Exploring more Realistic Evaluation Measures for Collaborative Filtering

Giuseppe Carenini, Rita Sharma
Computer Science Dept. University of British Columbia
2366 Main Mall, Vancouver, B.C. Canada V6T 1Z4
carenini,rsharma@cs.ubc.ca

Abstract

Collaborative filtering is a popular technique for recommending items to people. Several methods for collaborative filtering have been proposed in the literature and the quality of their predictions compared in empirical studies. In this paper, we argue that the measures of quality used in these studies are based on rather simple assumptions. We propose and apply additional measures for comparing the effectiveness of collaborative filtering methods which are grounded in decision-theory.

[**keywords:** information agents, human-computer interaction, recommender systems, evaluation]

Introduction

Recommenders are computer systems designed to help people find preferred items within a very large set of available alternatives, such as movies, books and news. Collaborative filtering (CF) is a technique used by recommenders to make predictions on how a user will rate a particular item, given his or her ratings on other items and given other people's ratings on various items including the one in question. CF has been an active area of research in recent years both in AI and HCI. Several collaborative filtering algorithms have been suggested, ranging from binary to non-binary ratings, and from implicit to explicit ratings (Resnick *et al.* 1994; Shardanand & Maes 1995),(Breese, Heckerman, & Kadie 1998), (Pennock *et al.* 2000), (Sharma & Poole 2001). Of all these algorithms, the most recent and most successful ones are based on a probabilistic approach.

(Breese, Heckerman, & Kadie 1998) presented the first attempt to perform a comprehensive evaluation of all the algorithms that had appeared in the literature up to that time. They compared each algorithm's performance according to two measures of prediction accuracy that have since become a standard for evaluating CF algorithms: the Mean Absolute Error (MAE) and the ranked scoring (RS). MAE has become especially popular. It has been used to evaluate new probabilistic CF algorithms (Sharma & Poole 2001), (Pennock *et al.* 2000) as well as techniques for eliciting ratings in CF (Rashid *et al.* 2002). In this paper, we examine the tacit assumptions underlying MAE and RS. MAE equates recommendation quality with the accuracy of the algorithm prediction of how a user will rate a particular item. The problem with this approach is that it makes an unrealistic assumption on how recommenders are used in practice. Users of a recommender are not prediction checkers. Rather, they use predictions to make decisions. So, a measure of recommendation quality should take into account the outcomes of the users' decisions. To address this problem, we propose a new measure for comparing CF algorithms which is grounded in decision-theory. Our new measure attempts to compute how useful a recommendation will be for the user by taking into account a user-specific *threshold* for accepting a recommendation. Also, we note that our new measure presents interesting similarity with RS (the other performance measure commonly used in CF). Based on this observation, we uncover tacit assumptions underlying RS and we conclude that our new measure can be effectively used to define a more plausible RS.

CF and its Evaluation

Filtering Problem and Notation

Let N be the number of users and M be the total number of items in the database. S is an $N \times M$ matrix of all user's ratings for all items; S_{ui} is the rating given by user u to item i. Ratings are typically on a cardinal scale with m values (e.g., $0, \ldots, 5$ in the movie domain). In collaborative filtering, S, the user-item matrix is generally very sparse since each user will only have rated a small percentage of the total number of items. Under this formulation, the collaborative filtering problem becomes predicting those S_{ui} which are not defined in S, the user-item matrix. The user for whom we are predicting a rating is called the active user a.

Algorithms for CF

Initial CF algorithms (Resnick *et al.* 1994; Shardanand & Maes 1995), referred in (Breese, Heckerman, & Kadie

1998) as memory based, predict the active user ratings as a *similarity-weighted sum* of the other users' ratings, with the most successful similarity measure being *correlation* (Breese, Heckerman, & Kadie 1998). Recently, AI researchers have investigated model based algorithms for CF. Model based algorithms build a model of the data and use the model for prediction. Breese et al. (Breese, Heckerman, & Kadie 1998) proposed and evaluated two probabilistic models: *cluster models* and *Bayesian networks*. Pennock et al. (Pennock *et al.* 2000) introduced another probabilistic model based CF algorithm called *personality diagnosis* (PD). And finally, (Sharma & Poole 2001) proposed a probabilistic approach based on a *noisy sensor model*.

Evaluation Strategy and Measures

Typically, a machine learning *training and test set* approach is used to evaluate the accuracy of CF algorithms. In this approach, the dataset of users and their ratings is divided into two: a *training set* and a *test set*. The *training set* is used as the CF dataset. The *test set* is used to evaluate the accuracy of the CF algorithm. When testing is performed, each user from the test set is treated as the *active* user. The ratings by each test user are divided into two sets: I_a and P_a. The set I_a contains ratings that are treated as observed ratings. The set P_a contains the ratings that the CF algorithm attempts to predict by using both the observed ratings (I_a) and the training set.

The Mean Absolute Error (MAE) is the most commonly used measure to evaluate the accuracy of CF algorithms. Let the number of predicted ratings in the test set for the active user be n_a; then the MAE for that user is given as follows:

$$MAE_a = \frac{1}{n_a} \sum_{j \in P_a} |o_{a,j} - p_{a,j}|$$

where $o_{a,j}$ is user a's observed rating for item j and $p_{a,j}$ is user a's predicted rating for item j. The MAE reported in CF evaluations is the average MAE for all users in the test set. Notice that the lower the MAE, the more accurate the collaborative filtering algorithm is.

A second, less commonly used measure of recommendation quality is the ranked scoring (RS). This measure assumes that the recommender presents a recommendation to the user as a list of items ranked by their predicted ratings. As described in (Breese, Heckerman, & Kadie 1998), RS assesses the expected utility of a ranked list of items, by multiplying the utility of an item for the user by the probability that the item will be viewed by the user. The utility of an item is computed as the difference between its observed rating and the default or neutral rating d in the domain (which, they suggest, can be either the midpoint of the rating scale or the average rating in the dataset), while the probability of viewing decays exponentially as the rank of items increases. Formally, the RS of a ranked list of items sorted according to the index j in order of declining $p_{a,j}$[1] is:

$$RS_a = \sum_j max(o_{a,j} - d, 0) * \frac{1}{2^{(j-1)/(\alpha-1)}}$$

The first term computes the utility of the item in the j^{th} position of the list, whereas the second term computes the probability of the user viewing that item (see (Breese, Heckerman, & Kadie 1998) for details on this second term). The RS reported in CF evaluations combines the RS_as for all the users in the test set:

$$RS = 100 \sum_a \frac{RS_a}{RS_a^{max}}$$

where RS_a^{max} is the utility of the best possible ranked list for user a (i.e., the utility of the ranked list that the recommender would generate if it knew all the $o_{a,j}$). Notice that the higher the RS, the more accurate the collaborative filtering algorithm is.

Table 1: MAE scores on the EachMovie data for *Noisy-Best* (Noisy sensor model), *PD* (Personality Diagnosis) and *Correl* (Correlation) - Note: lower scores are better.

Algorithm	*Protocol*			
	AllBut1	*Given10*	*Given5*	*Given2*
Noisy-Best	**.893**	**.943**	**.974**	**1.012**
PD	.964	.986	1.016	1.039
Correl	.999	1.069	1.145	1.296

Evaluation results using MAE and RS on the Eachmovie database[2]

Table 1 shows the results of a recent comparison of CF algorithms based on MAE (Sharma & Poole 2001)[3]. The algorithms were tested on the same subset of the EachMovie database as used by (Breese, Heckerman, & Kadie 1998) and (Pennock *et al.* 2000), consisting of 1,623 movies, 381,862 ratings, 5,000 users in the training set, and 4,119 users in the test set. Ratings in the EachMovie database are elicited on an integer scale from zero to five. In the table, *AllBut1* means that the test set P_a for each test user contains a single randomly selected rating and the observed set I_a contains the rest of the ratings. In contrast, *GivenX* means that X ratings are randomly placed for each test user in the observed set I_a, while the rest of the ratings are placed in the test set P_a. As shown in the

[1] (Breese, Heckerman, & Kadie 1998) says that items are sorted in order of declining $o_{a,j}$, but we believe that was a typo because the recommender cannot present a list ordered by the true ratings.

[2] http://research.compaq.com/SRC/eachmovie/

[3] (Sharma & Poole 2001) proposed two algorithms based on a noisy sensor model. In the table, we report only the performance of the best one, that we call Noisy-Best.

Table 2: RS scores on the EachMovie data when d is equal to 2.5 (the midpoint of the 0-5 rating scale) and 3.04 (the average rating of the dataset) - Note: higher scores are better.

Algorithm	*d*	*Protocol*			
		AllBut1	*Given10*	*Given5*	*Given2*
Noisy-Best	*2.5*	**77.42**	**76.82**	**76.80**	**74.09**
PD	*2.5*	69.58	69.20	69.08	**75.72**
Correl	*2.5*	**76.81**	71.17	67.86	59.70
Noisy-Best	*3.04*	**72.49**	**71.91**	**71.90**	**68.74**
PD	*3.04*	63.18	62.78	62.71	**70.66**
Correl	*3.04*	**71.89**	65.53	61.88	52.94

table, *Noisy-Best* performed better than *PD* and *Correlation* in all these settings. And since *Correlation* had been shown to either perform similarly or outperform (with respect to MAE) all other CF methods studied in (Breese, Heckerman, & Kadie 1998) (e.g., Bayesian Networks), it appears that *Noisy-Best* is the best performing CF algorithm according to MAE (when tested on the EachMovie database).

We also compared CF algorithms with respect to the RS measure. Table 2 shows the results of this comparison. *Noisy-Best* performed similarly or outperformed *PD* and *Correlation* in most settings (more later on statistical significance). Notice that these are new results, because neither (Sharma & Poole 2001) nor (Pennock *et al.* 2000) considered RS. As we did for MAE, the algorithms were tested on the same dataset and following the same procedure used by (Breese, Heckerman, & Kadie 1998). However, for *Correlation* (the only algorithm that both we and (Breese, Heckerman, & Kadie 1998) investigated) we obtained rather different results (our RS for *Correlation d=2.5* is in the 67-77 interval while RS for the same protocol in (Breese, Heckerman, & Kadie 1998) is in the 23-43 interval). We are not quite sure of the source of the discrepancy. Randomization cannot be the culprit because the differences are quite large. Therefore, this issue requires further investigation. Nevertheless, since in (Breese, Heckerman, & Kadie 1998) *Correlation* had been shown to either perform similarly or outperform all other CF methods studied with respect to RS, it appears that *Noisy-Best* is overall the best performing CF algorithm also according to RS (when tested on the EachMovie database).

Tables 1 and 2 also summarize the statistical significance of the results. In each protocol, the score(s) of the algorithm(s) that significantly outperformed the other algorithms is(are) in bold. In Table 1 statistical significance was based on randomized paired sample tests ($p < .05$) (Sharma & Poole 2001), while in Table 2 it was based on ANOVA with the Bonferroni procedure at 90% confidence level (as in (Breese, Heckerman, & Kadie 1998)).

Exploring New Evaluation Measures

Although MAE has become a standard metric for comparing CF algorithms, we argue that it is based on an unrealistic assumption about what it means for a recommendation to be effective. MAE relies on viewing the recommendation process as a machine learning problem, in which recommendation quality is equated with the accuracy of the algorithm's prediction of how a user will rate a particular item. This perspective is missing a key aspect of the recommendation process. The user of a recommender system is engaged in deciding whether or not to experience an item (e.g., whether or not to watch a movie). So, the value of a recommendation critically depends on how the recommendation will impact the user decision making process and only indirectly on the accuracy of the recommendation. For illustration, consider a user who will watch only movies whose predicted rating $p_{a,j}$ is greater than 3.5. Now, consider the following two predictions for that user:

(i) $p_{a,j} = 0$; when $o_{a,j} = 2$; (Absolute Error = 2)

(ii) $p_{a,j} = 3$; when $o_{a,j} = 4$; (Absolute Error = 1)

The Absolute Error in *(i)* is greater than the one in *(ii)*. However, in terms of user decision quality, *(i)* leads to a good decision because it entails that the user will avoid watching a movie that she would not have liked, while *(ii)* leads to a poor decision, because it entails that the user will miss a movie that she would have enjoyed.

The key point of this example is that in measuring the quality of a recommendation we should take into account the criterion used by the user in deciding whether or not to accept the recommendation. When the recommendation is provided as a predicted rating, a simple plausible criterion is a user specific threshold in the range of possible ratings. The user will accept a recommendation when the prediction is greater than the threshold.

More formally, let θ_a be a user specific threshold (in the range of possible ratings) such that:
$p_{a,j} \geq \theta_a \Rightarrow select(a, j)$

where, as before, $p_{a,j}$ is user a's predicted rating for item j, and $select(a, j)$ means that user a will select item j (e.g., the user will watch the movie).

Then, the quality of a recommendation $p_{a,j}$, which we call User Gain (UG), can be defined as:

$$UG(p_{a,j}) = \begin{cases} o_{a,j} - \theta_a & \text{if } p_{a,j} \geq \theta_a \\ \theta_a - o_{a,j} & \text{otherwise} \end{cases}$$

where, as previously defined, $o_{a,j}$ is user a's observed rating for item j.

The first condition covers the situation in which, since the prediction is greater than the threshold, the user will decide to experience the item and will enjoy it to the extent that its true rating is greater than the threshold. The second condition covers the situation in which the user will decide not to experience the item. In this case, the user will gain to the extent that the item's true rating is smaller than the threshold.

Similarly to MAE_a, we can also define the active user Mean User Gain (MUG_a) as:

$$MUG_a = \frac{1}{n_a} \sum_{j \in P_a} UG(p_{a,j})$$

and MUG as the average MUG_a for the test set

A comparison between the UG measure and the RS_a measure (see section on CF evaluation strategy and measures) reveals that the two measures are interestingly related. And, we argue, UG can be effectively used to improve RS_a. In RS_a, the utility of an item is computed as $max(o_{a,j} - d, 0)$. If we substitute in this expression d (the default rating) with θ_a (the user's threshold) and we admit negative utilities (by removing the max operator), the utility computed in RS_a is the same as the one computed in the first condition of UG (i.e., when $p_{a,j} \geq \theta_a$). Now let's examine these transformations and their effect in order.

- θ_a is just a more adequate version of d. While d is a neutral rating for the dataset [4], θ_a is a neutral rating for the active user decision strategy.
- If the user experiences an item whose true rating is below θ_a she will be disappointed. So, it is sensible to admit negative utilities.

Now, why is RS_a covering only the first condition of UG? We believe the underlying assumption is that, since the ranked list is presenting the items with the highest predicted rating in the dataset, for all these items $p_{a,j}$ is assumed to be greater than θ_a. However, when that is not necessarily the case a more comprehensive definition of RS_a would be:

$$RS_a^{UG} = \sum_j UG(p_{a,j}) * \frac{1}{2^{(j-1)/(\alpha-1)}}$$

In order to evaluate a CF algorithm, the RS_a^{UG}s for all the users in the test set need to be combined. Unfortunately, we cannot apply the same formula we used in RS to combine all the RS_a because RS_a^{UG} can be negative and this would lead to unintuitive results. A preliminary more intuitive way to combine the RS_a^{UG}s is simply to compute their average. We call this summary measure RS^{UG}. Notice that the higher the RS^{UG}, the more accurate the collaborative filtering algorithm is.

As a final note, we discuss how previous attempts to use MAE by taking into account the user decision process did not really address the core of the problem. Shardanand and Maes (Shardanand & Maes 1995) and Pennock et al. (Pennock *et al.* 2000) proposed that the accuracy of a CF algorithms should be evaluated only on items whose observed ratings are extreme (very high or very low). The supposition is that, most of the time, people are interested in suggestions about items they might like or dislike, but not about items they are unsure of. Pennock et al. (Pennock *et al.* 2000) defined the extreme ratings as those which are 0.5 above and 0.5 below the average rating in the dataset. In light of our analysis of MAE, this proposal is clearly problematic. Since intuitively θ_as should correspond to non-extreme ratings (more on this in the following section), errors on items with non-extreme ratings are more likely to lead the user to make poor decisions (e.g., watching movies she will not enjoy). So, if you wanted to use MAE trying to take into account the user decision process, you should focus on non-extreme ratings, rather than on extreme ones.

[4] Typically, d is either the midpoint of the rating range or the average of all ratings in the dataset (2.5 and 3.04 respectively in the Eachmovie database). So, d may not have any relation with any user decision criterion.

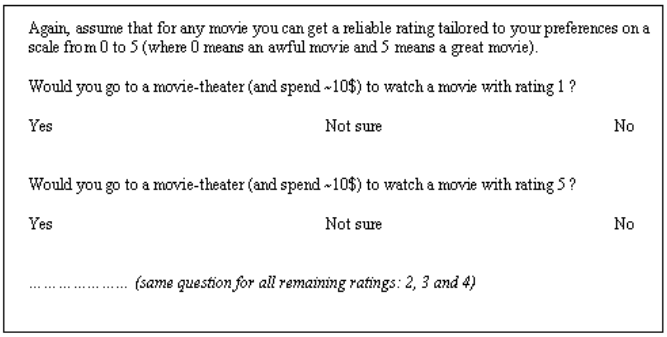

Figure 1: Portion of the questionnaire to assess θ_a

Elicitation Study

The MUG_a measure relies on a user specific threshold θ_a, which determines whether the user will choose to experience an item given its predicted rating. Thus, in order to apply UG in any domain, you need either to elicit θ_a from all users, or to elicit θ_a from a sample of users and then use the sample to compute a $\hat{\theta}$ that can be used as a default θ_a for all users [5]. To obtain a reasonable $\hat{\theta}$ for the movie domain, we have performed a user study in which participants filled out a questionnaire about their preferences and decision strategies in that domain.

The questionnaire was first checked with a few pilot participants and then was administered to 27 participants (students and faculties at UBC). Of these 27, 5 were eliminated from the study because of procedural faults. The questionnaire consists of two parts: one to elicit the user specific θ_a; the other to elicit the user utility function (in the decision-theoretic sense) for movie ratings. The reason for assessing the user's utility function was to obtain further independent support for the estimate of $\hat{\theta}$.

Figure 1 shows the part of the questionnaire that assesses θ_a. We left the definition of the ratings as general as possible *"..where 0 means an awful movie and*

[5] $\hat{\theta}$ should not be confused with the d in RS. $\hat{\theta}$ is based on a user decision criterion, while d only depends on the dataset.

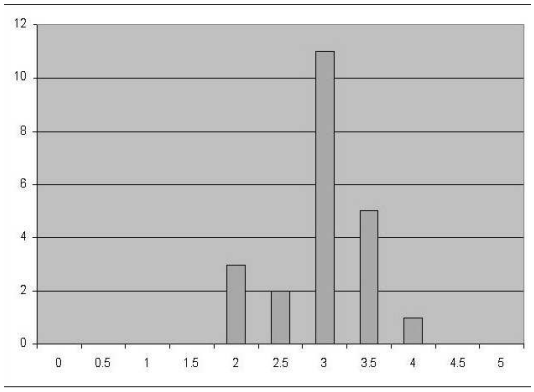

Figure 2: Counts for values of θ_a.

5 means a great movie" to preserve as much as possible the generality of the assessment. In interpreting the participant answers, we applied the following schema. If the participant marks "not sure" for a certain rating, θ_a is assigned that rating. Whereas, if the participant does not mark any rating as "not sure", then θ_a is assigned the midpoint between the highest rating for which the participant answered "no" and the lowest rating for which the participant answered "yes". The results of the analysis of this portion of the questionnaire are shown in Figure 2. The mean of our sample for θ_a is 2.98. So, we assume 2.98 as a reasonable estimate for $\hat{\theta}$ in the movie domain.

A second portion of the questionnaire elicited the participants' utility function for movie ratings. This was based on the classic probability-equivalent (PE) procedure.

The outcomes of the utility elicitation process are summarized in Figure 3. The figure compares a linear utility function with what we call the "average" utility function for our participants. This function is computed by averaging for each rating the utility of all participants.

When aggregated in this way, the users' utility functions provide independent evidence on the position of $\hat{\theta}$ on the rating scale. Several studies in behavioral decision theory have shown that people are $risk-adverse$ (their utility function is concave) when they believe they are gaining, while they are $risk-seeking$ (their utility function is convex) when they believe they are losing (Clemen 1996). Figure 3 indicates that people on average believe they are gaining when they watch a movie whose rating is greater than 3 (the average utility function is concave) and believe they are losing when they watch a movie whose rating is less than 2 (the average utility function is convex). Thus, a reasonable $\hat{\theta}$ must lie in that interval.

Applying MUG and RS^{UG}

We have applied MUG and RS^{UG} in comparing the same three CF algorithms that were compared with

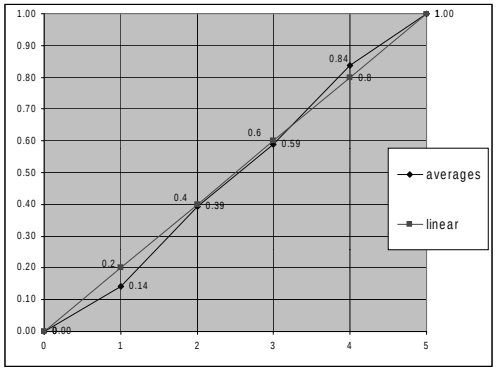

Figure 3: A comparison between a linear utility function and the one computed by averaging for each rating the utility of all participants.

Table 3: MUG scores on the EachMovie data. Note: scores are expressed on the $[-max(\hat{\theta}, 5-\hat{\theta}), +max(\hat{\theta}, 5-\hat{\theta})]$ rating interval ([-2.98,+2.98] in our experiments) and higher scores are better.

Algorithm	*Protocol*			
	AllBut1	*Given10*	*Given5*	*Given2*
Noisy-Best	0.71	**0.70**	**0.62**	**0.52**
PD	0.58	0.54	0.52	0.09
Correl	**0.78**	0.60	0.52	0.37

MAE in (Sharma & Poole 2001) and that we compared with RS in this paper. As it was the case for (Sharma & Poole 2001), we run our testing on the EachMovie database. This database specifies the user-item matrix, but does not provide any information on user specific θ_as. Therefore, we used the default estimates we obtain from our elicitation study. $\hat{\theta} = 2.98$ was used in all the experiments.

Results for MUG and RS^{UG} are shown in Table 3 and Table 4 respectively. In all these tables, as we did before, in each protocol, the score(s) of the algorithm(s) that significantly outperformed the other algorithms are in bold. Statistical significance was based on ANOVA with the Bonferroni procedure at 90% confidence level (as in (Breese, Heckerman, & Kadie 1998)).

Let us start by examining the results for MUG in Table 3. *Noisy-Best* is the best performer in all protocols except for *AllBut1*, in which *Correlation* is the best. When these results are compared with what was obtained with MAE (see Table 1), the most noticeable difference is that in the *AllBut1* protocol *Correlation* is the winner for MUG, whereas it was the loser for MAE. The explanation for this (which we verified empirically) is that in the *AllBut1* protocol *Correlation* is the most precise algorithm in predicting movies whose observed

Table 4: RS^{UG} scores on the EachMovie data. Note: the maximum value for this measure in our experiments was 1.72 (which is the average of the utilities of the best possible ranked list for all users in the test set) - higher scores are better.

Algorithm	*Protocol*			
	AllBut1	*Given10*	*Given5*	*Given2*
Noisy-Best	**1.04**	**0.96**	**0.95**	**0.83**
PD	0.77	0.73	0.71	0.35
Correl	**1.04**	0.75	0.61	0.43

value is close to $\hat{\theta}$ (i.e., observed-value= 3). And UG, by definition, is more sensitive to errors in that proximity.

The results for RS^{UG} mirror the ones that were obtained for RS. This is not surprising given that we had to use $\hat{\theta}$ instead of user-specific θ_as and given how close $\hat{\theta}$ was to the neutral rating of the dataset [6]. In detail, *Noisy-Best* is the best performer for RS^{UG} in all protocols with *Correlation* performing at the same level in the *AllBut1* protocol. A plausible explanation for *Correlation* performing less well in RS^{UG} than in MUG is the following. In RS-measures items are ranked with respect to their predicted value. This ranking is then used to weight the utility gain of each prediction, with more weight given to higher ranked items. Since *Correlation* is more precise only in predicting movies whose observed value is close to $\hat{\theta}$, it will be more precise only in predicting items that will tend to be ranked in the middle of the ranking and therefore will tend to be modestly weighted. As a result, the advantage of *Correlation* in MUG plays a more limited role in RS^{UG}.

In general, we claim the outcomes of our evaluation using MUG and RS^{UG} are more valuable than the ones presented in previous studies using MAE and RS, because MUG and RS^{UG} are arguably more realistic measures of recommendation quality than MAE and RS.

Conclusions and Future Work

Performance measures for testing algorithms should be based on realistic assumptions on how the output of those algorithms is used in practice. Algorithms for CF predict how a user will rate a particular item and these predictions are used by users to decide whether or not to select that item. Thus, performance measures for CF should take into account how predicted ratings influence the user's decision.

We noted that MAE, the most popular performance measure for CF, is inadequate in this respect, because it focuses exclusively on the accuracy of the predictions, without consideration for their influence on the user decision. To overcome this limitation, we have proposed the new measure of performance MUG that tries to quantify how much a user is going to gain from decisions based on CF predictions.

[6]But this may not be the case for other datasets or domains

We have shown that UG (the basic component of MUG) presents interesting similarity with RS, another performance measure commonly used in CF, and we have described how UG can be used to define a more plausible RS.

We have also presented the results of a user study in which we elicited from actual users their decision thresholds and their utility functions in the movie domain. This information was then used in performing a comparative evaluation of CF algorithms with respect to our new measures of recommendation quality. The main outcome of this evaluation is that while according to MAE *Noisy-Best* is the top performer in all protocols, according to MUG *Correlation* wins in the *AllBut1* protocol. As for RS^{UG}, the results mirror the ones obtained for RS. However, this may change if user-specific utility thresholds are used.

We envision at least two future extensions for our research. First, we plan to collect a new dataset that includes not only the user/rating matrix but also user specific θ_as. On this dataset, it will be possible to evaluate CF algorithms with MUG without having to rely on a default threshold. A second direction, we intend to pursue, consists of applying our new measures to both other CF algorithms (e.g., Bayesian Networks) and other datasets (e.g., the Nielsen TV viewing dataset) in order to verify the generality of our findings.

References

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical ananlysis of predictive algorithms for collaborative filtering. In *Proc. of the 14th Conf. on Uncertainity in AI*, 43–52.

Clemen, R. T. 1996. *Making Hard Decisions*. Belmont CA: Duxbury Press, 2nd edition edition.

Pennock, D. M.; Horvitz, E.; Lawrence, S.; and Giles, C. L. 2000. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In *Proc. of the 16th Conf. on Uncertainty in AI*, 473–480.

Rashid, A.; Albert, I.; Cosley, D.; Lam, S.; McNee, S.; Konstan, J.; and Riedl, J. 2002. Getting to know you: learning new user preferences in recommender systems. In *Proc. of the Int. Conf. on Intelligent User Interfaces*, 127–134.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; and Riedl, J. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of Conf. on Computer Supported Cooperative Work*, 175–186.

Schafer, J. B.; Konstan, J.; and Riedl, J. 1999. Recommender system in e-commerce. In *Proc. of the ACM Conf. on E-Commerce (EC-99)*, 158–166.

Shardanand, U., and Maes, P. 1995. Social information filtering: Algorithms for automating "word of mouth". In *Proc. of ACM CHI'95 Conf. on Human Factors in Computing Systems*, 210–217.

Sharma, R., and Poole, D. 2001. Symmetric collaborative filtering using the noisy sensor model. In *Proc. of the 17th Conf. on Uncertainty in AI*, 488–495.