PolyLens: A Recommender System for Groups of Users

Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA
{oconnor;cosley;konstan;riedl}@cs.umn.edu

Abstract. We present PolyLens, a new collaborative filtering recommender system designed to recommend items for groups of users, rather than for individuals. A group recommender is more appropriate and useful for domains in which several people participate in a single activity, as is often the case with movies and restaurants. We present an analysis of the primary design issues for group recommenders, including questions about the nature of groups, the rights of group members, social value functions for groups, and interfaces for displaying group recommendations. We then report on our PolyLens prototype and the lessons we learned from usage logs and surveys from a nine-month trial that included 819 users. We found that users not only valued group recommendations, but were willing to yield some privacy to get the benefits of group recommendations. Users valued an extension to the group recommender system that enabled them to invite non-members to participate, via email.

Introduction

Recommender systems (Resnick & Varian, 1997) help users faced with an overwhelming selection of items by identifying particular items that are likely to match each user's tastes or preferences (Schafer et al., 1999). The most sophisticated systems learn each user's tastes and provide personalized recommendations. Though several machine learning and personalization technologies can attempt to learn user preferences, automated collaborative filtering (Resnick et al., 1994; Shardanand & Maes, 1995) has become the preferred real-time technology for personal recommendations, in part because it leverages the experiences of an entire community of users to provide high quality recommendations without detailed models of either content or user tastes.

To date, automated collaborative filtering systems have focused exclusively on recommending items to individuals. In some domains, such as Usenet News (Konstan et al., 1997; Resnick et al., 1994), this limitation is understandable. Few users read articles collectively. In other domains such as books or music (Shardanand & Maes, 1995), it is common both to enjoy the media alone and in groups. (Indeed, the MusicFX system (McCarthy & Anagnost, 1998), which did not attempt to use collaborative filtering, was designed specifically to address the challenge of selecting music for the often-large groups of people using a corporate gym.) Moreover certain items, among them restaurants, board games, and movies (Hill et al., 1995), are more commonly enjoyed in groups. Recommender systems that identify items such as movies for individuals do not address the user's key question, which is not "what movie should *I* see?" but rather "what movie should *we* see?"

This paper explores the design space of collaborative filtering recommenders for groups and presents our experience deploying the PolyLens group recommender to over 800 MovieLens users. The design space includes issues such as: What is the nature of a group? How are groups formed? How are recommendations computed for groups? What interfaces are best for sharing recommendations with groups? What are the privacy issues in showing recommendations to groups?

In our field trial with PolyLens we explored user experiences with one set of design choices. We kept detailed logs to measure how users formed groups, how they used the groups, and how the experience of users who used groups differed from the experience of users who did not use groups. We also surveyed the group users to learn their reactions to group recommendations, including their opinions about the value of the recommendations and the tradeoff in lost privacy.

In the next section, we present related work in recommender systems and in other systems with related group dynamics issues. We then introduce PolyLens and review the design space for group recommenders, looking at group properties and member rights, algorithms for group recommendation, and interfaces for displaying group recommendations. We follow with results of our user trial and survey, and we conclude with a discussion of lessons learned.

Related Work

Though we know of no previously published studies of groups in recommender systems, the work is related to previously published work on collaborative filtering, group formation, roles in collaborative systems, and awareness in collaborative systems.

Collaborative filtering. Many different approaches have been applied to the basic problem of making accurate and efficient recommender systems, ranging from nearest neighbor algorithms to Bayesian analysis. The earliest recommenders used nearest neighbor collaborative filtering algorithms (Resnick et al., 1994; Shardanand & Maes, 1995). Nearest neighbor algorithms are based on computing the distance between users based on their preference history. Predictions of how much a user will like an item are computed by taking the weighted average of the opinions of a set of nearest neighbors for that product. Opinions should be scaled to adjust for differences in ratings tendencies between users (Herlocker et al., 1999).

Model-building methods work by creating a model offline, and then running the model online. The model may take hours or days to build. The goal is for the resulting model to be small, fast, and accurate. Several techniques have been shown to be successful, including: (1) Bayesian networks, which create a model based on a training set with a decision tree at each node and edges representing user information (Breese et al., 1998); (2) dimensionality reduction using eigenvectors (Goldberg et al., 2000) or singular value decomposition (Sarwar et al., 2000), which creates a low-dimensional space within which latent relationships between users or items can be discovered; (3) clustering techniques, which identify groups of users who appear to have similar preferences (Ungar & Foster, 1998); and (4) Horting, a graph-based technique in which nodes are users, and edges between nodes indicate the degree of similarity between two users (Aggarwal et al., 2000).

In this paper we focus on the basic question of whether group recommendations can be useful to users, so we use nearest neighbor algorithms. These algorithms are appropriate for our purpose since they are the most thoroughly studied, and since our users on MovieLens are most familiar with nearest neighbor algorithms.

Group formation. Many studies have examined systems that support group formation. Two interesting ends of the spectrum are Kansas and MusicFX. Kansas is a virtual world in which a user can join a group by moving towards other users (Smith et al., 1998). Kansas groups are similar to chatting in physical spaces in an office environment. MusicFX accidentally enabled group formation by creating a system in which the music in a corporate gym was selected according to the taste of the people working out at a given time (McCarthy & Anagnost, 1998). People began modifying their workout times to arrive at the center with other people, often strangers, who shared their music tastes.

Our work involves intentional groups, like Kansas, and unlike MusicFX. Unlike Kansas, the groups are explicitly selected for an external reason: these people want to see a movie together.

Roles. Roles for participants in collaborative systems have been studied by many researchers. For instance, Kansas includes an object-oriented programming language within which arbitrary roles can be described (Smith et al., 1998; similar to Edwards, 1996). Roles confer rights and responsibilities on the users. Other collaborative systems provide mechanisms to directly control the operations users can perform (Dewan & Shen, 1998).

Experimental work on roles in practice suggests that complicated roles are not necessary in many cases, since social protocols will evolve to manage the interaction tasks. For instance, studies of the Grove and Aspects multi-user editors show that users avoid conflict effectively without system support (Ellis et al., 1991). Further, one study of a large number of Lotus Notes databases shows that even though users say they expect moderators to increase the amount of communication in Notes discussion boards, the moderators actually significantly decrease the amount of communication (Whittaker, 1996). For these reasons, our work uses only very simple roles and permissions, which we address when we discuss design issues.

Awareness. One of the necessary conditions for social protocols to evolve is awareness of other users in the system. For instance, user group drawing tools often show where other users are in the drawing space, and what objects they are currently manipulating (Gutwin & Greenberg, 1998). Experience with group writing tools shows that awareness is important for group dynamics (Mitchell et al., 1995). The Prep collaborative editor treats awareness of other users explicitly, by creating separate columns to record the edits performed by each user (Neuwirth et al., 1994). Our work follows the Prep model of making awareness explicitly visible through separate columns of data about each user.

Designing PolyLens

PolyLens is a group recommender extension to the MovieLens recommender system (<http://movielens.umn.edu/>). MovieLens is a free movie recommender site with over 80,000 users and their ratings of over 3,500 movies (with a total of nearly 5 million ratings). MovieLens users rate movies on a five-star scale.

The MovieLens front page shows users several lists of recommended films, including movies in theaters and movies recently released on video tape or DVD formats. The front page also provides access to special features, experiments, and a query interface. Users may search for movies by title, retrieving a list of matching movies with predicted ratings, or may select categories of movies by date and genre, retrieving lists of recommendations sorted by prediction.

PolyLens was integrated with MovieLens in three places. New links were added to the front page to allow users to create or manage groups; a new field was added to the query interface to allow users to select whether they were looking for group or individual recommendations; and a membership consent interface was added to alert users who were invited to join groups of their pending invitation.

Our goals in designing PolyLens were to:

- gain experience with the design and use of group recommenders;
- create a system that MovieLens users would find valuable;
- simplify the implementation by using our existing infrastructure; and
- keep the trust of MovieLens users by implementing policies and algorithms that respected their privacy and presented accurate recommendations.

We explicitly were not trying to experimentally identify the best design or to compare design alternatives in any systematic way. We knew that several of our goals depended upon having happy users, and therefore focused our efforts on designing a system that would satisfy our users.

Given the novelty of group recommenders, we were forced to start design from scratch. We identified five specific questions about groups and membership, group recommendations, and group interfaces. We reviewed the design alternatives in each area, choosing those that seemed appropriate for our goal of supporting groups of people who were going to see a movie together. In this section, we discuss those five design questions, the alternatives we explored, the choices we made for PolyLens, and different design goals for which alternative designs would be more appropriate.

What is the nature of a group?

Are groups ephemeral or persistent? Public or private?

The persistence of groups is an issue related to both usage patterns and privacy issues. If users want to repeatedly receive recommendations for the same group of people, it saves time and effort to make the group persistent. On the other hand, if groups form and dissociate for a single use, ephemeral approaches better meet the need. This issue interacts with consent (discussed below) as a time-intensive consent process may render ephemeral groups inconvenient, but users may be willing to reduce the amount of consent required if they know the group is only used once, or only in their presence.

A related issue is whether groups are private, known only to group members, or public and accessible to all. Public groups can become community meeting places of sorts (e.g., the *Titanic-haters* group) or even soapboxes where famous critics such as Roger Ebert and Richard Roeper might define a new taste. Private groups serve as clubs where a group of friends or family may share tastes away from the noise and scrutiny of the masses.

In PolyLens, we chose to design the system to support many persistent private groups. Given the nature of movie going, we believed users would often choose to go to the movies with members of the same groups. We also expected that any individual user would be part of only a few groups, but that the system might have many small groups. Private groups minimized both naming problems and concerns about privacy.

Alternative designs would be more appropriate in other cases. A bookseller creating on-line book clubs, for example, might prefer to have a small number of public groups, both to make it easier to find a group and to focus attention on the selected books. Systems to support casual recommendations, such as the workout music playing in a gym might be better designed to support rapid ephemeral group formation (McCarthy & Anagnost, 1998). Similarly, a movie recommender in a kiosk in a video store, rather than on the web, might make use of ephemeral groups by having group members scan their membership cards to receive impromptu group recommendations.

How do groups form and evolve?

All group support systems, from e-mail lists to physical invitations, must address the question of creation and maintenance of the groups. Three important issues are: who creates groups, who can join them, and how are they managed.

The decision on who can create groups frequently determines the nature and quantity of groups. An administrator who creates every group may provide a level of quality control, but may also be a bottleneck that reduces the number of groups created. At the other extreme, some systems allow any user to create groups, encouraging group formation but possibly resulting in a large number of duplicate or otherwise underused groups (particularly when they are public). A middle position limits group creation to people with certain privileges.

A related question is who may join groups. Some systems let users apply to join private groups while others require members to be invited. Systems with large, public groups often allow members to simply join without the group's consent. Other issues include the ability to set qualifications for group membership and the question of whether group membership is restricted to members of a larger community (e.g., members of the recommender system).

Group management ranges from anarchy (where all members have all rights, including the right to disband the group) to dictatorship (where a group's founder or administrator has all rights, and members have none) with a variety of compromises in the middle such as voting systems. Particular rights relevant to administering groups include adding and removing members, viewing the membership list, disbanding the group, and delegating administration privileges.

In PolyLens, we wanted to encourage group formation while keeping administration and membership simple. We decided to allow any MovieLens user to create a group, and limited membership to the people that the group creator invited. All group members could view the membership list and remove themselves from the group, but only the creator of the group could invite new members or remove another member. The creator of the group must already know the person she wants to invite; there is no mechanism for finding other users through PolyLens.

Most of our decisions make less sense for systems with public groups or larger groups. In particular, public group systems need either open joining or an application process to avoid in-band membership requests. For instance, mailing list systems such as majordomo usually have careful rules that separate membership requests from the list traffic. Systems with larger groups also need a way to delegate administrator privileges and may want to have additional levels of administration, such as a system-wide administrator who has the authority to disband a group that violates site policies.

How is privacy handled within a group?

Since the goal of a group recommender system is to recommend items for the group as a whole, membership entails at least some minimal loss of privacy. We look at what control users have over their own membership in groups as well as what control they have over sharing their personal data.

There are three common policies for membership control. One policy, often used in "groups" formed by junk mailers and spammers, neither notifies members nor asks their consent. This gives users no control and awareness of group membership but does make it easy to form groups. A second policy notifies the user that she has been added to a group, providing awareness but no explicit consent. This policy also makes it easy to form groups, but sacrifices user control over private data. The third policy requires consent from an invited group member before that member enters the group and before her data is used by the group. This policy makes it more difficult to create groups but provides the greatest privacy protection to the users.

Once a user joins a group, what control does he retain over his personal data? In recommender communities, data can be divided into two categories: ratings data and recommendation data. Ratings data comprises the opinions the user has entered about items he has experienced. Recommendation data comprises system-derived predictions of how well a user would like a particular item. In the example of movies, a user may have ratings data indicating that he liked *Titanic* very much (giving it five stars) as well as recommendation data indicating that the system thinks he'll dislike *Star Trek VI*.

A recommender system can use personal data without revealing it, or may reveal it to other users. Any user joining a recommender system community, with or without groups, consents to having his ratings used by the system to generate recommendations for others. This consent is fundamental to collaborative filtering recommenders and is the basis for forming the community. Similarly, in order to influence the items recommended to a group, a group member must implicitly allow the system to use his ratings (and possibly recommendations) in the formation of group recommendations, even if they are not explicitly revealed.[1] On the other hand, the ratings and recommendations of individual users need not be revealed to other group members. Particular implementations may make this information available, may hide this information, or may leave control over the information to each individual user.

In PolyLens, our concern for our members' privacy led us to require explicit consent before an invited member could be added to a group. We tried to make the process of consent as easy as possible. If the member is physically with the inviter, she can simply enter her ID and password to provide instant consent. Otherwise, the user is notified of the pending invitation when she next logs in to MovieLens. For people who are already MovieLens users, we also send a notification e-mail if they have consented to receive e-mail from us; for invitations to non-users, we have no choice but to send e-mail. During the consent process, the user is also asked whether she is willing to share her recommendations with the other group members. She can change this decision or leave the group at any time. We do not provide a way to share actual ratings.

How do we form recommendations for groups?

There are two issues to address when forming recommendations for groups. First, there is the general issue of defining a social value function that describes how the tastes and opinions of individuals affect the group's recommendation. Then there is the technical, algorithmic implementation of that social value function to create an efficient recommendation based on the tastes of many users.

The social value functions for group recommendations can vary substantially. Group happiness may be the average happiness of the members, the happiness of the most happy member, or the happiness of the least happy member (i.e., we're all miserable if one of us is unhappy). Other factors can be included. A social value function could weigh the opinion of expert members more highly, or could strive for long-term fairness by giving greater weight to people who "lost out" in previous recommendations.

Once a social value function is selected, an algorithm must be developed to implement it. Single-user collaborative filtering systems commonly use nearest neighbor algorithms that identify a set of community members most like the target user and evaluate items as a similarity-weighted average of the normalized ratings from those users. This algorithm is not directly applicable to group recommendations because each group member has different tastes and therefore different individual ratings profiles. There are two basic approaches for retaining

---

[1] We recognize that groups with few members can attempt to infer individual tastes from group predictions, particularly if members can form groups with and without a target member. This type of inferential attack on privacy is well known (Beck, 1980) and addressing it is beyond the scope of this work.

the essence of nearest neighbor algorithms in a group setting: creating a single neighborhood for the group and merging individual recommendations.

The most direct way to support group recommendations is to create a "pseudo-user" that represents the group's tastes, and to produce recommendations for the pseudo-user. The pseudo-user could be created manually by group members as a reflection of their shared tastes, or it could be created automatically by merging the rating profiles of the group members. Manual creation allows groups to come to explicit consensus on movie ratings (in a way, they can define their own social value functions), but this is time-consuming and hard to keep current. Automatic creation of pseudo-users from group member profiles is more practical, but it raises several issues. In particular, the formula for merging ratings may give unequal weight to the ratings of different users. For instance, using the union of the group's individual ratings leads to recommendations biased toward users with more ratings. A related approach is to avoid merging profiles and instead choose a set of neighbors to best fit the group overall, for some best-fit criteria. By representing the taste of the group before making recommendations, these single neighborhood approaches increase the chance of finding serendipitously valuable recommendations. On the other hand, these algorithms can produce recommendations that satisfy many, but not all, members of a group, which may not match the desired social value function. Furthermore, the group prediction may lie outside the range of any individual predictions, which may be disorienting to users and difficult to explain (Herlocker et al., 2000).

Instead of creating pseudo-users, the recommender system can generate recommendation lists for each group member and merge the lists. Merging strategies have several advantages. They present results that can be directly related to the results that would be seen by individual group members. This means that the results are relatively easy to explain (e.g., "the system believes that three of you would like it a lot, but two of you wouldn't like it at all"). Also, since these approaches compute individual recommendations it is efficient to display them alongside the group recommendation, giving users more information with which to make decisions. On the other hand, group recommendations based on merge strategies are less likely to identify unexpected, serendipitous items.

In PolyLens, we expected most groups to be small—just two or three users—and therefore chose to use a social value function where the group's happiness was the minimum of the individual members' happiness scores. We also decided not to recommend movies that any member of the group had already rated (and therefore seen). We used an algorithm that merged users' recommendation lists, and sorted the merged list according to the principle of least misery.

We know that both our social value function and our recommendation list merge algorithm are unlikely to work well for large groups. It is an open research question to understand the types of social value functions that best satisfy large groups and to implement them algorithmically.

What interfaces support group recommenders?

Two key components to a group recommender interface are the interface for requesting recommendations and the display of returned recommendations. We intentionally omit discussion of graphic design to focus on the data being presented and the queries supported. We explored three models of organizing the available information when displaying recommendations: a group-only interface, a composite interface, and an individual-focused interface.

Group-only interfaces display items with the group recommendation. These interfaces avoid revealing preference information from other group members, but also prevent group members from balancing the interests of others in selecting an item. In Figure 1 we show a modified group-only interface that also displays the predicted value for the group member who is requesting the recommendation.

TITLE	GENRE	GROUP	YOUR
King of Masks, The (Bian Lian) (1996)	Drama	★★★★½	★★★★½

Figure 1. A modified group-only interface.

Composite interfaces display a list of recommended movies with both group and individual member predictions. Depending on the privacy policy of the system, particular group members may be omitted from the listing. These interfaces allow group members to balance the system's estimate of group welfare with the predicted happiness of each group member. Figure 2 depicts a simple composite interface for a two-member group.

TITLE	GENRE	REVIEWS	GROUP	YOUR	lam@cs.umn.edu
Frequency (2000)	Drama, Thriller	MOVIE REVIEW	★★★★½	★★★★½	★★★★½

Figure 2. A two-member composite group interface.

Individual-focused interfaces show the items in the context of individual user preferences. They may even entirely omit group recommendations, though such recommendations could be either displayed or used to filter the movies being displayed. In Figure 3 we illustrate a "manual" group recommender that simply brings together individual recommendation list displays.

TITLE	GENRE	REVIEWS	oconnor@cs.umn.edu
B. Monkey (1998)	Romance, Thriller		★★★★½
Last Night (1998)	Children's, Comedy	MOVIE REVIEW	★★★★½
TITLE	**GENRE**	**REVIEWS**	**lam@cs.umn.edu**
Get Bruce (1999)	Documentary		★★★★
Last Night (1998)	Children's, Comedy	MOVIE REVIEW	★★★½

Figure 3. A manual group recommender.

We implemented a composite interface for displaying recommendations in order to provide maximum information while minimizing the load on users. Recommendations are sorted in order of decreasing group prediction. The interface for making recommendation requests is the same as the standard MovieLens interface: users can request group recommendations by searching for titles or by specifying restrictions on date and genre. The only difference is that with PolyLens the user may also specify whether to receive group or individual recommendations. We expect that group recommender applications, including PolyLens, may warrant more advanced, custom interfaces, but have deferred study of these until after establishing the value of group recommendations *per se*.

The PolyLens Field Trial

We conducted a nine-month (and still ongoing) field trial of the PolyLens system beginning in May 2000. Figure 4 shows the progress of the field trial.

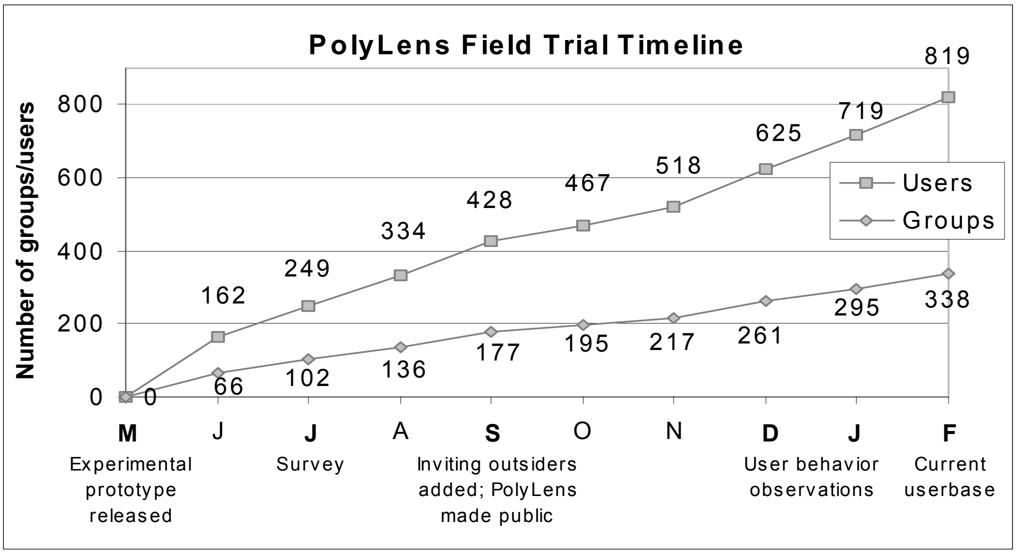

Figure 4. Timeline of the PolyLens Field Trial.

We released PolyLens as an experimental feature of MovieLens in May. This meant that users who consented were allowed to use the feature. In July, we surveyed the experimental users and revised the prototype based on their comments, allowing group creators to invite people who were not yet MovieLens members. We released the revised prototype to the entire MovieLens community in September. After three months, we conducted observations of how the group recommendation features affected user behavior. As of February 2001, PolyLens has 338 groups with 819 members. They have made about 7,000 requests for group recommendations, and have received over 114,000 recommended movies.

Several sources provide the data cited below. Group membership, creators, and names were pulled directly from our database. We logged the user, group, and time when users created groups, joined groups, and made individual or group recommendation requests. In July we conducted a survey of group users. The survey received a total of 143 responses (57% of the experimental users). The survey contained six "agree/disagree" type questions as well as a question soliciting additional comments.

How Groups Were Formed and Used

Most PolyLens groups are small, and they are made of up people who know each other. Table I shows the distribution of group sizes.

Members	Number of groups
2	257
3	53
4	16
More	12
Total	338

Table I. Group sizes.

Group names followed several common patterns. A pair of proximate personal names (*Jess and Wes*; *Brett&Corrie*) accounts for about 35% of all groups. Families (*The Vails*; *maxwell family*) created another 10%. The rest fall into several categories:

- in-jokes (*monkeychickens*; *Ninety-Percenters Know nothing!*);
- salient group characteristics (*College Swimmers*; *MommiesWeBe)*; and
- names about being a movie group (*Must See*; *Film, thy name is... film*).

Both the distribution of group sizes and the characteristics of group names suggest that PolyLens groups are composed of people who go to the movies together rather than people drawn by film characteristics (e.g., sci fi groups, Danny Elfman fans). Since movie-going groups tend to be small, PolyLens groups were small as well. This is not surprising, since MovieLens has no mechanism to help users to find other, like-minded users.

What was surprising was that 492 people tried to form groups and failed. Misspelled e-mail addresses, users who ignored invitations, duplicate groups, and flaws in the group interface explain some of these cases. We suspect, however, that many of these one-member "groups" were born when people thought that a movie group would be neat, created the group, and only then realized that they would not be able to recruit MovieLens users that they did not already know.

We wondered whether requiring the consent of members to join groups caused them to form slowly. We measured this consent lag during the prototype period. The median lag was 46 minutes, the mean 4.9 days. The shortest time was 16 seconds, the longest 110 days. Once formed, groups rarely formally disband: only 14 groups (4%) did so.

Groups follow one of three lifecycles. About one-quarter of all groups are used only on the day they form. These are probably either ephemeral groups, meant to be used only once, or else groups that users tried and decided they did not like. Another set of groups appears to be "permanent". One-quarter of the groups created in the first two months of the trial were still in use as of February, seven to nine months later. This may be a high estimate for the entire population of groups, as these groups belong to enthusiastic experimental "early adopters"—but it does imply that groups give long-term value to many users. The rest of the groups are used for a few weeks or months and then lay fallow, with the break-even point around 55 days: half of all groups are used for less than 55 days, half for more. Groups may be abandoned for several reasons: the members decide that group recommendations are not worth the effort; the members stop seeing movies as a group; the members leave MovieLens altogether; the group is dormant; etc.

Most PolyLens users joined (89%) or created (93%) exactly one group; Table II gives a detailed breakdown.

n	Member of n groups	Creator of n groups
1	628	286
2	55	19
3	19	2
More	6	2
Total	708	309

Table II. Number of groups users belong to and create.

There are 708 unique users filling the roles of 819 members, so there is about a 14% overlap between groups. Much of this overlap occurred in situations where a set of users created several overlapping groups.

Across the entire field study, the group's creator was the only member who requested group recommendations for 73 of 338 groups (22%). On the other hand, the group's creator never requested recommendations for 34 groups (10%). Overall, 659 of 819 (80%) of group members requested group recommendations.

Users have made a total of about 7,000 group recommendation requests, or about 10 per user. We do not know the exact number because of a data collection failure; our estimate is based on 11,432 pages of group recommendations viewed and the average number of pages viewed per request (about 1.7).

How Groups Affect the Way Users Use MovieLens

So how do 10 group requests per user stack up against their overall MovieLens usage? And how do new users who came to MovieLens via a group invitation behave differently than others? We made three comparisons to find out how the group recommendation features affected the way users use MovieLens:

- new users invited to join MovieLens through groups vs. other new users;
- new invited users vs. established users who joined groups; and
- established users who joined groups vs. other established users.

The statistics in this section are mostly descriptive; those that are statistically significant are noted. It is hard to establish statistical significance for behavioral differences among MovieLens users because their behavior has high variance. Some rate ten movies while others rate hundreds; some log in once per month, others, once per day.

We studied the 77 “invitees” (users who came to MovieLens because of group invitations) who joined MovieLens between September and January. These users differed from other new users in two ways: friends invited them to join groups, and they had to learn how to use both MovieLens and its group features. Because our server’s response time varies (publicity causes usage spikes) and could affect whether new users continue to use the system, we chose a control group of 77 users who joined at about the same times as the invitees. We measured system use for their first 30 days; Table III gives the results.

Use	Invitees	Other new users
Recommendation requests	872	459
Total logins after first visit	103	61
Users who returned after their first visit	36 (47%)	25 (32%)

Table III. New invitees’ use of MovieLens vs. that of other new users.

Invitees use the system more actively than other new users, making significantly more recommendation requests ($t(152)=2.01$, $p<0.05$). We believe several factors caused this. First, people who refer invitees may have introduced a selection bias, only inviting users they thought would like MovieLens. Their endorsement of MovieLens also probably carried weight, making invitees more disposed to experiment with the system. Finally, users responding to the survey said that group recommendations add value. Invitees saw a more useful system than did the other new users, and so used it more actively.

We next compared how these invitees used MovieLens to how established users who joined groups used the system. A total of 59 invitees used the system to get recommendations, so we chose a control group of 59 established users who joined groups during the same period and measured requests for group and individual recommendations for 30 days. Table IV shows what happened.

Use	Invitees	Established users
Group recommendation requests	112 (18%)	356 (23%)
Average movies recommended/group request	20	17
Individual recommendation requests	527 (82%)	1193 (77%)
Average movies recommended/individual request	101	46

Table IV. New invitees’ use of MovieLens vs. that of established users.

These data show that established MovieLens users made more overall requests than the invitees ($t(116)=2.68$, $p<0.01$). This is sensible, since some invitees never returned to MovieLens, while the non-returners had already been shaken out of the group of established users.

Established users used groups for a greater percentage of recommendation requests than did invitees, although the difference was not statistically significant. We believe that the difference is not statistically significant because user behavior will be different between the short term and the long term. In the short term, invitees had more to do—rate movies, learn how MovieLens works, and explore the group features—while established users could concentrate on using group recommendations. In the long term, we expect that invitees would use the system for more group recommendations than other group users because that is how the system was first presented to them.

Both invitees ($t(58)=4.14$, $p<0.01$) and experienced users ($t(58)=4.34$, $p<0.01$) view more movies per individual recommendation request than they do per group request. Three factors contribute to this. First, users can rate movies that MovieLens recommends and they have seen, but only when they request individual recommendations. Many users like to rate movies; these users would tend to pursue individual searches. Second, for group recommendations, the system's predicted score falls off quickly because of the "minimum individual score" social value function we use. Some users probably cut off searches once the prediction falls low enough. Finally, users may be able to make decisions based on group recommendations without looking at as many movies as they need to when they make decisions based on individual recommendations.

We wanted to see how joining groups affected MovieLens use among established users. We compared the 59 established users mentioned above who joined groups to another set of 59 established users who only used the system for individual recommendations. We measured system use for 30 days. Users who joined groups made a total of 1,549 recommendation requests. Users who were not group members made a total of 1,666 requests, slightly more than the group joiners, but the difference was not statistically significant.

User Satisfaction

Four of our survey questions focused on how well users liked the group recommendation features. Table V shows their responses.

	Strongly agree	**Agree**	**Disagree**	**Strongly disagree**
I found the process of creating groups easy.	45%	50%	4%	0%
I found it easy to add members to groups.	33%	45%	21%	0%
I found group recommendations more helpful than individual recommendations when deciding on a movie to see.	22%	55%	23%	0%
	Very satisfied	**Satisfied**	**Dissatisfied**	**Very Dissatisfied**
Overall how satisfied are you with MovieLens groups?	41%	54%	5%	0%

Table V. Survey results for user satisfaction questions.

These results suggest that users were pleased with the ability to receive group recommendations, with 95% satisfied or very satisfied with the group recommendation feature, and 77% finding group recommendations more useful than individual ones. Several users emphasized their approval in their comments:

> *"A delightful and substantive addition to your offering. I love movielens. And now even more."*

However, 21% of users found it difficult to add members to groups. Quotes from respondents indicated that the main problem was that in order to add a member to a group, that person had to already be a MovieLens user:

> *"I'd like very much to be able to invite non-movielens users to a group. Then when they receive the invitation, they can sign up. It's a lot easier than convincing someone to go to the site, sign up, and give me the e-mail address they used for your site."*

Outreach

Group creators invited non-MovieLens members to join their groups 391 times. The invitees accepted 95 of these invitations. In 15 cases, one group creator brought in two or more new users (the highest was 12). These invitees add value to the system. As shown above, they are more active than other new users. Recommender systems do better as more people use them, so when users are active the entire community benefits. Group recommenders also provide an opportunity to make explicit the normally anonymous underlying community that allows recommender systems to work.

Privacy

Two survey questions asked users how they felt about seeing others' recommendations and sharing their own. Table VI gives the results.

	Strongly agree	**Agree**	**Disagree**	**Strongly disagree**
I prefer being able to see each group member's personal recommendations.	60%	34%	4%	2%
I prefer having other group members see my personal recommendations.	47%	46%	4%	3%

Table VI. Survey results for privacy-related questions.

Nearly all users preferred to allow other group members to see their personal recommendations (93%) and to see the personal recommendations of other group members (94%). The MovieLens database shows that over 97% (798 of 819) of group members actually do share their recommendations, confirming the survey results. Some users also commented that seeing individual recommendations was essential to making good use of group recommendations.

Mailing group invitations to new users presents a privacy issue. We e-mailed group invitations to MovieLens users only if they had consented to receive e-mail from us. Since we could only contact non-users by e-mail, we designed the invitations to be as personal as possible. We allow group creators to include a personal message, and we send the mail in the name of the group creator if the name appears to be a valid e-mail address. To date, we have received no spam complaints resulting from sending group invitation e-mails.

Lessons Learned

We studied a number of aspects of group recommenders during the field trial, including how groups formed and who used them, how group recommenders affect the use of a recommender system, how satisfied users were with group recommendations, the effect of being able to invite members from outside the recommender system, and how users reacted to the loss of privacy required when joining a group. Here we summarize key lessons learned.

Users like and use group recommenders. PolyLens users expressed a clear desire for group recommendations. Both survey results and observations of user behavior support this claim. We believe that the utility of group recommender systems will generalize to most domains where groups consume entertainment together, such as book clubs, dining out, travel, and concerts. Whether group recommendations are useful in non-entertainment domains is less certain, as there is less shared consumption.

Users trade privacy for utility. The vast majority of PolyLens users were willing to trade privacy for group recommendations. Three factors contributed. First, users had direct control over sharing recommendations. Second, people in PolyLens groups already know each other and probably discuss their reactions to movies. Third, personalized movie recommendations have limited intrinsic value to others. All of these factors are important: if users do not control their data, if they must share their data with strangers, or if the items recommended are of a sensitive nature (e.g., stock picks), people will be less likely to share personal recommendations.

For maximum group use, users must be able to find each other. Most PolyLens groups were very small, and many groups were stillborn. This is partly because of the nature of our userbase, and partly because users were required to know each other outside of MovieLens. We feel that most group recommender systems should have features to help users find each other. These features would help users form groups and would make the community aspect of recommender systems more explicit, although they would raise new privacy issues.

Better social value functions for group predictions are needed. Several users disliked the "minimize misery" policy that PolyLens uses, and one pointed out that our implementation does not take into account differences in rating scales (e.g., Mark's "5" is Dan's "3"). We got away with this simple method for combining recommendations because the groups were small and because users could review the individual recommendations to make a final decision.

Groups are permanent, but also ephemeral. We were right that groups would be permanent, but failed to address situations where only part of the group wanted to go to a movie. Several sets of users created multiple groups, each of which contained a subset of the members, to support temporarily removing group members. One user explicitly asked for such a feature:

> *"We need to be able to select certain group members and generate suggestions specifically for those members only. For instance, if only three of us are going to the movies, and there are five in the group, we don't want the other two skewing our average."*

Group recommenders should support temporary removal of group members.

Groups are valuable to all members, not just the creator. We expected that group creators might have the role of primary decision maker for movie selections for their groups and might therefore be the main users. We found to the contrary that 80% of group members requested group recommendations. This suggests that the group recommendation interface should not require administrative privileges to use. Also, a group administration policy that allows all group users to add members might be effective and might encourage groups to grow.

Using the group mechanism to reach out to new users is effective. New users who came to MovieLens through group invitations used MovieLens more actively than other new users. Group creators no doubt did some filtering of users who were not likely to use the system—but that's a good thing.

Conclusion

We presented the first example of a collaborative filtering recommender system that recommends items to groups of people based on their collective preferences. Surveys and usage studies show that users like group recommendations. Further, even though group recommendations require users to give up some of their privacy, our users indicate that the tradeoff was worthwhile for them. One reason the reduced privacy was less of a problem in this study is that most of the groups were very small, probably comprised of a group of close friends. Future work is needed to understand the tradeoffs for larger or more anonymous groups, as well as to establish appropriate social value functions for such groups.

While the PolyLens system was designed specifically for users of a movie recommendation site, we also reviewed the design space for group recommenders to help others design group recommender applications. The results of our field trial are limited to the particular set of design decisions we made for PolyLens. Further study is needed to understand which designs best serve users of other recommender applications.

Acknowledgements

We would like to thank members of the GroupLens Research Group for their contributions to this research, and particularly Tony Lam for developing the MovieLens experimental framework. We would also like to thank the members of MovieLens.

This work was supported by grants from the National Science Foundation (IIS 96-13960, IIS 97-34442, and IIS 99-78717) and by Net Perceptions, Inc.

References

Aggarwal, C., Wolf, J., Wu, K., & Yu, P.S. (1999): 'Horting hatches an egg: A new graph-theoretic approach to collaborative filtering', in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999, pp. 201-212.

Beck, L. (1980): 'A security mechanism for statistical databases', *ACM Transactions on Database Systems*, vol. 5, no. 3, 1980, pp. 316-338.

Breese, J.S., Heckerman, D., & Kadie, C. (1998): 'Empirical analysis of predictive algorithms for collaborative filtering', in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence,* San Francisco, CA, 1998, pp. 43-52.

Dewan, P., & Shen, H. (1998): 'Flexible Meta Access-Control for Collaborative Applications', in *Proceedings of the ACM 1998 Conference on CSCW,* Seattle, WA, 1998, pp. 247-256.

Edwards, W.K. (1996): 'Policies and roles in collaborative applications', in *Proceedings of the ACM 1996 Conference on CSCW,* Boston, MA, 1996, pp. 11-20.

Ellis, C.A., Gibbs, S.J., & Rein, G. (1991): 'Groupware: Some Issues and Experiences', *Communications of the ACM*, vol. 34, no. 1, January 1991, pp. 39-58.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2000): 'Eigentaste: A Constant Time Collaborative Filtering Algorithm', UCB ERL Technical Report M00/41, August 2000.

Gutwin, C., & Greenberg, S. (1998): 'Design for individuals, design for groups: tradeoffs between power and workplace awareness', in *Proceedings of the ACM 1998 Conference on CSCW,* Seattle, WA, 1998, pp. 207-216.

Herlocker, J.L., Konstan, J.A., Borchers, A., & Riedl, J. (1999): 'An algorithmic framework for performing collaborative filtering', in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999, pp. 230-237.

Herlocker, J.L., Konstan, J.A., & Riedl, J. (2000): 'Explaining Collaborative Filtering Recommendations', in *Proceedings of the ACM 2000 Conference on CSCW,* Philadelphia, PA, 2000, pp. 241-250.

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995): 'Recommending and Evaluating Choices in a Virtual Community of Use', in *Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, 1995, pp. 194-201.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., & Riedl, J. (1997): 'GroupLens: Applying Collaborative Filtering to Usenet News', *Communications of the ACM*, vol. 40, no. 3, March 1997, pp. 77-87.

McCarthy, J., & Anagnost, T. (1998): 'MusicFX: An arbiter of group preferences for computer supported collaborative workouts', in *Proceedings of the ACM 1998 Conference on CSCW,* Seattle, WA, 1998, pp. 363-372.

Mitchell, A., Posner, I., & Baecker, R. (1995): 'Learning to Write Together Using Groupware', in *Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, 1995, pp. 288-295.

Neuwirth, C.M., Kaufer, D.S., & Chandhok, R. (1994): 'Computer Support for Distributed Collaborative Writing: Defining Parameters of Interaction', in *Proceedings of the Conference on CSCW,* Chapel Hill, NC, 1994, pp. 145-152.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994): 'GroupLens: An Open Architecture for Collaborative Filtering of Netnews', in *Proceedings of the Conference on CSCW,* Chapel Hill, NC, 1994, pp. 175-186.

Resnick P., & Varian H. (1997): 'Recommender Systems', *Communications of the ACM*, vol. 40, no. 3, March 1997, pp. 56-58.

Sarwar, B.M., Konstan, J.A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998): 'Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system', in *Proceedings of the ACM 1998 Conference on CSCW,* Seattle, WA, 1998, pp. 345-354.

Sarwar, B.M., Karypis, G., Konstan, J.A., & Riedl, J. (2000): 'Application of dimensionality reduction in recommender systems—a case study', in *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, Boston, MA, 2000.

Schafer, J.B., Konstan, J., & Riedl, J. (1999): 'Recommender Systems in E-Commerce', in *Proceedings of the First ACM Conference on Electronic Commerce*, Denver, CO, 1999, pp. 158-166.

Shardanand, U., & Maes, P. (1995): 'Social Information Filtering: Algorithms for Automating "Word of Mouth"', in *Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, 1995, pp. 210-217.

Smith, R.B., Hixon, R., & Horan, B. (1998): 'Supporting Flexible Roles in a Shared Space', in *Proceedings of the ACM 1998 Conference on CSCW,* Seattle, WA, 1998, pp. 197-206.

Ungar, L.H., & Foster, D.P. (1998): 'Clustering Methods for Collaborative Filtering', *in AAAI Workshop on Recommendation Systems*, Menlo Park, CA, 1998.

Whittaker, S. (1996): 'Talking to Strangers: An Evaluation of the Factors Affecting Electronic Collaboration', in *Proceedings of the ACM 1996 Conference on CSCW,* Boston, MA, 1996, pp. 409-418.