Social Information Filtering: Algorithms for Automating "Word of Mouth"

Upendra Shardanand
MIT Media-Lab
20 Ames Street Rm. 305
Cambridge, MA 02139
shard@media.mit.edu
(617) 253-7441

Pattie Maes
MIT Media-Lab
20 Ames Street Rm. 305
Cambridge, MA 02139
pattie@media.mit.edu
(617) 253-7442

ABSTRACT

This paper describes a technique for making personalized recommendations from any type of database to a user based on similarities between the interest profile of that user and those of other users. In particular, we discuss the implementation of a networked system called Ringo, which makes personalized recommendations for music albums and artists. Ringo's database of users and artists grows continuously as more people use the system and enter more information. Four different algorithms for making recommendations by using social filtering were tested and compared. We present quantitative and qualitative results obtained from the use of Ringo by more than 2000 people.

KEYWORDS: social filtering, personalized recommendation systems, user modeling, information retrieval, intelligent systems, CSCW.

INTRODUCTION

Recent years have seen the explosive growth of the sheer volume of information. The number of books, movies, news, advertisements, and in particular on-line information, is staggering. The volume of things is considerably more than any person can possibly filter through in order to find the ones that he or she will like.

People handle this information overload through their own effort, the effort of others and some blind luck. Most items and information are removed from the stream simply because they are either inaccessible or invisible to the user. Second, a large amount of filtering is done for us. Newspaper editors select what articles their readers want to read. Bookstores decide what books to carry. However with the dawn of the electronic information age, this barrier will become less and less a factor. Finally, we rely on friends and other people whose judgement we trust to make recommendations to us.

We need technology to help us wade through all the information to find the items we really want and need, and to rid us of the things we do not want to be bothered with. The common and obvious technique used to tackle the problem of information filtering is *content-based filtering*[1]. A system recommends items for the user's consumption based on correlations between the *content* of the items and the user's preferences. For example, the system may try to correlate the presence of keywords in an article with the user's taste. However, content-based filtering has limitations:

- Either the items must be of some machine parsable form (e.g. text), or attributes must have been assigned to the items by hand. With current technology, media such as sound, photographs, art, video or physical items cannot be analyzed automatically for relevant attribute information. Often it is not practical or possible to assign attributes by hand due to limitations of resources.
- Content-based filtering techniques have no inherent method for generating serendipitous finds. The system recommends more of what the user already has seen before (and indicated liking). In practice, additional hacks are often added to introduce some element of serendipity.

A complementary filtering technique is needed to address these issues. This paper presents *social filtering*, a general approach to personalized information filtering. Social filtering essentially automates the process of "word-of-mouth" recommendations: items are recommended to a user based upon values assigned by other people with similar taste. The system determines which users have similar taste via standard formulas for computing statistical correlations.

Social filtering overcomes some of the limitations of content-based filtering. Items being filtered need not be amenable to parsing by a computer. Furthermore, the system may recommend items to the user which are very different (content-wise) from what the user has indicated liking before.

This paper details the implementation of a social filtering system called Ringo, which makes personalized music recommendations to people on the Internet. Results based on the use of this system by thousands of actual users are presented. Various social filtering algorithms are described, analyzed and compared. These results demonstrate the strength of social filtering and its potential for immediate application.

RINGO: A PERSONALIZED MUSIC RECOMMENDATION SYSTEM

Social filtering exploits similarities between the tastes of different users to recommend (or advise against) items. It relies on the fact that people's tastes are not randomly distributed: there are general trends and patterns within the taste of a person and as well as between groups of people. Social filtering automates a process of "word-of-mouth" recommendations. A significant difference is that instead of having to ask a couple friends about a few items, a social filtering system can consider thousands of other people, and consider thousands of different items, all happening autonomously and automatically. The basic idea is:

1. The system maintains a *user profile*, a record of the user's interests (positive as well as negative) in specific items.
2. It compares this profile to the profiles of other users, and weighs each profile for its degree of similarity with the user's profile. The metric used to determine similarity can vary.
3. Finally, it considers a set of the most similar profiles, and uses information contained in them to recommend (or advise against) items to the user.

Ringo[7] is a social filtering system which makes personalized music recommendations. People describe their listening pleasures to the system by rating some music. These ratings constitute the person's *profile*. This profile changes over time as the user rates more artists. Ringo uses these profiles to generate advice to individual users. Ringo compares user profiles to determine which users have similar taste (they like the same albums and dislike the same albums). Once similar users have been identified, the system can predict how much the user may like an album/artist that has not yet been rated by computing a weighted average of all the ratings given to that album by the other users that have similar taste.

7 : BOOM! One of my FAVORITE few!
Can't live without it.
6 : Solid. They are up there.
5 : Good Stuff.
4 : Doesn't turn me on, doesn't bother me.
3 : Eh. Not really my thing.
2 : Barely tolerable.
1 : Pass the earplugs.

Figure 1: Ringo's scale for rating music.

Ringo is an on–line service accessed through electronic mail. Users may sign up with Ringo by sending e-mail to *ringo@media.mit.edu* with the word "join" in the body. People interact with Ringo by sending commands and data to a central server via e-mail. Once an hour, the server processes all incoming messages and sends replies as necessary. Soon, users will be able to interact with Ringo via its World Wide Web interface, which is currently being tested.

When a user first sends mail to Ringo, he or she is sent a list of 125 artists. The user rates artists for how much they like to listen to them. If the user is not familiar with an artist or does not have a strong opinion, the user is asked not to rate that item. Users are specifically advised to rate artists for how much they like to *listen* to them, not for any other criteria such as musical skill, originality, or other possible categories of judgment.

The scale for ratings varies from 1 "pass the earplugs"' to 7 "one of my favorite few, can't live without them". A seven point scale was selected since studies have shown that the reliability of data collected in surveys does not increase substantially if the number of choices is increased beyond seven[6]. Ratings are not normalized because as we expected, users rate albums in very different ways. For example, some users only give ratings to music they like (e.g. they only use 6's and 7's), while other users will give bad as well as good ratings (1's as well as 7's). Figure 2 illustrates the distribution of mean scores given by every user to all artists. Since the peak of this distribution is not sharp, an absolute scale was employed and descriptions for each rating point were provided to make it clear what each number means (Figure 1).

The list of artists sent to a user is selected in two parts. Part of the list is generated from a list of the most often rated artists. This ensures that a new user has the opportunity to rate artists which others have also rated, so that there is some commonality in people's profiles. The other part of the list is generated through a random selection from the (open) database of artists. Thus, artists are never left out of the loop. A user may also request a list of some artist's albums, and rate that artist's al-

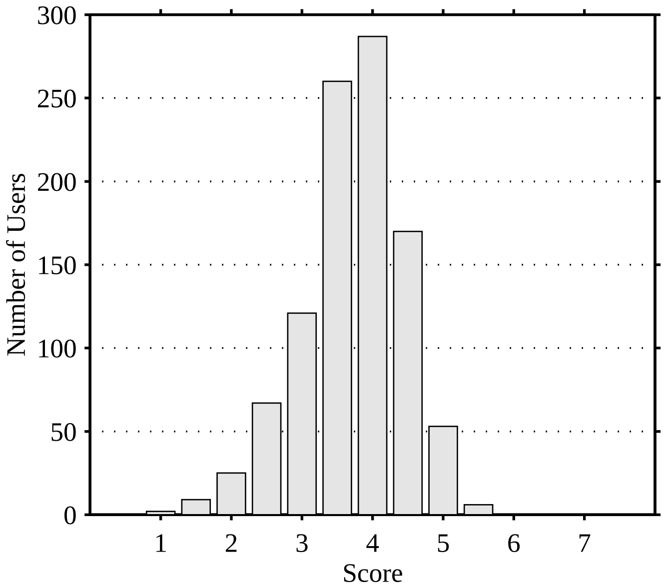

Figure 2: The distribution of mean scores of user profiles.

6	"10,000 Maniacs"
3	"AC/DC"
3	"Abdul, Paula"
2	"Ace of Base"
1	"Adams, Bryan"
	"Aerosmith"
	"Alpha Blondy"
6	"Anderson, Laurie"
5	"Arrested Development"
	"Autechre"
3	"B-52s"
	"Babes in Toyland"
	"Be Bop Deluxe"
5	"Beach Boys, The"
	"Beastie Boys"
4	"Beat Happening"
7	"Beatles, The"
1	"Bee Gees"

Figure 3: Part of one person's survey.

bums on an individual basis. The procedure for picking an initial list of artists for the user to rate leaves room for future improvement and research, but has been sufficient for our early tests. Figure 3 shows part of one user's ratings of the initial 125 artists selected by Ringo.

Once a person's initial profile has been submitted, Ringo sends a help file to the user, detailing all the commands it understands. An individual can ask Ringo for predictions based upon their personal profile. Specifically, a person can ask Ringo to (1) suggest new artists/albums that the user will enjoy, (2) list artists/albums that the user will hate, and (3) make a prediction about a specific artist/album. Ringo processes such a request using its social filtering algorithm, detailed in the next section. It then sends e-mail back to the person with the result. Figure 4 provides an example of Ringo's suggestions. Every recommendation includes a measure of confidence which depends on factors such as the number of similar users used to make this prediction, the consistency among those users' values, etc. (cfr. [7] for details.) Ringo's reply does not include any information about the identity of the other users whose profiles were used to make the recommendations.

Ringo provides a range of functions apart from making recommendations. For example, when rating an artist or album, a person can also write a short review, which Ringo stores. Two actual reviews entered by users are shown in Figure 5. Notice that the authors of these reviews are free to decide whether to sign these reviews or keep them anonymous. When a user is told to try or to avoid an artist, any reviews for that artist written by similar users are provided as well. Thus, rather than one "thumbs-up, thumbs-down" review being given to the entire audience, each user receives personalized reviews from people that have similar taste.

In addition, Ringo offers other miscellaneous features which increase the appeal of the system. Users may add new artists and albums into the database. This feature was responsible for the growth of the database from 575 artists at inception to over 2500 artists in the first 6 weeks of use. Ringo, upon request, provides dossiers on any artist. The dossier includes a list of that artist's albums and straight averages of scores given that artist and the artist's albums. It also includes any added history about the artist, which can be submitted by any user. Users can also view a "Top 30" and "Bottom 30" list of the most highly and most poorly rated artists, on average. Finally, users can subscribe to a periodic newsletter keeping them up to date on changes and developments in Ringo.

ALGORITHMS AND QUANTITATIVE RESULTS

Ringo became available to the Internet public July 1, 1994. The service was originally advertised on only four specialized USENET newsgroups. After a slow start, the number of people using Ringo grew quickly. Word of the service spread rapidly as people told their friends, or sent messages to mailing lists. Ringo reached the 1000-user mark in less than a month, and had 1900 users after 7 weeks. At the time of this writing Ringo has 2100 users and processes almost 500 messages a day.

Like the membership, the size of the database grew

Artist	Rating	Confidence
"Orb, The"	6.9	fair
"Negativland"	6.5	high
Reviews for "Negativland" They make you laugh at the fact that nothing is funny any more. — user@place.edu		
"New Order"	6.5	fair
Reviews for "New Order" Their albums until 'Brotherhood' were excellent. Since then, they have become a tad too tame and predictable. — lost@elsewhere.com		
"Sonic Youth"	6.5	fair
Reviews for "Sonic Youth" Confusion is Sex: come closer and I'll tell you.		
"Grifters"	6.4	fair
"Dinosaur Jr."	6.4	fair
"Velvet Underground, The"	6.3	low
Reviews for "Velvet Underground, The" The most amazing band ever.		
"Mudhoney"	6.3	fair

Figure 4: Some of Ringo's suggestions.

Tori Amos has my vote for the best artist ever. Her lyrics and music are very inspiring and thought provoking. Her music is perfect for almost any mood. Her beautiful mastery of the piano comes from her playing since she was two years old. But, her wonderful piano arrangements are accompanied by her angelic yet seductive voice. If you don't have either of her two albums, I would very strongly suggest that you go, no better yet, run down and pick them up. They have been a big part of my life and they can do the same for others. —- user@place.edu

I'd rather dive into a pool of dull razor blades than listen to Yoko Ono sing. OK, I'm exaggerating. But her voice is *awful* She ought to put a band together with Linda McCartney. Two Beatles wives with little musical talent.

Figure 5: Two sample reviews written by users.

quickly. Originally, Ringo had only 575 artists in its database. As we soon discovered, users were eager to add artists and albums to the system. At the time of this writing, there are over 3000 artists and 9000 albums in Ringo's database.

Thanks to this overwhelming user interest, we have an enormous amount of data on which to test various social information filtering algorithms. This section discusses four algorithms that were evaluated and gives more details about the "winning" algorithm. For our tests, the profiles of 1000 people were considered. A profile is a sparse vector of the user's ratings for artists. 1,876 different artists were represented in these profiles.

To test the different algorithms, 20% of the ratings in each person's profile were then randomly removed. These ratings comprised the *target set* of profiles. The remaining 80% formed the *source set*. To evaluate each algorithm, we predicted a value for each rating in the target set, using only the data in the source set. Three such target sets and data sets were randomly created and tested, to check for consistency in our results. For brevity, the results from the first set are presented throughout this paper, as results from all three sets only differed slightly.

In the source set, each person rated on average 106 artists of the 1,876 possible. The median number of ratings was 75, and the most ratings by a single person was 772! The mean score of each profile, i.e. the average score given all artists by a user, was 3.7.

Evaluation Criteria

The following criteria were used to evaluate each prediction scheme:

- The mean absolute error of each predicted rating must be minimized. If $\{r_1, \ldots, r_N\}$ are all the real values in the target set, and $\{p_1, \ldots, p_N\}$ are the predicted values for the same ratings, and $E = \{\varepsilon_1, \ldots, \varepsilon_N\} = \{p_1 - r_1, \ldots, p_N - r_N\}$ are the errors, then the mean absolute error is

$$\overline{|E|} = \frac{\sum_{i=1}^{N} |\varepsilon_i|}{N} \tag{1}$$

  The lower the mean absolute error, the more accurate the scheme. We cannot expect to lower $\overline{|E|}$ below the error in people's ratings of artists. If one provides the same list of artists to a person at different points of time, the resulting data collected will differ to some degree. The degree of this error has not yet been measured. However we would expect the error to at least be ± 1 unit on the rating scale (because otherwise there would be 0 or no error).

- The standard deviation of the errors,

$$\sigma = \sqrt{\frac{(\sum (E - \overline{E})^2)}{N}} \tag{2}$$

  should also be minimized. The lower the deviation, the more consistently accurate the scheme is.

- Finally, $\mathcal{T}$, the percentage of target values for which the scheme is able to compute predictions should be maximized. Some algorithms may not be able to make predictions in all cases.

Base Case Algorithm

A point of comparison is needed in order to measure the quality of social filtering schemes in general. As a base case, for each artist in the target set, the mean score received by an artist in the source set is used as the predicted score for that artist. A social filtering algorithm is neither personalized nor accurate unless it is a significant improvement over this base case approach.

Figure 6 depicts the distribution of the errors, E. $\overline{|E|}$ is 1.3, and the standard deviation σ is 1.6. The distribution has a nice bell curve shape about 0, which is what was desired. At first glance, it may seem that this mindless scheme does not behave too poorly. However, let us now restrict our examination to the *extreme* target values, where the score is **6** or greater or **2** or less. These values, after all, are the critical points. Users are most interested in suggestions of items they would love or hate, not of items about which they would be ambivalent.

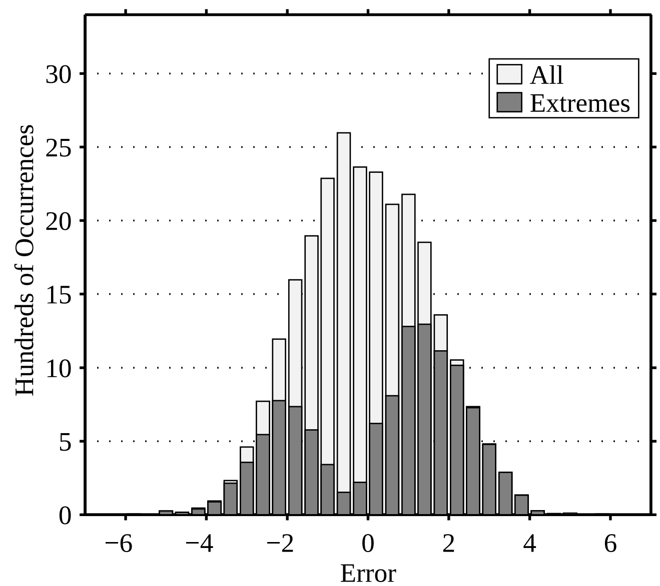

Figure 6: The distribution of errors in predictions of the Base Algorithm.

The distribution of errors for extreme values is shown by the dark gray bars in Figure 6. The mean error and standard deviation worsen considerably, with $\overline{|E|} = 1.8$ and $\sigma = 2.0$. Note the lack of the desired bell curve shape. It is in fact the sum of two bell curves. The right hill is mainly the errors for those target values which are **2** or less. The left hill is mainly the errors for those target values which are **6** or greater.

For the target values **6** or greater, the mean absolute error is much worse, with $\overline{|E|} = 2.1$. Why the great discrepancy in error characteristics between all values and only extreme values? Analysis of the database indicates that the mean score for each artist converges to approximately **4**. Therefore, this scheme performs well in cases where the target value is near **4**. However, for the areas of primary interest to users, the base algorithm is useless.

Social Filtering Algorithms

Four different social filtering algorithms were evaluated. Due to space limitations, the algorithms are described here briefly. Exact mathematical descriptions as well as more detailed analysis of the algorithms can be found in [7].

The Mean Squared Differences Algorithm. The first algorithm measures the degree of dissimilarity between two user profiles, U_x and U_y by the *mean squared difference* between the two profiles:

$$\overline{(U_x - U_y)^2} \tag{3}$$

Predictions can then be made by considering all users with a dissimilarity to the user which is less than a certain threshold L and computing a weighted average of the ratings provided by these most similar users, where the weights are inverse proportional to the dissimilarity.

The Pearson r Algorithm. An alternative approach is to use the standard *Pearson r* correlation coefficient to measure similarity between user profiles:

$$\frac{\sum(U_x - \overline{U_x})(U_y - \overline{U_y})}{\sqrt{\sum(U_x - \overline{U_x})^2 \times \sum(U_y - \overline{U_y})^2}} \quad (4)$$

This coefficient ranges from -1, indicating a negative correlation, via O, indicating no correlation, to +1 indicating a positive correlation between two users. Again, predictions can be made by computing a weighted average of other user's ratings, where the Pearson r coefficients are used as the weights. In contrast with the previous algorithm, this algorithm makes use of negative correlations as well as positive correlations to make predictions.

The Constrained Pearson r Algorithm. Close inspection of the Pearson r algorithm and the coefficients it produced prompted us to test a variant which takes the *positivity* and *negativity* of ratings into account. Since the scale of ratings is absolute, we "know" that values below **4** are negative, while values above **4** are positive. We modified the Pearson r scheme so that only when there is an instance where both people have rated an artist positively, above **4**, or both negatively, below **4**, will the correlation coefficient increase. More specifically, the standard Pearson r equation was altered to become:

$$\beta_{xy} = \frac{\sum(U_x - 4)(U_y - 4)}{\sqrt{\sum(U_x - 4)^2 \times \sum(U_y - 4)^2}} \quad (5)$$

To produce recommendations to a user, the constrained Pearson r algorithm first computes the correlation coefficient between the user and all other users. Then all users whose coefficient is greater than a certain threshold L are identified. Finally a weighted average of the ratings of those similar users is computed, where the weight is proportional to the coefficient. This algorithm does not make use of negative "correlations" as the Pearson r algorithm does. Analysis of the constrained Pearson r coefficients showed that there are few very negative coefficients, so including them makes little difference.

The Artist-Artist Algorithm. The preceding algorithms deal with measuring and employing similarities between *users*. Alternatively, one can employ the use of correlations between *artists or albums* to generate predictions. The idea is simply an inversion of the previous three methodologies. Say Ringo needs to predict how a user, Murray, will like "Harry Connick, Jr". Ringo examines the artists that Murray has already rated. It weighs each one with respect to their degree of correlation with "Harry Connick, Jr". The predicted rating is then simply a weighted average of the artists that Murray has already scored. An implementation of such a scheme using the constrained Pearson r correlation coefficient was evaluated.

	All		Extremes		
Method	$\overline{\lvert E\rvert}$	σ	$\overline{\lvert E\rvert}$	σ	$\mathcal{T}$
Base Case	1.3	1.6	1.8	2.0	90
Mean Sq. Diff., $L = 2.0$	1.0	1.3	1.2	1.6	70
Pearson r	1.1	1.4	1.5	1.7	99
Pearson r, $L = 0.35$	1.0	1.3	1.4	1.6	99
Pearson r, $L = 0.5$	1.0	1.3	1.3	1.6	95
Pearson r, $L = 0.65$	1.1	1.4	1.3	1.6	73
Pearson r, $L = 0.75$	1.1	1.5	1.3	1.7	41
Con. Pearson r, $L = 0.5$	1.1	1.3	1.3	1.6	97
Con. Pearson r, $L = 0.6$	1.1	1.4	1.2	1.6	91
Con. Pearson r, $L = 0.7$	1.1	1.3	1.3	1.6	70
Artist-Artist, $L = 0.6$	1.1	1.4	1.3	1.6	89
Artist-Artist, $L = 0.7$	1.1	1.4	1.1	1.5	65

Table 1: Summary of results.

Results

A summary of some of our results (for different values of the threshold L) are presented in table 1. More details can be found in [7]. Overall, in terms of accuracy and the percentage of target values which can be predicted, the constrained Pearson r algorithm performed the best on our dataset if we take into account the error as well as the number of target values that can be predicted. The mean square differences and artist-artist algorithms may perform slightly better in terms of the quality of the predictions made, but they are not able to produce as many predictions.

As expected, there is a tradeoff between the average error of the predictions and the percentage of target values that can be predicted. This tradeoff is controlled by the parameter L, the minimum degree of similarity between users that is required for one user to influence the recommendations made to another.

Figure 7 illustrates the distribution of errors for the best algorithm with the threshold L equal to 0.6. The distribution for extreme values approaches a bell curve, as desired. The statistics for all values and extreme values are $\overline{|E|} = 1.1$, $\sigma = 1.4$ and $\overline{|E|} = 1.2$, $\sigma = 1.6$, respectively. These results are quite excellent, especially as the mean absolute error for extreme values approaches that of all values. At this threshold level, 91% of the target set is predictable.

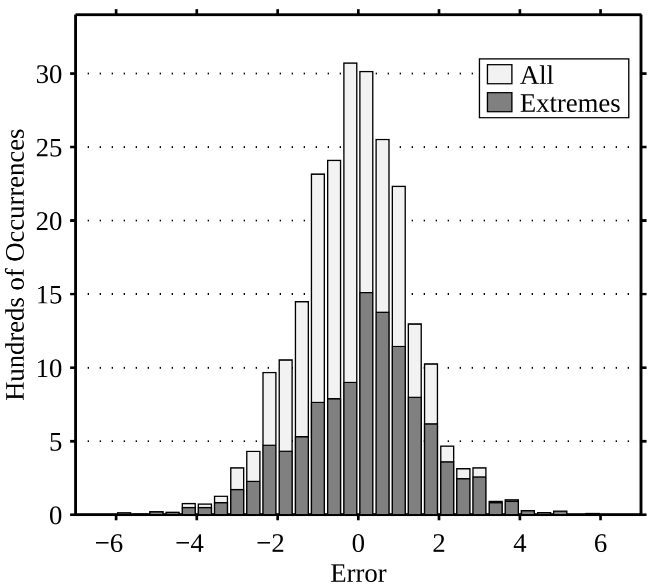

Figure 7: The distribution of errors for the Constrained Pearson r algorithm with $L = 0.6$.

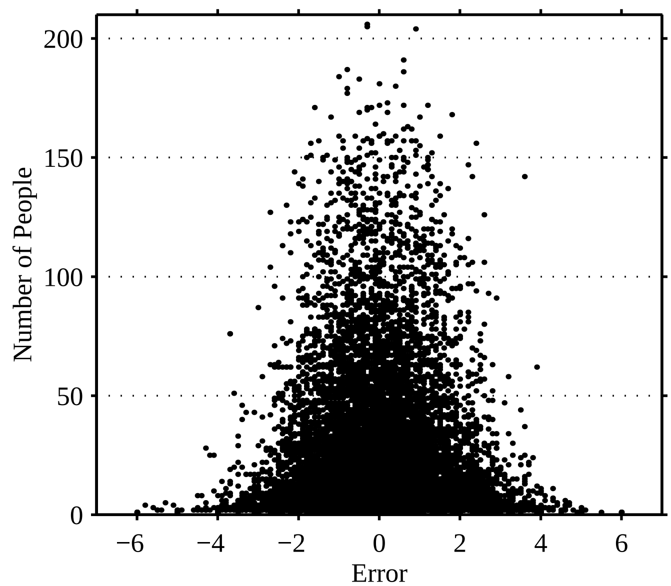

Figure 8: The scatter plot of the error vs. the number of people consulted to make the prediction.

QUALITATIVE RESULTS

Ultimately, what is more important than the numbers in the previous section is the human response to this new technology. As of this writing over 2000 people have used Ringo. Our source for a qualitative judgment of Ringo is the users themselves. The Ringo system operators have received a *staggering* amount of mail from users— questions, comments, and bug reports. The results described in this section are all based on user feedback and observed use patterns.

One observation is that a social filtering system becomes more competent as the number of users in the system increases. Figure 8 illustrates how the error in a recommendation relates to the number of people profiles consulted to make the recommendation. As the number of user scores used to generate a prediction increases, the deviation in error decreases significantly. This is the case because the more people use the system, the greater the chance are of finding close matches for any particular user. The system may need to reach a certain *critical mass* of collected data before it becomes useful. Ringo's competence develops over time, as more people use the system. Understandably then, in the first couple weeks of Ringo's life, Ringo was relatively incompetent. During these days we received many messages letting us know how poorly Ringo performed. Slowly, the feedback changed. More and more often we received mail about how "unnervingly accurate" Ringo was, and less about how it was incorrect. Ringo's growing group of regular "customers" indicates that it is now at a point where the majority of people find the service useful.

However, many people are disappointed by Ringo's initial performance. We are often told that a person must do one or two iterations of rating artists before Ringo becomes accurate. A user would rate the initial set, then receive predictions. If the user knows any of the predicted artists are not representative of their personal taste, they rate those artists. This will radically alter the members of the user's "similar user" neighborhood. After these iterations, Ringo works satisfactorily. This indicates that what is needed is better algorithm for determining the "critical" artists to be rated by the user so as to distinguish the user's tastes and narrow down the group of similar users.

Beyond the recommendations, there are other factors which are responsible for Ringo's great appeal and phenomenal growth. The additional features, such as being a user-grown database, and the provisions for reviews and dossiers add to its functionality. Foremost, however, is the fact that Ringo is not a static system. The database and user base is continually growing. As it does, Ringo's recommendations to the user changes. For this reason, people enjoy Ringo and use it on a regular basis.

RELATED WORK

Several other attemps have been made at building filtering services that rely on patterns among multiple users. The Tapestry system [2] makes it possible to request Netnews documents that have been approved by other users. However, users must themselves know who these similar people are and specifically request documents annotated by those people. That is, using the Tapestry system the user still needs to know which other people may have similar tastes. Thus, the social information filtering is still left to the user.

During the development of Ringo, we learned about the existence of similar projects in a similar state of development. One such example is Grouplens [4], a system applying social filtering to the personalized selection of Netnews. GroupLens employs Pearson r correlation coefficients to determine similarity between users. On our dataset, the algorithms described in this paper performed better than the algorithm used by Grouplens.

Two other recently developed systems are a video recommendation service implemented at Bellcore, Morristown, NJ and a movie recommendation system developed at ICSI, Berkeley, CA. Unfortunately, as of this writing, there is no information available about the algorithms used in these systems, nor about the results obtained.

The user modeling community has spawn a range of recommendation systems which use information about a user to assign that user to one of a finite set of hand-built, predefined user classes or *stereotypes*. Based on the stereotype the user belongs to, the system then makes recommendations to the user. For example [5] recommends novels to users based on a stereotype classification. This method is far less personalized than the social filtering method described in this paper. The reason is that in social filtering, in a sense every user defines a stereotype that another user can to some degree belong to. The number of stereotypes which is used to define the user's taste is much larger.

Finally, some commercial software packages exist that make recommendations to users. An examples is Movie Select, a movie recommendation software package by Paramount Interactive Inc. One important difference is that these systems use a data set that does not change over time. Furthermore, these systems also do not record any history of a person's past use. As far as can be deduced from the software manuals and brochures, these systems store correlations between different items and use those correlations to make recommendations. As such the recommendations made are less personalized than in social filtering systems.

CONCLUSIONS AND FUTURE WORK

Experimental results obtained with the Ringo system have demonstrated that social filtering methods can be used to make personalized recommendations to users. Ringo has been tested and used in a real-world application and received a positive response. The techniques employed by the system can be used to recommend books, movies, news articles, products, and more.

More work needs to be done in order to make social filtering applicable when dealing with very large user groups and a less narrow domain. Work is currently under way to speed up the algorithm by the use of clustering techniques, so as to reduce the number of similarity measures that need to be computed. We are also using clustering techniques among the artist data so as to identify emergent musical genres and make use of these distinct genres in the prediction algorithms.

Finally, we haven't even begun to explore the very interesting and controversial social and economical implications of social filtering systems like Ringo.

ACKNOWLEDGEMENTS

Carl Feynman proposed the mean squared differences algorithm and initially developed some of the ideas that led to this work. Karl Sims proposed the artist-artist algorithm. Lee Zamir implemented the WWW interface to Ringo and is currently responsible for maintaining the Ringo system. Alan Wexelblat and Max Metral provided useful comments on an earlier version of the paper. This work was sponsored by the National Science Foundation under grant number IRI-92056688 and by the "News in the Future" Consortium of the MIT Media Lab.

REFERENCES

[1] Special Issue on Information Filtering, Communications of the ACM, Vol. 35, No. 12, December 1992.

[2] David Goldberg, David Nichols, Brian M. Oki and Douglas Terry, "Using Collaborative Filtering to Weave an Information Tapestry", Communications of the ACM, Vol. 35, No. 12, December 1992, pp. 61–70.

[3] Pattie Maes, "Agents that Reduce Work and Information Overload", Communications of the ACM, Vol. 37. No. 7, July 1994, pp. 31–40.

[4] Paul Resnick, Neophytos Iacovou, Mitesh Sushak, Peter Bergstrom, John Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", to be published in the Proceedings of the CSCW 1994 conference, October 1994.

[5] Elaine Rich, "User Modeling via Stereotypes", Cognitive Science, Vol. 3, pp 335-366, 1979.

[6] Robert Rosenthal and Ralph Rosnow, "Essentials of Behavioral Research: Methods and Data and Analysis", McGraw Hill, second edition, 1991.

[7] Upendra Shardanand, "Social Information Filtering for Music Recommendation", MIT EECS M. Eng. thesis, also TR-94-04, Learning and Common Sense Group, MIT Media Laboratory, 1994.