Joseph A. Konstan, Bradley N. Miller, David Maltz,
Jonathan L. Herlocker, Lee R. Gordon, and John Riedl

GroupLens: Applying Collaborative Filtering to Usenet News

High volume and personal taste makes Usenet news an ideal candidate for collaborative filtering techniques.

Recommender Systems

THE GROUPLENS PROJECT DESIGNED, IMPLEMENTED, AND EVALUATED a collaborative filtering system for Usenet news—a high-volume, high-turnover discussion list service on the Internet. Usenet newsgroups—the individual discussion lists—may carry hundreds of messages each day. While in theory the newsgroup organization allows readers to select the content that most interests them, in practice most newsgroups carry a wide enough spread of messages to make most individuals consider Usenet news to be a high noise information resource. Furthermore, each user values a different set of messages. Both taste and prior knowledge are major factors in evaluating news articles. For example, readers of the rec.humor newsgroup, a group designed for jokes and other humorous postings, value articles based on whether they perceive them to be funny. Readers of technical groups, such as comp.lang.c++ value articles based on interest and usefulness to them—introductory questions and answers may be uninteresting to an expert C++ programmer just as debates over subtle and advanced language features may be useless to the novice.

The combination of high volume and personal taste made Usenet news a promising candidate for collaborative filtering. More formally, we determined the potential *predictive utility* for Usenet news was very high. The GroupLens project started in 1992 and completed a pilot study at two sites to establish the feasibility of using collaborative filtering for Usenet news [8]. Several critical design decisions were made as part of that pilot study, including:

- The requirement that GroupLens integrate with

existing news reading applications, since users are extremely reluctant to change news reader programs.
- The requirement that GroupLens support a single keystroke rating input (or, when possible, replacing an existing keystroke) since users typically spend very little time or attention on any particular article. (Other research has shown that more extensive textual ratings can be effective in close-knit communities [2, 4].
- The requirement that GroupLens provide predictions of the rating the system expects the user will give each article, rather than only winnowing down the list of articles. We consider it very important to provide advice rather than exercise censorship.

The pilot study, successful yet limited in scope, demonstrated that collaborative filtering could be implemented for Usenet news. Since then, the project has continued forward to undertake the challenge of applying collaborative filtering to a larger set of users and on a larger scale. Moreover, we have focused our efforts on overcoming some of the challenges of applying collaborative filtering to Usenet news, including:

- Integration of collaborative filtering into an information system with existing users, existing applications and interfaces, and an open architecture that supports many news reader applications.
- Addressing the dynamic, distributed nature of Usenet news. Articles have short lifetimes and there is no central repository of news articles.
- Working with extremely sparse sets of ratings. Typical users read only a tiny fraction of Usenet news articles.
- Delivering acceptable performance to users and providing mechanisms to scale the system as the number of users and articles grows.

This article discusses the challenges involved in creating a collaborative filtering system for Usenet news. The public trial of GroupLens invited users from over a dozen newsgroups selected to represent a cross-section of Usenet (listed in Table 1) to apply our news reader software to enter ratings and receive predictions (we provided GroupLens-adapted versions of Gnus, xrn, and tin). Over a seven-week trial starting February 8, 1996, we registered 250 users who submitted a total of 47,569 ratings and received over 600,000 predictions for 22,862 different articles. These users were volunteers who saw our announcement postings or our Web page. They downloaded specially modified news browsers that accepted ratings and displayed predictions on a 1–5 scale where 1 was described as "this item is really bad! a waste of net.bandwidth" and 5 as "this article is great, I would like to see more like it." For privacy reasons, users were known to us only by pseudonyms. Qualitative results are therefore the compilation of feedback from the GroupLens mailing list and private email rather than a comprehensive survey. In [5] we present a more detailed summary of the trial results, along with comparisons with noncollaborative approaches to managing Usenet news.

	Predict Good	Predict Bad
Desirable	HIT Movie: +high Legal Cite: +high Sci. Art.: +high Restaurant: +med	MISS Movie: −low Legal Cite: −very high Sci. Art.: −low Restaurant: −low
Undesirable	False Positive Movie: –$7+30 min Legal Cite: –med Sci. Art.: –5 min Restaurant: –high	Correct Rejection Movie: +med/high Legal Cite: +low/med Sci. Art.: +med/high Restaurant: +high

▲ **Figure 1.** Predictive utility cost/benefit analyses for four selected tasks. Different domains have different values for correct and incorrect predictions. Missing a desirable legal citation can be extremely costly, while missing a good movie is not since there are many desirable movies. Similarly, the cost of mistakenly picking an undesirable restaurant is higher than the cost of picking an undesirable science article due to the time and money invested.

rec.humor
rec.food.recipes
rec.arts.movies.current-films
comp.lang.c++
comp.lang.java
comp.groupware
comp.human-factor
mn.general
all groups in comp.os.linux.*

Table 1. Newsgroups supported in the public trial

Assessing Predictive Utility

Predictive utility refers generally to the value of having predictions for an item before deciding whether to invest time or money in consuming that item. For Usenet, the items are news articles, but the concept is general enough to include physical items such as books or videotapes as well as other information items. In each domain predictive utility is not sim-

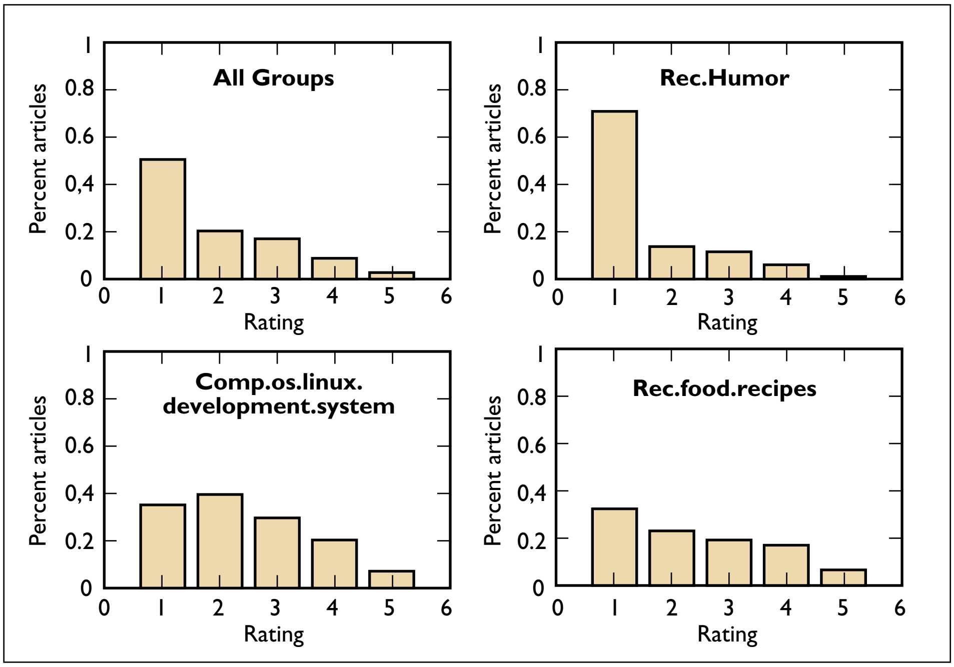

Figure 2. Ratings profiles for four Usenet news groups The percentage of articles assigned each rating varies significantly from newsgroup to newsgroup. Most articles in rec.humor were given the worst rating (1 out of a possible), while the ratings in comp.os.linux.development.system were distributed more uniformly.

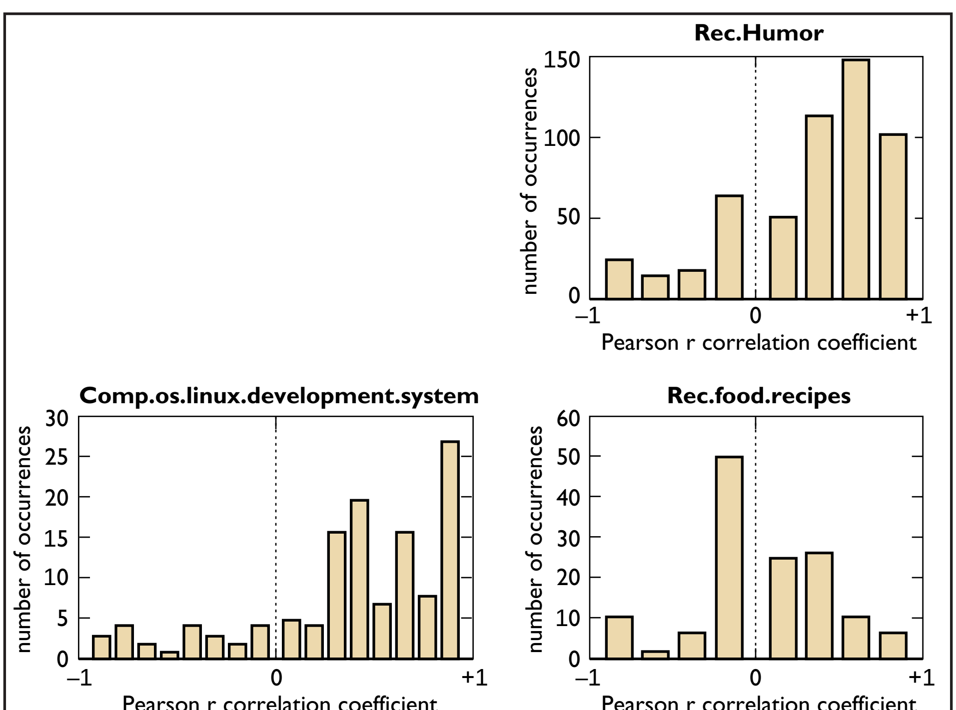

Figure 3. **User pair correlations for three newsgroups. One way to compare the similarity of users is to compute the Pearson coefficient between their ratings. Here, the number of user pairs with each Pearson coefficient is plotted for three different newsgroups. The presence of many high correlations in the rec.humor newsgroup indicates general agreement about quality in that domain. In the moderated newsgroup rec.food.recipes correlations are nearly evenly distributed about the origin, suggesting that individual taste matters more in this domain.**

ply a measure of accuracy; it is a measure of how effectively predictions influence user consumption decisions. A domain with high predictive utility is one where users will adjust their decisions a great deal based on predictions. A domain with low predictive utility is one where predictions will have little effect on user decisions.

Predictive utility is a function of the relative quantity of desirable and undesirable items and the quality of predictions. The desirability of an item is a measure of a particular user's personal value for that item. Items are not intrinsically good or bad.

The cost-benefit analysis for a consumption decision compares the value of consuming a desirable item (a hit), the cost of missing a desirable item (a miss), the value of skipping over an undesirable item (a correct rejection), and the cost of consuming an undesirable item (a false positive). Figure 1 shows four cost-benefit analyses. For watching a movie, the value of finding desirable movies is high to movie fans, but the cost of missing some good ones is low since there are many desirable movies for most movie fans. The cost of false positives is the price of the ticket plus the amount of time before the watcher decides to leave. The value of correct rejections is high because there are so many undesirable movies that it would be impractical to see movies at all without rejecting many of them.[1] Similarly, finding desirable general-interest scientific articles benefits from predictions since there are so many to select from (even though many are good thanks to peer review and editors). Restaurant selection follows a similar pattern though the risk of going to an undesirable restaurant is higher since you typically still have the meal and the bill. Legal research is very different. The cost of missing a relevant and important precedent is very high, and may outweigh the cost of sifting through all of the potentially relevant cases (especially when that cost is being billed to the client and serves as protection against malpractice).

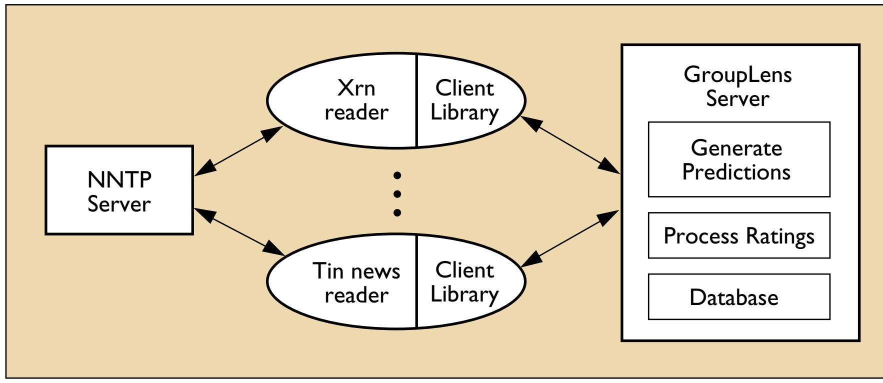

Figure 4. GroupLens architecture overview. Usenet clients connect to the GroupLens server through the GroupLens client library, and to a separate NNTP server as usual. The GroupLens Server accepts ratings and provides predictions for articles delivered by the NNTP server.

The costs of misses and false positives represent the *risk* involved in making a prediction. The values of hits and correct rejection represent the *potential benefit* of making predictions. Predictive utility is the difference between the potential benefit and the risk. Thus, the risk of mistakes is lowest for movies or scientific articles, and the potential benefit is highest for movies, articles, and restaurants.

One important component of the cost-benefit analysis is the total number of desirable and undesirable items. If 90% of the items being considered are desirable, filtering will generally not add much value over simply predicting that all items are desirable because there are few correct rejections and the probability of a hit is high even without a prediction. Of course when there are many desirable items, users may refine their desires to select only the most interesting of the interesting ones given their limited time. On the other hand, if there are many items and only 1% are good, then filtering can add significant value because the aggregate value of correct rejections becomes high requiring a very high miss cost before it becomes preferable to predict that all items are desirable.

Usenet news is a domain with extremely high predictive utility. While statistics vary by newsgroup, we have found that users generally consider only 5% to 30% of articles in typical newsgroups to be desirable. (Figure 2 shows the distribution of ratings for the most widely rated technical, recreational, and moderated newsgroups from the trial.) Because of the high volume of news, the value of correct rejections is high (in many groups it is infeasible to read the entire group). At the same time, the fact that so many users read Usenet articles implies the value of a hit is also moderately high. Thus, Usenet has a high potential benefit. It also has low risk. False positives are certainly annoying, but it takes only a few seconds for a user to dismiss an unwanted article. And misses turn out to be low cost as well since truly valuable articles tend to reappear in follow-up discussion, reducing the chance of missing something particularly important. Later, we show the effect of predictions on user behavior to confirm high-predictive utility.

We should point out that high-predictive utility implies that any accurate prediction system will add significant value—why then do we need a personalized collaborative filtering system? Would it not be easier to simply calculate average ratings across all users as was done by Maltz [3] and reap the benefits of high-predictive utility? We have found that personalized predictions are significantly more accurate than nonpersonalized averages. In general, users do not agree on which articles are desirable. Figure 3 shows that users do not agree overall. The group rec.humor has unusually high agreement, primarily due to a large number of cross-posted articles that do not even attempt to be funny, but there are a substantial number of low and negative user-pair correlations. Rec.food.recipes, a group in which agreement literally is based on taste, has a large number of near-zero correlations that we believe represent people with overlapping but different tastes, such as a vegetarian and a

[1]Our analysis includes the effect of frequency of occurrence in the cost or benefit. Hence, correct rejections are worth more when there are many undesirable items. If we isolate frequency of occurrence, then the benefit of a correct rejection is zero since its value is simply the absence of the cost of a false positive. We find the combined analysis more intuitive, though separating the frequency from the per-item cost can be useful for some analyses.

meat-eater who both enjoy chocolate desserts. Hence, it is better not to lump all votes together since there are systematic differences in taste. Moreover, even in an area where users agree overall, such as rec.humor, Table 2 shows that correlation between ratings and predictions is dramatically higher for personalized predictions than for all-user average ratings.

GroupLens Architecture Overview

The GroupLens system architecture is designed to blend into the existing Usenet client-server architecture. At a high level, Figure 4 shows that a news reader such as xrn, tin, or Gnus connects to two servers: The NNTP server that holds Usenet news articles and the GroupLens server that holds ratings and generates predictions. The GroupLens client library encapsulates the interface to the server. The typical usage pattern is for a news reader to request a set of headers for unread articles from the NNTP server and pass the article identifiers to the GroupLens client library to obtain predictions. As the user reads articles in the newsgroup, the news reader records ratings with the client library which sends them back to the server. The server uses these ratings both to provide predictions to other users and to better capture this user's tastes.

One of the major challenges distinguishing Usenet news from other domains that have been used to demonstrate the value of collaborative filtering is that Usenet is a real, preexisting system with millions of users and hundreds of software components already written. In some ways, building collaborative filtering into an existing domain provided us with significant benefits. We already knew the information resource was useful, as attested to by the millions of users already reading Usenet news. We also did not have to worry about content creation, since tens of thousands of articles are posted daily. We already had a natural partitioning of content into hierarchical newsgroups that evolved through a democratic voting process and were likely to represent real clusters of content and interest.

In other ways, however, working with Usenet news raised research problems. Two important problems were the need to integrate into preexisting clients and the integration of predictions with different news presentation models.

The problem of integrating with the sheer volume and diversity of news readers led us toward the client library and an open architecture model [6]. A quick survey showed over a dozen widely used news readers, and typically several versions of each in active use. These news readers ranged from

Newsgroup	Avg	Pers
rec.humor	0.49	0.62
rec.food.recipes	0.05	0.33
comp.os.linux.development.system	0.41	0.55

Table 2. Correlations between ratings and predictions for average and personalized predictions

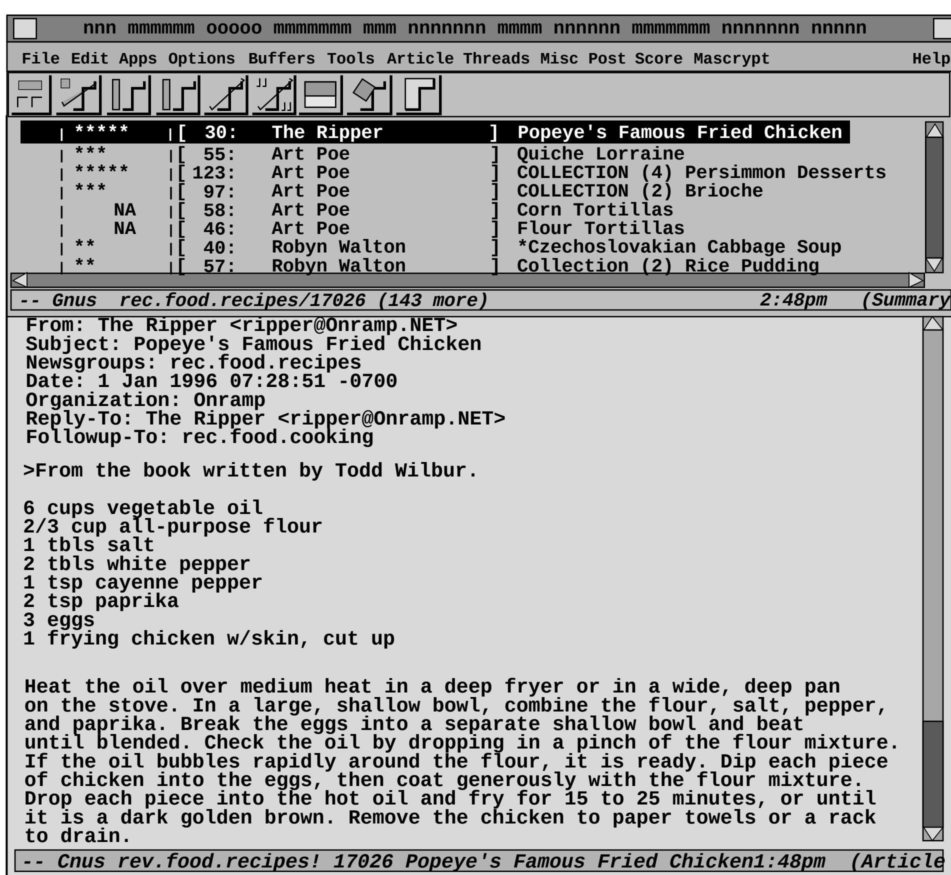

Figure 5. The Gnus interface with GroupLens predictions are shown here. Predictions are indicated as an ASCII bar-chart on the left edge of the summary part of the interface. The longer bars indicate articles that are predicted to be of greater interest.

text-only to graphical to web-based and ran on every platform including Macintosh, DOS, Windows, and Unix platforms. We quickly determined it was infeasible for us to update and maintain a fleet of news readers. Instead, we would need to make it easy for news reader authors to incorporate GroupLens into their own code. Since there was no standard protocol for exchanging ratings and predictions, we defined an open protocol for communication between news readers and the GroupLens server. To further simplify the task of caching data and following the protocol, we implemented and distributed client libraries written in C and in Perl.

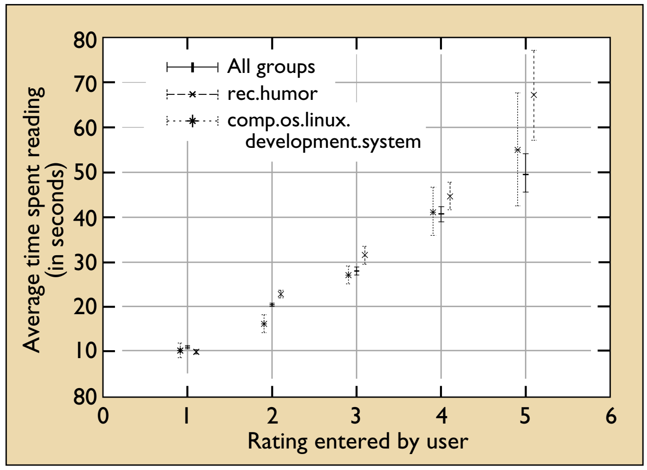

Figure 6. Correlation between time spent reading and explicit ratings. Readers who spend a long time with an article are more likely to rate it highly. The points mark the average time spent reading an article for each rating, while the ranges span the 95% confidence interval from that mean.

The client libraries define a simple API that news readers can use to request predictions and to transmit ratings. They also define utility functions to manage a user's initialization file and to provide user-selectable display formats for predictions. We consider the client library and its API to be a substantial success as we've found several news reader authors and one user willing to use it to provide GroupLens support. (GroupLens support is provided or forthcoming in Gnus 5.2 and SLRN 0.8.8.5.) One of our test users in Poland wrote a proxy GroupLens server to download ratings and predictions each evening to help him deal with network throughput as low as 10bps. This type of user participation can only come about with an open and protocol and a usable API.

The problem of integrating predictions into different presentation models was more formidable. The original GroupLens system was designed for news readers in which the user selected a newsgroup and was then given a split screen with one part containing a list of unread articles (in either chronological or discussion-thread order) and the other part showing the text of the currently selected article. In this presentation model, it is simple and effective to display predictions along with other header information to help users choose which articles to read and which to skip. An example of this interface is the Gnus interface shown in Figure 5. Several news readers have adopted other interface models that are more difficult to integrate predictions into. Some discussion-thread news readers, for example, show only a single entry for each thread. It is not clear what prediction value should be shown for this entry: the average prediction, the first prediction, the maximum, the range, or some other value. This problem requires further research.

In part, the challenge of effectively integrating predictions into different presentation models stems from competing goals of users reading news. Users typically want to read news in roughly chronological order, grouped by discussion thread. When predictions are provided, users add the goal of reading news in order of decreasing quality, so they can read the good things first and then bail out of the newsgroup. We found that a new interface component added to one news reader, a keystroke to move to the highest-predicted unread article, was extremely popular in the rec.humor newsgroup where discussion threads were rarely rated highly and chronological order was less important.

The diversity and sheer number of installed news readers led us to adopt a library and open protocol approach. With this approach the implementers of each news reader could easily add access to the GroupLens server and could also use the returned predictions in whatever manner they found to be most consistent with their news reader interface.

A Dynamic and Fast-Paced Information System

Item volume and lifetimes are another way in which Usenet news differs from other domains where collaborative filtering has been applied. Across all newsgroups, users will see 50,000 to 180,000 new messages each day, and the volume of postings is doubling each year. The useful lifetime of a Usenet message is short; most sites expire messages after approximately one week. Furthermore, there is no central authority or official repository of Usenet news articles. Usenet is a truly distributed system where articles appear at different sites at different times, and there is no unique timestamp or sequence.

The implications of the high volume and fast pace of Usenet news include:

- The need for GroupLens to discover new content when it first learns about it, that is at the first rating or request for predictions.
- The need for ratings to affect subsequent predictions almost immediately—a delay of a full day would result in no predictions for as many as half of the system users, and even a delay of 5-10 minutes would result in large prediction gaps during the "morning rush" when many users read news.

To address these implications, the GroupLens server has a two-part database (shown in Figure 6). The *ratings* database stores all ratings that users have given to messages. The *correlations* database stores information about the historical agreement of pairs of users. The GroupLens architecture has three separate process pools that access these databases. The prediction processes always have the highest priority. They read both correlations and ratings and generate predictions in real time based on the latest available data. The ratings processes have the next highest priority. They write ratings into the ratings database and are expected to do so quickly to ensure that current data is available for generating predictions. The ratings processes are also responsible for identifying new articles and adding them into the database. Ratings, for both existing and new articles, are almost always stored into the database within 60 seconds of the time they are received. Finally, the correlation process reads the ratings database to update the correlations database. This process is scheduled so that each user pair's correlation is updated approximately every 24 hours. Since correlations are measures of historical agreement, they should not change rapidly. New users can be correlated individually after their first batch of ratings to make it possible for them to use the system quickly.

Ratings Sparsity

Users of Usenet news read only a small fraction of the articles posted to the system. Our studies found that users take an average of 10- to 60-seconds to read an article. Even using the conservative estimate of 10-seconds, users can read only 360 articles in an hour. Even a user reading news several hours each day will struggle to read 1% of all articles posted. Of course, we are heartened by this fact because it points to the value of filtering. But sparsity also poses a problem for collaborative filtering:

- When each user has read a tiny percentage of the total number of articles, it becomes more difficult to find other users with whom to correlate, since the overlap between users is small on average and we devalue correlations with too few common ratings to avoid spurious correlations. Worse yet, there is not a set of very popular news articles, unlike box office hits for the movie domain or best-sellers in the book domain.
- A consequence of sparsity is that an enormous number of raters is needed to cover all of the articles. Until then, many users will experience the "first-rater problem" of finding articles with no prediction whatsoever.

GroupLens addresses the challenge of sparsity: algorithmically and at the user interface. The primary algorithmic technique for attacking sparsity is partitioning the set of Usenet news articles into clusters that are commonly read together. The newsgroup hierarchy provides a natural partitioning that successfully identifies clusters of articles. We partition our ratings database by newsgroup and thereby improve the local density of ratings. We also partition our correlations database by newsgroup to ensure that users can be clustered with other users who have read and rated the same articles. Essentially, we have created a subset of Usenet news where users are known to read a greater percentage of content, compared with Usenet overall, and therefore where there are likely to be enough common ratings to compute meaningful correlations. Partitioning the database by newsgroup also provides more accurate predictions. The user pair correlations shown in Figure 3 provide sufficient agreement to generate meaningful predictions. Retrospectively, using the data to make predictions based on correlation across all newsgroups provided lower correlations and less accurate predictions. This data confirms our hypothesis that agreement in one domain (such as humor) is not necessarily predictive of agreement in a different domain (such as recipes) and suggests that

One other approach to sparsity that we are examining is the incorporation of agent-style filter-bots into the GroupLens framework.

approaches that simply model users uniformly across domains are diluting their predictive power.

Even partitioning articles into newsgroup clusters doesn't fully address the sparsity concerns. During our trial, it was still the case that users rated as few as 1% to 2% of articles in high-volume newsgroups. While a Pearson correlation coefficient-based prediction algorithm was able to generate useful predictions (as shown in Table 2), we identified opportunities for increased accuracy if the ratings density could be improved. We identified two causes for this sparsity:

- Efficiently reading high-volume groups requires being highly selective. We apply collaborative filtering specifically to help users be selective, but the result is they skip over articles that don't interest them, either due to topic or low prediction.
- Even users who have read articles often do not rate them, even though the ratings interface involves at most one additional keystroke. Informal feedback suggests users are "lazy" in that they would prefer not to even think about the appropriate rating. Being advised that each rating helps perfect their own profile motivates some users, but others will avoid rating nonetheless.

Some researchers have proposed compensation systems that reward users for entering ratings. While the economic consequences of this solution are interesting, we wonder whether compensation would be necessary if ratings could be captured without any effort on the part of the user.[2] (We believe an ideal solution is to improve the user interface to acquire *implicit* ratings by watching user behaviors. Implicit ratings include measures of interest such as whether the user read an article and, if so, how much time the user spent reading it. Our initial studies show that we can obtain substantially more ratings by using implicit ratings and that predictions based on time spent reading are nearly as accurate as predictions based on explicit numerical ratings. Figure 6 shows an analysis of the relationship between time spend reading and explicit ratings. Our results also provide large-scale confirmation of the work of Morita and Shinoda [7] in finding the relationship between time and rating holds true without regard for the length of the article. We are continuing to explore further implicit ratings

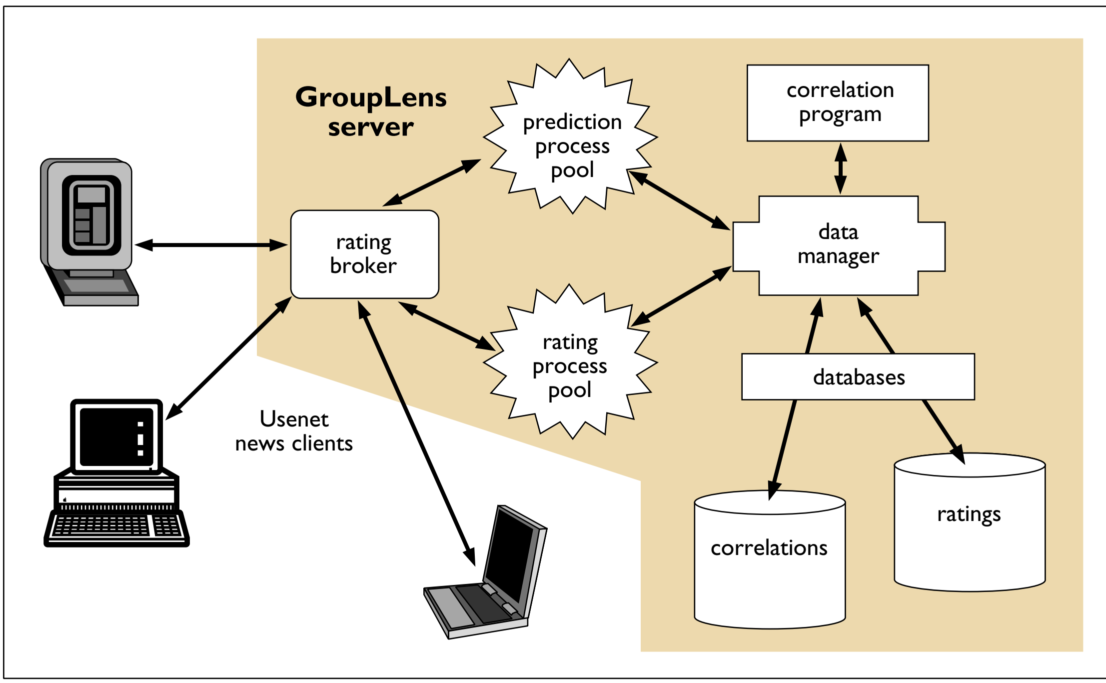

Figure 7. GroupLens server architecture. The beige box encloses the GroupLens server. The ratings broker serves as a single point of contact for clients to the server.

[2]Indeed, in this issue Avery speculates that even no-cost rating may not be cheap enough, since there is a positive benefit to waiting long enough for others to filter information for you. While we have observed this phenomenon, we expect that other factors, including the desire of many readers to read the most current articles at specific times of the day, will mitigate this desire to wait for predictions.

for Usenet including using actions such as printing, saving, forwarding, replying to, and posting a follow-up message to an article. Of course, other domains also have their own implicit ratings (for example, a library may record borrowing a book as an implicit rating in favor of the book).

One other approach to sparsity that we are examining is the incorporation of agent-style *filter-bots* into the GroupLens framework. Filter-bots are programs that read all articles and follow an algorithm to rate them systematically. Since they are automated, they can read and rate each article as soon as it is visible at their location. In GroupLens, they are treated as just another set of ordinary users; if a user correlates well with a filter-bot, then the filter-bot will contribute to predictions for that user. We are experimenting with a range of simple filter-bots that examine syntactic properties such as whether an article is a reply or an original message, degree of cross-posting to different newsgroups, the length and reading level of an article, among others.

> **Once users invest time in GroupLens, we have found they like the system and are likely to continue using it.**

Performance Challenges

The final set of challenges inherent in the Usenet news domain are the severe demands for low latency and high throughput to make it feasible to attract and serve a large number of users. The critical performance measures are the latency for handling prediction requests and ratings submissions and the throughput of the system measured by the number of users and articles that a GroupLens server can handle before performance degrades unacceptably. After examining the critical path at the user interface, we discovered that most news readers would be unable to request predictions or send ratings asynchronously. Accordingly, we established these performance goals based on the assumption that requesting predictions would delay the appearance of the articles in a newsgroup and that transmitting ratings would delay the return to newsgroup selection mode:

- A request for predictions for 100 articles in a newsgroup should complete in under two seconds (end to end) at least 95% of the time.
- A transmission of ratings for 100 articles (including any implicit ratings) should complete in under one second (end to end) at least 95% of the time.

There are several techniques that we were able to employ to help improve latency. A newsgroup can have several ratings and prediction processes active so multiple requests can be handled concurrently. The GroupLens ratings broker assigns each incoming request to a free process which can then fulfill the request as shown in Figure 7. Ratings processes release the client as soon as the ratings are received and write the ratings to the database afterwards, allowing the user to return to reading news as quickly as possible. Finally, we organized our database to store ratings so the correlation and prediction processes can efficiently retrieve either all ratings from a given user or all ratings for a given message.

Using a Sun Sparcstation 5 workstation as the server, we were able to surpass the ratings latency goal (100 ratings required approximately 250 ms) during the trial. We did not meet our prediction latency goal, however, as 100 predictions averaged just over four seconds. Later performance tuning, including the use of more memory, has allowed us to reduce the latency to approximately 150 ms for 100 ratings and below 500 ms for 100 predictions.

The primary throughput goal for the trial was to be able to handle 10,000 users for up to 20 Usenet groups. While we never had active usage at that level, we ran several experiments with simulated users (that interacted through the standard client library interface) and found that 10,000 users was realistic even if users concentrated their news reading into only 1/3 of the day. Obviously 10,000 users and 20 newsgroups are only a tiny fraction of Usenet. To achieve the scale needed for Usenet as a whole requires applying additional throughput enhancements:

- **Partitioning the server by newsgroup**. Separate servers can handle different newsgroups with nearly perfect parallel speed-up (only log-in costs are replicated).
- **Partitioning the server by user.** Different clusters of users can be assigned to different servers. Partitioning would be particularly effective if user clusters are based on historical agreement, but our trial suggests that even random assignment within a newsgroup would provide enough agreement to obtain useful predictions.

- **Use of composite users.** When millions of users are involved, even replication may be impractical. In that case, prototype users can be defined and users can be defined as combinations of those prototypes. Readers would obtain predictions based on the prototypes and their ratings would feed back into the prototypes to update the prediction profile. Users would still receive personalized predictions, but these predictions would be based on a personal combination of composite user opinions rather than a combination of individual user ratings.

Discussion and Conclusions

Usenet news is a domain that can greatly benefit from collaborative filtering, but it poses many challenges that will help us build more efficient and effective collaborative filtering systems. The GroupLens project is a notable success in collaborative filtering. Usage data gathered during a seven- week public trial shows that predictions are meaningful and valuable to users. To verify that this success was not caused by the bias of a prediction on a user's rating, we repeated our analysis retrospectively on users who did not see predictions before entering a ratings, and found the same results. Most notably, however, we found that users valued prediction because they tended to read and rate articles with high predictions more than those with low predictions as shown in Figure 8.

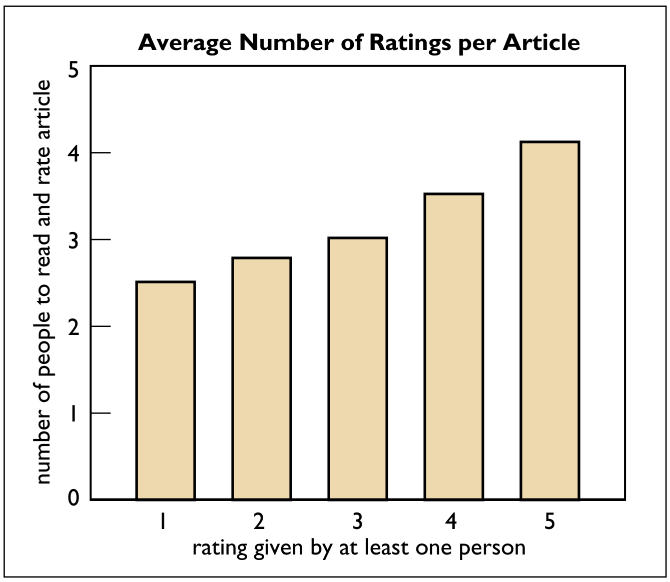

Figure 8. Number of people who read an article based on the rating it was given by some other user. For each rating of an article, the number of users who read and rate it is counted. The totals show that highly rated articles are read more often than less highly rated articles.

In addition to quantitative results, we gathered substantial anecdotal evidence about the challenges and successes of providing collaborative filtering for Usenet news. The start-up problem is composed of two parts:

- Users need to rate several articles before they can receive predictions. Accordingly, many users abandon the system before ever receiving benefits from it because they perceive effort without reward.
- Early adopters find there are not many other raters and therefore they receive predictions for only a fraction of the articles that they read.

We can address these problems in three ways. First, we can provide some predictions, if only the average rating for all users, so new users see some value in the system. Second, the use of implicit ratings reduces or eliminates the perceived effort, making it more likely that users will continue using the system. Third, we can combine the use of implicit ratings and the use of filter-bots to create faster perceived payback for reduced effort. We are experimenting with these approaches now.

Once users invest time in GroupLens, we have found they like the system and are likely to continue using it. While we found that more than half of the users who signed up for GroupLens discontinued active rating after a couple of weeks, many of the trial users were still using the system six months after the trial ended. Users often commented they would like GroupLens for all of their Usenet newsgroups, though we do not have the resources to serve that large a population and data set except perhaps with an overall average prediction rather than personalized predictions.

Usenet presents a different set of challenges to collaborative filtering than domains such as music [12] or movies [4] where new items are relatively infrequent and lifetimes are relatively long. In addition to addressing critical performance issues, the GroupLens system continues to address several key problems involving ratings sparsity and start-up usage by applying techniques including partitioning the system by newsgroup (which provides more accurate predictions), using implicit ratings, and exploring the use of filter-bot rating agents. We still have several interface challenges to address, including filtering and display interfaces that handle threads, integration with search engines such as InReference and DejaNews,[3] and other ways of making predictions more useful to users. We also are very interested in comparing GroupLens with, and exploring the integration of collaborative filtering with, information retrieval approaches to filtering information such as the SIFT system [10].

We are often asked "What would it take to make all of Usenet use GroupLens?" The answer involves performance, availability, and convincing users to use the system. Our current architecture and implementations support 10,000 users for 10 to 20 newsgroups on a single economical workstation. Partitioning could allow us to economically expand to cover all of Usenet for tens of thousands of users, or to cover specific newsgroups for all users, but probably will not allow us to support all groups for all users. Considering that Usenet news already relies upon a wide network of servers, we believe that creating a worldwide network of GroupLens servers is a practical and feasible approach to collaborative filtering for all of Usenet.

Availability is determined almost entirely by the willingness of news reader authors to incorporate GroupLens into their systems. We have received very positive feedback on both our client library and our open architecture. These tools make adding GroupLens quite easy, especially compared with the effort undertaken to communicate with the NNTP (news) server. The remaining hurdle is to provide the ground swell of support that requires the existence of servers supporting most or all Usenet newsgroups. We believe most users will prefer having GroupLens predictions, though they may prefer not to have to do any work to enter ratings. For this reason, we believe implicit ratings are critical for convincing users to use the system.

In conclusion, GroupLens collaborative filtering for Usenet news is an experimental success and it shows promise as a viable service for all Usenet news users. We are currently conducting a second public trial. This trial will test the effect of providing predictions to new users more rapidly by providing overall averages until the user has rated enough articles to correlate, and it will make full use of time spent reading measures to capture implicit ratings unobtrusively. Readers interested in using GroupLens, in adapting their own news readers to use GroupLens, or in following the ongoing trial, are invited to the GroupLens home page at http://www.cs.umn.edu/Research/GroupLens. C

[3]InReference (http://www.reference.com/) DejaNews (http://www.dejanews.com/)

Acknowledgments

Many people participated in making GroupLens a success. Paul Resnick deserves special recognition for cofounding the project with John Riedl. Also, Danny Iacovou, Mitesh Susak, and Pete Bergstrom who worked on earlier versions of the system.

References

1. Goldberg, D., Nichols, D., Oki, B. and Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM 35*, 12. (1992) pp. 61–70.
2. Hill, W., Stead, L., Rosenstein, M. and Furnas, G. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems* (1995). ACM, New York, pp. 194–201.
3. Maltz, D. Distributing information for collaborative filtering on Usenet net news. Master's thesis, M.I.T. Department of EECS, Cambridge, Mass. May 1994.
4. Maltz, D. and Ehrlich, K. Pointing the way: Active collaborative filtering. In *Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems*. (1995). ACM, New York.
5. Miller, B., Riedl, J. and Konstan, J. Experiences with GroupLens: Making Usenet useful again. In *Proceedings of the 1997 Usenix Winter Technical Conference*. Jan. 1997.
6. Miller, B., Riedl, J., Konstan, J., Resnick, P., Maltz, D. and Herlocker, J. The GroupLens Protocol Specification. http://www.cs.umn.edu/Research/GroupLens/protocol.html.
7. Morita, M. and Shinoda, Y. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of SIGIR '94*. ACM, New York.
8. Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., and Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 Computer Supported Cooperative Work Conference*. (1994) ACM, New York.
9. Shardanand, U. and Maes, P. Social information filtering: Algorithms for automating "word of mouth." In *Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems*. (1995) ACM, New York, pp. 210–217.
10. Yan, T. and Garcia-Molina, H. SIFT: A tool for wide-area information dissemination. In *Proceedings of the USENIX 1995 Winter Technical Conference*. (Jan. 1995) Usenix Assoc., New Orleans, La .

Joseph A. Konstan (konstan@cs.umn.edu) is an asistant professor of computer science at the University of Minnesota, Minneapolis. He is also cofounder and consulting scientist at Net Perceptions, a new company developing and marketing GroupLens collaborative filtering software based in Eden Prairie, Minn.

Bradley N. Miller (bmiller @netperceptions.com) is cofounder and vice presidnet of product development at Net Perceptions, Inc., Eden Prairie, Minn.

David Maltz is a Ph.D. student in computer science at Carnegie Mellon University studying mobile networking and computer-supported cooperative work.

Jonathan L. Herlocker (herlock@cs.umn.edu) is a Ph.D. student in computer science at the University of Minnesota where he is conducting research on flexible multimedia authoring and playback systems.

Lee R. Gordon (lrg@netperceptions.com) is a senior software engineer at Net Perceptions, Inc., Eden Prairie, Minn., and a M.S. candidate at the University of Minnesota, Minneapolis.

John Reidl is an associate professor of computer science at the University of Minnesota. He is also cofounder and chief technical officer of Net Perceptions. Inc. (http://www.netperceptions.com)