# A Framework for Collaborative, Content-Based and Demographic Filtering

Michael J. Pazzani
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697
pazzani@ics.uci.edu
phone: (714) 824-5888
fax (714) 824-4056
http://www.ics.uci.edu/~pazzani/

**Abstract**

*We discuss learning a profile of user interests for recommending information sources such as Web pages or news articles. We describe the types of information available to determine whether to recommend a particular page to a particular user. This information includes the content of the page, the ratings of the user on other pages and the contents of these pages, the ratings given to that page by other users and the ratings of these other users on other pages and demographic information about users. We describe how each type of information may be used individually and then discuss an approach to combining recommendations from multiple sources. We illustrate each approach and the combined approach in the context of recommending restaurants.*

## 1 Introduction

Users are constantly confronted with situations in which they have many options to choose from and need assistance exploring or winnowing down the possibilities. Internet Search Engines commonly find thousands of potentially relevant sites. Each year journals and conference proceedings report on thousands of research results of potential interest. There are hundreds of articles in each newspaper. Someone trying to escape this volume of information is confronted with selecting from hundreds of restaurants and dozens of television shows and movies.

Intelligent agents (Maes, 1994) have been proposed as a means of sorting through potentially relevant information and making recommendations customized to the individual user. Intelligent agents collect user ratings and create a profile of the user. Users may explicitly give ratings or they may be inferred implicitly from the user's actions. For example, in NewsWeeder (Lang, 1995) users rate Internet news articles on a five-point scale. In Syskill & Webert (Pazzani & Billsus, 1997) users click on a thumbs up symbol when visiting a web site they like and a thumbs down symbol when visiting a web site they don't like (see Figure 1). The agent, Letizia (Leiberman, 1995), infers a user's ratings for web pages by a set of heuristics (e.g., visiting a page for a short period of time is a sign that the page wasn't liked while saving the page indicates it was liked).


**Figure 1.** The Syskill & Webert interface for collecting user feedback.

When some feedback has been collected from the user, an intelligent agent can make recommendations for the user. Two basic approaches have emerged for making recommendations: content-based filtering and collaborative filtering. Content-based filtering analyzes the content of information sources (e.g., the HTML source of Web pages) that have been rated to create a profile of the user's interests in terms of regularities in the content of the information that was rated highly. This profile may be used to rate other unseen information sources or to construct a query of a search engine. Collaborative approaches find and recommend information sources for an individual user that have been rated highly by other users who have a pattern of ratings similar to that of the user. In this paper, we also consider making recommendations based upon demographic information concerning the types of users that have rated particular web pages highly. We explore the advantages and disadvantages of alternative approaches to making recommendations and we argue that the strengths of the different approaches are complementary. We explore approaches to combining recommendations from multiple approaches.

To make this paper more concrete, we present data and results from a group of 44 users of Syskill & Webert. These users were students at the University of California, Irvine. The users all provided ratings for 58 Web pages that describe restaurants in Orange County, California. Figure 1 shows an example of one such restaurant description. The descriptions typically contain a short overview of the restaurant, its cuisine, atmosphere, location, and a detailed menu of the restaurant. The users were asked to indicate whether they would like to eat at the restaurant described by the web page. Approximately 53.4% of the ratings were positive. The demographic information used in this study is obtained from the home page of the user.

In this paper, we will present a variety of approaches that use this data to recommend web pages. The approaches will be trained on a subset of each user's ratings. From this subset, they will recommend three restaurants. The approaches will be evaluated on the precision of the top three recommendations. A value of 66.7% indicates that two of the three restaurants selected were ones that the user would like. We show that approaches that combine multiple sources of information have higher precision than approaches based upon a single type of information.

## 2. Learning User Profiles

In this section, we discuss approaches to learning user profiles. Each approach uses a different type of information and has a different representation of a user profile.

### 2.1 Collaborative Recommendations

Collaborative filtering (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994; Shardanand & Maes, 1995) is an approach to making recommendations by finding correlations among users of a recommendation system. It presents a uniform approach to finding items of potential interest (i.e., items not seen by the current user which have been rated by other users) and predicting the rating that the current users would give to an item. To see how such a prediction could be made, consider the example in Table 1. This gives the ratings of 5 restaurants by 5 users. A "+" indicates that the user liked the description of the restaurant and a "–" indicates that the user did not like the restaurant.

**Table 1. Ratings of five users of five restaurants.**

| | Karen | Lynn | Chris | Mike | Jill |
|---|---|---|---|---|---|
| Kitima | – | + | + | + | – |
| Marco Polo | + | + | + | + | + |
| Spiga | + | – | + | – | + |
| Thai Touch | – | + | – | + | – |
| Dolce | + | – | + | – | ? |

To predict the rating that Jill would give to Dolce, we can look for users that have a similar pattern of ratings with Jill. In this case, Karen and Jill have identical tastes and one might want to predict that Jill would like Dolce because Karen does. A more general approach would be to find the degree of correlation between Jill and other users. Rather than relying on just the most similar user, a weighted average of the recommendations of several users can be found. The weight given to a user's rating would be found by degree of correlation between the two users. In the most general case, the rating could also be a continuous number rather than just +1 and –1. The Pearson $r$ is a measure of correlation that can be used in these circumstances. Let $R_{i,j}$ be the rating of user $i$ on document $j$. Then the correlation between user $x$ and user $y$ is given by:

$$r(x,y)=\frac{\sum_{d\in documents}\left(R_{x,d}-\overline{R_x}\right)\left(R_{y,d}-\overline{R_y}\right)}{\sqrt{\sum_{d\in documents}\left(R_{x,d}-\overline{R_x}\right)^2\sum_{d\in documents}\left(R_{y,d}-\overline{R_y}\right)^2}}$$

Where $\overline{R_x}$ is the mean value of ratings by user $x$.

In the above example, the correlation between Jill and Karen is 1.0, between Jill and Lynn is –0.577, between Jill and Chris is 0.577, and between Jill and Mike is –0.577. Therefore, the weight average of the product of each user's rating for Dolce and the correlation between Jill and that user is 0.682. A collaborative algorithm would predict that Jill would like Dolce based upon the recommendations of these other users. Note that in part this recommendation makes use of the fact that Jill and Mike have nearly opposite tastes and that Mike doesn't like Dolce.

We conducted an experiment in which we randomly deleted half of each user's ratings. Then, for each user, we used collaborative filtering to find the three restaurants (whose rating had been deleted) with the highest recommended rating. We compared the predicted rating of these three restaurants with the actual rating. We repeated this process of randomly deleting ratings 20 times for each user. On average, 67.9% of the restaurants in the top three restaurants recommended via this collaborative process were actually liked by the user. This average is the result of making 24640 predictions (28 ratings, 20 times for each of 44 users).

Although collaborative filtering is most commonly used to find correlations among user rating objects, it may also be used to find collaborations among the objects rated. For example, there is a perfect correlation between the ratings of Dolce and Spiga in Table 1. As a consequence, one might predict that Jill would like Dolce given that Jill likes Spiga. Similarly, this may be generalized by finding the correlations between restaurants (again using Pearson r) and making predictions based upon the weighted average of ratings for other restaurants. Once again, taking the weighted average of all restaurants in Table 1 would yield the result that Jill would like Dolce.

We repeated the experiment described above using correlations among restaurants as the basis of predictions. Under these conditions, 59.8% of the restaurants in the top three restaurants were actually liked by the user. Although basing recommendations on correlations among restaurants does not yield as high a precision as correlations among users in this problem, we shall show in Section 3 that they may be combined with other sources of information to provide a better overall recommendation.

### 2.2 Content-Based Recommendations

Content-based methods make recommendations by analyzing the description of the items that have been rated by the user and the description of items to be recommended. A variety of algorithms have been proposed for analyzing the content of text documents and finding regularities in this content that can serve as the basis for making recommendations. Many approaches are a specialized versions of classification learners, in which the goal is to learn a function that predicts which class a document belongs to (i.e., either liked or not-liked). Other algorithms would treat this as a regression problem in which the goal is to learn a function that predicts a numeric value (i.e., the rating of the document). There are two important subproblems in designing a content-based filtering system. The first is finding a representation of documents. The second is to create a profile that allows for unseen documents to be recommended.

All of the content-based approaches represent documents by the "important" words in the documents. For example, Fab (Balabanovic, 1997) represents documents in terms of the 100 words with the highest TF-IDF weights (Salton, 1989), i.e., the words that occur more frequently in those documents than they do on average. Syskill & Webert (Pazzani & Billsus, 1997) represents documents by the 128 most informative words, i.e., the words that are more associated with one class of documents than another. Table 2 shows an example with 5 restaurants and 5 words that appear in descriptions of the restaurants. The ratings of Jill on these pages are also shown in this table.

**Table 2. The words contained in the description of 5 restaurants together with the ratings of a user for those restaurants.**

| | noodle | shrimp | basil | exotic | salmon | Jill |
|---|---|---|---|---|---|---|
| Kitima | Y | Y | Y | Y | Y | – |
| Marco Polo | | Y | Y | | | + |
| Spiga | Y | | Y | | | + |
| Thai Touch | Y | Y | | Y | | – |
| Dolce | | Y | Y | | Y | ? |

Once a representation has been found for documents, a classification algorithm can learn a profile to distinguish representations of highly rated documents from others. Fab uses Rocchio's algorithm (Rocchio, 1971) to learn a TF-IDF vector that is the average of the documents that are highly rated. Syskill & Webert uses a Bayesian classifier to estimate the probability that a document is liked. Both of these approaches have a shortcoming in that they require prespecifying the number of terms used in the profile. In this research, we take an alternate approach by using the Winnow algorithm (Littlestone & Warmuth, 1994; Blum, Hellerstein & Littlestone, 1995). Winnow is designed to identify relevant features when there are many possible attributes. Prior experimental research has demonstrated that Winnow works well on text classification (Lewis, Schapire, Callan, & Papka, 1996; Blum, 1997) in which each word $x_i$ (or pair of adjacent words) is treated as a Boolean feature. Winnow learns the weight $w_i$ associate with each word to form a linear threshold function:

$$\sum w_i x_i > t$$

where $\tau$ is the threshold. The weights are initialized to 1. Then, each training example is evaluated by finding the sum of the weights of the words that are present in the document (i.e, the presence of a word sets the variable $x_i$ to 1 and the absence sets it to 0). If the sum is above the threshold and the user did not like the document, the weight associated with each word in the document is divided by 2. If the sum is below the threshold, and the document was liked by the use, the weight associated with each word is multiplied by 2. Otherwise, the example is classified correctly and no change is made to weights. The set of training examples is cycled through adjusting the weights if necessary until all examples are processed correctly (or until the examples are cycled through 10 times with no change in accuracy on the training set). Due to the multiplicative update rule, Winnow quickly converges on a set of weights that typically assign high weights to a small percentage of the words.

We repeated the experiment that was described in Section 2.1 using Winnow as a representative content based learning algorithm. The three restaurants that had the highest sum of weights associated with the terms describing the restaurant are recommended to the user. We used Winnow in two ways. In the first, each individual word was treated as a separate term. No stemming or other processing of the words occurred. In this case, an average of 61.2% of the restaurants in the top three were actually liked by the user. In the second condition, individual words and pairs of adjacent words were used as terms. Using adjacent terms allowed Winnow to give high weight to terms such as "shrimp appetizer" while "shrimp" and "appetizer" alone had low weights. Word pairs had a negligible effect on accuracy: an average of 61.5% of the restaurants in the top three were actually liked by the user.

We use a postprocessor to identify which terms play a significant role in the classification. The postprocessor sorts the terms by their weights. Then for each positive training example, it identifies the terms with the highest weights that would need to be present to classify the example correctly. Only those terms that are needed by some positive example are considered relevant by this process. In our experiments, this process usually identifies a core set of 50-125 relevant terms for each user profile. For example, when using single words as terms, one user's profile gave very high weights to "farm," "sirloin," "private," "milk," "exquisitely," "old," "chefs," "delicacies," "scallop," and "fed." When pairs of words were included as terms, the same user's profile included "prime rib," "white fish," "old English," "charbroiled served." Our experience is that profiles that include word pairs make more sense to people, although they do not have a substantial effect on the precision of the recommendations.

### 2.3 Demographic-Based Recommendations

Demographic information can be used to identify the types of users that like a certain object. For example, Table 3 shows information on the age, gender, education, etc. of people that rated a restaurant together with their rating of the restaurant. One might expect to learn the type of person that likes a certain restaurant. Similarly, LifeStyle Finder (Kruwlich, 1997) attempts to identify one of 62 pre-existing clusters to which a user belongs and to tailor recommendations to users based upon information about others in this cluster. Obtaining demographic information can be difficult. LifeStyle Finder enters into dialog with the user to help categorize the user.

**Table** 3**.** Demographic information on the users who rated a restaurant together with the ratings of the users for that restaurant**.**

| | gender | age | area code | education | employed | Dolce |
|---|---|---|---|---|---|---|
| Karen | F | 15 | 714 | HS | F | + |
| Lynn | F | 17 | 714 | HS | F | – |
| Chris | M | 35 | 714 | C | T | + |
| Mike | F | 40 | 714 | C | T | – |
| Jill | F | 10 | 714 | E | F | ? |

In this work, we consider an alternative approach to obtaining demographic information in which in which we minimize the effort required to obtain information about user by leveraging the work the user has already expended in creating a home page on the World Wide Web. Therefore, instead of using approaches to learning from a structured database, we also use text classification to classify users. The positive examples are the HTML home pages of users that like a particular restaurant and the negative examples are the HTML home pages of users that do not like that restaurant. The Winnow algorithm can be used to learn the characteristics of home pages associated with users that like a particular restaurant.

We ran an experiment in the exact same manner as the previous experiments. There were six users that did not have or report home pages and we used the text “File not found” for these home pages. On average, 57.7% of the restaurants in the top three restaurants recommended using a demographic profile were actually liked by the user. Common terms in the profiles included words that referred to the ethnicity of the user or the users home town. While this precision is not as high as other methods, there is an increase over randomly guessing and that information may be combined with other information exploited to increase the precision of predictions.

## 2.4 Collaboration via content

Collaborative methods look for similarities between users to make predictions. Typically, the pattern of ratings of individual users is used to determine similarity. Such a correlation is most meaningful when there are many objects rated in common between users. For example, in our experimental situation, half of each user's ratings are used in the training data. Because the training data is selected from a uniform distribution, on average one quarter of the restaurants will be rated by both users. Since there are 58 restaurants in our sample, approximately 15 are used as the basis of this correlation. In some real situations, we'd expect there to be a smaller number of ratings in common. For example, for someone visiting a city for the first time, there may not be any users with a rating in common. In such a situation, collaborative methods might be expected to fail. We have created a approach we call "collaboration via content" to address this issue.

In collaboration via content, the content-based profile of each user is exploited to detect similarities among users. Table 4 shows an example of the type of data inspected by collaboration via content. Recall that the user's content-based profile contains weights for the terms that indicate that a user will like an object. Before calculating the similarity between profiles, terms that are irrelevant are deleted from each profile using the approach described in Section 2.2. This avoids considering two users to be very similar if they share a large number of terms that are irrelevant and have low weights. When computing Pearson's *r* between two profiles, any word in one profile but not another is treated as having a weight of 0 in the other profile. As in collaborative filtering, the prediction made is determined by a weighted average of all users' predictions for that restaurant using the correlation between profiles as the weight.

**Table** 4. Content-based profiles of five users plus their ratings for a particular restaurant**.**

| | noodle | shrimp | basil | exotic | salmon | Dolce |
|---|---|---|---|---|---|---|
| Karen | 2.5 | 0 | .2 | 0 | 0 | + |
| Lynn | 1.1 | 0 | 1.1 | 1.5 | 0 | – |
| Chris | 1.5 | 0 | 3.5 | 1.5 | .5 | + |
| Mike | 1.1 | 1.1 | 2.1 | 2.0 | 2.5 | – |
| Jill | 1.1 | 2.2 | 0 | 0 | 3.5 | ? |

We ran an experiment in the exact same manner as the previous experiments. On average, 70.1% of the restaurants in the top three restaurants recommended using a collaborative via content were actually liked by the user. The precision obtained by this method is higher than those achieved by either content-based or collaborative methods. We attribute this to the fact that the approach has a greater number of items from which to determine similarity than collaborative filtering. Furthermore, unlike content based filtering, this is also sensitive to whether users in general prefer the restaurant. In contrast, purely content-based methods would give identical ratings to restaurants with identical menus, ignoring other users' impression of the quality of the restaurant.

To illustrate the difference between collaborative filtering, content-based filtering, and collaboration via content, we ran an experiment in which we varied the percentage of restaurants that a user had in common with other users. In particular, the task was to select restaurants in southern Orange County for a user. For all other the users, the training data had only their recommendations for all 30 restaurants in southern Orange County. For the user for whom the system would make a recommendation, the training set consisted of thirty restaurants, and we varied the number of restaurants from southern Orange County from 3 to 20 and used the users ratings for northern Orange County restaurants for the remainder. For each number of restaurants from southern Orange County (i.e., restaurants in common with other users), we measured the precision of the top three restaurants as recommended by content-based, collaborative, and collaboration via content filtering. We repeated this process 20 times for each user (and each number of restaurants in common). Figure 2 plots the precision averaged over all users and all trial as a function of the number of restaurants in common.

As might be expected, the content-based method is relatively insensitive to the distribution of the user's ratings between southern and northern Orange County. An examination of the terms used in profiles rarely found words that referred to specific cities or geographic regions. On the other hand, when the user had few ratings in common with other users, the collaborative method performed poorly. As the number of ratings in common with other users increased, the precision of the collaborative method increased until it was eventually substantially higher than the content-based method. The collaboration via content method had higher precision than the other two methods regardless of the distribution of the training examples. This occurs because it measures similarity between users on the content-based profile, which is not very sensitive to the distribution of training examples. However, it also makes recommendations based upon the experiences of users that may not be reflected by the content of the description. In this experiment, the combination of these methods proved more effective at making recommendations than purely content-based or purely collaborative methods.

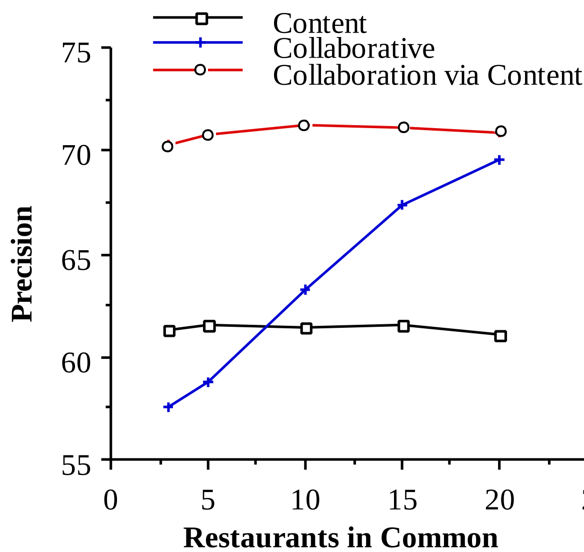


**Figure 2.** The precision of the three learning methods when learning from sparse data.

### 2.5 Summary: Profiles of user's interests

Each of the methods discussed in the previous sections uses different information to create a profile of a user's interests. We illustrate this in the restaurant domain. Demographic methods attempt to find a regularity among the descriptions of users that like particular restaurants. Content-based methods find regularity among the descriptions of restaurants liked by a particular user. Collaborative methods find a correlation between the rating of a particular user and the ratings of other users to make a prediction for that user. Alternatively, collaborative methods could also be used to find a correlation between the user ratings of a particular restaurant and the user ratings of other restaurants. Collaboration via content finds a correlation between the content-based profile of a particular user and the content-based profile of other users. Since each type of profile is learned from a different type of information, we would not expect them necessarily to make the same predictions for the rating of a particular user for a particular restaurant. Under these circumstances, we might expect that combining the predictions of each type of model could make a better prediction. We explore this possibility in the next section.

### 3.0 Combining recommendations from multiple profiles

We have identified five different approaches that may be used to make recommendations. Combining the recommendations of these approaches has the potential of improving the precision of the recommendations by finding consensus among the approaches. Since the classifiers operate differently and have different scales in which they make a recommendation, we only consider the rank of each item recommended and not the strength with which it is recommended. The approach we take is straightforward. An object with the highest recommendation receives 5 points, the next receives 4 points, etc. until the fifth most liked from each source receives 1 point. We total the number of points received by each and order the objects by the total points.

We ran an experiment in the same manner as the previous experiments in which we combined all five algorithms discussed in this paper. This resulted in the most precise predictions on the restaurant data: an average of 72.1% of the restaurants rated among the top 3 by the consensus method were liked by the user. To judge the influence of each learning method on the consensus, we also ran the consensus algorithm five times, each time leaving out one of the constituent learners. As would be expected, leaving out the most precise method (collaboration via content) results in the greatest loss of precision, while leaving out the least precise method (demographic profiling) resulted in only a slight loss of precision. In particular, without collaboration via content, the average precision of the top 3 predictions formed by consensus was 70.4%, without collaborative filtering (correlating among people), average precision was 71.3%, without collaborative filtering (correlating among restaurants) average precision was 71.8%, without content-based filtering average precision was 71.8% and without demographic profiling average precision dropped to 71.7%. This experiment demonstrates that the consensus-based method is effective at combining the strengths of the individual methods.

## 4.0 Discussion

We have reviewed existing approaches for learning user profiles based upon collaborative, content-based and demographic filtering. Table 5 summarizes the information available to a learning algorithm. Although each approach attempts to perform the same task, each approach uses a mutually exclusive subset of the available information:

- Content-based approaches use descriptions of the items rated to learn a relationship between the ratings of a single user and the description of the items rated. The information used is in the upper right portion of Table 5.
- Demographic based approaches use descriptions of the people rating to learn a relationship between a single item and the type of people that like that object. The information used is in the lower left portion of Table 5.
- Collaborative approaches use the rating of a set of people on a set of items to make recommendations, but ignore the content of the items or the descriptions of the people. The information used by collaborative methods is in the upper left of Table 5.

Placing these algorithms within a common framework has several advantages. First, since the algorithms use different forms of information, it naturally leads to the question of how the algorithms might be combined. Here, we have proposed two new combination methods. One method, collaboration via content, uses a different measure of similarity between users than existing collaborative methods. Another method combines the results of individual algorithms looking for consensus among the algorithms.

**Table 5.** The information available for inducing a user's rating for a restaurant.

| | People | | | Content | | |
|---|---|---|---|---|---|---|
| **Restaurants** | Karen | Lynn | Jill | noodle | shrimp | basil |
| Kitima | – | + | – | Y | Y | Y |
| Marco Polo | + | + | + | | Y | Y |
| Dolce | + | – | **?** | | Y | Y |
| gender | F | F | F | | | |
| age | 15 | 17 | 10 | | | |
| area code | 714 | 714 | 714 | | | |
| **Demographics** | | | | | | |

A second reason for placing the algorithms within a common framework is that it suggests future research directions by transferring approaches used in one method to another. For example, latent semantic indexing (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) is an extension to content based approaches in which an attempt is made to represent the content by a set of independent terms that are derived from the content. The mathematical technique underlying latent semantic indexing, singular value decomposition (Press, Flannery, Teukolsky, & Vetterling, 1990) reduces the dimensionality of the matrix used by content-based filtering. Although no systems currently use this approach, singular value decomposition could also be used to derive new dimensions for demographic filtering that represent underlying commonalties among users. Similarly, singular value decomposition might be applied to the matrix of user ratings for web pages. Such an approach might be an alternative to the collaboration via content approach proposed here for addressing the sparse matrix problem.

So far, we have explored collaboration via content only using binary features (indicating the presence or absence of a word), and only using binary ratings (liked or not liked). It would be fairly simple to extend the approach to the more general case in which feature values are continuous (e.g., indicating the TF-IDF weight of a term) and ratings are continuous (indicating the degree to which an item is liked). In this case, rather than using the Winnow algorithm for creating a content-based profile, a related algorithm (EG) for regression could be used (Kivinen & Warmuth, 1995). Like Winnow, such an algorithm would also result in a set of term weights that could be used as the basis of computing the similarity between users.

Collaboration via content is similar in some respects to the approach advocated by Balabanovic (1997) and implemented in the Fab system. Fab addresses a slightly different problem than considered here in that it both searches the Web for relevant pages (via a set of collection agents) and recommends the most highly rated objects (via an individual selection agent for each user). We have concentrated on the recommendation process and have not addressed the search process in this paper. As described in Balabanovic (1997), Fab uses only a content-based approach to selection, in which items found by any collection agent are rated by the user's content based profile and the most highly rated items are recommended to the user. In contrast, collaboration via content uses collaboration among users to determine the ratings of predicted pages and uses the content-based profile only to compute similarity among users.

We have also shown in this paper that the rankings of individual algorithms can be combined to increase the precision of predictions. If the classifiers all returned a ranking on the same scale, (e.g., a probability that the user would like the page), then methods for combining predictions could be used (e.g., Larkey, & Croft 1996). However, the constituent learners differ drastically in their output and instead we relied only on the ordering returned by the constituent learners. In spite of this limitation, the combined method produced the rankings with the highest precision in our experiments.

## 5.0 Conclusion

As the amount of information available to users increases, methods are needed to assist the user in finding relevant information. Intelligent agents that learn a profile of the user are one solution to this problem. We have reviewed approaches to learning user profiles based upon collaborative, content-based and demographic filtering. We have shown how the approaches use different types of information and put the approaches within a common framework. We explored two hybrid approaches for recommendation that use more of the available information and consequently have more precise recommendations.

## References

Blum, A. (1997) Empirical Support for Winnow and Weighted-Majority based algorithms: results on a calendar scheduling domain. *Machine Learning* 26:5-23.

Blum, A., Hellerstein, L. & Littlestone, N. (1995). Learning in the Presence of Finitely or Infinitely Many Irrelevant Attributes. *Journal of Computer and System Sciences* 50(1): 32-40.

Balabanovic, M. (1997). An Adaptive Web Page Recommendation Service. *First International Conference on Autonomous Agents*, Marina del Rey, CA.

Balabanovic, M. & Shoham, Y. (1997). Fab: Content-based, Collaborative Recommendation *Communications of the ACM 40(3),* March 97

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing By Latent Semantic Analysis. *Journal of the American Society For Information Science, 41*, 391-407.

Kivinen, J. & Warmuth, M. (1995). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation,* 132(1): 1-63.

Krulwich, B. (1997) LIFESTYLE FINDER: Intelligent User Profiling Using Large-Scale Demographic Data, *Artificial Intelligence Magazine* 18(2). 37-45.

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

Larkey, L. & Croft, B. (1996). Combining Classifiers in Text Categorization in *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, pp. 289-297.

Lewis, D., Schapire, R., Callan, J., & Papka, R. (1996). Training algorithms for linear text classifiers. In Hans-Peter Frei, Donna Harman, Peter Schauble, and Ross Wilkinson, editors, *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298-306.

Lieberman, H. (1995). Letizia: An agent that assists web browsing. *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal.

Littlestone, N. & Warmuth, M. (1994). The Weighted Majority Algorithm. *Information and Computation* 108(2): 212-261.

Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM,* Vol. 37, No.7, 31-40.

Miller, G. (1991). WordNet: An on-line lexical database. *International Journal of Lexicography,* 3(4).

Pazzani, M. & Billsus, D.. (1997). Learning and Revising User Profiles: The identification of interesting web sites. *Machine Learning, 27,* 313-331.

Press, W., Flannery, B., Teukolsky, S., & Vetterling, W. (1990). *Numerical Recipes in C.* Cambridge University Press.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of Netnews, *Proceedings of Conference on Computer Supported Cooperative Work (CSCW'94),* Chapel Hill, NC, ACM Press, 175-86.

Rocchio, J. (1971). Relevance feedback information retrieval. In Gerald Salton (editor), *The SMART retrieval system- experiments in automated document processing* (pp. 313-323). Prentice-Hall, Englewood Cliffs, NJ.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D. Rumelhart and J. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations,* (pp. 318-362). Cambridge, MA: MIT Press.

Salton, G. (1989). *Automatic Text Processing*. Addison-Wesley.

Salton, G. & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science, 41, 288-297.*

Shardanand, U. & Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth", *Proceedings of the Conference on Human Factors in Computing Systems* (CHI'95), Denver, CO, ACM, 1995, 210-217.