Using Groups of Items for Preference Elicitation in Recommender Systems

Shuo Chang, F. Maxwell Harper, Loren Terveen

GroupLens Research
University of Minnesota
{schang, harper, terveen}@cs.umn.edu

ABSTRACT

To achieve high quality initial personalization, recommender systems must provide an efficient and effective process for new users to express their preferences. We propose that this goal is best served not by the classical method where users begin by expressing preferences for individual items - this process is an inefficient way to convert a user's effort into improved personalization. Rather, we propose that new users can begin by expressing their preferences for *groups* of items. We test this idea by designing and evaluating an interactive process where users express preferences across groups of items that are automatically generated by clustering algorithms. We contribute a strategy for recommending items based on these preferences that is generalizable to any collaborative filtering-based system. We evaluate our process with both offline simulation methods and an online user experiment. We find that, as compared with a baseline rate-15-items interface, (a) users are able to complete the preference elicitation process in less than half the time, and (b) users are more satisfied with the resulting recommended items. Our evaluation reveals several advantages and other trade-offs involved in moving from item-based preference elicitation to group-based preference elicitation.

Author Keywords

Recommender System; Cold Start Problem; Interaction Design

ACM Classification Keywords

H.1.2. User/Machine Systems

INTRODUCTION

Collaborative filtering-based recommender systems [26] are among the most successful social computing applications. They automate "word-of-mouth" recommendations by first learning about the preferences of users, then matching these preferences against those of all other users of the system to provide personalized recommendations. From recommending videos on YouTube to suggesting people to connect with on Facebook, collaborative filtering is widely applied in various popular online platforms.

Despite the popularity of collaborative filtering (CF), there are some open problems, one of which is *cold start problem*. There are three types of cold start problem. In the *user cold start* problem, new users enter into a recommender system with only existing preference data about other users. In the *item cold start* problem, new items are added to a system with no preference data on them. The combination of the two is the *system cold start* problem, which occurs when a system is first released. [25] In particular, for *user cold start problem*, the system cannot provide personalized recommendations until it collects enough preference information from users. In practice, there are two common approaches to addressing this problem. In the first approach, users are admitted directly into a system with no requirements, leading to an initial experience that is non-personalized. In the second approach, users are asked to complete a *preference elicitation process* before gaining access to a personalized view of the site. In this work, we are primarily concerned with the second approach, as we focus our attention on systems where the core features are built around personalization.

There are two critical challenges in designing a new user bootstrapping process in a personalization-oriented system: providing an *efficient and easy* mechanism for eliciting preferences, and delivering *high quality recommendations* based on relatively little data. Systems with difficult or slow preference elicitation processes may waste user time and effort, and may see decreased activity. For example, in the movie recommendation website MovieLens[1], new users are required to provide ratings on at least 15 movies as part of the sign-up process. Our analysis of MovieLens data shows that users take an average of 6.8 minutes to complete this process, and 12.6% of these users fail to complete the process and never even get to the front page! But on the other hand, systems that do not learn user preferences will deliver poorer

CSCW 2015, March 14–18, 2015, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2675133.2675210

[1] www.movielens.org

recommendations, may not gain users' trust, and therefore may receive less use. As evidence, an analysis of MovieLens shows that users who receive bad recommendations initially are less active than other users.

There has been substantial research interest in improving these aspects of the user cold start problem [8, 12, 13, 24, 25]. These studies generally focus on algorithmic solutions that maximize the information gained about new users from item ratings. Though further optimizing the selection of items for rating remains an open problem, we posit that larger advances may be possible by re-thinking the user interaction model.

We are motivated by a simple intuition that runs contrary to the assumptions in these studies: perhaps the process of rating items to bootstrap user profiles is inherently inefficient. An obvious alternative to rating individual items is to rate groups of items. We hypothesize that if we can derive groups of items that are understandable, recognizable, and that span the preference space, then perhaps we can improve upon the classical methods for new user preference elicitation. Such a system would ask relatively few questions to minimize the time required, but would be able to turn the answers to those questions into a valuable source of information for bootstrapping personalization.

In this paper, we evaluate the potential of a new user preference elicitation process that combines automatic clustering algorithms with interfaces for capturing preferences about groups of items. We hypothesize that this type of system will allow users to accurately express their preferences, and quickly give them high quality personalization. Specifically, we evaluate our group-based process with two research questions:

RQ1-Minimal user effort. Is group-based preference elicitation efficient, flexible, and easy for users to understand ?

RQ2-High quality recommendation. Does group-based preference elicitation lead to an accurate model of user preferences and therefore high-quality personalized recommendations?

We study these research questions using multiple methods. To determine the theoretical feasibility of our algorithmic approach, we run an offline simulation analysis based on data from MovieLens, our research platform. To better evaluate the real-world characteristics of our system, we follow this with a user experiment with MovieLens users where we compare our process to a baseline process that has been in production for over ten years.

This paper is structured as follows. We first survey related work and situate our contribution. Then, we describe the design challenges related to generating high-quality groups of items, eliciting preferences for those items, and turning those preferences into recommendations. We follow this by describing the methods and results of two related experiments: a simulation-based feasibility study, and an online user experiment. We conclude with a discussion of the lessons learned from these experiments and the advantages and trade-offs we perceive in using a group-oriented preference elicitation process.

RELATED WORK

Most existing work on preference elicitation for new users in recommender systems focuses on algorithmic solutions that maximize the information gained about new users. However, recent work has called attention to the need for research on user experience that includes both interaction design and algorithm development to address this problem [17].

Designing for New Users

Dealing with newcomers in online communities has been an active research area [18]. Prior work looked at various interventions that can improve user retention. For example, authors in [6] conducted a field experiment in MovieLens. They found that fewer users completed the sign-up process when they were required to do additional work. However, they found that the users who finished the sign-up process contributed more value to the community.

More recently, research in [33] studied "social bootstrapping", a common method whereby users import their social connections from other social networking sites. For example, users signing up for Pinterest are asked to import their network from Facebook. Authors in [2] point out that importing social connections is not necessarily effective in personalizing content. They suggest that this might be improved by asking users' preferences on existing topics in Pinterest by clustering content as part of sign up process.

This work. Inspired by [6], we seek to strike a balance between the effort required from new users and participation of users over the long term. And similar to the suggestion in [2], we study a preference elicitation process on group of items that is based on automatically clustering content.

Preference Elicitation in Recommender System

Research on bringing new users into recommender systems has traditionally been focused on effective algorithms for optimizing the process of collecting high-quality preference information from users. Further, prior research in CF recommender systems focuses on methods to elicit users' preferences on individual items or pairs of items.

Collaborative filtering algorithms can provide personalized recommendations based only on user-item rating data and are therefore unable to make recommendations for new users until they have rated some number of items. To collect these ratings, most CF recommender systems ask new users to go through a preference elicitation process. The first batch of research on this type of process [8, 24, 25] framed the problem as finding the optimal item sequence for new users to rate, so that users

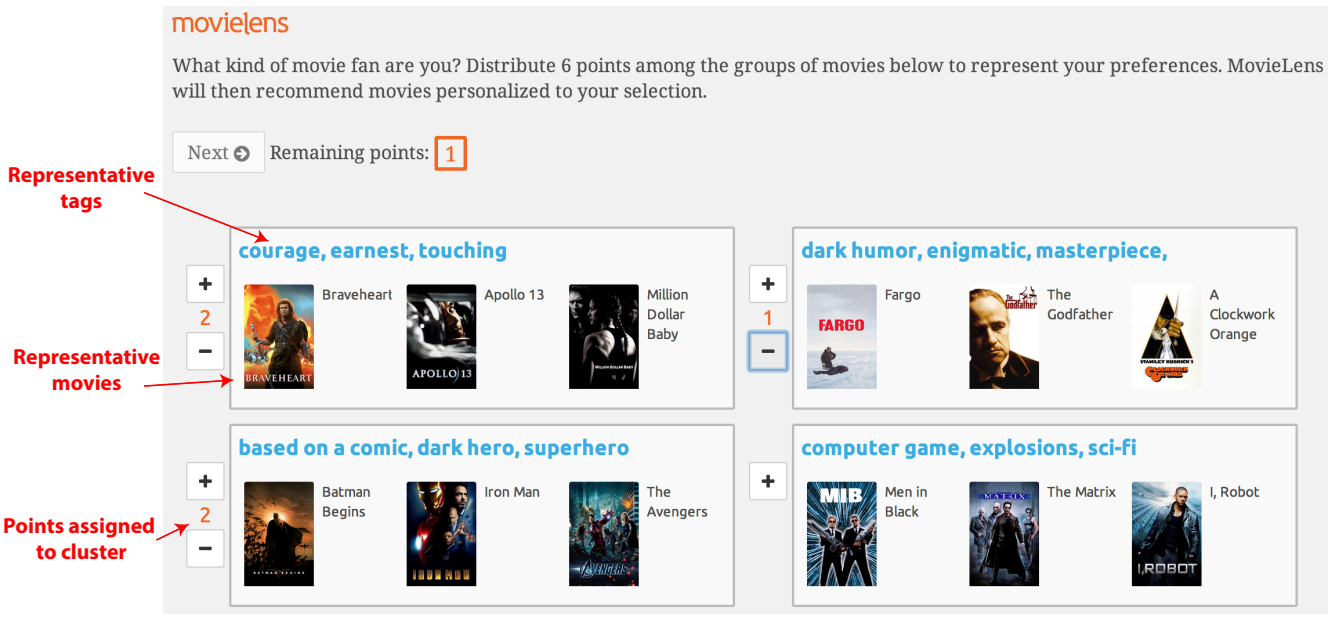

Figure 1. Screen shot of the interface for 14 movie groups, 4 of which are visible, and 6 preference points to allocate.

spend the least effort finishing the required number of ratings, and so that the system can optimize the resulting prediction accuracy. Authors in [24,25] showed that ranking movies based either on popularity or a combination of popularity and entropy of ratings led to good performance in both offline simulation and in a user experiment. Another study [8] used a different simulation method and more comprehensive evaluation metrics to shed light on the performance of the above two ranking strategies and further proposed ranking strategies based on highest predicted rating and lowest predicted rating.

Commercial systems have introduced a variety of user experiences for new user preference elicitation. Some systems use explicit ratings data. For example, Netflix asks users to step through a variety of pages where they rate genres and movies. Other systems use implicit feedback to infer users' preferences for recommendation: Youtube and Netflix use viewing history, Amazon uses clicking and purchasing history, and Facebook uses user activities on news feed items. However, implicit feedback is not as useful for systems that wish to allow users to control the personalization. Content-based recommender systems can directly elicit preferences on content features. For example, both Facebook Paper and Pinterest ask new users about the categories they are interested in before recommending content. However, similar content features are not necessarily available in other application domains, and these features may not offer the flexibility to control the granularity of users' stated preferences.

Under the same problem definition, other research has recently developed decision tree-based preference elicitation methods [12, 13, 30], whereby the recommender system repeatedly asks new users to pick their preference from a pair of movies. According to the user's previous choice, the system adaptively generates movie pairs for comparison. In the end, the system categorizes users into a user group with similar taste and then uses an average of the group's taste for generating recommendations.

However, algorithmic approaches have limitations in real-world applications. There is little transparency and limited control for users in the process of rating movies from a list or comparing a series of movie pairs. As is shown in [20], less control results in more perceived effort even though users spend less time. And in the case of decision tree-based methods, there is a high probability that users will be asked for movie pairs that they have not seen, because the algorithm is optimized only to pick movie pairs to best divide user groups.

Recently, some research has focused more on interaction in recommender systems [1, 14, 19]. SmallWorlds [14] provides interactive graph visualization for social recommendation on Facebook. Through a user study, authors find that users think the system is transparent and are satisfied with the system. TasteWeights [1], an interactive music recommender, allows users to control their recommendations by choosing their preferred artists or by leveraging the preferences found in their social network.

The most directly relevant work to this paper [19] studied a choice-based preference elicitation process. Similar to decision tree-based methods, the system iteratively asks users to compare two groups of movies before making any recommendations. In contrast with decision tree-based methods, movies are picked to represent two opposite values of a latent factor, which is com-

puted from a matrix factorization collaborative filtering algorithm. In picking movies to present, they consider both the algorithmic latent space and the transparency to users. Through a user study, they showed that the system has advantages on 15 parameters over both a manual system where users search for items and an automatic recommender system with no interaction.

This work. Similar to [19], we propose an interactive preference elicitation process that strikes a balance between user effort and quality of recommendation by asking users to rate groups of items. Compared to [19], we provide a different presentation of movie tastes with tag-labeled movie clusters. Moreover, our approach is generalizable to any collaborative filtering recommender system, while [19] is constrained to matrix factorization-based systems. And, similar to decision tree-based methods [12, 13, 30], we represent users' tastes as the average preference of users who are similar.

DESIGN SPACE ANALYSIS

In this section, we describe the design space surrounding our preference elicitation process, guided by our two research questions. As part of this analysis, we articulate several design challenges related to the creation and display of groups of items, ways of allowing users to express preferences for those items, and how we might turn these preferences into item recommendations. Specifically, we target our analysis on the design of a new user process for MovieLens, a CF-based movie recommender system that recommends movies based on movie ratings.

Design space

Designing a preference elicitation process involves making a choice along a spectrum of user effort, and a correlated spectrum of personalization. On the one hand, a system may require a demanding process that asks users for lots of information - such as ratings or survey responses - in exchange for high-quality personalization. On the other hand, a system might admit users directly, making the process fast and easy at the cost of foregoing the opportunity for personalized recommendations. In this research, we are interested in developing a part of the design space that is both low-effort and high-personalization.

The standard preference elicitation process for CF systems extracts preferences from the most basic unit - ratings on individual items. However, some ratings contain highly overlapping information. For example, if a user has rated "Toy Story", it may be the case that rating "The Lion King" does not add much additional information for the purposes of personalization, since they both reflect a preference for children's animation. Research on active learning in recommender system [28] explores a similar idea that ratings on similar items represent correlated information about user preference. Therefore, we believe that asking users for preferences on *groups* of similar movies is more efficient at the cost of some information loss. Higher quality clusters, where movies in the same cluster are more similar, will result in less information loss. One obvious way to group items in the movie domain is to use existing categorization such as genres. However, categorization is not necessarily available in other item domains and existing categorization has no flexibility to control the granularity of groupings. Therefore, we propose a preference elicitation method based on groups of items. The item groups (or item *clusters* for the rest of this paper) are identified by automatic clustering algorithm.

Design challenges

To accomplish our goal of achieving a new user preference elicitation process that is both minimal effort and leads to high quality recommendations, we must address several design challenges. In this section, we detail the following challenges (shown in Figure 2), along with our proposed solutions:

- DC1: Generating groups,
- DC2: Describing groups,
- DC3: Eliciting preferences,
- DC4: Recommending based on preferences.

DC1: Generating groups

We use an unsupervised clustering method to identify non-overlapping groups of item. In MovieLens, the items are movies. As mentioned above, the quality of clusters is critical to the effectiveness of preference elicitation: good clustering puts similar movies in the same cluster, dissimilar movies in different clusters, and covers the whole preference space.

Selecting data for clustering. We use only the MovieLens user-item rating matrix for clustering movies. Because rating information is available for all CF-based recommender systems (attributes of items are not always available), this method is generalizable. However, ratings matrices in CF systems are sparse, because both ratings from users and ratings on items follow a long tail distribution. We can get data with higher quality on movies with more ratings from seasoned users who have rated more movies, resulting in better movie clusters. Based on this intuition, we extract a dense ratings matrix to serve as the input to the clustering algorithm. To build this dense matrix, we pick the 200 most frequently-rated movies in MovieLens, and those users who have rated more than 75% of those movies. These numbers are chosen according to the size of the dataset and sparsity of the ratings. Our intuition is confirmed by manual inspection of clustering results using dense rating matrix and the full matrix.

We further experimented with data transformation techniques such as normalizing ratings and dimension reduction on the rating matrix. Based on extensive offline simulation, we chose to use mean subtracting normalization as our technique [5].

Clustering algorithm. After selecting data for the clustering, we compute pairwise cosine similarities [26]

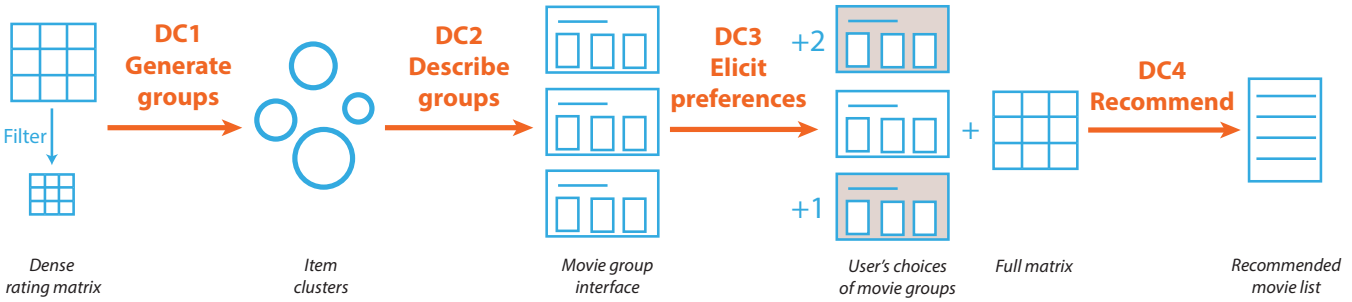

Figure 2. An overview of the design challenges in our group-based preference elicitation process.

between all movie vectors, and run the Spectral Clustering algorithm [23] to partition movies into clusters. Spectral Clustering is a standard graph-based clustering technique commonly applied to similar domains such as document clustering and community detection [3, 9]. Again, we evaluate the performance of Spectral Clustering by manual inspection of clustering result and find it to generate more sensible clusters in comparison to Affinity Propagation [10], another popular graph-based clustering algorithm.

DC2: Describing groups

To help users better understand the preference elicitation process, we need to present the movie groups in a comprehensible way. In our implementation, we use both movies and tags to represent movie groups in the user interface (see Figure 1). Tags, which are created and applied to movies by users in MovieLens, provide high quality descriptions for the movie groups. However, they can be replaced by manual labeling for a system that does not have tags.

For picking representative movies and tags, there are two options: (1) select representative movies then find the most descriptive tags, and (2) pick representative tags then find the most relevant movies. We decide to use the latter because it results in more comprehensible representations of groups in our informal analysis.

For each movie group, we first pick the top-three tags that both *uniquely describe* and are *highly relevant* to the group. Therefore, we define the measure of tag uniqueness as Equation 1 and tag relevance as Equation 2. We pick the three tags with the highest multiplication of uniqueness and relevance. (Multiplication is used to handle different scales of the two metrics.)

$$unique(t,c) = \frac{rel(t,c)}{\sum_{c_i \in C} rel(t,c_i)} \quad (1)$$

$$relevance(t,c) = \frac{rel(t,c)}{\sum_{t_i \in T_c} rel(t_i,c)} \quad (2)$$

where t denotes one of the tags T_c that appears in cluster c, and C denotes all the clusters. Note that $rel(t,c)$ is the aggregated relevance of tags t to all movies in cluster c. In our implementation, we use relevance between a tag and a movie generated from the Tag Genome [31], but other systems could replace this data with a count of applications of the tag to the movie.

DC3: Eliciting preferences

There are many possible user interfaces that support the design goal of eliciting user preferences for movie groups. For example, users might rank the movie groups, or they might provide 5-star ratings for each group. In this work, our goal is to minimize effort while achieving high personalization. Therefore, we start with a very simple interface: users choose one favorite group.

Intuitively, one movie group may not be sufficient to represent movie taste, e.g., a user might enjoy both horror movies and sci-fi movies, which are likely to be in different groups. To give users more control of how they express their preferences for movie groups, we further propose experimenting with an interaction technique that asks users to allocate a fixed number of points across one or more movie groups.

We use a data-driven approach to decide on the appropriate number of points to give users to allocate. Again, we analyze ratings from users in the dense matrix and check how many groups their top-rated movies[2] fall into. As a result of this analysis, we are able to pick the number of points that covers all the top-rated movies for 80% of the users for any number of groups. The results of this analysis inform the design of our user experiment (described in the user experiment section below) where we examine the impact of point allocation versus an interface where users simply pick their favorite movie group.

DC4: Recommending based on preferences

The next step, given a user's chosen preference for one or more movie groups, is to generate personalized recommendations from the full item space. One important feature of our recommendation process is that it can be generalized to any standard collaborative filtering algorithm.

Our algorithm, summarized in Algorithm 1, represents the preferences of a new user as the average rating from users in the dense matrix who share a similar taste.

[2]movies rated ≥ 4 on a 5 star rating scale

To recommend based on one "favorite" cluster, we find users from the dense matrix who have the highest ratings for movies in that cluster and generate a *pseudo rating profile* by taking the average ratings of those users. Though this technique builds the pseudo rating profile from data in the dense matrix, it is able to generate recommendations and predicted ratings that cover the full item space using standarad collaborative filtering.

Let r_{um} be the dense rating matrix, where $u \in U$ denotes user and $m \in M$ denotes movie. Assume there are K clusters of movies $C_k, k \leq K$. The new user u_0 has distributed p_k on each cluster k. Picking one favorite cluster is the special case where only one $p_k = 1$ is non zero.

Step 1: Compute average rating vectors
foreach *cluster* C_k **do**
  1. Find the set of users U_k where average rating $\bar{r}_{um}, m \in C_k$ is the highest
  2. Compute $\vec{r}_{U_k}$, which is the average rating for $u \in U_k$ on all movies $m \in M$
end
Step 2: Make recommendations
foreach *cluster k where* $p_k > 0$ **do**
  Make recommendation Rec_k for the pseudo user rating profile $\vec{r}_{U_k}$ using a standard CF algorithm
end
Step 3: Aggregate recommendations Rec_k using weights p_k

Algorithm 1: Algorithm for the recommendation process

To aggregate recommendations in the case where users allocate multiple points across clusters, there are two types of recommendation results to combine: predicted ratings and a top-N list. For predicted ratings, we compute a simple weighted average of multiple predicted ratings, based on different pseudo rating profiles. For generating a top-N list, we aggregate multiple ranked item lists by assigning scores $(N - rank + 1)$ to items based their rank in the list, then computing weighted averages of the scores, and finally ranking items accordingly. An example is shown in Table 1.

FEASIBILITY STUDY

To evaluate the effectiveness of group-based preference elicitation, we conduct an offline simulation study. We focus on an evaluation of *RQ2-High quality recommendation* using prediction accuracy and top-N recommendation accuracy [29], and leave *RQ1-Minimal user effort* for the user experiment. We also evaluate the trade-offs inherent in increasing the number of clusters through an analysis of resulting recommendation quality and of clustering quality metrics.

Table 1. Example of weighted aggregation of ranked item lists. List 1 has 2 points and list 2 has 1 point. We assign scores to items in the lists based on ranking. We re-rank items based on the weighted average of item scores in two lists and take top 3.

	List1(2 points)		**List2**(1 point)		**Combined list**	
Rank	**movie**	**score**	**movie**	**score**	**movie**	**score**
1	A	3	D	3	A	2.33
2	B	2	B	2	B	2
3	C	1	A	1	D	1

Data

We construct a data set of 2.2M ratings from 5,018 users on 22,115 movies from the MovieLens database. We select users who have provided at least 50 ratings and who have rated at least one movie since June 1, 2013. We choose users based on their ratings behavior because we can more accurately simulate user preferences and evaluate quality-based metrics for users with many ratings.

Method

Our simulation study has two parts: first, we build item groups and simulate user group choices; second, we use those simulated group choices to evaluate the quality of the resulting recommendations. To fairly evaluate the results, we conduct a 5-fold cross validation on the data set, where 80% of the users (and their ratings) are used to build the item groups, and the remaining 20% are used to simulate group choices and evaluate recommendation quality. Further, for each user in the test set, we randomly select 80% of their ratings to guide the simulated group choices, while the remaining 20% are left for evaluating recommendation quality. We simulate users picking one favorite movie group. We assume that users will choose the movie group for which they have the highest average rating across items in that group. Because this assumption is a crude approximation of actual user behaviour, we also include an oracle strategy that assumes users will always pick the group that has the best prediction accuracy, so as to approximate the theoretically optimal performance. Then, we make recommendations using a standard CF algorithm.

Baseline

We compare group-based preference elicitation to the standard method used in MovieLens for the last 10 years - asking users to rate 15 movies from a long list of movies generated by an algorithm. We analyzed our data set to compare various movie sorting algorithms [25] as well as several state of the art CF algorithms - item-item [5] and FunkSVD [11] - with respect to their resulting recommendation quality after 15 ratings. Item-item CF has the best recommendation quality in combination with popularity-based sorting. [3] Therefore, we will use item-item CF both to serve as our rate-15 baseline recommender, and to recommend movies based on users' choices of movie groups.

Evaluation

For prediction accuracy, we measure RMSE [16] of predictions on ratings from the set of test users. Lower

[3]All evaluations are carried out in Lenskit [7] and the scripts with details of the algorithm configurations will be shared online.

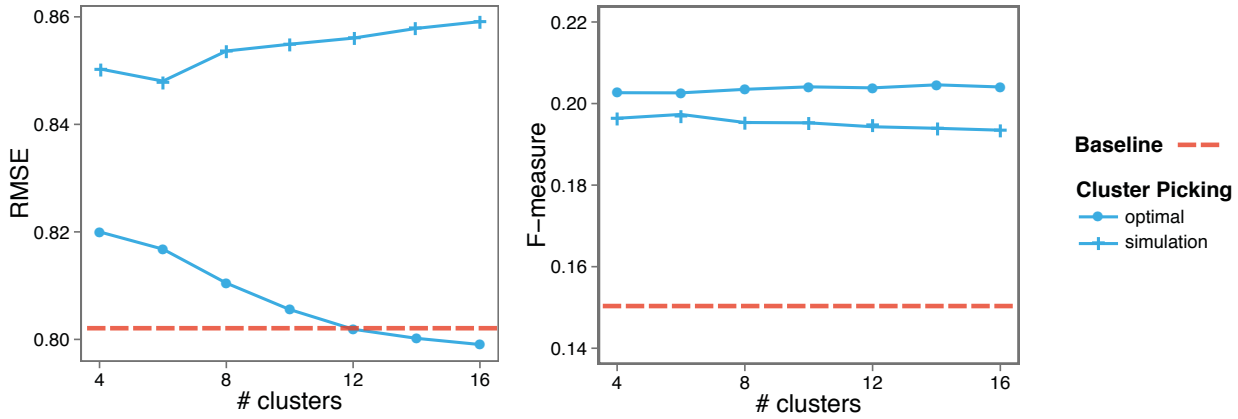

Figure 3. Results of simulated group-based preference elicitation compared to a baseline. "Baseline" shows the performance of the rate 15 movies process. The two simulated cluster-picking strategies are "simulation", where users pick the cluster with movies they've rated the highest, and "optimal", where users pick the cluster that results in the best predictions. Lower RMSE scores are better and higher F-measure scores are better. The chart on the left shows that prediction accuracy is slightly worse than the baseline in most cases while the chart on the right shows that recommendation accuracy is significantly improved compared with the baseline.

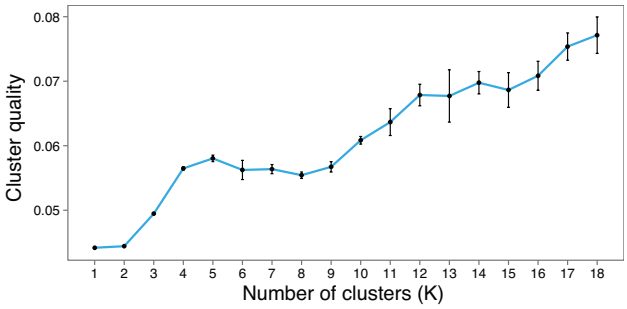

Figure 4. Cluster quality measured by Silhouette Coefficient vs. number of clusters. For each cluster number, we run the clustering algorithm 10 times and plot the standard deviation with error bars. This figure shows two local maxima around 5 and 14 clusters.

RMSE scores are better. For top-N recommendation accuracy, we follow the methodology of [4]. For easy interpretation, we report F-measure scores[4] to combine precision and recall. Higher F-measure scores are better.

Results

The results of the simulation is summarized in Figure 3.

RQ2-Quality of recommendation.

Group-based preference elicitation has better average top N recommendation accuracy (in aggregate, $F > 0.19$ for both optimal and simulation conditions) than the baseline (in aggregate, $F = 0.15$). In other words, the top N recommendation list generated from our group-based process contains more movies that users may be interested in.

We find that the optimal prediction accuracy of group-based preference elicitation varies with the number of clusters, but in general is close to the baseline (in aggregate, $RMSE = 0.804$ for the baseline condition, and $RMSE = 0.801$ for the group-based process when the number of clusters is 12). However, the actual prediction accuracy is likely to be worse than baseline. E.g., the RMSE is 0.856 when simulating 12 clusters. This is as expected, because 15 ratings provide information about user rating patterns, such as the tendency to rate higher or lower on average than other users; cluster-based preference elicitation doesn't provide such information.

Our simulation gives us evidence that the group-based method will provide better top N recommendation accuracy but slightly worse prediction accuracy. This tells us that our method is feasible, especially considering prior research [15, 21] that argues prediction accuracy is less relevant to the end user experience than recommendation accuracy.

Number of clusters.

One goal of this simulation is to inform our design choices concerning the best number of groups to display to users. As shown in Figure 3, the theoretical bound for prediction accuracy improves with the number of clusters. This is intuitive because as the number of clusters increases, the groups of items become smaller and more homogeneous - these smaller groups may better capture user preferences.

However, we see two potential downsides to increasing the number of clusters. First, this improvement is likely gained at the cost of increasing user effort: the act of picking one movie group from sixteen choices is intuitively harder than picking one movie group from four choices. Second, when we simulate cluster-picking behaviour using highest average ratings rather than the optimal method, we find that RMSE increases slightly with the number of clusters. This indicates the possibil-

[4] $F - measure = \frac{Precision * Recall}{Precision + Recall}$

ity that in a real-world system, increasing the number of choices will decrease the chances of users picking the best cluster in terms of the resulting prediction accuracy.

We also measure the quality of clusters using the Silhouette coefficient [27]. Silhouette coefficient values range from -1 for poor clustering to 1 for good clustering. We plot the Silhouette coefficient for a varying number of clusters (K) in Figure 4. This figure shows a general trend of increasing cluster quality as K increases, and the presence of two local maxima around 5 and 14 clusters.

Combining these results, we find that there is a trade-off in increasing the number of groups for our preference elicitation process: we can have high quality groups and the best prediction accuracy with many groups, or we can minimize user effort with few groups. We investigate this trade-off in our user experiment (below).

USER EXPERIMENT

We conduct an online user experiment to evaluate both *RQ1-Minimal user effort* and *RQ2-Quality of recommendation* with real users. We invite MovieLens users to participate in this experiment, comparing the performance of our group-based process with a baseline process that is currently used in MovieLens. In addition to the offline simulation work discussed above, this user experiment can shed light on user effort and subjective perception of recommendation quality.

Method.

We invite recently active MovieLens users to participate, using the same selection criteria as the feasibility study (see above). We send emails to 2.8K users and receive 342 responses between May 18 and May 23, 2014. This group of users is already familiar with the MovieLens recommender system.

We ask users to complete two tasks, using an experimental interface designed for this evaluation.[5] First, users pick one or more groups (see experimental conditions below) from our prototype interface, shown in Figure 1. Users then answer survey questions about the process of picking among the groups, and evaluate a top 10 list of recommended movies that is generated based on their group choices. Second, users evaluate a top 10 movie list generated based on their historical first 15 ratings after signing up for MovieLens. The order of the above two tasks is randomized. Users are not aware of how the recommendation lists are generated. We can evaluate *RQ2-Quality of recommendation* based on survey responses about the two recommendation lists, and we can evaluate *RQ1-User effort* by measuring the time users take picking groups and based on survey responses about perceived ease-of-use.

[5]The experimental interface was available by invitation only on a members-only preview version of Movielens (http://beta.movielens.org) that was not publicly available at the time of writing.

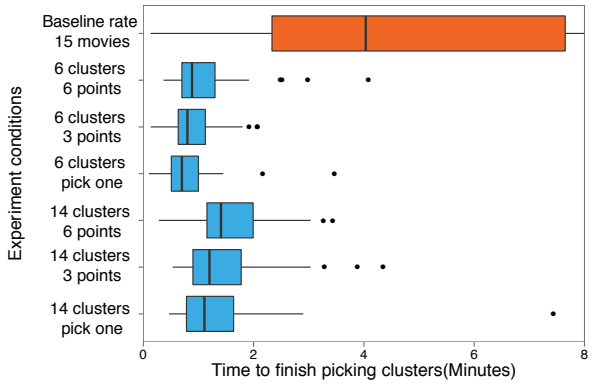

Figure 5. Time to finish preference elicitation process. Time is measured in minutes. The group-based approach takes less than half as long as the baseline, on average.

For the group-based process, we have a 2x2x3 between-subjects design with the following three factors:

1. *Order.* The order in which the user evaluates the two top-10 movie lists. We include this condition to make sure that there is no order effect.

2. *Number of groups.* The number of movie groups to display to the user. We experiment with 6 groups and 14 groups, based on our findings and open questions from the quantitative analysis discussed above. Further, considering UI design, we pick an even number of groups to better organize them in the interface.

3. *Group-picking interface.* The interface given to users for picking groups. We experiment with three conditions: picking one favorite movie group, and distributing either 3 or 6 points across movie groups (multiple points may be given to a single group). These values are chosen based on the design space analysis discussed above.

Users are randomly assigned to one of the twelve experimental conditions. All survey questions have likert scale answers ranging from strongly disagree to strongly agree.

Results

RQ1 - Minimal user effort.

Time. We examine the time spent by subjects picking movie groups as an objective metric measuring user effort. We compare the time required in the new process to the time required by the baseline rate 15 process, as measured by a historical analysis of 27,226 new users who signed up to MovieLens between Jan. 1, 2008 and Dec. 31, 2010. We show a summary of the results in Figure 5. Users spend more time as the interface becomes more expressive (more groups or more points). However, even for the most time-consuming task - distributing 6 points across 14 groups - the median time is only 1.5 minutes. This is less than half of the median time (4 minutes) that users have historically spent in the rate 15 baseline process!

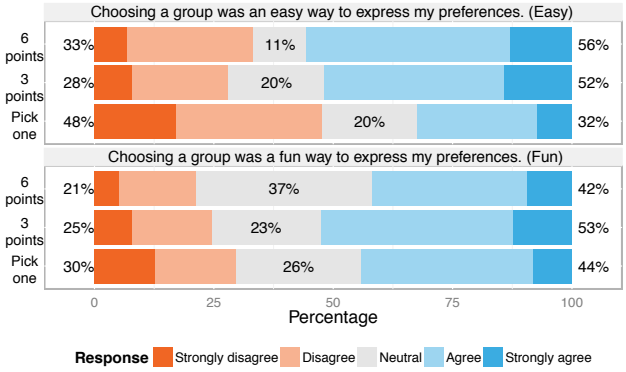

Figure 6. Survey results about the easiness and fun of our group-picking process. We compare picking one group with distributing 3 or 6 points across groups. The percentages summarize the responses after combining disagree/strongly disagree, and agree/strongly agree. Respondents think point allocation is easier and more fun than picking a single favorite.

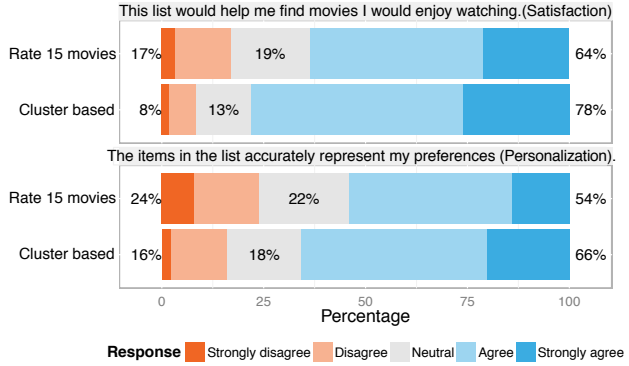

Figure 7. Survey results about recommendation quality, comparing our group-based process with the baseline rate 15 process. The percentages summarize the responses after combining disagree/strongly disagree, and agree/strongly agree. Respondents answer more positively concerning the group-based recommendations.

Subjective feedback. To evaluate whether the group-picking interface is easy and fun, we asked users two likert scale questions about the process. We did not find statistically significant differences in user responses when comparing users in the 6 and 14 group conditions. However, we did find that varying the group-picking interface had an effect on perceived ease of use. Interestingly, although picking one group is the task that takes the least time to finish (Figure 5), users think that this is the hardest task (p value < 0.001, Wilcoxon test [32]). In fact, we received an email from one user speaking to this point:

"*I'm finding that my movie preferences cannot be accurately described by selecting just one group of three movies.*"

Point allocation gives more flexibility and more control to users in expressing their preferences.

Table 2. User-expressed familiarity with recommended movies and prediction accuracy. Familiarity is represented by the average number of movies (out of 10) that users had heard of or seen. Prediction accuracy is measured by average RMSE.

Method	Heard of	Seen	RMSE
Rate 15 movies baseline	8.5	6.7	0.784
Group-based	9.0	7.4	0.797

RQ2 - Quality of recommendation.

Top N recommendation accuracy. To evaluate the resulting recommendation accuracy of the group-based process, we ask users to evaluate two lists of recommendations in randomized order. One of these lists is generated based on the user's group choice(s), and the other is generated based on their initial 15 ratings in MovieLens. Based on these data, we can perform a within-subjects comparison of user perceptions of recommendation quality. Figure 7 shows the responses to two questions related to this goal - one about general satisfaction and another about personalization. Users rate both factors higher for the group-based method (p values ~ 0, paired Wilcoxon test). Specifically, users felt that lists from the group-based recommender would better help them "find movies I would enjoy watching" and "accurately represent my preferences". These results echo the earlier findings from our offline simulation of top N recommendation accuracy. Note that satisfaction and personalization are both rated higher on average by users in the point allocation conditions (as compared with the pick one condition), but these differences are not statistically significant.

We also ask users to count the number of movies that they have heard of or seen from the list of recommended movies. Results are shown in Table 2. The group-based method recommends movies that users are more familiar with.

Prediction accuracy. We also evaluate the prediction accuracy of the group-based method. For each user, we make predictions on all movies except their first 15 rated movies. These prediction are either based on their group choices or their historical first 15 ratings. The average RMSEs are summarized in Table 2. Consistent with findings from the offline simulation study, the group-based method has slightly worse prediction accuracy (0.797 vs. 0.784). It is, however, unclear if this small difference in accuracy would be perceptible to users.

To summarize, the group-based method reduces user effort and improves top N recommendation quality, while potentially losing some prediction accuracy. Most importantly, users are more satisfied and get better personalization from the resulting top N recommendations, which is the fundamental goal of most recommender systems.

DISCUSSION

In this research, we develop a new process for preference elicitation in recommender systems based on the idea

that recommender systems may bootstrap new users more effectively by having them express preferences for groups of items rather than individual items. We evaluate this process with the dual goals of *minimal user effort* and *high quality recommendation*. A user experiment showed that our method succeeded. Compared to a baseline condition where users rate 15 movies - mirroring a process that has been in production on movielens.org for over a decade - the new process is much faster and leads to more highly-evaluated recommendation lists.

We claim that the process described here is generalizable to any collaborative filtering recommender system. Implementers must create groups of ratable entities, and then present those groups to users in an understandable fashion to collect their expressions of preference. Once users have expressed their preferences for those item groups, our algorithm for building a pseudo-ratings profile can be used to bootstrap personalization.

Our process depends on high-quality methods for grouping ratable entities. In this work, we apply a Spectral Clustering algorithm. In addition to the evaluation already presented, we asked subjects in our experiment to evaluate the scrutability of the movie groups. Figure 8 summarizes their responses. 80% of subjects understood the types of movies in each group, giving us confidence that our clustering algorithm resulted in good separation. Additionally, most subjects agreed that displaying three movies (89%) and three tags (74%) helped them to understand the movie groups. In other questions, we learned that subjects felt that three tags was about the right number to display, while showing one or two additional movies for each group might be better. Future work might explore these or other interface-oriented questions in more depth. By extension, as noted in the design challenges section, there are many possible interfaces for the display and collection of preference data on groups of items - future work might explore different alternatives such as ratings- or ranking-based interfaces.

Why was our group-based method highly evaluated by users? One explanation is that our method resulted in more familiar recommendations (see Table 2) - recent work [22] found that too much novelty in top-N recommendation lists harms user perceptions of those lists. It is possible that this increased familiarity would be lead to lower evaluations over time; however, in the context of new user recommendations, we are more interested in maximizing first impressions to establish trust in the system. A second explanation is that our method was perceived by users to be more transparent because of its group-oriented interaction design. We are not sure how users perceive the inner workings of these types of algorithms; future work might investigate how users perceive the relationship between their expressions of preferences and algorithmic "decision making". A third explanation is that by clustering on dense rating matrix from active users, we achieve high quality representation of the movie space. Asking new users to express their preference on the movie groups is more efficient than eliciting ratings on individual movies. It is worth pointing out that we can not decouple these possible reasons for better user satisfaction, because our experiment is only one exploration in the design space of preference elicitation process. We need controlled experiment over these factors to understand the keys to a successful design.

LIMITATIONS AND FUTURE WORK

One limitation of our cluster-based process is that users exit the preference elicitation process with no ratings data. Therefore, a user's initial experience is semi-personalized, based on a weighted combination of pseudo-profiles. To achieve fully ratings-based personalization, users still must express their preferences about items directly. This leads to a practical question: how should the system transition users away from this semi-personalized profile to a ratings-based profile? One approach would be to simply cut over to a user model based on ratings data at a discrete cutoff point (e.g., 20 ratings). Alternatively, the system could use some weighted combination of the outputs of the group-based and ratings-based models until users hit the cutoff point. This question of how to transition gracefully from one recommendation model to another as a user adds data is contextually broader than other questions we have addressed in this paper (i.e., it applies to all recommender systems, not just ratings-based ones) and is an interesting direction for future work.

There are several limitations of the methods employed in this research. Most importantly, we studied experienced users of MovieLens rather than new users as they entered the system. This granted us more confidence in subjects' abilities to evaluate items and recommendation lists, but did not allow us to directly compare their evaluation of user experience between the rate-15 process and our new process. In addition, we chose to perform our evaluation in the movie domain, which is different from other domains. Although we suspect our fundamental idea of picking favorite groups of items is generalizable, we have no evidence of that. Finally, it is important to note that we only compare top-10 recommendation lists - we have not examined whether system recommendations are better or worse as users venture deeper into lists of recommended items.

Future work might extend the ideas presented here more broadly to a variety of social computing systems. Many existing systems already ask users to import their contacts from other social networking platforms such as Facebook or Twitter. It is an interesting question whether this social network-based approach to bootstrapping personalization results in recommendations that are as high quality as those that would result from choosing among groups of content. For example, the social curation site Pinterest asks new users to choose among a set of predefined categories to specify their

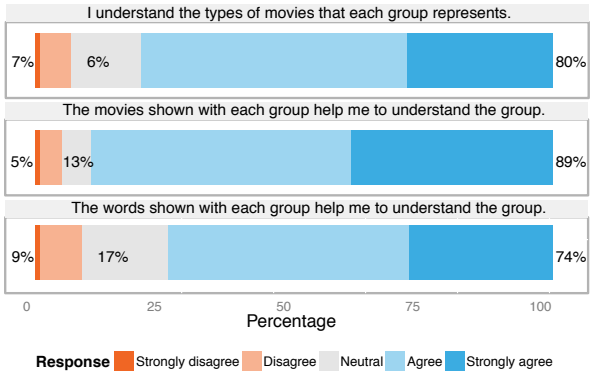

Figure 8. Survey response regarding the whether the groups are comprehensible. The percentages summarize the responses after combining disagree/strongly disagree, and agree/strongly agree. The movie groups presented to users were scrutable; both the example movies and the example tags contributed to this outcome.

interests, and to import their social network from Facebook. Could this process be improved though the use of automated clustering algorithms? What are the most effective ways of combining users' social networking information with the results of the content-based preference elicitation process to form a coherent user profile?

CONCLUSION

Research in bootstrapping new users in personalized systems has to date been largely dominated by algorithmic work seeking to optimize interfaces that elicit item ratings. In this work, we advance the idea that this classical interface design does not necessarily lead to the best user experience, nor the best personalization outcomes. Clearly, the process introduced in this paper is simply one of many possible improvements, and we expect that our process will be improved upon. Further exploratory work is needed to find interaction designs that make the new user experience more efficient (and fun!) while contributing rich information towards user profiles.

ACKNOWLEDGEMENT

We thank anonymous reviewers for providing valuable feedback to our paper. We also acknowledge the thoughtful discussions and helpful feedback from Daniel Kluver, Michael Ekstrand, and other members of GroupLens Research. Finally, we thank National Science Foundation for funding this research with awards IIS-1212338 and IIS-1210863.

REFERENCES

1. S. Bostandjiev, J. O'Donovan, and T. Höllerer. TasteWeights. In *RecSys*, New York, New York, USA, 2012.
2. S. Chang, V. Kumar, E. Gilbert, and L. Terveen. Specialization, homophily, and gender in a social curation site: Findings from pinterest. In *CSCW*, 2014.
3. S. Chang and A. Pal. Routing questions for collaborative answering in community question answering. ASONAM, Niagara, Ontario, Canada, 2013. ACM.
4. P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, Barcelona, Spain, 2010. ACM.
5. C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.
6. S. Drenner, S. Sen, and L. Terveen. Crafting the initial user experience to achieve community goals. In *RecSys*. ACM Press, 2008.
7. M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl. Rethinking the recommender research ecosystem: Reproducibility, openness, and lenskit. RecSys, Chicago, Illinois, USA, 2011. ACM.
8. M. Elahi, F. Ricci, and N. Rubens. Active learning strategies for rating elicitation in collaborative filtering. *ACM Transactions on Intelligent Systems and Technology*, 5(1):1–33, 2013.
9. S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75 – 174, 2010.
10. B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976.
11. S. Funk. Netflix update: Try this at home. *http://sifter.org/ simon/journal/20061211.html*, 2006.
12. N. Golbandi, Y. Koren, and R. Lempel. On bootstrapping recommender systems. In *CIKM*, New York, New York, USA, 2010.
13. N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. *WSDM*, 2011.
14. B. Gretarsson, J. O'Donovan, S. Bostandjiev, C. Hall, and T. Höllerer. SmallWorlds: Visualizing Social Recommendations. *Computer Graphics Forum*, 29(3):833–842, 2010.
15. A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.*, 10:2935–2962, Dec. 2009.
16. J. L. Herlocker, J. A. Konstan, L. G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.
17. J. Konstan and J. Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1-2):101–123, 2012.

18. R. Kraut, M. Burke, and J. Riedl. Dealing with newcomers.
19. B. Loepp, T. Hussein, and J. Ziegler. Choice-based preference elicitation for collaborative filtering recommender systems. In *CHI*, pages 3085–3094, New York, New York, USA, 2014.
20. S. M. McNee, S. K. Lam, J. A. Konstan, and J. Riedl. Interfaces for eliciting new user preferences in recommender systems. pages 178–187, 2003.
21. S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI Extended Abstracts*. ACM, 2006.
22. M. W. Michael Ekstrand, F. Maxwell Harper and J. Konstan. User perception of differences in movie recommendation algorithms. In *Recsys*, 2014.
23. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856. MIT Press, 2001.
24. A. Rashid, I. Albert, and D. Cosley. Getting to know you: learning new user preferences in recommender systems. *IUI*, 2002.
25. A. Rashid, G. Karypis, and J. Riedl. Learning preferences of new users in recommender systems: an information theoretic approach. *ACM SIGKDD Explorations Newsletter*, 2008.
26. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW*, 1994.
27. P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(0):53 – 65, 1987.
28. N. Rubens, D. Kaplan, and M. Sugiyama. Active learning in recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 735–767. Springer US, 2011.
29. G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer US, 2011.
30. M. Sun, F. Li, J. Lee, and K. Zhou. Learning multiple-question decision trees for cold-start recommendation. *WSDM*, 2013.
31. J. Vig, S. Sen, and J. Riedl. The Tag Genome. *ACM Transactions on Interactive Intelligent Systems*, 2(3):1–44, 2012.
32. F. Wilcoxon. Individual comparisons by ranking methods. In S. Kotz and N. Johnson, editors, *Breakthroughs in Statistics*, Springer Series in Statistics, pages 196–202. Springer New York, 1992.
33. C. Zhong, M. Salehi, S. Shah, M. Cobzarenco, N. Sastry, and M. Cha. Social bootstrapping: how pinterest and last. fm social communities benefit by borrowing links from facebook. In *WWW*, 2014.