Recommendations in Taste Related Domains: Collaborative Filtering vs. Social Filtering

Georg Groh
TU München, Department of Informatics
Boltzmannstr. 3
85478 Garching, Germany
grohg@in.tum.de

Christian Ehmig
TU München, Department of Informatics
Boltzmannstr. 3
85478 Garching, Germany
ehmig@in.tum.de

ABSTRACT

We investigate how social networks can be used in recommendation generation in taste related domains. Social Filtering (using social networks for neighborhood generation) is compared to Collaborative Filtering with respect to prediction accuracy in the domain of rating clubs. After reviewing background and related work, we present an extensive empirical study where over thousand participants from a social networking community where asked to provide ratings for clubs in Munich. We then compare a typical traditional CF-approach to a social recommender / social filtering approach where friends from the underlying social network are used as rating neighborhood and analyze the experiments statistically. Surprisingly, the social filtering approach outperforms the CF approach in all variants of the experiment. The implications of the experiment for professional and private-life collaborative environments and services where recommendations play a role are discussed. We conclude with future perspectives on social recommender systems, especially in upcoming mobile environments.

Categories and Subject Descriptors

H.5.3 [**Group and Organization Interfaces**]: Collaborative Computing; H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.3.5 [**Online Information Services**]: Data Sharing

General Terms

Design, Experimentation, Human Factors

1. INTRODUCTION

Platforms and communities on the web offering services based on explicit social networking models have become a major trend. In this article we will investigate, how models for social relations can be incorporated into recommender systems. Having reviewed basic notions in social network analysis and recommender systems in sections 2 and 3, we present in section 4 a detailed empirical study on how social relations can improve traditional collaborative filtering based recommenders in a taste related domain. In sections 5 and 6 we will take a look at future perspectives and discuss how social relation models may be incorporated into information systems.

GROUP'07, November 4–7, 2007, Sanibel Island, Florida, USA.

2. MODELS FOR SOCIAL ENTITIES AND RELATIONS

The main subject of social network analysis is to investigate properties of networks comprised of actors and relations. Actors re typically persons or groups of persons (teams, organizations, nations etc.) and relation-types include individual evaluations (e.g. friendship), transfer of resources, movement (physical or social), roles, kinship etc. [28].

According to [7] "Relations [...] are characterized by content, direction and strength." Most of the existing social network platforms offer only very few if not only one type of relation (one type of content), do not allow to specify strengths (weight) for relations and usually only provide for undirected relations between the actors.

The sum of all relations between two individuals is usually referred to as a tie: "A tie connects a pair of actors by one or more relations." [7][9].

Considering groups a usual definition is given in [29]: "A group is a social network whose ties are tightly-bounded within a delimited set and are densely-knit so that almost all network members are directly linked with each other."

Usually, social relations are mathematically modeled as a graph. a undirected uni-mode network with only one (unweighted) relationship type would be modeled as simple graph $G(V, E)$ with $E \subseteq \binom{U}{2}$. For multiple types of relations one would have to use multigraphs, directed and weighted edges are also straightforward to model.

For modeling groups, the most intutive model are maximal cliques [14][28] but other locally dense subgraphs such as LS-sets, k-clans, k-clubs and k-plexes are also possible [14][28].

[4] gives an overview of data which are provided by Internet users (without even realizing it) and which can be used for the purpose of analyzing relationships. E.g. several research projects have been conducted about the usage of computer-networks as a source of relationship data. [13] use websites as sources of social network data by scanning for co-occurrences of names. Newsgroups and discussion forums were e.g. analyzed by [25] and [26]. Apart from names, the

hierarchical structure of reply relations can be used to derive relationships ([10]).

Furthermore, Orkut [17] and Multiply [19] are examples of large online-communities where users have the possibility to state several aspects about their personal relationships. While Orkut offers friend-list services, group membership and rating of other users in three categories (trusty, sexy, cool), Multiply steps ahead by allowing users to define relationships in a more detailed way (three main categories - family, friends and professional contacts - with several subcategories each). Content may be shared based on relationship degrees the possible recipients need to be within in order to be able to access the shareable content.

3. RECOMMENDER SYSTEMS

Recommender systems record a user's and other user's past preferences in profiles in order to (mostly pro-actively) recommend new items to the user (see e.g. [5]). Montaner et al. [15] provide a useful taxonomy of existing recommender systems. The first large classification dimension is profile maintenance with the subfields representation of profiles and profile data generation (initial and adaptational). Having an operational profile for each users, one has to exploit these profiles in an appropriate way. For profile exploitation there are, again, several sub-dimensions: type of information filtering (e.g. demographic filtering, content-based filtering or collaborative filtering), profile matching (via keywords, cosine-similarity, nearest neighbors etc.), neighborhood creation (if applicable)(e.g. correlation thresholding or best-n-neighbors) and prediction computing (e.g. most-frequent-item recommendation or weighted average).

A general disadvantage of recommender systems is the portfolio effect: A user is recommended items which he already knows or which are too similar to those he already knows (see [1] for a workaround). Social recommenders are a possibility to cope with this (see 5).

In this article, we will mainly focus on collaborative filtering. The basic idea is to let users rate items and predict for an active user ratings of previously unrated items by determining a neighborhood of other raters which have exhibited a similar rating behavior to the active user [11]. Its benefits are that in contrast to content-based filtering, it is (to a certain extend) able to create cross-genre or serendipitous recommendations [11], because it does not rely on item specific similarities (as in content-based filtering) but on user-rating-similarities. Nevertheless, social recommenders are even more capable of this type of recommendations (see 5). As a drawback, collaborative filtering can exhibit a severe cold-start/early-rater problem. If not enough ratings are available the results will be poor, thus nobody will enter new ratings and so forth.

3.1 Social Recommenders

Human decision making is (contrary to what classical utility theory suggests) more often based on heuristics than on pure logic and mathematically weighing advantages and disadvantages. [31] describes types of heuristics on which human decision making is based. Furthermore, "the use of advice is a fundamental practice in making real-life decisions" ([30]). "Generally, advice-seeking is seen as a problem of combining and weighting different information sources to come to a final conclusion" ([2]). Decision making (e.g. on the usefulness of items) and recommending are inherently intertwined social processes. Especially in taste related domains, it is beneficial to know the advice-givers / recommending persons in order to perform this "combining" and "weighting" on the basis of e.g. social estimations. Thus, for recommender systems trust is an important goal to achieve ([21][8]). [8] shows how to improve trust in a movie recommender system by listing and weighting movie ratings of friends and non-friends separately.

With respect to decision making in groups two aspects are important: normative influence, which is based on the desire to conform to the expectations of others, or informational influence, which is based on the acceptance of information from others ([6]). Group members may strongly identify with their group and may fear isolation if they act against the group. This behavior is explained in detail by the spiral of silence model introduced by [20].

Previous work on social recommender systems include [24] and [2]. [24] compares the recommendations of online recommender systems to recommendations made by the user's friends regarding books and movies. In an empirical study they discover that friends are preferred over online recommender systems considering good (recommendations of interest) and useful (recommendations of interest which have not been experienced yet) recommendations. [2] investigates the nature of taste-related recommendations (movies) by an empirical study and find users to prefer recommendations from people they know regarding this domain. Furthermore, they make several proposals on how and why to combine social networks with recommender systems. We will now investigate the influence of social relations on recommender systems in more detail with the help of a empirical study, where we will compare a collaborative filtering approach (neighborhoods for recommendation generation are computed on the basis of rating similarity) with a social filtering approach (neighborhoods for recommendation generation are computed on the social network).

4. EMPIRICAL STUDY

In this section we will first describe the data set of our empirical study and then we discuss two experiments on the collected data set corresponding to two research questions: (1) Is rating behavior statistically dependent on friendship and if so: How strongly does rating behavior correlate with the size of the social structures investigated? (2) How does a social filtering approach perform compared to a collaborative filtering approach?

4.1 Basic Data Set Properties

The German community Lokalisten [18] is a Munich-based German language virtual community, which was founded in May 2005 and has (April 2007) approximately 700000 users all over Germany [18]. A very large and active local part of the user community is still based in the Munich area. The focus of the community is best described as communication- and spare-time-oriented. Central feature of the community is a simple social network model where two-way-handshake confirmed personal friendship relations can be created managed and visualized by the users. Besides "'friendship" no further types of relations exist. No weighting of relations is provided.

As a possible domain of interest, clubs in the Munich area where chosen. Lokalisten well correlated with the goal to investigate social recommenders and their relation to usual

Rating	Frequency	Percent
0	57566	69.4
1	2861	3.4
2	1796	2.2
3	2241	2.7
4	2430	2.9
5	3232	3.9
6	3203	3.9
7	3409	4.1
8	2974	3.6
9	1755	2.1
10	1517	1.8
Total	82984	100.0

Table 1: Distribution of Ratings

collaborative filtering recommenders in such a taste-related domain because it offered a social network representation and has a strong Munich user base that potentially know the clubs and have interest in them.

Using a breadth first search starting from one user (one of the authors), the friendship graph was traversed until depth 4 which in our experiment led to 4249 users. Their publicly available information (nicknames, friendship-relations and supplemental profile data) were downloaded and stored in a relational database. We only used nickname and friendship relations for our experiment.

In order collect suitable data to be able to compare collaborative filtering with social filtering operating on the taste related domain of clubs in Munich, an online-survey was constructed where users where asked to rate 82 clubs in the Munich area on a discrete scale from 0 to 10 where 0 (which was preselected) indicates that a user does not know the club. Thus 1 is the worst rating and 10 is the best rating. The users were sent a URL pointing to the Online-survey questionnaire contained in a message asking them to rate all the clubs they know.

Of the contacted users, 1012 users (aged between 16 and 47) completed the Online-questionnaire. 524 (51.8 %) of them were male, 488 (48.2 %) were female. Thus we received $82 * 1012 = 82948$ ratings. Table 1 shows the distribution of the ratings.

4.1.1 Basic Notations and Approaches

The social relationship graph $G(U, E)$ (nodes U correspond to users and undirected edges $E \subseteq \binom{U}{2}$ to friendship relations) is stored in an adjacency matrix A_{ij} ($A_{ij} = 1$ if $\{u_i, u_j\} \in E$, $A_{ij} = 0$ if $\{u_i, u_j\} \notin E$).

Denoting a 82-dimensional rating vector of a user u_i as

$$r^{(u_i)} = (r_0^{(u_i)}, r_1^{(u_i)}, r_2^{(u_i)}, ...r_{81}^{(u_i)}) \quad (1)$$

with each element $r_k^{(u_i)} \in [0, 10]$, we can build a 82×1012 rating Matrix M_{ur} with these rating vectors as columns.

We can build a user-user-similarity matrix $S_{ij} = sim(u_i, u_j)$ with respect to ratings only by

$$sim(u_i, u_j) = cos(r^{(u_i)}, r^{(u_j)}) = \frac{r^{(u_i)} \bullet r^{(u_j)}}{||r^{(u_i)}||||r^{(u_j)}||} \quad (2)$$

or

$$sim'(u_i, u_j) = sim(u_i, u_j)\ w_{co}(r^{(u_i)}, r^{(u_j)}) \quad (3)$$

respectively.

Using the cosine instead of the Pearson correlation (recommended for neighborhood-based CF in [11]) has two advantages: it is easier to compute and does not inadequately include 'missing data' (the zero values in M_{ur}). According to [22] both approaches should be equivalent.

The co-occurrence weight $w_{co}(r^{(u_i)}, r^{(u_j)})$ (or significance weight as it is called in [11]) is introduced to account for the fact that e.g. two vectors with only one item rated equally and all other items not rated at all $r^{(u_i)} = r^{(u_j)} = (0, 0, \ldots, 0, x, 0, \ldots, 0)$ would yield $sim(u_i, u_j) = 1$ while two vectors with lots of co-occurring very similar but not identical ratings might yield $sim(u_i, u_j) < 1$. That is, the similarity's trustworthiness should increase with the number of times the same item is rated by both users. For the sake of simplicity we incorporate the co-occurrence weight into the similarity.

Following in principle the suggestions in [11] the co-occurrence weight introduces a linear decrease $w_{co}(r^{(u_i)}, r^{(u_j)}) = co(r^{(u_i)}, r^{(u_j)})\ 1/\overline{co}$ if the number of co-occurring ratings is below the average number of co-occurrences $\overline{co} = 1/0.5 * U(U-1)\ \sum_{k<l} co(r^{(u_k)}, r^{(u_l)})$ between all users.

4.2 Experiment One

In the first experiment we investigate the statistical dependence of the rating behavior of the users and their social relations (groups and pairs).

As has been discussed in 2, we assume cliques in the relationship graph as good models for social groups. We have to compute sufficiently many cliques for our experiments. How this can be done and how time-complex state of the art algorithms are, is explained in appendix A.

For the social network in our experiment retrieved the number of cliques shown in table 2. In the following sections

Type of Group	Quantity
Pairs of Friends	561
Cliques (3 Members)	94
Cliques (4 Members)	58
Cliques (5 Members)	21
Cliques (6 Members)	3

Table 2: quantity of groups extracted from relationship data

we will first analyze whether the rating behavior in groups of friends is more similar than in arbitrarily chosen groups of people who do not know each other. To be more precise such a group of 'strangers' are usually called independent. We can find such independent sets or 'anti-cliques' by searching for cliques in the complement graph of G.

4.2.1 Rating Behavior: Friends vs. Non-Friend-Pairs

First we will investigate whether the independent variable 'rating similarity' is statistically dependent of the variable 'social relationship' (friends or no friends). In order to do this, we will use the χ^2 Test ([3]). We partition the continuous variable rating similarity's domain $r \in [0, 1]$ into 11 partitions, each having a similar number of total cases (friend-pairs and non-friend pairs). The results are shown in table 4 1. The content of this table is visualized in the upper sub-figure of figure 1. The H_0 Hypothesis is *The rating-similarity is independent from the social relationship between rater-pairs.* For the crosstable 4 1, a χ^2 value of 173.401 was computed. To accept the null hypothesis on a level $\alpha = 0.01$ with 10 degrees of freedom, the computed χ^2 should not be greater than 29.59. Since 173.401 > 29.59, the null hypothesis is rejected. Thus the two variables are statistically dependent. In order to verify the strength of

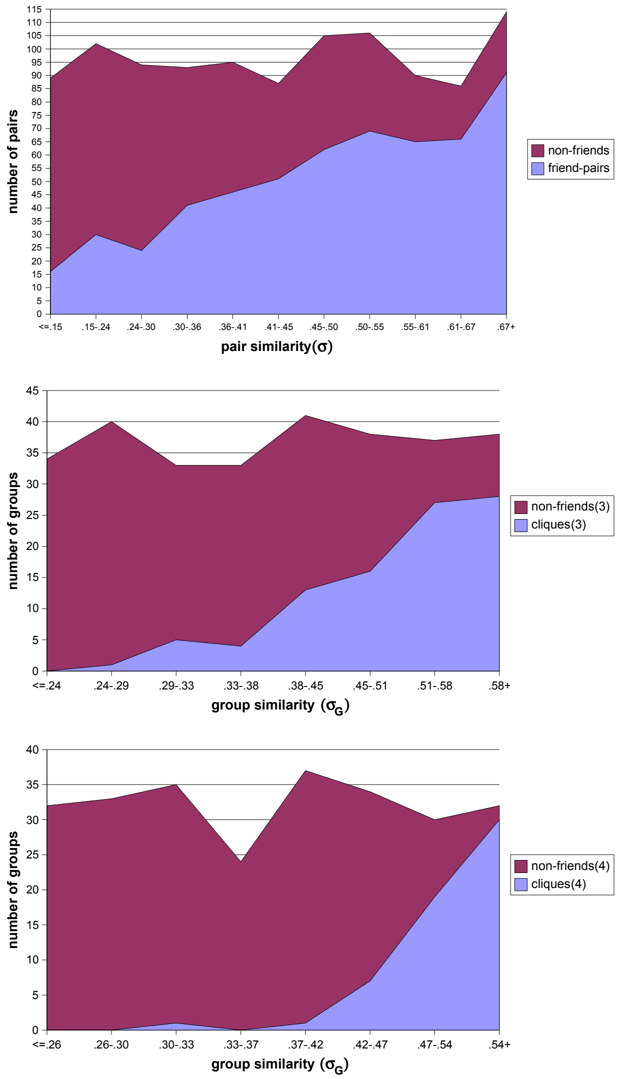

Figure 1: Comparison Chart of Friend-Pairs to Non-Friend Pairs (upper subfigure), Cliques(3) to Non-Friends(3)(middle subfigure) and Cliques(4) to Non-Friends(4) (lower subfigure)

the statistical dependency, we compute a Pearson correlation coefficient [3] the range of which is $[-1, 1]$. A value of 0 indicates no correlation while any absolute value near 1 indicates strong correlation. From table 4 ① we get a value of -0.397 indicating a rather strong negative correlation between ratings and friends. The coefficient is negative since a higher value for the first variable (rating similarity) is correlated with a lower value for the second variable (relation type) (friend-pairs are mapped to 0, non-friends to 1).

4.2.2 Rating Behavior: Cliques vs. Independent Sets

Since the small number of present cliques larger than 4 is statistically not significant, we confined ourselves to investigating the rating similarity in cliques of sizes 3 and 4 and comparing it to groups of non-friends (independent sets) of the same sizes.

The H_0 Hypothesis is *The rating-similarity is independent from the social relationship between rater-groups.*

In order to compare ratings of sets of users we have to define an average rating similarity σ_U for a group U as

$$\sigma_U = \frac{2}{|U| * |U-1|} \sum_{i=0}^{|U-1|} \sum_{k=i+1}^{|U-1|} sim(u_i, u_k) \qquad (4)$$

	Pearson Correlation w.r.t. Group-Similarity(σ_U)
Pairs	$^{**}-0.397$
Cliques(3)	$^{**}-0.568$
Cliques(4)	$^{**}-0.661$

**Correlation is significant at the 0.01 level (2-tailed)

Table 3: Pearson Correlation Results: Pairs, Cliques(3) and Cliques(4)

The values 164.2 (computed χ^2) and 18.5 (threshold χ^2) from table 5 show that the H_0 hypothesis is again rejected ($\alpha = 0.001$) and table 3 shows that the correlation between the average rating similarity and the variable differentiating between 3-cliques and 3-anti-cliques is even stronger than in case of the 2-cliques (pairs of friends) case from above.

Investigating the same dependencies for cliques of size 4 we find similar results (103.0 (computed χ^2) and 18.5 (threshold χ^2) from table 5 $\rightarrow$ H_0 hypothesis is again rejected ($\alpha = 0.001$)). Table 3 and lower sub-figure of figure 1 show that the correlation is even stronger for the case of 4-cliques than for 3-cliques.

4.3 Discussion of Experiment One

Arguably, the data set is not completely unbiased, since we started from one user only and crawled the graph of friendships from there. The missing 'total' representativeness of the chosen set of people is nevertheless not a big problem in the experiment since we can assume that for measuring the connection between social relatedness and rating behavior the *overall* taste-'bias' possibly present in the set does not influence the investigated differences. The possible bias or 'hidden' social relations in the set could even have been expected to influence the results in the other direction compared to what has been observed. Furthermore, the community chosen is definitely not night-life or club-centered per se.

Considering [28] p.256, the distribution of cliques that we found is congruent to what is usually observed in social network data, where the number of large cliques tends to be small.

From tables 4 ①-③ and the corresponding visualisations in figure 1 and from the Pearson correlation in table 3, we can very clearly see that groups of friends are more similar in ratings of taste related domains than groups of people that do not know each other (independent sets). This is in correspondence to observations in [2] and [30]. Even more so, the correlation becomes increasingly stronger when the number of people in the groups is raised from 2 to 3 to 4. The overall similarity average of cliques(4) of 0.556 is the highest observed value compared to cliques(3) with 0.526 and friend-pairs with a mean value of 0.485.

It may well be the case, that considering taste related domains with a lower 'social' relevance, the results may differ. But considering that for example clothing-, tv-, cinema-, and music-tastes also have a significant social component it may be difficult to find taste related domains without such a 'social' relevance.

Groups being 'centers of taste' is a phenomenon which has been reviewed in [10] and is well known in social sciences [10]. Among other reasons this is due to a 'normative' effect that group-taste may have on members of the group [10].

To sum it up, with the necessary caution considering generalizing, several conclusions may be drawn from these empirical results:

- Virtual friend-relationships are capable of providing similar ratings in taste related domains
- Even binary friend-relations on average show more rating similarity than disconnected pairs
- Cliques tend to be centers of taste showing a higher similarity than mere friend-pairs on the average
- Based on similarity comparisons, cliques and friend-pairs might prove as suitable recommendation source since they share a common taste regarding the investigated domain.

4.4 Experiment Two: Collaborative vs. Social Filtering

In this part of the experiment's analysis we investigate how a classical collaborative filtering algorithm performs in comparison to a social filtering or social recommendation algorithm.

As has been explained in section 3, in a classic collaborative approach, we have to perform three basic steps: computing similarities (matching), correlation-thresholding (neighborhood creation) and weighted average ratings (prediction computing). We can then compare the generated predictions (in our case predictions of club-ratings) with the true values (the true ratings). We will apply the same three step procedure also for a simple social filtering approach. The only difference will be in the neighborhood creation step, where social relations are used instead of correlation-thresholding.

Before we present the details of both approaches and the results of the experiment, we have to note down some basic preliminaries.

4.4.1 Notation and Quality Measures

As has been explained above, each user u_i has provided a rating vector $r^{(u_i)} = (r_0^{(u_i)}, r_1^{(u_i)}, r_2^{(u_i)}, ...r_{81}^{(u_i)})$ with $r_k^{(u_i)} \in [0, 10]$. A recommender system can (in principle) predict every one of these ratings. Thus the predicted rating for a user can be denoted as

$$pr^{(u_i)} = (pr_0^{(u_i)}, pr_1^{(u_i)}, r_2^{(u_i)}, ...pr_k^{(u_i)}) \tag{5}$$

with $0 \leq k \leq 81$.

We will have to distinguish between *adequate*, *inadequate* and *novel* predictions. An adequate prediction is a prediction for a club-rating $r_j^{(u_i)} \geq 1$ (that is for a club the user knows and has 'actively' rated) that is close to that original club-rating. A novel prediction is a prediction for a club-rating that was $r_j^{(u_i)} = 0$ (that is for a club the user does not know).

Formally this is expressed as:

A prediction $pr_j^{(u_i)}$ is *adequate* (denoted as $pr_j^{(u_i)} \in P_a^{(u_i)}$) if

$$pr_j^{(u_i)} \in P_a^{(u_i)} \iff (r_j^{(u_i)} \geq 1 \wedge pr_j^{(u_i)} \in [r_j^{(u_i)} - \delta, r_j^{(u_i)} + \delta]) \tag{6}$$

A prediction $pr_j^{(u_i)}$ is *inadequate* (denoted as $pr_j^{(u_i)} \in P_{ia}^{(u_i)}$) if

$$pr_j^{(u_i)} \in P_{ia}^{(u_i)} \iff (r_j^{(u_i)} \geq 1 \wedge pr_j^{(u_i)} \notin [r_j^{(u_i)} - \delta, r_j^{(u_i)} + \delta]) \tag{7}$$

A prediction $pr_j^{(u_i)}$ is *novel* (denoted as $pr_j^{(u_i)} \in P_n^{(u_i)}$) if

$$pr_j^{(u_i)} \in P_n^{(u_i)} \iff (r_j^{(u_i)} = 0 \ \wedge \ pr_j^{(u_i)} \neq 0) \tag{8}$$

The tolerance δ is a free parameter.

In our experiment, we distinguished, with respect to δ, between favorite clubs and non-favorite clubs. The logic is that favorite clubs need to be more precisely predicted (smaller δ (denoted as δ_f) by the algorithm in order to call the algorithm's performance 'good' than the less favored clubs (larger δ (denoted as δ_r). Thus we determined for every user her (at most) three favorite clubs to be able to compute for every user the sets $P_a^{(u_i)}$, $P_{ia}^{(u_i)}$ and $P_n^{(u_i)}$ and the overall sets $P_a = \bigcup_i P_a^{(u_i)}$, $P_{ia} = \bigcup_i P_{ia}^{(u_i)}$, and $P_n = \bigcup_i P_n^{(u_i)}$.

We furthermore define the set of all 'adequately predictable' ratings R_{ap} as the set of all the original ratings of all users ≥ 1.

We can then define *precision*, *recall*, *f-measure*, and *mean absolute error (MAE)* of a total run for the recommender system in a standard way:

$$Precision = \frac{|P_a|}{|P_a| + |P_i|} \tag{9}$$

$$Recall = \frac{|P_a|}{|R_{ap}|} \tag{10}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{11}$$

If we let the set $A = \{(i,j) | pr_j^{(u_i)} \notin P_n\}$ denote the set of all user-club-combination-indices for which the recommender actually produces a non-novel prediction, the mean absolute error is defined as:

$$MAE = \frac{1}{|A|} \sum_{\{(i,j) \in A} |pr_j^{(u_i)} - r_j^{(u_i)}| \tag{12}$$

(see [12] for an detailed discussion of these measures).

As has been explained above we implemented two variants of the recommender system: A conventional collaborative filtering system and a social recommender.

4.4.2 Neighborhood Creation and Prediction Computing

The conventional CF system's threshold-based neighborhood creation is governed by a parameter λ. All users u_j which have a rating similarity $S_{ij} = \text{sim}(u_i, u_j) \geq \lambda$ to user u_i are taken into account when computing the predictions (recommendations) for that user u_i. Thus the neighborhood $N_{\text{coll}}^{(u_i)}$ is defined as

$$N_{\text{coll}}^{(u_i)} = \{u_j \mid S_{ij} \geq \lambda\} \tag{13}$$

Note the following: In contrast to the first experiment where we (for the sake of simplicity) used equation (2) for S_{ij}, for the second experiment we use the co-occurrence weighted similarity (equation (3)) following the suggestions in [11] for a good CF algorithm.

In our experiment, depending on λ these sets can be much larger than the set of friends of a user

$$F^{(u_i)} = \{u_j \mid A_{ij} = 1\}. \tag{14}$$

Thus we took as a socially defined neighborhood the friends of a user together with the friends of these friends:

$$N^{(u_i)}_{\text{social}} = F^{(u_i)} \cup \{u_k \mid \exists u_j \in F^{(u_i)} : A_{jk} = 1\}. \tag{15}$$

Having defined the neighborhoods we can easily compute the prediction-vector for a user u_i in a standard way. Instead of simply averaging the ratings within the neighborhood

$$r^{(u_i)} = \frac{1}{|N^{(u_i)}|} \sum_{\{j|u_j \in N^{(u_i)}\}} r^{(u_j)} \tag{16}$$

we compute the predictions as similarity weighted average over the rating-vectors of the users in his neighborhood:

$$pr^{(u_i)}_m = \overline{r^{(u_i)}} + \frac{1}{\sum\limits_{\{j|u_j \in N^{(u_i)}\}} S_{ij}} \sum_{\{j|u_j \in N^{(u_i)}\}} (r^{(u_j)}_m - \overline{r^{(u_j)}})\, S_{ij} \tag{17}$$

with $N^{(u_i)} \in \{N^{(u_i)}_{\text{social}}, N^{(u_i)}_{\text{coll}}\}$.
The average rating for a user u is denoted by $\overline{r^{(u)}}$.

This approach takes into account that different users my have a substantially different rating bias (Thus the difference from the average rating is used in the sum). We weight this difference with the co-occurrence weighted rating similarity. These suggestions are due to [11]. In case of N_{social} we would naturally substitute the rating similarity with the strengths of the respective social relations, which is not present in our data set. To have a fair comparison between the two approaches, we also compute predictions using no similarity weighting:

$$pr'^{(u_i)}_m = \overline{r^{(u_i)}} + \frac{1}{|N^{(u_i)}|} \sum_{\{j|u_j \in N^{(u_i)}\}} (r^{(u_j)}_m - \overline{r^{(u_j)}}) \tag{18}$$

4.4.3 *Creating Sparse Training-Sets*

A Recommender System can be viewed as a classifier where training phase and classification phase are united in one step. We can thus think of the set of ratings as a 'training' set, allowing the recommender system to guess a rating for a user (make a prediction or recommendation).

We created 'sparse versions' of our original rating matrix M_{ur} (containing 25418 non-zero ratings). Randomly choosing $n * 1000$ ratings ($n \in \{1, 2, \ldots, 25\}$ we constructed 25 'sparse versions' of M_{ur} with decreasing sparseness in order to investigate the influence of sparseness on the two recommender variants. In order to be able to better compare the collaborative results with the social results we considered for every run of the sparse training data (that is for every $n \in [1, 25]$) the *best* of the ten runs for the collaborative recommender (that is the best out of all runs for each $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$) with respect to the f-measure values.

Furthermore, in order to be not to generous considering the quality of the predictions we chose to set very strict values for the parameter(s) controlling the adequateness of predictions ($\delta_f = 0.5$, $\delta_r = 1.0$).

To be better able to understand the MAE values from table 6 we computed an MAE resulting from a continuous

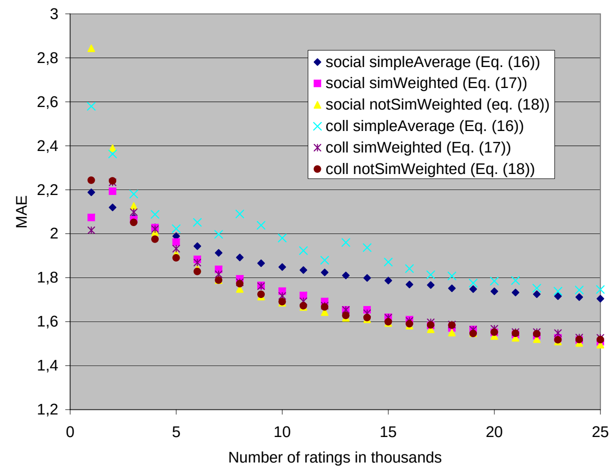

Figure 2: Comparison of mean absolute error(MAE) for social and collaborative predictions

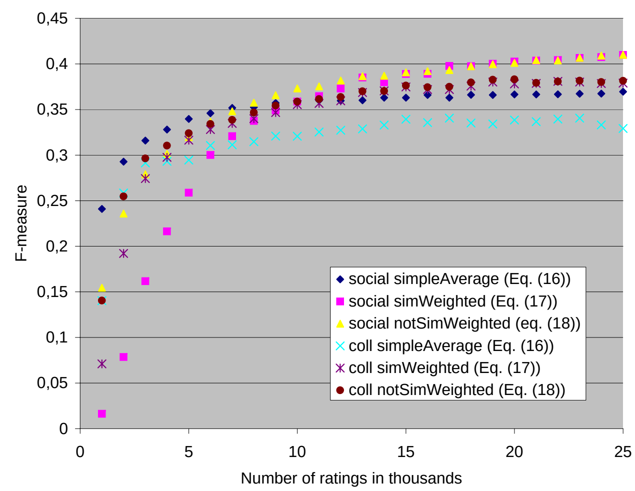

Figure 3: Comparison of F-measure for social and collaborative predictions

uniform random distribution for pr on [0,10] and a discrete uniform random distribution for r on $\{0, 1, \ldots, 10\}$. Simple calculations show that in this random case we would get a MAE of ≈ 3.35. This also confirms that our recommenders perform reasonably.

4.5 Discussion of Experiment Two

The collaborative results (table 6: ①, ② and ③ (visualized in figures 2 and 3)) show relatively low best-thresholds returning the best f-measure for a specific training set: Starting with 0.1 for $n = 1$ and $n = 2$ the maximum best threshold level was found from $n = 21$ upwards with a value of 0.6.

The observation of relatively low collaborative thresholds returning the best results was also made by [27] and [12]. A high similarity between the users in the neighborhood and the target user does not imply an equally high similarity among the neighborhood users themselves. As a result, their potential disagreement about some items and the small neighborhood size due to high correlation values lead to predictions which deviate to a great extent from the target users original rating ([27]).

Considering the f-measure results from table 6 which are visualized in figure 3, some interesting aspects show up: First of all, the social neighborhood with simple averaging prediction calculation (eq. (16)) is best by far for few ratings (high sparseness), being then beaten for low sparseness (many ratings) by social neighborhood with non-similarity weighted (eq. (18)) and similarity weighted (eq. (17)) predictions computing (which then almost coincide)). The similarity weighted version is substantially worse at high sparsity (few ratings) than the other two social approaches. Since according to [11] the similarity weighted collaborative filtering approach with similarity weighting that we use is one of the best neighborhood based filtering algorithms known, the social approach works excellently.

The social neighborhood with simple averaging prediction calculation (eq. (16)) is substantially better at high sparseness (cold-start-situations) than the social neighborhood with non-similarity weighted (eq. (18)) and especially than the similarity weighted (eq. (17)) versions. While the peer group of friends and friend-friends *as a whole* (on average) is obviously a good indicator of personal taste even in very sparse situations, the *single* friends and friends-friends may have a slightly different taste then the active user which becomes especially apparent in the highly sparse situations where few ratings are available. Another explanation for the similarity weighted approaches (both social and collaborative) being worse than non similarity weighted approaches is that the similarity measure becomes increasingly unreliable with increasing sparseness.

The collaborative approach (both sim.-weighted (eq. (17)) and non-sim.-weighted (eq. (18))) is only slightly better than the *worst* social version (simple averaging) but only when the sparseness is very low (many ratings). As has been confirmed by [11], simple averaging (eq. (16)) is not an option for collaborative filtering which shows up in our experiments too.

Another aspect is that the collaborative approach is very sensitive to the λ-threshold. We 'generously' took the values of the best run (w.r.t. to λ) for the collaborative approach. Deviating from this best λ by only 0.2 delivers devastatingly bad results in all 4 quality measures (Precision, Recall, F-measure, MAE). While fine tuning λ may be possible by observing the system, the optimal λ value increases substantially with decreasing sparseness and in a real recommender system may be very hard to determine. This also strongly speaks in favor of the social approach.

Considering the MAE results from table 6 which are visualized in figure 2, we find very similar results. Only the collaborative approaches with appropriate prediction calculation (both similarity weighted (eq. (17)) and non-similarity weighted (eq. (18)) are as accurate than the social approach for most levels of sparseness. But the accuracy comes at a price: The collaborative approach (all three versions) start with a very low λ (a large neighborhood) being able to make many novel predictions which are extremely in-accurate. At lower levels of sparseness (more ratings) and thus at higher λ (smaller neighborhoods) the number of novel predictions is substantially lower (by almost a factor of 25 %) than in case of the social approach (stable neighborhood) which, in turn starts with comparatively few novel predictions (high sparseness) and substantially increases this number. This is a clear advantage for the social approach.

Another more general criticized aspect of collaborative filtering systems is the lack of transparency. Collaborative systems today are black boxes, computerized oracles which give advice but cannot be questioned [15]. This lack of transparency can be overcome by a social recommender because the origin of recommendations (the set of users used) is transparent, which can to some extent be used to indicate to the user *why* a certain recommendation was made (see e.g. [8] for an example). Social recommendations may thus induce a higher level of trust in the system itself.

5. FUTURE APPLICATIONS

So besides mere 'performance' (e.g. with respect to F-measure) where do the social approaches have their particular strengths with respect to *applications*?

Social environment-models (such as cliques) can certainly represent valuable ordering and filtering primitives in case no other such criteria are present. For example, designing a mobile application which aims at bridging the gap between public transport (no personalization (e.g. w.r.t. to the travel destinations and travel times) cheap and environmentally acceptable) and individual transport (maximum personalization but expensive and environmentally not acceptable with a mid term future perspective). Such approaches could function like an agency for arranged lifts: Automatically offering an arranged lift, every time a user enters her destination into her navigation system. Other persons are the automatically informed if their personal context (travel destination, time etc.) matches the offer. Since not everybody is willing to take every person with her in her car, social groupings may be a more suitable target for the offering.

In general, sharing parts of the personal information space, as it is the case with any recommender system, is an interesting application for social models. It can be subjectively more appropriate to be presented elements from other user's information space (e.g. ratings) if these users have a social relation to the active user.

But quality of information is usually judged in a sense 'the system should provide me with ratings (or more general with information) as close to what i would have rated (or more general information that i would have chosen) if i had known this in advance'. Obviously this is not the only way in which an information could be *useful.* we can also imagine information to be useful which is of poor 'quality' (in the above sense). Think of novel recommendations that fall out of the scope of items that you would normally think of as being interesting but that would nevertheless *broaden your horizon.* As an example think of drinking beer: Most children would consider beer to be just bitter, but nevertheless once have grown up to adults many of them start drinking beer occasionally. If their beverage recommendations would solely have been based on their former tastes, most adults would still be mainly be drinking hot chocolate or fruit juice. So how do these new influences come into their lives? One important mechanism is due to social recommendations.

In other words your social groupings provide you all the time with horizon broadening information. That is novel information that is not just novel in the sense defined in the previous sections (a novel prediction was a prediction that 'you would have done yourself that way') but that is horizon broadening (a prediction/rating/information that 'you would *not* have done/found yourself that way'). As another example think of information retrieval and the difference between useful information that you find and that you have been searching for and of useful information that you did not search for. (see [10] for more on that).

Thus a group's recommendation also have a normative effect on the group members [10], which is not all bad, since everybody has (to a certain extent) a tendency to behave in a way that she is accepted in a group. Thus knowing what others of your group know or like is beneficial.

Without social recommenders, such an effect could not be reached because we would lack criteria on how to chose these horizon broadening recommendations.

In order to use that social structures for context-aware recommendation generation, we are currently working on an agent-based mobile peer-to-peer framework for generating not only context-aware recommendations but to generate *social-sensitive* AND context-aware recommendations.

6. SUMMARY AND CONCLUSIONS

We have shown that social networks can be used to in recommender systems in taste related domains With good success. As we have seen, the study we have conducted hints that social recommenders perform as good as the best CF approaches (in some situations (e.g. sparsity) and under some aspects (e.g. novel predictions) even clearly better). Future work will have to continue and deepen the investigation of how social models in general can be used to improve recommenders and information systems in general.

7. REFERENCES

[1] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.

[2] P. Bonhard and M. A. Sasse. 'knowing me, knowing you' – using profiles and social networking to improve recommender systems. *BT Technology Journal*, 24(3):84–98, 2006.

[3] J. Bortz. *Statistik*. Springer, Heidelberg, 2005.

[4] D. Boyd. Faceted id/entity: Managing representation in a digital world, 2001.

[5] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[6] M. F. Caplan and C. E. Miller. Group decision making and normative versus informational influence effects of type of issue and assigned decision rule. *Pers. Soc. Psychol. Bull*, pages 306–313, 1987.

[7] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks, 1997.

[8] J. A. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, College Park, MD, USA, 2005. Chair-James Hendler.

[9] M. S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[10] G. Groh. *Ad-Hoc-Groups in Mobile Communities - Detection, Modeling and Applications*. PhD thesis, 2005.

[11] J. L. Herlocker, J. A. Konstan, and J. T. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering systems. *Information Retrieval*, 5:287–310, 2002.

[12] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. 2(1):5–53, 2004.

[13] H. Kautz, B. Selman, and M. Shah. Referralweb: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[14] S. Kosub. Local density. *in: U. Brandes, T. Erlebach 'Network Analysis – Methodological Foundations', Springer LNCS 3418, 2004*, pages 112–140, 2004.

[15] M. Montaner, B. Lopez, and J. L. de la Rosa. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19(4):285–330.

[16] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.

[17] N.N. www.google.com/orkut. May 2007.

[18] N.N. www.lokalisten.de. May 2007.

[19] N.N. www.multiply.com. May 2007.

[20] E. Noelle-Neumann. The theory of public opinion: the concept of the spiral of silence. 1991.

[21] J. O'Donovan and B. Smyth. Trust in recommender systems. In *IUI '05: Proc. 10th intern. conf. on Intelligent user interfaces*, pages 167–174, New York, NY, USA, 2005. ACM Press.

[22] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proc. 16th. Ann. Conf. on Uncertainty in AI (UAI-2000)*, pages 473–480. Morgan Kaufman, 2000.

[23] H. A. Shuji Tsukiyama, Mikio Ide and I. Shirakawa. A new algorithm for enumerating all the maximal independent sets. *Siam Journal on Computing*, 6(3):505–517, 1977.

[24] R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. 2001.

[25] M. Smith. Invisible crowds in cyberspace: Measuring and mapping the social structure of usenet. in marc smith and peter kollock, editors. *Communities in Cyberspace*, 1999.

[26] M. A. Smith and A. T. Fiore. Visualization components for persistent conversations. in proceedings of the sigchi conference on human factors in computing systems. *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 136143, 2001.

[27] E. G. Vozalis and K. G. Margaritis. Recommender systems: An experimental comparison of two filtering algorithms. 2003.

[28] S. Wasserman and K. Faust. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, 1994.

[29] B. Wellman. An electronic group is virtually a social network. *In Sara Kiesler, editor*, pages 179–205, 1997.

[30] I. Yaniv. Receiving other people's advice: Influence and benefit. Discussion paper series, Center for Rationality and Interactive Decision Theory, Hebrew University, Jerusalem, Sep 2005.

[31] P. G. Zimbardo and R. J. Gerrig. *Psychologie*. Springer-Verlag Berlin Heidelberg, 1999.

APPENDIX

A. DETECTING CLIQUES

In an undirected graph with n vertices (nodes) a naive upper bound for the number of cliques is obviously 2^n. A theorem By Moon and Moser [16][14] gives a tighter upper bound ('An undirected graph with n vertices has at most $3^{\lceil \frac{n}{3} \rceil}$ maximal cliques'), but thus one will have to expect, in general, an exponential number of cliques, which will, in the general case, prevent the existence of a polynomial time algorithm for the enumeration of all maximal cliques. An obvious naive approach is exhaustive search, for the time complexity of which a simple upper bound is $O(n^2 2^n)$.

Even searching for specific cliques is generally computationally hard. Deciding for a given Graph and a number $k \in \mathbb{N}$ whether there is a clique of lat least size k is an NP-complete problem [14].

But there are easy other problems which can be solved for a graph $G(V,E)$ in time $O(|V| + |E|)$. E.g. determining for a set $U \subseteq V$ if it is a clique in G or determining for a clique $U \subseteq V$ if it is maximal or (assuming an ordering on V) determining for a clique U' the lexicographically smallest clique containing U'. See [14] for details.

Modern algorithms for the enumeration of all maximal cliques exist that run in polynomial total time. That is these algorithms output all ν existing maximal cliques in a time bounded by a polynomial in $|V|$ and ν [14]. If the number of maximal cliques in the graph is comparatively small (the nodes have small neighborhoods on average) then these algorithms can be expected to deliver all maximal cliques in reasonable time.

Since we are only interested in computing 'enough' cliques to test the connection between rating behavior and social relations, we want an algorithm that will not run a long time before putting out cliques. Thus we are interested in a polynomial total time algorithm with polynomial delay. Polynomial delay means that the delay until the first output, the delay between consecutive output and the delay until the last output are polynomially bound [14].

Such an algorithm was published in [23]. It enumerates all maximal cliques of a graph with polynomial delay $O(n^3)$ using $O(|V| + |E|)$ space. It works like this [14]:

First construct a binary tree with $|V| = n$ levels having leaves only at level n. The nodes at level i represent all the maximal cliques of the induced graph $G[\{v_1, ..., v_i\}]$. Thus the leaves of the tree represent all maximal cliques of $G[\{v_1, ..., v_n\}] = G$. Thus a suitable tree traversal that outputs all leaves does the job.

In order to construct the tree we will have to do the following: Considering level i and a maximal clique U (in $G[\{v_1, ..., v_i\}]$) on that level. When determining the at most two children of U on level $i+1$, two cases arise:

(1) Either all vertices of U are adjacent to v_{i+1} in G. Thus $U \cup \{v_{i+1}\}$ is a maximal clique in $G[\{v_1, ..., v_i, v_i + 1\}]$. U then has only this one child $U \cup \{v_{i+1}\}$.

(2) Or there are vertices in U not adjacent to v_{i+1}. Certainly U itself is then a maximal clique in $G[\{v_1, ..., v_i, v_i + 1\}]$. If the clique $U \setminus \bar{N}(v_{i+1}) \cup \{v_{i+1}\}$ (where $\bar{N}(v_{i+1})$ denotes all the nodes not adjacent to v_{i+1}) is maximal (which does not necessarily needs to be the case) it is also a potential child for U. Since this node could potentially be the child of many other nodes on level i we place it under the lexicographically smallest U on level i (if it is maximal).

A simple further consideration yields the delay estimation $O(n^3)$ and the space complexity $O(n + |E|)$ [14]. Thus if we have ν maximal cliques in the graph we have a polynomial total time complexity of $O(n^3 \nu)$.

In our case, the average neighborhood of a node (the average number of friends is 2.48 thus we expect not too many cliques and will therefore be able to enumerate all of them in reasonable time.

B. RESULT TABLES

	Number of Pairs		
Similarity	friend-pair	non-friend pair	Total
$\leq .15$	16	73	89
$.15 - .24$	30	73	102
$.24 - .24$	24	72	94
$.30 - .36$	41	52	93
$.36 - .41$	46	49	95
$.41 - .45$	51	36	87
$.45 - .50$	62	43	105
$.50 - .55$	69	37	106
$.55 - .61$	65	25	90
$.61 - .67$	66	20	86
$\geq .67$	91	23	114
Total	561	500	1061

Crosstable ①: Similarity, Friend-Pairs and Non-Friends

	Number of Groups		
Similarity	cliques(3)	non-friends(3)	Total
$\leq .24$	0	34	34
$.24 - .29$	1	39	40
$.29 - .33$	5	28	33
$.33 - .38$	4	29	33
$.38 - .45$	13	28	41
$.45 - .51$	16	22	38
$.51 - .58$	27	10	37
$.58 \geq$	28	10	38
Total	94	200	294

Crosstable ②: Similarity, Cliques(3) and Non-Friends(3)

	Number of Groups		
Similarity	cliques(4)	non-friends(4)	Total
$\leq .26$	0	32	32
$.26 - .30$	0	33	33
$.30 - .33$	1	34	35
$.33 - .37$	0	25	25
$.37 - .42$	1	36	37
$.42 - .47$	7	27	34
$.47 - .54$	19	11	30
$.54 \geq$	30	2	32
Total	58	200	258

Crosstable ③: Similarity, Cliques(4) and Non-Friends(4)

Table 4: Crosstables

Pairs	Significance Level	0.01
	Degrees of Freedom	10
	Chi-Square	29.59
	Computed Pearson Chi-Square	^{a}173.401
Cliques(3)	Significance Level	0.01
	Degrees of Freedom	7
	Chi-Square	18.48
	Computed Pearson Chi-Square	^{a}164.209
Cliques(4)	Significance Level	0.01
	Degrees of Freedom	7
	Chi-Square	18.48
	Computed Pearson Chi-Square	^{a}103.016

a. 0 cells have expected count less than 5.
The minimum expected count is 40.53.

Table 5: χ^2-Test - Friend-Pairs, Cliques(3) and Cliques(4)

λ	n	P_a	P_{ia}	P_n	*Prec.*	*Recall*	*F-meas.*	*MAE*
0.1	1	2305	5109	7099	0.3109	0.0907	0.1404	2.5798
0.1	2	5856	14055	26224	0.2941	0.2304	0.2584	2.3630
0.2	3	7110	16252	36428	0.3043	0.2797	0.2915	2.1808
0.2	4	7398	17653	43266	0.2953	0.2911	0.2932	2.0879
0.2	5	7528	18163	47405	0.2930	0.2962	0.2946	2.0227
0.3	6	7864	17372	43921	0.3116	0.3094	0.3105	2.0517
0.3	7	7965	17786	46418	0.3093	0.3134	0.3113	1.9974
0.4	8	7642	15514	34689	0.3300	0.3007	0.3147	2.0896
0.4	9	7896	15894	34858	0.3319	0.3106	0.3209	2.0385
0.4	10	8025	16601	37065	0.3259	0.3157	0.3207	1.9806
0.4	11	8246	17032	39685	0.3262	0.3244	0.3253	1.9234
0.4	12	8335	17166	40334	0.3268	0.3279	0.3273	1.8798
0.5	13	7612	13282	23256	0.3643	0.2995	0.3287	1.9603
0.5	14	7992	14584	25887	0.3540	0.3144	0.3330	1.9369
0.5	15	8296	15186	28704	0.3533	0.3264	0.3393	1.8712
0.5	16	8330	15883	30827	0.3440	0.3277	0.3357	1.8413
0.5	17	8513	16058	32513	0.3465	0.3349	0.3406	1.8131
0.5	18	8427	16436	34241	0.3389	0.3315	0.3352	1.8077
0.5	19	8414	16540	35804	0.3372	0.3310	0.3341	1.7755
0.6	20	7803	12887	20409	0.3771	0.3070	0.3385	1.7846
0.6	21	7979	14000	23113	0.3630	0.3139	0.3367	1.7869
0.6	22	8152	14457	25293	0.3606	0.3207	0.3395	1.7524
0.6	23	8274	14892	27563	0.3572	0.3255	0.3406	1.7384
0.6	24	8160	15419	29052	0.3461	0.3210	0.3331	1.7443
0.6	25	8117	15776	31102	0.3397	0.3193	0.3292	1.7467

① Collaborative Prediction Results: 25 test sets with tolerance ranges $\delta_f = 0.50$ and $\delta_r = 1.00$ (best-threshold results only) using the simple averaging over neighborhoods (equation(16)) and co-occurrence weighted cosine similarity (equation(3))

λ	n	P_a	P_{ia}	P_n	*Prec.*	*Recall*	*F-meas.*	*MAE*
0.1	1	999	1679	1875	0.3730	0.0393	0.0711	2.0160
0.1	2	3588	8364	12291	0.3002	0.1412	0.1921	2.2337
0.1	3	6275	14087	27282	0.3082	0.2469	0.2742	2.0983
0.1	4	7364	16705	38282	0.3060	0.2897	0.2976	2.0209
0.1	5	8045	17386	45277	0.3163	0.3165	0.3164	1.9319
0.2	6	8372	17259	45897	0.3266	0.3294	0.3280	1.8689
0.2	7	8601	17374	49006	0.3311	0.3384	0.3347	1.8157
0.2	8	8767	17384	50989	0.3352	0.3449	0.3400	1.7792
0.3	9	8880	16925	44855	0.3441	0.3494	0.3467	1.7616
0.3	10	9125	16823	46206	0.3517	0.3590	0.3553	1.7173
0.3	11	9183	16890	47774	0.3522	0.3613	0.3567	1.6928
0.3	12	9258	16829	48065	0.3549	0.3642	0.3595	1.6732
0.4	13	9397	16158	41257	0.3677	0.3697	0.3687	1.6539
0.4	14	9468	16230	41815	0.3684	0.3725	0.3704	1.6384
0.4	15	9594	16197	43608	0.3720	0.3774	0.3747	1.6186
0.4	16	9534	16323	43982	0.3687	0.3751	0.3719	1.6037
0.4	17	9533	16314	44439	0.3688	0.3750	0.3719	1.5969
0.4	18	9629	16216	44650	0.3726	0.3788	0.3757	1.5869
0.5	19	9562	15328	35477	0.3842	0.3762	0.3802	1.5635
0.5	20	9553	15566	36949	0.3803	0.3758	0.3780	1.5679
0.5	21	9558	15493	37674	0.3815	0.3760	0.3787	1.5530
0.5	22	9644	15589	38695	0.3822	0.3794	0.3808	1.5530
0.5	23	9625	15595	39306	0.3816	0.3787	0.3801	1.5483
0.6	24	9269	14292	28974	0.3934	0.3647	0.3785	1.5281
0.6	25	9338	14517	31019	0.3914	0.3674	0.3790	1.5268

② Collaborative Prediction Results: 25 test sets with tolerance ranges $\delta_f = 0.50$ and $\delta_r = 1.00$ (best-threshold results only) using similarity weighted averaging over neighborhoods (equation(17)) and co-occurrence weighted cosine similarity (equation(3))

λ	n	P_a	P_{ia}	P_n	*Prec.*	*Recall*	*F-meas.*	*MAE*
0.1	1	2266	4564	6218	0.3318	0.0891	0.1405	2.2435
0.1	2	5722	13797	25296	0.2932	0.2251	0.2547	2.2405
0.1	3	7338	16751	39245	0.3046	0.2887	0.2964	2.0519
0.2	4	7824	17170	42967	0.3130	0.3078	0.3104	1.9751
0.2	5	8277	17399	47218	0.3224	0.3256	0.3240	1.8901
0.2	6	8583	17377	49863	0.3306	0.3377	0.3341	1.8283
0.2	7	8728	17408	51514	0.3339	0.3434	0.3386	1.7888
0.3	8	8916	17111	48296	0.3426	0.3508	0.3467	1.7726
0.3	9	9128	16978	48390	0.3497	0.3591	0.3543	1.7244
0.3	10	9246	16885	48841	0.3538	0.3638	0.3587	1.6909
0.3	11	9320	16844	49386	0.3562	0.3667	0.3614	1.6723
0.4	12	9257	16220	40270	0.3633	0.3642	0.3637	1.6671
0.4	13	9462	16282	42882	0.3675	0.3723	0.3699	1.6284
0.4	14	9482	16326	42851	0.3674	0.3730	0.3702	1.6185
0.4	15	9646	16223	44355	0.3729	0.3795	0.3762	1.5995
0.4	16	9606	16294	44628	0.3709	0.3779	0.3744	1.5905
0.4	17	9620	16279	44846	0.3714	0.3785	0.3749	1.5855
0.5	18	9543	15303	34200	0.3841	0.3754	0.3797	1.5842
0.5	19	9637	15311	35769	0.3863	0.3791	0.3827	1.5462
0.5	20	9687	15463	37195	0.3852	0.3811	0.3831	1.5524
0.5	21	9583	15521	37835	0.3817	0.3770	0.3793	1.5471
0.5	22	9651	15627	38941	0.3818	0.3797	0.3807	1.5444
0.6	23	9267	13890	27516	0.4002	0.3646	0.3816	1.5177
0.6	24	9311	14266	29037	0.3949	0.3663	0.3801	1.5191
0.6	25	9406	14484	31080	0.3937	0.3701	0.3815	1.5185

③ Collaborative Prediction Results: 25 test sets with tolerance ranges $\delta_f = 0.50$ and $\delta_r = 1.00$ (best-threshold results only) using similarity weighted averaging over neighborhoods (equation(18)) and co-occurrence weighted cosine similarity (equation(3))

n	P_a	P_{ia}	P_n	*Prec.*	*Recall*	*F-meas.*	*MAE*
1	4459	7131	13670	0.3847	0.1754	0.2410	2.1886
2	6046	9827	20534	0.3809	0.2379	0.2928	2.1199
3	6885	11280	25108	0.3790	0.2709	0.3159	2.0739
4	7395	12274	28501	0.3760	0.2909	0.3280	2.0288
5	7829	12866	30629	0.3783	0.3080	0.3396	1.9891
6	8112	13374	32582	0.3775	0.3191	0.3459	1.9436
7	8369	13799	34345	0.3775	0.3293	0.3517	1.9128
8	8468	14153	35741	0.3743	0.3331	0.3525	1.8923
9	8668	14443	37308	0.3751	0.3410	0.3572	1.8656
10	8738	14665	38623	0.3734	0.3438	0.3580	1.8485
11	8829	14819	39592	0.3734	0.3474	0.3599	1.8346
12	8861	14991	40344	0.3715	0.3486	0.3597	1.8239
13	8889	15064	41084	0.3711	0.3497	0.3601	1.8105
14	9008	15210	41912	0.3720	0.3544	0.3630	1.7990
15	9012	15240	42055	0.3716	0.3546	0.3629	1.7871
16	9122	15308	42723	0.3734	0.3589	0.3660	1.7693
17	9062	15442	43388	0.3698	0.3565	0.3630	1.7667
18	9158	15453	44095	0.3721	0.3603	0.3661	1.7516
19	9161	15494	43975	0.3716	0.3604	0.3659	1.7484
20	9197	15554	44796	0.3716	0.3618	0.3666	1.7383
21	9206	15593	45048	0.3712	0.3622	0.3666	1.7328
22	9222	15655	45373	0.3707	0.3628	0.3667	1.7247
23	9248	15692	45866	0.3708	0.3638	0.3673	1.7160
24	9258	15712	46042	0.3708	0.3642	0.3675	1.7117
25	9322	15698	46343	0.3726	0.3667	0.3696	1.7049

④ Social Prediction Results: 25 test sets with tolerance ranges $\delta_f = 0.50$ and $\delta_r = 1.00$ using the simple averaging over neighborhoods (equation(16))

n	P_a	P_{ia}	P_n	*Prec.*	*Recall*	*F-meas.*	*MAE*
1	211	361	278	0.3689	0.0083	0.0162	2.0737
2	1135	2349	2290	0.3258	0.0447	0.0786	2.1932
3	2708	5368	6443	0.3353	0.1065	0.1617	2.0643
4	4077	8202	11560	0.3320	0.1604	0.2163	2.0278
5	5294	10221	16435	0.3412	0.2083	0.2587	1.9623
6	6531	11572	20986	0.3608	0.2569	0.3001	1.8830
7	7291	12751	24687	0.3638	0.2868	0.3207	1.8378
8	7910	13566	28007	0.3683	0.3112	0.3374	1.7950
9	8357	14107	30804	0.3720	0.3288	0.3491	1.7647
10	8716	14483	33242	0.3757	0.3429	0.3586	1.7386
11	8974	14840	35373	0.3768	0.3531	0.3646	1.7190
12	9262	14992	37034	0.3819	0.3644	0.3729	1.6914
13	9642	15016	38351	0.3910	0.3793	0.3851	1.6531
14	9565	15389	40103	0.3833	0.3763	0.3798	1.6536
15	9823	15262	40291	0.3916	0.3865	0.3890	1.6197
16	9872	15453	41186	0.3898	0.3884	0.3891	1.6092
17	10137	15412	42531	0.3968	0.3988	0.3978	1.5831
18	10149	15503	43403	0.3956	0.3993	0.3974	1.5713
19	10233	15496	43334	0.3977	0.4026	0.4001	1.5645
20	10323	15556	44410	0.3989	0.4061	0.4025	1.5513
21	10363	15574	44997	0.3995	0.4077	0.4036	1.5419
22	10395	15619	45406	0.3996	0.4090	0.4042	1.5364
23	10470	15626	45821	0.4012	0.4119	0.4065	1.5262
24	10507	15641	46101	0.4018	0.4134	0.4075	1.5167
25	10582	15629	46537	0.4037	0.4163	0.4099	1.5086

⑤ Social Prediction Results: 25 test sets with tolerance ranges $\delta_f = 0.50$ and $\delta_r = 1.00$ using the similarity weighted approach (equation(17))

n	P_a	P_{ia}	P_n	*Prec.*	*Recall*	*F-meas.*	*MAE*
1	2674	6528	10118	0.2906	0.1052	0.1545	2.8433
2	4784	10350	18811	0.3161	0.1882	0.2359	2.3894
3	6069	12040	24157	0.3351	0.2388	0.2789	2.1262
4	6852	13221	28167	0.3414	0.2696	0.3013	2.0063
5	7451	13735	30349	0.3517	0.2931	0.3197	1.9089
6	8013	14188	32488	0.3609	0.3152	0.3365	1.8414
7	8419	14526	34304	0.3669	0.3312	0.3481	1.7886
8	8745	14778	36013	0.3718	0.3440	0.3574	1.7491
9	9046	15026	37663	0.3758	0.3559	0.3656	1.7143
10	9295	15111	39068	0.3808	0.3657	0.3731	1.6863
11	9405	15281	39920	0.3810	0.3700	0.3754	1.6668
12	9612	15340	40765	0.3852	0.3782	0.3817	1.6435
13	9758	15301	41493	0.3894	0.3839	0.3866	1.6177
14	9828	15546	42382	0.3873	0.3867	0.3870	1.6109
15	9931	15467	42630	0.3910	0.3907	0.3908	1.5942
16	10000	15603	43293	0.3906	0.3934	0.3920	1.5820
17	10049	15646	43932	0.3911	0.3953	0.3932	1.5652
18	10182	15625	44680	0.3945	0.4006	0.3975	1.5507
19	10247	15604	44655	0.3964	0.4031	0.3997	1.5475
20	10311	15672	45364	0.3968	0.4057	0.4012	1.5348
21	10395	15635	45709	0.3993	0.4090	0.4041	1.5264
22	10402	15699	46064	0.3985	0.4092	0.4038	1.5210
23	10493	15687	46508	0.4008	0.4128	0.4067	1.5102
24	10563	15662	46708	0.4028	0.4156	0.4091	1.5032
25	10599	15680	47055	0.4033	0.4170	0.4100	1.4956

⑥ Social Prediction Results: 25 test sets with tolerance ranges $\delta_f = 0.50$ and $\delta_r = 1.00$ using the approach without similarity (equation(18))

Table 6: Prediction Results