Marko Balabanović
and Yoav Shoham

Fab: Content-Based, Collaborative Recommendation

By combining both collaborative and content-based filtering systems, Fab may eliminate many of the weaknesses found in each approach.

ONLINE READERS ARE IN NEED OF TOOLS TO HELP THEM COPE with the mass of content available on the World-Wide Web. In traditional media, readers are provided assistance in making selections. This includes both implicit assistance in the form of editorial oversight and explicit assistance in the form of recommendation services such as movie reviews and restaurant guides. The electronic medium offers new opportunities to create recommendation services, ones that adapt over time to track their evolving interests. Fab is such a recommendation system for the Web, and has been operational in several versions since December 1994.

The problem of recommending items from some fixed database has been studied extensively, and two main paradigms have emerged. In *content-based* recommendation one tries to recommend items similar to those a given user has liked in the past, whereas in *collaborative* recommendation one identifies users whose tastes are similar to those of the given user and recommends items *they* have liked. Our approach in Fab has been to combine these two methods. Here, we explain how a hybrid system can incorporate the advantages of both methods while inheriting the disadvantages of neither.

In addition to what one might call the "generic advantages" inherent in any hybrid system, the particular design of the Fab architecture brings two additional benefits. First, two scaling problems common to all Web services are addressed—an increasing number of users and an increasing number of documents. Second, the system automatically identifies emergent communities of interest in the user population, enabling enhanced group awareness and communications.

Here we describe the two approaches for content-based and collaborative recommendation, explain how a hybrid system can be created, and then describe Fab, an implementation of such a system. For more details on both the implemented architecture and the experimental design the reader is referred to [1].

The content-based approach to recommendation has its roots in the information retrieval (IR) community, and employs many of the same techniques. Text documents are recommended based on a comparison between their content and a user profile. Data

structures for both of these are created using features extracted from the text of the documents. Often some weighting scheme is used which gives high weights to discriminating words. For instance, Fab's five top-weighted words from the IRS Forms and Publications page are "faint-of-heart" (0.33), "tax" (0.28), "regulations" (0.25), "tax-payer" (0.23) and "commissioner" (0.22). When a page for a user has been picked, it can be shown to them and feedback of some kind elicited. If the user liked a page, weights for the words extracted from it can be added to the weights for the corresponding words in the user profile. This process is known as relevance feedback. As well as being simple and fast, it is empirically known to give improved results in a normal IR setting [2]. Many alternative methods exist both for weighting words or other features from the text and for updating user profiles. The choice of methods does not affect our analysis.

When we contrast content-based and collaborative recommendations we need to be clear what we mean by the terms. Systems in industry and academia exist which combine elements of the two approaches, so it would be useful to define a "pure" case of each. We consider a pure content-based recommendation system to be one in which recommendations are made for a user based solely on a profile built up by analyzing the content of items which that user has rated in the past. Examples of such systems are InfoFinder [5], NewsWeeder [6], and systems developed for the routing task at the TREC conferences [3].

A pure content-based system has several shortcomings. Generally, only a very shallow analysis of certain kinds of content can be supplied. In some domains the items are not amenable to any useful feature extraction methods with current technology (such as movies, music, restaurants). Even for text documents the representations capture only certain aspects of the content, and there are many others that would influence a user's experience. For Web pages, for instance, IR techniques completely ignore aesthetic qualities, all multimedia information (including even text embedded in images), and network factors such as loading time.

A second problem, which has been studied extensively both in this domain and in others, is that of over-specialization. When the system can only recommend items scoring highly against a user's profile, the user is restricted to seeing items similar to those already rated. Often this is addressed by injecting a note of randomness. In the context of information filtering, for example, the crossover and mutation operations (as part of a genetic algorithm) have been proposed as a solution [9].

Finally, there is a problem common to most recommendation systems—eliciting user feedback. Rating documents is an onerous task for users, so the fewer ratings required the better. With the pure content-based approach, a user's own ratings are the only factor influencing future performance, and there seems to be no way to reduce the quantity without also reducing performance.

Collaborative Recommendation

The collaborative approach to recommendation is very different: Rather than recommend items because they are similar to items a user has liked in the past, we recommend items other similar users have liked. Rather than compute the similarity of the items, we compute the similarity of the users. Typically, for each user a set of "nearest neighbor" users is found with whose past ratings there is the strongest correlation. Scores for unseen items are predicted based on a combination of the scores known from the nearest neighbors.

As for the content-based case, it will be useful to define a pure version of collaborative recommendation. A pure collaborative recommendation system is one which does no analysis of the items at all—in fact, all that is known about an item is a unique identifier. Recommendations for a user are made solely on the basis of similarities to other users. Examples of systems taking this approach include GroupLens [7], the Bellcore video recommender [4], and Ringo [8].

Pure collaborative recommendation solves all of the shortcomings given for pure content-based systems. By using other users' recommendations, we can deal with any kind of content and receive items with dissimilar content to those seen in the past. Since other users' feedback influences what is recommended, there is the potential to maintain effective performance given fewer ratings from any individual user.

However, this approach does introduce certain problems of its own. If a new item appears in the database there is no way it can be recommended to a user until more information about it is obtained through another user either rating it or specifying which other items it is similar to. Thus, if the number of users is small relative to the volume of information in the system (because there is a very large or rapidly changing database), then there is a danger of the coverage of ratings becoming very sparse, thinning the collection of recommendable items. A second problem is simply that for a user whose tastes are unusual compared to the rest of the population there will not be any other users who are particularly similar, leading to poor recommendations.

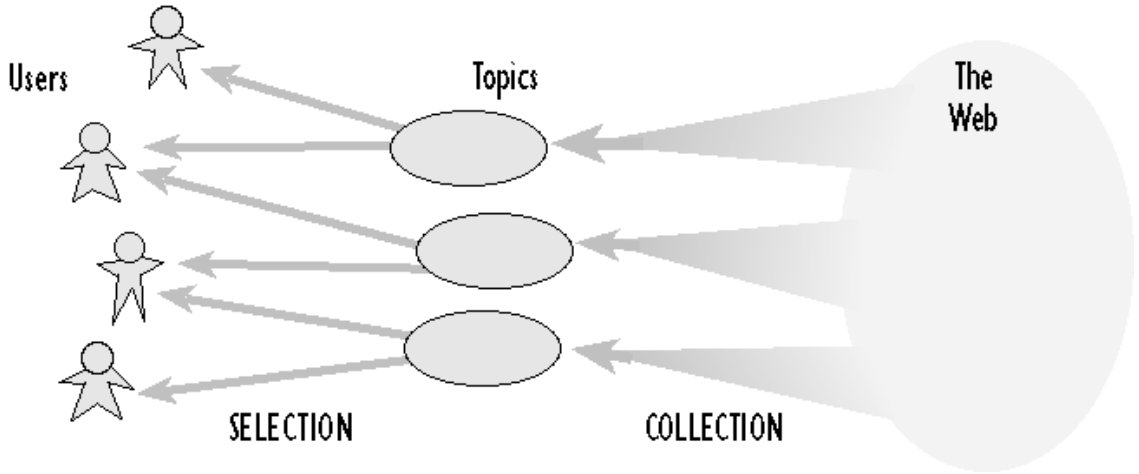

Figure 1. Pages relevant to specific topics are collected from the Web. Selections for individual users are made among these pages.

The last two problems critically depend on the size and composition of the user population, which also influence a user's group of nearest neighbors. In a situation where feedback fails to cause this group of nearest neighbors to change, expressing dislike for an item will not necessarily prevent the user from receiving similar items in the future. Furthermore, the lack of access to the content of the items prevents similar users from being matched unless they have rated the exact same items. Therefore, if one user liked the CNN weather page and another liked the MSNBC weather page, the two would not necessarily end up being nearest neighbors.

for content-based and collaborative systems, as well as adding important benefits.

One can consider both pure approaches we have discussed to be special cases of this new scheme. If the content analysis component returns just a unique identifier rather than extracting any features, then it reduces to pure collaborative recommendation; if there is only a single user, it reduces to pure content-based recommendation.

The Fab System

Fab is a distributed implementation of a hybrid system, and is part of the Stanford University digital

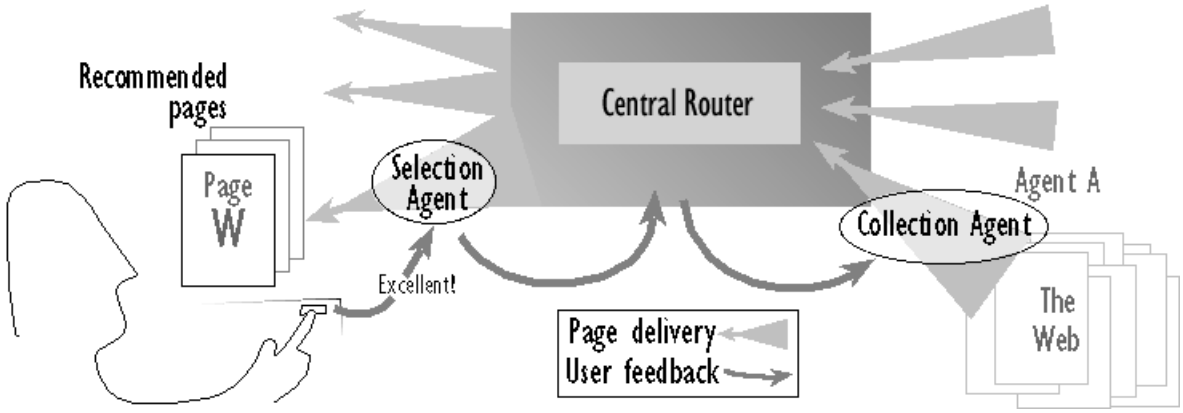

Figure 2. Overview of the Fab architecture

To create a hybrid content-based, collaborative system, we maintain user profiles based on content analysis, and directly compare these profiles to determine similar users for collaborative recommendation. Users receive items both when they score highly against their own profile, and when they are rated highly by a user with a similar profile. The hybrid approach avoids the limitations mentioned

library project.[1] In order to understand Fab it is useful to make the following practical distinction. The process of recommendation can be partitioned into two stages: collection of items to form a manageable database or index, and subsequently selection of items from this database for particular users. In some

[1]Fab can be accessed at http://fab.stanford.edu.

instances the collection stage is trivial or performed by a third party, but in the case of the Web it is a real problem faced by the system designer. Figure 1 shows our underlying model. The collection stage gathers pages relevant to a small number of topics, computer-generated clusters of interests which track the changing tastes of the user population. These pages are then delivered to a larger number of users via the selection stage. One topic can be of interest to many users, and one user can be interested in many topics.

The implemented architecture (Figure 2) closely mirrors this model. There are three main components: collection agents (that find pages for a specific topic), selection agents (that find pages for a specific user) and the central router. Every agent maintains a profile, based on words contained in Web pages which have been rated. A collection agent's profile

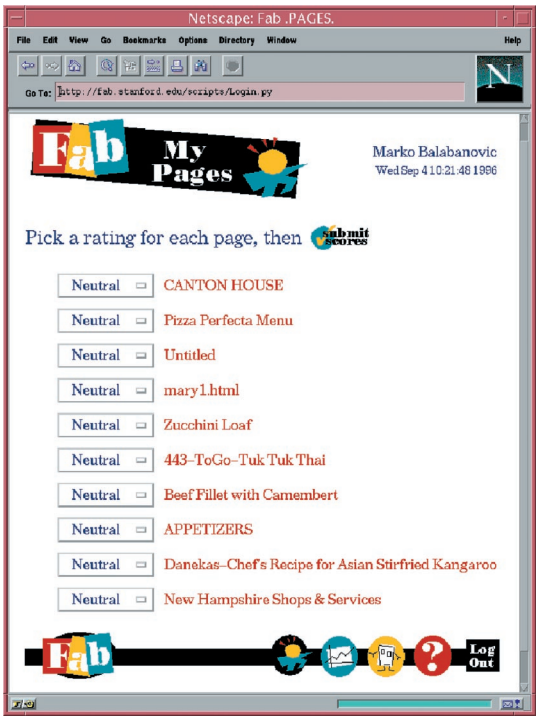

Figure 3. An example set of recommendations as they appear to a Fab user. In this case the user has already rated all of the pages, and is about to submit the scores.

represents its current topic, whereas a selection agent's profile represents a single user's interests.

Pages found by the collection agents are sent to the central router, which forwards them on to those users whose profiles they match above some threshold. Thus, each user receives pages matching their profile from the collection agents. Additional functionality is located within the user's personal selection agent: pages the user has already seen are discarded, and in any single batch of recommendations (usually 10-pages) we insure there is at most one page from any site. The user's feedback represents a significant investment in time and effort. By storing it in their own private selection agent's profile, we insure it can never be "drowned out" by other users' feedback. In fact, it is easily exportable for use in other applications.

When the user has requested, received, and looked over their recommendations, they are required to assign appropriate ratings from a 7-point scale. An example set of recommendations illustrating the Fab interface is shown in Figure 3. The user's ratings are used to update their personal selection agent's profile, and are also forwarded back to the originating collection agents, which will use them to adapt their profiles. Additionally, any highly rated pages are passed directly to the user's nearest neighbors—other people with similar profiles. These collaborative recommendations are processed by the receiving user's selection agent in the same way as the pages from the central router.

The construction of accurate profiles is a key task—the system's success will depend to a large extent on the ability of the learned profiles to represent the users' actual interests. Accurate profiles enable both the content-based component (to insure recommendations are appropriate) and the collaborative component (to insure users with similar profiles are indeed similar).

The collection agents' profiles represent a topic of interest to a dynamically changing group of users, as opposed to a user's profile, which represents multiple interests possibly served by several collection agents. The population of collection agents as a whole adapts to the population of users, not to any specific user. To aid this process, unpopular collection agents (whose pages are not seen by many users) or unsuccessful ones (who receive low median feedback scores) are regularly weeded out and the best ones duplicated to take their places. Thus, the collection agents' specializations need not be fixed in advance, but are determined dynamically and change over time. In effect, our system engages in two different and simultaneous load-balancing acts, reflected in the two dynamically changing sets of linkages: those between documents and collection agents, and those between collection agents and users. One of our goals is to investigate the properties of this mutual adaptation.

We have implemented several different kinds of collection agents. *Search agents* perform a best-first search of the Web, trying to find pages best matching their profiles. Their assumption is that a page will have links to similar pages, and so by following links from page to page they can uncover information pertinent to a particular topic. *Index agents* con-

struct queries to pass to various commercial Web search engines that have already performed exhaustive indexing. For comparative purposes we have also included agents that supply randomly picked pages, agents that collect various human-picked "cool sites of the day," and agents that attempt to serve an average user (with an average of all the user profiles in the system), rather than maintaining their own specialized profile.

The system exhibits all of the advantages hybrid systems bring to the selection process:

- By making collaborative recommendations, we can use others' experiences as a basis rather than the incomplete and imprecise content analysis methods at our disposal.
- By making content-based recommendations as well, we can deal with items unseen by others.
- We can use the profile we build from the content of items to make good recommendations to users, even if there are no other users similar to them. We can also filter out items.
- We can make collaborative recommendations between users who have not rated any of the same items (as long as they have rated similar items), extending the reach of collaborative systems to include databases which change quickly or are very large with respect to the number of users.
- By utilizing group feedback we potentially require fewer cycles to achieve the same level of personalization.

Additionally, the adaptation of the collection agents enables some features impossible with the pure collaborative or content-based approaches alone:

- We can instantiate a smaller number of collection agents than there are users, perhaps even a fixed number. This should allow the system to scale gracefully as the number of users and documents rise. The exact number of collection agents required is determined by several factors, including the extent of the overlaps between users' interests and the tradeoff between the available computing resources and the quality of recommendations required.
- The collection agents automatically identify emergent communities of interest, allowing us to support social interactions between like-minded people and to automatically provide group as well as individual recommendations. Effectively, like-minded users are pooling their resources, as each collection agent will be receiving feedback from all users interested in a topic. Both of these features rely crucially on the ability of the collection agents to specialize and learn profiles which do indeed represent areas where users' interests overlap.

Experiments

We have conducted evaluations of several aspects of the Fab system. Here we present three sets of results—two statistical in nature and one anecdotal—from a controlled experiment with a small number of users. All of our tests have been in real-world settings, recommending current Web pages to real users.

Since accurate profiles based on the content of Web pages are a cornerstone of our design, we set out to measure with our first experiment the predictive power of the learned profiles: How well can they predict the user's ranking of a set of items? If they cannot predict well they may still be usable to provide a similarity measure for collaborative recommendation, but they would certainly not be able to provide good content-based recommendations.

We asked 11 users to declare in advance a single topic of interest (to allow easier postliminary analysis of the resulting profiles). Only nine were sufficiently frequent users for their results to be interpretable. Their topics were: computer graphics and game programming, library cataloging and classification, post-industrial music, sports information and gaming, Native American culture, cookery, 1960s music, hiking, and evolution. On every fifth set of evaluations (roughly every five days), the users were shown a special selection of items and informed their ratings were being used only for evaluation purposes, and would not influence their profiles. The composition of this special selection is not crucial to this experiment, but plays a significant role in the final experiment to be described, and so will be explained in that section.

We used each user's ratings to order the documents they had seen, creating a preference ranking (possibly including ties). For each point in time we then measured the distance between the users' rankings and the rankings predicted from their profiles, using the *ndpm* measure as defined by Yao [10].[2] The duration of the experiment was approximately one month. Figure 4 shows how the profiles, given more and more examples, become much better predictors of the users rankings over time. In particular, the

[2]Briefly, whereas traditional methods of IR evaluation assume absolute relevance judgments are available, Yao's proposal is to require only comparative judgments—how documents rank relative to each other. Among other advantages, these rankings prove to be more consistent over long periods of time, both for a given user and between users.

ndpm value of approximately 0.02 arrived at by evaluation 25 is equivalent to a difference between 16-item predicted and actual rankings of just a single item misplaced by two positions.

One of the hypothesized merits of our system is leveraging the common interests of users, with collection agents specializing to topics and serving multiple users where appropriate. While we have no statistical results on this issue, we do have anecdotal evidence the system is performing in this fashion. In a clear case of automatic specialization, one agent became a "cooking expert:" 77% of the top 400 terms in its profile are obviously cooking-related (Table 1). It mainly serves the user interested in cooking, who receives 50–90% of his or her documents from this one agent. The common interests of the two users interested in music are reflected in the fact there are three agents with an approximately equal number of obviously music-related terms in their profiles, and the two users receive their music-related pages from a mix of these three agents.

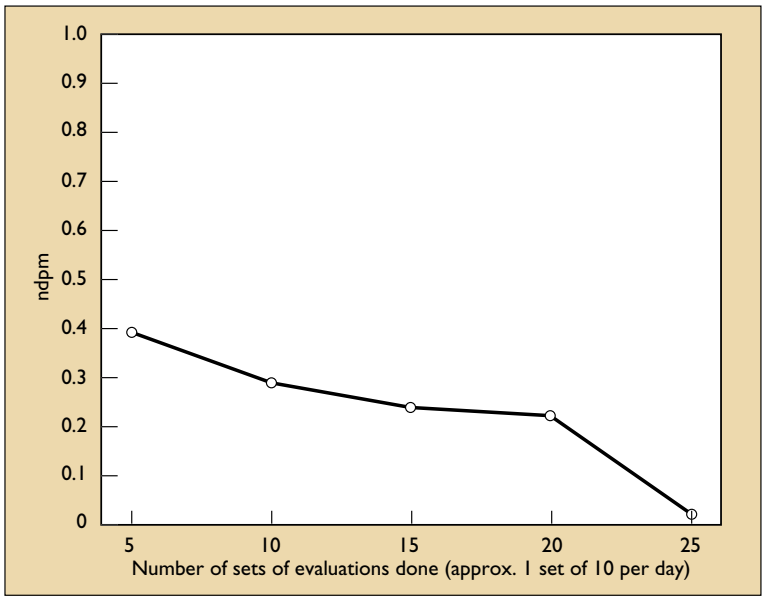

Figure 4. Distance between actual and predicted rankings, averaged over all users at each evaluation point.

Despite the small number of seemingly disparate topics, the system still managed to pick out some areas of overlap, where an agent specialized to a topic of interest to several users. The best example of this was an agent serving pages about India (resulting from a confusion with the topic of Native American cultures). This agent delivered pages on biodiversity in India to the user interested in evolution and on Indian recipes to the user interested in cooking. Similarly, the users interested in Web development and computer graphics received pages on computing textbooks relevant to both their topics.

These examples show the agents can specialize to specific topics over time, and automatically converge to areas of overlap between the users. Our aim is to utilize this feature to discover how many users we can serve successfully from a fixed pool of agents.

Overall Performance

The final results are again statistical in nature, and look at the performance of the Fab system as a whole. In this experiment the special sets of evaluation pages shown to users consisted of pages from four different sources: regular "personal" Fab recommendations, randomly selected pages, pages from human-selected "cool sites of the day," and pages best matching an average of all user profiles in the system ("public" pages).

While there are a number of ways the results of the users' rankings of the pages from the four sources could be presented, we have chosen to use the *ndpm* measure again. In order to do this we need to define an ideal ranking for each source. An ideal ranking of some batch of pages for source *S* is one where the user prefers every page from *S* to every page not from *S*. Note that this notion is intentionally underspecified—it does not matter how the user ranks the pages from *S* relative to one another, nor the pages not from *S*. The greater the preference the user expresses for pages from *S* over the other pages supplied, the smaller the *ndpm* distance between the user's actual ranking and the ideal ranking for *S*.

Figure 5 plots this distance between the users' actual rankings and the ideal ranking for each source. It shows the personal pages provided by Fab clearly

Table 1. Top 20 words and associated weights from the profile of a collection agent specializing in cooking. Some of the word endings have been removed (e.g., "mince", "minced" and "mincing" all become "minc") or altered (e.g., "parsley" becomes "parslei") as part of the stemming process which reduces words to their roots.

tablespoon	2.95	sprinkl	1.95	tomato	1.58
teaspoon	2.44	saut	1.92	cup	1.51
onion	2.16	chop	1.92	stir	1.44
flour	2.13	parslei	1.92	preheat	1.37
minc	2.09	saucepan	1.79	pepper	1.34
garlic	2.06	sauc	1.71	parmesan	1.33
clove	2.00	butter	1.59		

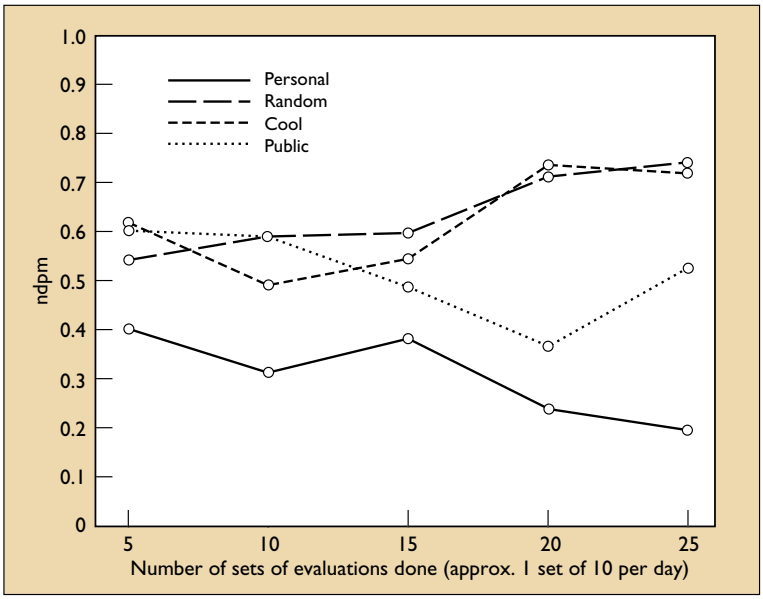

Figure 5. For each source, distance between users' rankings and its ideal ranking, averaged over all users at each evaluation point.

outperform the other sources, improving over the course of the experiment. The public pages represent a system which adapts but is not personalized to individual users. Although not as good as the performance of the regular Fab system, the public pages still rank higher than the random and cool pages, which end up equally poor.

Future Work

The Web is an intimidatingly large information space, and an effective service providing personalized recommendations is of undisputed value. Both content-based and collaborative systems can provide such a service, but individually they both face shortcomings. Fab is an implementation of a hybrid content-based, collaborative Web-page recommendation system that eliminates many of the handicaps of the pure versions of either approach.

As well as embodying the advantages of a hybrid scheme, the Fab architecture brings added benefits, which are made possible by using the overlaps between users' interests for more than just collaborative selection. The design of the adapting population of collection agents takes advantage of these overlaps to dynamically converge on topics of interest, both automatically identifying communities of interest and providing the possibility of significant resource savings when increasing the numbers of users and documents.

Initial experiments validate our profile construction methods, and show anecdotally that the emergent properties we postulated for collection agents are indeed being exhibited, namely agents specializing to topics and serving multiple users where appropriate. In a comparison relative to three benchmarks, the Fab system has been shown to improve its performance over time, while consistently producing pages users ranked higher than pages from the other three systems.

We are currently deep into our next set of experiments. In this next phase there are two main research issues we wish to tackle. We aim to study the effects of massively scaling up the number of users, and we plan to continue our investigation of the dynamic processes involved, in particular to further elucidate the roles of the collaborative and content-based components by measuring their relative performance. C

This research was supported in part by the NSF/ARPA/NASA Digital Library project (NSF IRI-9411306) and in part by NSF contract IRI-9220645.

References

1. Balabanovic, M. An adaptive web page recommendation service. In *Proceedings of the 1st International Conference on Autonomous Agents* (Marina del Rey, Calif., Feb.1997).
2. Buckley, C., and Salton, G. Optimization of relevance feedback weights. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, July 1995).
3. Harman, D. Overview of the 3rd Text REtrieval Conference (TREC-3). In *Proceedings of the 3rd Text REtrieval Conference* (Gaithersburg, Md, Nov. 1994).
4. Hill, W., Stead, L., Rosenstein, M., and Furnas, G. Recommending and evaluating choices in a virtual community of use. In *Conference on Human Factors in Computing Systems—CHI '95.* (Denver, May 1995).
5. Krulwich, B., and Burkey, C. Learning user information interests through extraction of semantically significant phrases. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access* (Stanford, Calif., March 1996).
6. Lang, K. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning* (Tahoe City, Calif.) 1995.
7. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work* (Chapel Hill, NC) 1994.
8. Shardanand, U., and Maes, P. Social information filtering: Algorithms for automating "word of mouth." In *Conference on Human Factors in Computing Systems—CHI '95.* (Denver, May 1995).
9. Sheth, B., and Maes, P. Evolving agents for personalized information filtering. In *Proceedings of the 9th IEEE Conference on Artificial Intelligence for Applications* (Orlando, Fla, March 1993).
10. Yao, Y. Y. Measuring retrieval effectiveness based on user preference of documents. *J. Amer. Soc. Info. Sci. 46*, 2 (1995), 133–145.

Marko Balabanović (marko@cs.stanford.edu) is a Ph.D. student in the computer science department at Stanford University, Stanford, Calif.
Yoav Shoham (shoham@cs.stanford.edu) is an associate professor, and the director of the Robotics Laboratory and AI division, at Stanford University, Stanford, Calif.