *"Syskill & Webert"*, and the *"UCI GrantLearner"*, that are based on the probabilistic learning algorithm proposed in this paper.

2.1 Syskill & Webert

Syskill & Webert is an intelligent agent that learns a model of a user's interest in a specific topic, based on rated World Wide Web pages. The system is implemented as a web browser extension and provides an interface that allows users to rate web pages as either "hot" or "cold". A typical Syskill & Webert session proceeds in the following way: First, the user chooses an arbitrary topic of interest (say e.g. *"machine learning",* or *"user modeling"*). The user can then browse or search the web and rate any page that she finds interesting with respect to the topic as hot, and any unrelated or uninteresting page as cold. As soon as Syskill & Webert has collected a minimum number of ratings (currently set at 10 pages), it can use the rated web pages to induce a probabilistic model of the user's interests. This model can then be used in two different ways: First, Syskill & Webert can annotate the links on any web page with its prediction of the probability that the user will be interested in the corresponding page. Second, Syskill & Webert can use the learned model to automatically construct a search engine query, send the query to the search engine, collect the results, and finally evaluate them with respect to the user model. In essence, the system aims to filter the enormous amount of information available on the World Wide Web with respect to a personalized model of the user's interests.

2.2 UCI GrantLearner

The UCI GrantLearner uses the same user model learning technique as Syskill & Webert. The goal of the UCI GrantLearner project is to provide a system that learns to distinguish between interesting and uninteresting funding opportunities, based on users' ratings of these descriptions. Our current implementation of the GrantLearner is a standalone application that connects to a regularly updated grant database maintained at UCI and provides a user interface to browse this database. The system can display a list of all grant subjects available in the database, and the user can retrieve corresponding full text descriptions that provide information about the grant's subject area, funds, eligibility and deadlines. Once the user has rated a number of grants, GrantLearner uses the grant descriptions to learn a model of the user's interests. The learned model is then used to generate a list of funding opportunities sorted according to GrantLearner's relevance prediction. First experiments and feedback from different users suggest that GrantLearner is a helpful tool to locate potentially interesting funding opportunities.

3. The Simple Bayesian Classifier

In previous work (Pazzani, Muramatsu, Billsus, 1996) we compared standard machine learning algorithms, including the SBC, artificial neural networks, decision trees and nearest neighbor approaches, on text classification tasks. Using rated text documents as training examples, we induced Boolean concepts that distinguish between interesting and uninteresting documents. We found that the SBC achieved on average the highest classification accuracy. In this section we give a brief outline of the way we apply the SBC to this task.

Since the SBC requires that training examples be represented as a set of feature vectors, we convert text documents to Boolean feature vectors, where each feature indicates whether a particular word is present or absent in a particular text document. Not all words that appear in a text 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 to select words that occur frequently in pages rated as interesting, but infrequently on pages rated as uninteresting. This is accomplished by finding the expected information gain (e.g. Quinlan 1986) that the presence or absence of a word gives toward the classification of elements of a set of documents. Using this approach, we find the k most informative words of the current set of rated pages. In the experiments discussed in this paper, we use the 96 most informative words, because previous experiments with the SBC (Pazzani, Muramatsu, Billsus, 1996) resulted in the highest average accuracy for this value of k. We convert all text documents in the training set to Boolean feature vectors, which can then be used as training examples for the SBC. The SBC (Duda & Hart, 1973) is a probabilistic method for classification. It is used to determine the probability that an example belongs to class C_j given feature values of the example. Applied to our text classification task, this means that we are interested in the probability of a document being interesting or uninteresting, given that it contains or does not contain specific words:

$$P(class_j \mid word_1 \,\&\, word_2 \,\&\, ... \,\&\, word_k)$$

where $word_1$ to $word_k$ are Boolean features that indicate whether a certain word appears or does not appear in the document. Under the assumption that words appearing or not appearing in a document are independent events given the class of the document, the probability of an example belonging to $class_j$ is proportional to:

$$P(class_j)\prod_i^k P(word_i \mid class_j)$$

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. We provide more details on how to obtain these estimates in section 4. 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.

4. Improving the SBC

Apart from its good performance on text classification tasks, the SBC has several additional advantages that make it a useful technique for user model induction. 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. The SBC can be cast as an on-line algorithm, which is particularly valuable for tasks such as user modeling, where additional knowledge about the user has to be added to the model incrementally. In addition, rather than computing binary classifications, the SBC can be used to obtain probabilities that allow to rank order items that might be of interest to the user.

However, an obvious problem of the SBC in the user modeling context is that its predictions are based on probabilities estimated from very small datasets. Since the SBC multiplies probability estimates to obtain joint probabilities, one must be careful that no probability estimate is equal to zero. This would result in a joint probability of zero, and thus completely rule out one class on the basis of only one feature. Kohavi et. al. (1997) compare 8 different estimation methods that prevent probability estimates from being zero and vary in the strength of their estimation bias. For example, the commonly used probability correction based on Laplace's law of succession (Good, 1965) has the following form for two-class problems: $p = (N + 1) / (n + 2)$, where N is the number of matches, i.e. in our context the number of examples from one class that have a certain attribute value, and n is the overall number of examples. If probabilities are estimated from small datasets, the Laplace correction can be interpreted as a relatively strong bias towards an uninformed prior (0.5). A different method to prevent probabilities from being zero is to replace probability estimates that are zero with a small constant ε, in the experiments reported here set to 0.001. Note that these two approaches differ significantly with respect to their estimation bias, i.e. probabilities are biased away from zero by different amounts.

As our experiments suggest (see section 5), the applied estimation bias affects classification accuracy significantly. We implemented all the different techniques described in (Kohavi et. al., 1997), but noticed that there is not one single approach that performs well throughout our whole set of different text classification problems. Essentially, the tested methods differ in the strength of their estimation bias. In this paper we only report results for Laplace correction, and the ε approach as described above. These approaches can be seen as representatives for techniques with different estimation biases, and they were among the top performers for different domains in our experiments. Our experiments indicate that there are some domains where a strong estimation bias is very helpful, while it can significantly hurt performance in other domains. These observations suggest that a number of questions need to be addressed. What is it about our data at hand that causes the estimation bias to affect classification accuracy so strongly? Is it possible to analyze the data and decide which estimation bias to apply based on some characteristics of the data?

The estimation bias plays a significant role for our class of learning problem due to several reasons. First, the datasets we are looking at are relatively small (on the order of about 30 to 100 examples), which is a reasonable assumption for user modeling tasks in general, since one would not want the user to train a system for an extended period of time before any pay-off can be observed. Due to the small size of the datasets, Laplace corrected probabilities differ significantly from probabilities estimated with methods that apply a weaker bias, such as replacing zeros with a small constant. In addition, low frequency counts or zero counts for attribute values might occur more often in our text classification approach than in other domains. One reason for this is that we select more features than we have examples in our training set. Thus, it is likely that the frequency counts for many of the chosen features will be small or even zero. Furthermore, features are selected according to expected information gain. In most cases this results in features that are indicators for interesting pages, because what a user considers to be interesting is usually much better defined than what she considers to be uninteresting. Therefore, certain words tend to appear more frequently among interesting pages, which results in high information gain values for these words. This effect causes a lot, oftentimes nearly all of the frequency counts for the "uninteresting" class to be zero, and hence it becomes increasingly important to which extent these probabilities are biased away from zero. The reason why a strong estimation bias is beneficial for some domains, but hurts performance on others, seems to be linked to the "quality" of extracted features. Intuitively, we would like to apply a bias towards an uninformed prior of 0.5 in cases where features do not discriminate well between classes. Poor discriminatory value can be caused by noise, i.e. the feature was selected because it appeared coincidentally a bit more frequently in one class than in the other. Similarly, this effect occurs in cases where a certain feature value occurs about equally often in all classes. On the other hand, we would like to keep probability estimates unbiased for cases where features tend to discriminate well between classes, because this can be seen as an indicator that these features carry more information, which should remain unchanged.

The approach we have taken to address this problem is to decide which estimation bias to use on a per feature basis, rather than applying the same bias to all probability estimates. In our current implementation, we treat this as a binary decision per feature, and base our decision whether we should apply a strong or a weak bias to the estimate on the expected information gain of the feature. Our current approach is rather simple: if the expected information gain of a feature is below a predefined threshold t (0.1 in the experiments reported here), we use Laplace correction for the probability estimates of the two conditional probabilities associated with the feature. On the other hand, if the expected information gain is greater than t, we replace zero probabilities with a small constant value ε (0.001). Using this approach, we take different characteristics of the datasets into account in order to improve the probability estimation.

5. Experiments and Results

The algorithms described in this paper were tested in an empirical study based on datasets collected from different users who trained our "Syskill & Webert" web page classification system on different topics they were interested in. 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 by used by the SBC to learn a user model. Next, the test data was converted to feature vectors using the features found informative on 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 48 trials of each algorithm. Figure 1 shows the average accuracy of each algorithm as a function of the number of training examples for four domains. The algorithms compared are "Epsilon" (i.e. ε corrected probabilities), "Laplace" (Laplace corrected probabilities), "Information Gain" (our feature based estimation approach), and "BaseRate" which is the accuracy obtained from assigning the most frequently occurring class to all test examples.

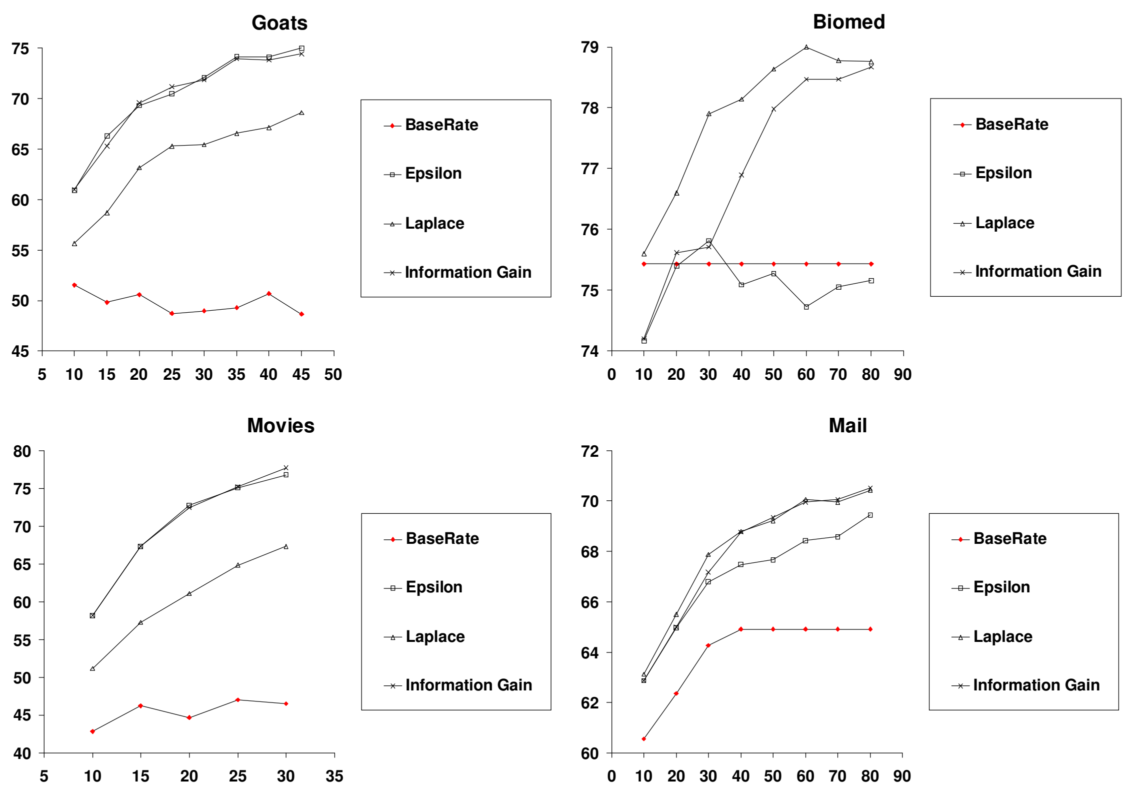

Figure 1: Experimental Results

Note that "Laplace" achieves a much higher test accuracy than "Epsilon" in the *biomed* and *mail* domains. On the other hand, "Epsilon" performs better than "Laplace" in the *goats* and *movies* domains. The differences between these two static methods range between two and ten percent. The information gain approach appears to be a solution to this problem. In all cases this approach is either better or very close to the best static method, and thus has the highest accuracy on average over all domains. We observed similar results in five other domains, which we do not report here due to space limitations.

6. Future Work

The information gain based estimation approach has thus far only been tested on our text classification domains. We suspect that the effect of the algorithm will not be as significant on larger machine learning domains, but it will still be interesting to compare our approach to other estimation methods over a broad range of machine learning problems. Several other methods are potentially useful when learning from small sets of text documents. We will experiment with feature selection methods facilitated by semantic knowledge (e.g. word hierarchies), to restrict the selection of features to words that are semantically related to the topic of interest. In addition, we will perform further experiments with additional knowledge directly obtained from the user (e.g. in form of a more detailed rating scale or features explicitly provided by the user). Experiments reported in (Billsus, Pazzani, 1996) suggest that explicit user knowledge can substantially increase predictive accuracy.

7. Conclusions and Summary

We have motivated the use of a probabilistic learning algorithm, the Simple Bayesian Classifier (SBC), for user model induction based on rated text documents. We have discussed advantages and disadvantages of the SBC, and proposed an information gain based method for probability estimation from data. An experimental study indicated that this approach is superior to commonly used static estimation methods. Overall, the SBC appears to be a useful tool for the induction of user preferences from small sets of rated text documents.

References

Billsus, D. & Pazzani, M. (1996) Revising User Profiles: The Search for Interesting Web Sites. In *Proceedings of the Third International Workshop on Multistrategy Learning*, AAAI Press.

Domingos, P., and Pazzani, M. (1996) Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. *Proceedings of the International Conference on Machine Learning*. Bari, Italy.

Duda, R. & Hart, P. (1973) *Pattern classification and scene analysis.* New York: John Wiley & Sons.

Good, I. (1965) *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. M.I.T. Press.

Kobsa, A., Wahlster, W. (eds.), (1989) *User Models in Dialog Systems* New York: Springer 1989

Kohavi, R., Becker, B., and Sommerfield, D. (1997) Improving Simple Bayes, *ECML-97 Technical Report* (ftp://starry.stanford.edu/pub/ronnyk/impSBC.ps.Z).

Lang, K. (1995) NewsWeeder: Learning to filter news. *Proceedings of the Twelfth International Conference on Machine Learning*. Lake Tahoe, CA.

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

Quinlan, J.R. (1986) Induction of decision trees. Machine Learning, 1, 81–106.