The Effect of Correlation Coefficients on Communities of Recommenders

Neal Lathia
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
n.lathia@cs.ucl.ac.uk

Stephen Hailes
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
s.hailes@cs.ucl.ac.uk

Licia Capra
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
l.capra@cs.ucl.ac.uk

ABSTRACT

Recommendation systems, based on collaborative filtering, offer a means of sifting through the enourmous amounts of content on the web by composing user ratings in order to generate predicted ratings for other users. These kinds of systems can be viewed as a network of interacting peers, where each user is a node and the links to all other nodes are weighted according to how similar the corresponding users are. Predicted ratings are generated for a user for unknown items by requesting and aggregating rating information from the surrounding neighbors. However, the different methods of computing user similarity, or weighting the network links, very often do not agree with each other, and, as a result, the structure of the network of recommenders changes completely. In this work we perform an analysis of a range of similarity measures, comparing their performance in terms of prediction accuracy and coverage. This allows us to understand the effect that similarity measures have on predicted ratings. Based on the obtained results, we argue that user-similarity may not sufficiently capture the relationships that recommenders could otherwise share in order to maximise the utility of these communities.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Collaborative Filtering

Keywords

Correlation, Recommender Systems

1. INTRODUCTION

Recommendation systems, based on collaborative filtering techniques, aim at relieving the problem of information overload [1] on the web, by providing users with personalised recommendations. Since the founding principle of collaborative filtering is to take advantage of like-minded individuals to generate recommendations, a recommendation system can be viewed as a network of users, linked according to their similarity, who exchange rating information with each other in order to produce predicted ratings of unknown items.

Recommendations are created for a user following a procedure that can be decomposed into three steps. In the first step, rating information, or *opinions*, of a set of items are collected from a set of users, or selected *neighborhood*. The opinions are then combined to generate *predicted ratings*. Lastly, the predicted ratings are sorted and *recommendations* are presented to the user. The starting point for generating recommendations is the opinions of other users; any user providing rating information thus becomes a *recommender*. Recommenders are linked to each other by measuring how similar they are to other recommenders, and a *community* is formed, actively collaborating with each other to filter the overwhelming amount of content available on the web.

The structure of these kinds of networks will be determined by the method that is used to measure user similarity, which is derived by comparing two user's historical ratings. The principle of using like-minded individuals is expressed by weighting opinions from the entire neighborhood based upon these similarities, that range from $+1$, or perfect correlation, to -1, or polar opposite preferences. The most well known techniques to computer user similarity are the Pearson Correlation Coefficient (PCC), and Vector Similarity (VS) [2], although many more exist. There are a wide range of techniques that can be implemented which achieve comparable results to the most prevalent and cited correlation coefficients, such as concordance-based measures [3]. Each method uses varying amounts of profile information, as will be explored in Section 2, but nevertheless produces very similar prediction accuracy results.

Although different coefficients will lead to comparable performance results, there is an underlying problem that is best demonstrated with an example. If Alice's rating history for five items, on a five-point rating scale, is $[2, 3, 1, 5, 3]$, and Bob's rating history for the same items is $[4, 1, 3, 2, 3]$, then the VS coefficient will be about 0.76. The PCC will return -0.50, while adding significance-weighting will produce -0.05. Other methods will result in equally different values. There is no consensus between the different methods as to the quality of recommendations that Alice and Bob could exchange. Just as the relationship between Alice and Bob

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil
Copyright 2008 ACM 978-1-59593-753-7/08/0003 ...$5.00.

will change from good to bad depending on how they compute their similarity, selecting different coefficients will alter the weightings of all the user-pairs in the community, and the network that represents the interacting recommenders will be remarkably different. The similarity coefficient values will, in turn, affect the prediction accuracy and coverage of the collaborative filtering process. However, no method significantly outperforms the others, which seems to contradict the dependence of predicted ratings on user similarity measures.

Any attempt at finding the "best" user weighting, to date, can only be done by conducting an analysis on comparative results of different techniques applied to the same dataset of user ratings; there is no way of measuring how close these algorithms are to an optimal answer. However, if these datasets are regarded as a community of recommenders, new performance benchmarks can be produced, highlighting two scenarios that are possible within a network of collaborating peers. The first is a worst-case scenario: we construct a network of recommenders based on random-valued links, and observe how accurately this scenario can generate predicted ratings. The second addresses prediction coverage, by comparing different coefficient's performance with respect to the optimal coverage that is possible on a dataset. These benchmarks are not new ways of measuring recommendation error, such as the tasks proposed in [4], but instead give researchers a basis for comparison that is not simply alternative algorithms to their own.

This work can be subdivided into three questions. We first look at a range of similarity measurements in Section 2, that each use varying amounts of profile information. We then address the issue of how a particular coefficient will affect the structure of the community of recommenders in Section 3. Following this, we evaluate the coefficients with respect to the benchmark results, and observe how well current collaborative methods perform relative to them using the MovieLens[1] dataset.

2. MEASURES OF SIMILARITY

The simplest similarity measure between two user profiles can be derived using information that disregards the actual ratings themselves, but considers two other factors. The act of rating an item is a conscious decision made by human users, and represents a judgment on a product that has been "consumed" (viewed, listened to, etc). Furthermore, the mere problem of information overload explains that recommendation systems are built to aide product-selection. Therefore, when two users have selected the same product, they already share a common characteristic: their choice to consume and rate that product. This similarity measure disregards each user's judgment of the item, and weights users according to the proportion of co-rated to rated items:

$$w_{a,b} = \frac{|R_{a,i} \cap_i R_{b,i}|}{|R_{a,i} \cup_i R_{b,i}|} \quad (1)$$

Concordance-based measures [3] build upon the idea of co-rating similarity to generate coefficients that aim at finding the degree of agreement on the intersection of two user's profiles. A pair of ratings, $r_{a,i}$ and $r_{b,i}$, for item i by users a and b is concordant if the difference between each rating and the respective user's mean shares the same sign. On the other hand, if one rating lies above its user's mean (difference > 0), and the other lies below (difference < 0), then the ratings are discordant- they disagree. Lastly, if one of the ratings is equal to the user's mean, or the item is unrated, then that particular user can not express any useful information about the item, and so the ratings are tied. By counting the number of concordant (C), discordant (D), and tied (T) pairs, and the number of rate-able items, N, proportions of agreement between the two users can be derived. As described in [5] there are varying ways of combining this information; in this work we focus on Somers' d:

$$w_{a,b} = \frac{C - D}{N - T} \quad (2)$$

Lastly, similarity measures such as the PCC aim to measure the degree of agreement between two users, thus including the idea of "how much" a user may have liked or disliked an item. It does so by measuring the extent to which a linear relationship exists between the two users' historical ratings [2].

$$w_{a,b} = \frac{\Sigma_{i=1}^{N}(r_{a,i} - \bar{r}_a)(r_{b,i} - \bar{r}_b)}{\sqrt{\Sigma_{i=1}^{N}(r_{a,i} - \bar{r}_a)^2 \Sigma_{i=1}^{N}(r_{b,i} - \bar{r}_b)^2}} \quad (3)$$

The PCC has been subject to a number of improvements. For example, [2] was the first to introduce significance- weighting: a coefficient would be scaled by $n/50$, where n is the number of co-rated items, if the two users had co-rated less than 50 items. This extension is based on the observation that although correlation coefficients demonstrate convergent behavior over time (as they are recomputed with growing profiles), the values it takes when very few items have been co-rated varies wildly. Significance-weighting, in essence, attempts to incorporate a degree a reliability into the coefficient, and, in fact, [2] (with more recent work in [6]) reported improved prediction results. There are also other heuristics that have been applied; for example, the constrained-PCC uses the rating scale midpoint, rather than the user's mean.

All of these measures are used to compute correlation coefficients $w_{a,b}$ that contribute to the generation of predicted ratings by acting as *weights* on the opinions received from neighbors [2]:

$$p_{a,i} = \bar{r}_a + \frac{\Sigma(r_{b,i} - \bar{r}_b)w_{a,b}}{\Sigma w_{a,b}} \quad (4)$$

The continuing research to improve the performance of recommender systems, based on profile-similarity, acknowledges and demonstrates that certain coefficients will perform better than others. However, there is a lack of a full description as to why such performance differences are occurring, either by investigating the nature of the underlying dataset, or the effect that the methods used for weighting user pairs is having on the classification of items.

3. DISTRIBUTION OF CORRELATION COEFFICIENTS

To gain insight into the effect of correlation coefficients on the collaborative filtering process, we return to the alternative view of the system, based on a network of recommenders, where each user is a node in the network. The nodes are linked to each other according to the similarity between the linked pair of users. By looking at the distribution of coefficient values over the entire range of users we can

[1]http://www.grouplens.org/

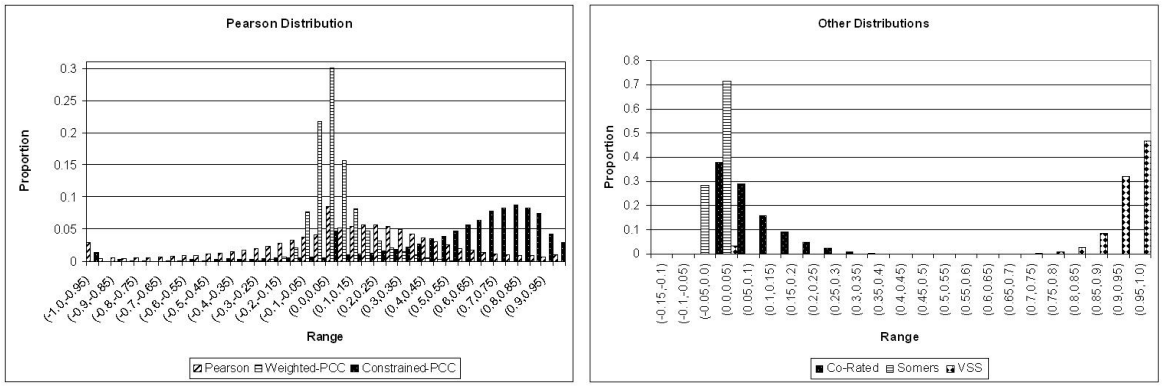

Figure 1: Distribution of Coefficients in the Community

visualise how any method of deriving coefficients affects the community, and its ability to generate recommendations.

We investigated the nature of these different similarity measures by looking at their distribution over the full range of available neighbors in the MovieLens dataset. We first computed all the coefficients between every pair of users, using all available profile information (i.e. without dividing the dataset into training/test sets). We then plotted the proportion of the total number of coefficients that fell within a given range (of size 0.05) to be able to see how these coefficients are shared out among all the available user pairs in Figure 1.

An analysis of the distribution of correlation coefficients in the community of recommenders may, at first glance, seem inappropriate, since the coefficient values will change over time, as they are recomputed with growing user profiles. However, as described above, these coefficients converge over time, and therefore the network of recommenders, in the MovieLens dataset, will tend towards the values shown here.

The PCC distribution has two interesting peaks: one in the range of $(0, 0.05)$, and the other between $(-1.0, -0.95)$. In other words, a relatively high proportion of coefficients fall between the two ranges covered by these points. The constrained-PCC skews the entire distribution toward the positive end; it seems thus that this variation of the PCC will increase the similarity between pairs of users that may otherwise have been deemed minimally similar with the standard PCC. Applying significance weighting to the coefficient changes the distribution drastically, by increasing the frequency of neighbors who have very low correlation. Nearly half of the user pairs are valued within $(0, 0.05)$, which implies that a high proportion of recommendations are weighted extremely lightly.

The Somers' d distributions show that all of the coefficients fall between -0.05 and 0.05. This implies extremely low similarity; in fact, the community seems to be composed of users who would be terrible recommenders for each other. This occurs because the value of a Somers' d coefficient (Equation 2) will depend on the number of rate-able items, N. As the problem of information overload outlines, the total number of available items is far greater than the number of items an average user will have rated. A small number of items would not give rise to the problem, and thus all Somers' d coefficients can be expected to fall within a very short range from 0.

On the other hand, the coefficients based on the proportion of co-rated items peaks at 0, for the number of users who do not share any rated items. The rest of the user-pairs all share a positive similarity. Since this coefficient is derived using the number of co-rated items that the user-pair share, this coefficient can not be negative, and thus a community of recommenders in this scenario will only have positive links. The VS coefficient had the largest number of coefficients within a very high range; 0.78, or nearly 80%, of the community is weighted between 0.9 and 1.0. This is the result of summing the proportion of coefficients between $(0.9, 0.95)$, 0.32, and $(0.95, 1.0)$, 0.46. In other words, VS coefficients will favor neighbor recommendations much higher than, for example, Somers' d coefficients. Much like the concordance-based measures, finding that majority of the population share similar coefficients may imply that the population is full of very similar users, but following this same analysis using the PCC yielded quite opposing results. Once again, we found that the distribution given by each similarity measure does not agree with any of the others. There does not seem to be any unifying behavior or descriptive characteristics, in terms of coefficient distribution, of the dataset, as the method for computing the coefficients are changed.

4. EXPERIMENTAL ANALYSIS

Recommendation system algorithms are often evaluated according to two criteria: how many recommendations they can generate (prediction *coverage*), and how good the generated recommendations are (prediction *accuracy*), although new error measures have been suggested [7]. In this section, we use the MovieLens dataset to conduct an evaluation of these techniques, and first report the accuracy results, followed by a section dedicated to the coverage results. The dataset is available both in its entirety, and divided into five disjoint training/test sets $(u1, u2, ..., u5)$. The training sets are used to build the community (by computing user similarity values), and the test sets are used to measure how well this community can generate predicted ratings. Due to lack of space, we report the full results from the $u1$ subset, showing the influence of an increasing neighborhood size on the results. We also report the results for all subsets when using the entire community as the neighborhood.

4.1 Accuracy

Table 1: MAE Prediction Error, MovieLens U1

Neighborhood	Co-Rated	Somers' d	PCC	Weighted-PCC	R(0.5, 1.0)	R(-1.0,1.0)	Constant(1.0)
1	0.9449	0.9492	1.1150	0.9596	1.0665	1.0341	1.0406
10	0.8498	0.8355	1.0455	0.8277	0.9595	0.9689	0.9495
30	0.7979	0.7931	0.9464	0.7847	0.8903	0.8848	0.9108
50	0.7852	0.7817	0.9007	0.7733	0.8584	0.8498	0.8922
100	0.7759	0.7728	0.8136	0.7647	0.8222	0.8153	0.8511
153	0.7725	0.7727	0.7817	0.7638	0.8053	0.8024	0.8243
229	0.7717	0.7771	0.7716	0.7679	0.7919	0.8058	0.7992
459	0.7718	0.7992	0.8073	0.8025	0.7773	0.7812	0.7769

Table 2: Accuracy of Coefficients

Dataset	Co-Rated	Somers' d	PCC	Weighted-PCC	R(0.5,1.0)	R(-1.0,1.0)	Constant(1.0)
u1	0.7718	0.7992	0.8073	0.8025	0.7773	0.7812	0.7769
u2	0.7559	0.7825	0.7953	0.7903	0.7630	0.7666	0.7628
u3	0.7490	0.7706	0.7801	0.7775	0.7554	0.7563	0.7551
u4	0.7463	0.7666	0.7792	0.7747	0.7534	0.7554	0.7531
u5	0.7501	0.7715	0.7824	0.7784	0.7573	0.7595	0.7573
Average	0.7548	0.7781	0.7889	0.7847	0.7613	0.7638	0.7610

Our first aim was to see how accurate the predictions are for the given coefficients. To do so, we measured the mean absolute error (MAE) of the predicted ratings, *only* in the case when a prediction was made. If no information was available, typical experiments will simply return the user mean, and this value is used when finding the MAE of the predictions. However, we measured the coverage and accuracy separately, in order to see the accuracy of the coefficient when it does result in predicted ratings. Since MAE measures the mean absolute deviation from the actual ratings, and the MovieLens dataset uses a five-point rating scale, the error measures can be expected to fall between 0, the optimal result, and 4.

The initial observations of user-pair coefficient distribution lead to the question: how do similarity measures affect the prediction accuracy of a CF method? In order to explore this question, we compare all previous similarity measures with constant similarity measures (say, all 1.0), and randomly selected measures between user pairs. These measures do not use any information from the dataset to find like-minded peers. We thus expected that the error reported on the prediction set would be devastatingly worse than when any similarity measures were used, as constant/random numbers do not consider how much users have co-rated items or how much these ratings agree with each other.

We experimented with three ranges: $(-1.0, 1.0)$, or randomly assigning relationships so that the distribution of coefficients over the community pairs is uniform over the full similarity scale, $(0.5, 1.0)$, i.e. giving all the user-pairs high similarity relationships, and all 1.0, giving all recommenders perfect correlation. Applying different constant values to the community did not alter the performance accuracy of the method. The reason for this can be seen in Equation 4, which combines many recommendations together. It is a weighted average of deviations from the recommender's mean, and thus if all recommenders are weighted equally (no matter what the weight value), then the results will be the same. Of course, the only exception to this is if all recommenders were weighted 0, a case that we do not consider here as it completely eliminates the effect of the community of recommenders when generating predicted ratings.

A recommender contributes to a predicted rating (again, in Equation 4) through two values: the opinion $(r_{u,i} - \bar{r}_u)$ it offers, and the similarity, $w_{a,u}$, it shares with the node creating the predicted rating. Table 1 shows the results from the $u1$ subset as the number of opinions considered when making a predicted rating is increased. The most accurate results were obtained when predicted ratings were derived using the all of the community member's opinions.

To our surprise, the results of the experiments using random-valued and constant relationships were not only comparable to the performance of the correlation coefficients, but on average they also performed slightly better than the tested similarity measures, as reported in Table 2. This once again points to the fact that when all of the community member's opinions is included, the accuracy of predicted ratings is independent of what similarity value is used.

Such results would be expected if there were a certain degree of homogeneity amongst the community members, regardless of whether the specific correlation values agreed or not. A simple popularity-based recommender, which returns the average rating of an item (using all available ratings of it) also performs within a small range of the correlation coefficients. The average MAE over all data subsets, in this case, is 0.8182, which is 0.04 less than the weighted-PCC's accuracy.

4.2 Coverage

The second part of the evaluation focuses on prediction coverage. Unfortunately, perfect coverage is not a realistic goal for these systems to strive for, due to persistent imperfect information within the datasets. For example, when a new movie is added to the system it does not have any user ratings, and thus there is no available information on that item. Recommendation systems that do not operate on a closed set of items will always include new items that have not been rated by any community member, and as such can not be recommended.

Table 3: Coverage of Coefficients, MovieLens U1

Neighborhood	Co-Rated	Somers' d	PCC	Weighted-PCC	Oracle
1	0.67795	0.57165	0.96725	0.61375	0.00495
10	0.15455	0.0999	0.80515	0.1114	0.00495
30	0.0512	0.0407	0.57225	0.04135	0.00495
50	0.03065	0.0266	0.3641	0.0251	0.00495
100	0.01515	0.01645	0.08345	0.01485	0.00495
153	0.00945	0.0122	0.0273	0.01135	0.00495
229	0.00715	0.00965	0.01165	0.00915	0.00495
459	0.00495	0.0054	0.00495	0.00495	0.00495

Table 4: Coverage of Coefficients

Dataset	Co-Rated	Somers' d	PCC	Weighted-PCC	Oracle
u1	0.00495	0.0054	0.00495	0.00495	0.00495
u2	0.0035	0.0037	0.0035	0.0035	0.00345
u3	0.00205	0.00215	0.00205	0.00205	0.00205
u4	0.00135	0.00145	0.00135	0.00135	0.00135
u5	0.0028	0.00215	0.0018	0.0018	0.0018
Average	0.00293	0.00297	0.00273	0.00273	0.00272

Apart from these unavoidable cases intrinsic in the dataset, coverage decreases as a result of zero-similarity relationship between a pair of users, unless a correlation threshold is used when deciding who to collect opinions from. Visualising the relationships within the community of recommenders as was done in Section 3 also demonstrates why high correlation thresholds will lead to impoverished coverage: the proportion of the community that falls within the "very similar" range (say, greater than 0.5 similarity) is very small for some coefficients.

The question we now want to answer is: what is the impact of different similarity techniques on the proportion of zero-similarity coefficients created? In other words, if we define a *useful relationship* as a link to a recommender who can give you an opinion about an item, how many zero-weighted relationships are discovered which eliminate potentially useful relationships?

To answer this question, we propose a new coverage target, called the coverage *oracle*. This value can be easily derived by counting the number of items in the test set that do not appear in the training set. The coverage that the oracle achieves is equivalent to the coverage of a k-nearest recommenders algorithm, where neighbors are selected on the basis of whether they have rated the active item, instead of being in the top-k most similar users to the active user.

In this section, we only report coverage error measures, found by dividing the number of uncovered predictions by the total number of predictions. In other words, we show the proportion of the dataset that is uncovered, and these error measures will range from 0, perfect coverage, to 1, if no predicted ratings can be made. Just like the accuracy metrics, the aim of collaborative filtering is to minimise this value. The results for MovieLens $u1$ are shown in Table 3, while the average results for the full neighborhood across all the subsets is shown in Table 4.

The results show that none of the coefficients achieve the target coverage until the full community is included, or when the neighborhood size is equal to the number of users. This means that the k-nearest neighbors of any active user will not have all the recommendation information the user requires and therefore similarity alone may not be the best way of connecting recommenders with each other. The optimal coverage value varied for each subset, as did the rate at which the coefficients converged toward the optimal as the neighborhood size increased. An important observation, in the $u1$ results, is that even the similarity measure that seems to carry the least amount of information (the proportion of co-rated items) was able to achieve optimal coverage, although different datasets may provide varying results. When considering the averaged values, not a single coefficient achieved the average oracle coverage.

Although the focus of these accuracy experiments was not on coverage, it is worth noting that using random or constant coefficients also achieved the same prediction coverage, a result that was equal to the oracle value when the neighborhood size, k, was the entire community.

We began this work by focusing on the analogy between collaborative filtering datasets and a network of linked user profiles that behave as a community of recommenders. The principle of using like-minded individuals to generate good recommendations translates to requesting rating information over the links that have the strongest weighting. However, in the experiments, we found that no matter what method of measuring user similarity was imposed on the community, the performance accuracy was no better than when the same network was created with random-valued weightings on each link. There are a number of reasons that may have led to these results.

5. DISCUSSION

These results may be a sign that the dominant error measures used to compare collaborative filtering algorithms are not sufficient. They contain very little information as to how much customers will be *satisfied* with their recommendations, and, moreover, can not be differentiated from the behavior of random coefficients. The need for better error

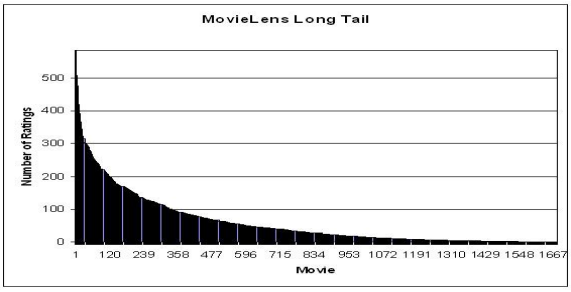

Figure 2: Long-tail of the MovieLens Dataset

measures has been suggested in [7], which further states that accuracy metrics actually hurt the development of recommender systems. However, these considerations do not remove the requirement that there must exist empirical measures, such as MAE and coverage, to compare the performance of different algorithms on datasets.

The datasets themselves may be to blame for the results. The MovieLens dataset we used does comply with the "long-tailed" characteristic which lead to the initial work on collaborative filtering; the number of movies that have been rated, say, between 0 and 30 times, is much greater than the number of movies rated between 90 and 120 times, as can be seen in Figure 2. However, it is relatively straightforward to show that there are many items that appeal to a small number of users, rather than a small number of popular items that appeal to the entire community. Popularity-based recommendations should thus not provide useful results. However, as discussed in Section 4.1, user correlation does not drastically outperform a simple popularity-based recommender on this dataset.

The dataset, along with the results, provides poor evidence to support the heuristic that item predictions should be weighted according to user similarity, rather than, for example, item-item similarity [8]. It will be interesting to see if such results persist over a wide range of datasets.

Rather than blaming the dataset, the results may highlight the fact that the current similarity measures are not strong enough to optimally represent a community of recommenders. The most accurate results are achieved when the size of the neighborhood, k, begins to approach the size of the entire community. Again, this seems to contradict the fact that like-minded individuals provide the best recommendations to each other, assuming that the community is not entirely constructed of like-minded individuals.

Other recent suggestions aim resolving these problems by incorporating multi-criteria ratings into recommender systems [9]. Rather than leaving it to the user to create a single rating for an item, these methods ask for multiple ratings, according to different characteristics. For example, a movie could be rated according to its story, acting, special effects, etc. Regardless of the difficulty of getting users to contribute these kinds of ratings, the main idea highlights an important part of rating profiles: we do not understand how users compose their judgment of the varying attributes of an item into a single rating, or how users' usage of rating schemes will vary between each other. The fact that an abstract human behavior is behind the generation of the datasets we apply collaborative filtering techniques to may also explain why such inconsistent results have been found. A deeper understanding of how these ratings are applied will shed light on how to translate them into useful and trustworthy recommendations for others.

We believe that there is more to user-similarity than a mere comparison of profile history can disclose, a topic we plan on addressing in future work. For example, a user-pair with unmeasurable similarity may still be able to exchange useful recommendation information. While current collaborative filtering techniques require co-rated items to compute a similarity value, we plan on applying the perspective that has, to date, been characteristic of trust management systems: a user expresses levels of *uncertainty* in other users, and updates these values by evaluating the individual experiences it has with them. In a peer-to-peer scenario, where we do not have the full user-rating matrix available for analysis, an appropriate solution may lie in incremental learning of user-relationships, by weight adjustment. This is equivalent to expressing a trust value in a recommender based on the historical opinions that have been received, rather than expressing similarity and trust as two distinct values, a topic that has been previously explored [10].

6. REFERENCES

[1] J.B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *Proceedings of the ACM Conference on Electronic Commerce*, 1999.

[2] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.

[3] N. Lathia, S. Hailes, and L. Capra. Private distributed collaborative filtering using estimated concordance measures. In *Proceedings of Recommender Systems (RecSys)*, 2007.

[4] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. In *ACM Transactions on Information Systems*, volume 22, pages 5–53. ACM Press, 2004.

[5] A. Agresti. *Analysis of Ordinal Categorical Data*. John Wiley and Sons, 1984.

[6] R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–336, 2004.

[7] S.M. McNee, J. Riedl, and J.A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems*. ACM Press, 2006.

[8] G. Linden, B. Smith, and Y. York. Amazon.com recommendations: Item-to-item collaborative filtering. In *IEEE Internet Computing*, pages 76–80, 2003.

[9] G. Adomavicius and Y. Kwon. New recommendation techniques for multicriteria rating systems. In *IEE Intelligent Systems*, pages 48–55, May 2007.

[10] P. Massa and B. Bhattacharjee. Using trust in recommender systems: An experimental analysis. In *iTrust International Conference*, 2004.