# A Single Strong Disagreement Ruins a Recommender: Improving Recommendation Accuracy with a Simple Statistic

**Jennifer Golbeck**
Human-Computer Interaction Lab
University of Maryland, College Park, MD
jgolbeck@umd.edu

## ABSTRACT

Research on the use of social trust relationships for collaborative filtering has shown that trust-based recommendations can outperform traditional methods in certain cases. This, in turn, lead to insights that tie trust to certain more subtle types of similarity between users which is not captured in the overall similarity measures normally used for making recommendations. In this study, we investigate the use these trust-inspired nuanced similarity measures directly for making recommendations. After describing previous research that identified these similarity statistics, we present an experiment run on two data sets: FilmTrust and MovieLens. Our results show that using a simple measure - the single largest difference between users - as a weight produces significantly more accurate results than a traditional collaborative filtering algorithm and in some cases also outperforms a model-based approach.

### Author Keywords

recommender systems, collaborative filtering, profile similarity, trust

### ACM Classification Keywords

H.3.4 Information Storage and Retrieval: Systems and Software - *Performance Evaluation (efficiency and effectiveness)*

## INTRODUCTION

Recommender systems rely on computing similarity, be it between people or items, to make recommendations. In this research, we take results from the literature on computing with social trust, and attempt to improve the quality of recommendations by using more nuanced similarity measures. We show particularly that the largest difference between users and a user's rating habits can improve the accuracy of predictive ratings.


*CHI 2009*, April 4 - 9, 2009, Boston, Massachusetts, USA.


**Table 1. Example movie ratings on a 1-5 scale from three hypothetical users**

| | | Allison | Ben | Catherine |
|---|---|---|---|---|
| 1 | Wizard of Oz | 5 | 1 | 5 |
| 2 | Gigli | 1 | 5 | 1 |
| 3 | Star Wars | 4 | 4 | 2 |
| 4 | Vertigo | 4 | 4 | 5 |
| 5 | High Noon | 3 | 3 | 4 |
| 6 | Over the Hedge | 4 | 4 | 2 |
| 7 | Goodfellas | 2 | 2 | 4 |
| 8 | Forest Gump | 3 | 3 | 5 |
| 9 | Clockwork Orange | 2 | 2 | 3 |
| 10 | Singin' in the Rain | 4 | 4 | 3 |

Over the past 5 years, trust derived from social networks has received much attention as a method for computing recommendations [9, 19, 21, 34]. Several experiments have shown how trust can be used for this purpose and cases where it outperforms more traditional collaborative filtering algorithms.

This previous work has addressed the relationship between trust and similarity. Certainly, we expect that if Allison highly trusts Ben, Ben is likely to be more similar to Allison than someone she does not trust. This has been confirmed in experiments [32, 33].

More recent work has shown that trust captures something more nuanced than similarity alone [1].

For example, consider the situation where Allison has rated a set of movies, and we choose a subset of ten that includes Allison's favorite movie, least favorite movie, and eight other films that Allison has seen and rated but about which Allison has no strong opinion. Now if Ben and Catherine rate those ten films, Allison can make a judgment from this about how much she trusts each of them about movies.

Consider the ratings in table 1. Ben gives the lowest possible rating to User Allison's favorite movie and the highest rating to Allison's least favorite movie, but they agree perfectly on the other eight. In this situation, similarity is very high. Contrast this with Catherine's

ratings. She agrees perfectly on Allison's favorite and least favorite films, but there is variation in their ratings of the other eight movies such that overall, their similarity is lower than in the first case. Who should Allison trust more about movies? Previous research has shown that Allison tends to trust Ben more since they agree on the favorite and least favorite movies [1].

This inspired a study we completed as previous work [1] that empirically showed that trust is related to the following factors between users:

- Overall similarity, as is used in traditional user-user recommender systems;
- Similarity on items to which the user has given an extreme rating, as in the example above. In this example, Allison has two movies with extreme ratings. Agreeing with her on those leads to higher trust;
- The largest difference between the users. This involves finding the one item where users have the largest disagreement. For example, in table 1, the largest difference between Allison and Ben is 4 on the Wizard of Oz and Gigli; the largest difference between Allison and Catherine is 2 on Star Wars and several other movies). The larger this difference is, the lower the resulting trust is;
- The individual's propensity to trust. Some users are more trusting than others.

This leads to the following question, which is the hypothesis of this paper: if trust can be used effectively to make recommendations, and we have identified a set of nuanced similarity measures that reflect trust, can we use those measures directly to make accurate recommendations? Essentially, we are replacing a social expression of trust with an approximation drawn from similarity on the underlying data.

To test this, we draw on the results of our previous work to compute these similarity measures between all pairs of users in two data sets - FilmTrust and MovieLens - and then compute predictive recommendations which are compared to the users' known ratings. We show in both data sets, the very simple statistic measuring the single largest difference between users outperforms correlation-based collaborative filtering techniques, and may also perform better than a well known model-based approach.

We begin by presenting a summary of the results of our previous study that identified the nuanced similarity measures that relate to trust. Then, we describe the experimental methodology and data sets, and present our results. Because it is somewhat surprising that such a simple measure performs so well, we analyze the differences in performance between the maximum difference and standard correlation measures to gain insights into why it performs better. Finally, we conclude with a discussion of how these measures can be incorporated into working recommender systems and the future work required.

## BACKGROUND

With the explosion of social networking websites, a wealth of publicly available information about people's relationships has become available. This has, in turn, led to the development of methods for estimating relationships and using them to improve the functionality of applications. Social trust has been of particular interest to the research community, and the most common application using trust is recommender systems [9, 21, 19, 31]. These results have shown that trust can provide significant benefits over traditional user-based collaborative filtering algorithms in certain cases. Those results suggest that trust captures something more than similarity alone. In this section, I describe the results of a previous study we conducted that identified several nuanced similarity measures that can be used to estimate social trust relationships. These results led directly to the hypothesis for the new experiments conducted in this article, where the nuanced similarity measures are used directly to make recommendations.

We conducted an experiment to delve further into the question of how trust and similarity relate, and the results are reported in [1]. It took place in the context of a movie rating website. First, subjects were asked to rate all the movies they had seen on a list of nearly 300 diverse films. The set was composed of the top 100 movies from the Internet Movie Database [1] top grossing, top rated, and worst rated films, as well as top 10 films from each genre.

Those ratings were then used to generate profiles of hypothetical users. Each profile consisted of ten movies where the hypothetical user differed from the subject in controlled ways. In particular, we tested the impact of differences on movies the subject had given extremely high or low ratings (values of 1, 2, 9, or 10 on a 1-10 scale), ratings more than two standard deviations from the average, and movies that fit both categories. In a profile, movies 1 - 4 were chosen from one category, and movies 5-10 came from the complementary category. So, for example, a profile would contain four movies that the subject had rated in the extreme and six movies that had non-extreme ratings. Another example profile could have four movies where the subject's ratings were were within two standard deviations of the mean and six movies where the subject's ratings were outside that range.

In the main part of this experiment, a predefined set of differences were applied to generate the hypothetical user's ratings. These created profiles that were different from the user in small, medium, and large amounts.

The users were asked to rate how much they trusted this hypothetical user based on the ratings. This method of generating ratings controlled for all factors - overall

[1] http://imdb.com

agreement, standard deviation, etc. Thus, if the movies were reordered,we could conclude that any difference in the trust rating must be due only to the size of the difference on specific movies.

Fifty-nine subjects participated in this study. They ranged in age from 20 to 52, with an average age of 32 (standard deviation of 8.5 years). On average, subjects reported watching movies about once a week, and looking at movie related media and websites every week or two. Each subject rated a total of 54 hypothetical profiles.

Results showed several factors that impacted the trust ratings that subjects assigned.

- Trust strongly correlates with overall similarity, reconfirming results found in [33, 32]. Correlation was measured by computing the Pearson correlation coefficient over the set of items rated in common by two users. We first computed the correlation between the trust rating assigned to a profile and the average difference between the subjects ratings and the ratings from the profile. That correlation was -0.65, indicating a strong negative correlation; that is, as the average absolute difference in ratings increases, trust decreases.
- Agreement on extremes impacts trust, even with overall similarity held constant. For both medium and large profile differences, the average trust ratings were significantly lower ($p < 0.05$) for profiles where movies with extreme ratings had the large differences.
- The largest single difference between subject and hypothetical user impacts trust, with all other factors held constant
- Subjects propensity to trust varies widely, and is a final significant factor predicting trust ratings they might assign. There are factors specific to each person and each relationship that affect trust independently of similarity. Adjusting the predicted trust values by how much the subject's average trust rating was higher or lower than the overall average trust assigned in the system significantly improved the accuracy of our predictions.

These factors all had statistically significant impacts on trust scores for $p < 0.05$.

To confirm the validity of these experimental results, a linear combination of these similarity measures was used to estimate trust. The accuracy of this estimate was tested in the FilmTrust system where users had rated both movies and their trust for their friends. That known trust value was used as the ground truth against which to compare the estimated value. As shown in Table 2, this estimate based on nuanced trust was a much more effective predictor of trust than overall similarity alone.

Interestingly, in this linear combination, more weight was given to the maximum difference than to the agreement on extremes factor, indicating that the maximum difference had more predictive power. Since this is such a simple measure, this was surprising.

Social trust has been used to produce recommendations because trust indicates similarity in taste. However, is it possible to use these nuanced similarity measures directly to produce recommendations? Inspired by the results from the controlled study, we investigate how accurately these factors may improve the accuracy of recommendations.

## RELATED WORK

### Recommender Systems

#### *User-Based Collaborative Filtering*

First introduced in the Tapestry system [10], user-based collaborative filtering generally computes the similarity or correlation between two users, and uses this measure to weight their ratings of items. One of the early applications was to filtering Usenet News with the GroupLens system [17]. [11] provided a framework for developing collaborative filtering algorithms, and investigated variations on these correlation and similarity measures, including the use of z-scores and alternative correlation measures (e.g. Spearman vs. Correlation). There have been countless variations and optimizations to improve the accuracy of these algorithms, including [18, 15, 16]. Other ways of improving systems to make them more satisfactory to the user rather than simply more accurate were also presented in the seminal article [13].

There have been a number of user-based collaborative filtering methods developed for movie recommendations in particular. MovieLens [12, 20], Recommendz [7], and Film-Conseil [23] are just a few of the websites that implement recommender systems in the context of films.

#### *Model-Based Collaborative Filtering*

One of the drawbacks of user-item collaborative filtering is the sparsity of data. In large systems, it is often the case that users have few, if any, items rated in common. Scalability is also an issue; the computational power required to find similar users increases as the size of the system increases. To address these issues, a number of model-based collaborative filtering algorithms have been developed.

Machine learning techniques have been an effective way to model user preference. Clustering is one technique that has shown improved performance over simpler models. Clustering can be treated as a classification problem, where users are grouped into classes [27, 4, 2]. Co-clustering methods, that cluster both users and items have been shown to be effective and computationally efficient [8]. Bayesian network models [4, 6] build a probabilistic model of the users' preferences, generating expected values for user ratings of items.

**Table 2. Statistics relating actual trust values and predicted trust values derived from overall agreement and a weighted average of nuanced similarity measures.**

| | Overall Similarity | Nuanced Similarity |
|---|---|---|
| Correlation With Actual Trust Value | 0.24 | 0.73 |
| MAE of Computed and Known Trust Value | 1.91 | 1.13 |
| Standard Deviation of Average Difference | 1.95 | 0.95 |

Matrix-factorization methods build a ratings matrix and predict the missing values. These approaches include singular value decomposition (SVD) [25] and non-negative matrix factorization (NNMF) [26]. Similar to clustering methods and using matrix factorization is latent semantic analysis [14], a technique translated from information retrieval to collaborative filtering.

While user-based collaborative filtering algorithms look for people most similar to the user, item-based collaborative filtering [24] works with the set of items. The algorithms look at items rated by the user, and then identify items similar to those to recommend. This approach has been shown to have good performance and quality. Finally, some work such as [29] and [28] has integrated both user and model based approaches.

#### Alternative Weighting Schemes

In addition to our previous work described above, there has been some research that addressed alternative features and weighting schemes. These algorithms weight items or users differently based on estimates of their importance.

One insight is that large variances in ratings may indicate than an item is more important, because it potentially inspires a greater range of opinions. However, in [11], adjusting the weights on items to take into account variance actually produced worse results than when no weighting scheme was used.

The concept of mutual information leads to another approach. The theory behind this suggests that if ratings for Item X are highly tied to those for Item Y, then Item Y may deserve more weight. Some results have shown improved accuracy with this approach [30], but other research has called these results into question, claiming that the improvement originates from the "hand crafted" nature of the analysis [28].

Inverse user frequency is yet another method for weighting that was introduced early in collaborative filtering research. This parallels the inverse document frequency measure in information retrieval systems. It gives more weight to items that are liked by specific groups of people, and less to items that are more universally liked. Essentially, items commonly agreed upon should have less impact than those which are liked more rarely. This method has been shown to improve the accuracy of recommendations [5].

The Leave-one-out (LOO) method has also been used [16]. The goal of this method is to give more weight to users who are more similar, and decrease the weight to less similar users. Results have shown it has some success in outperforming unweighted methods.

These methods all differ from our approach here, as they adjust the weights given to items based on estimates of importance while we adjust the weights given to users based on their similarity on a pre-defined subset of items. These also search the data to classify the importance of items while we have borrowed from empirical results to define these sets of important items *a priori*. However, we believe our techniques could be integrated easily with these weighting methods if needed.

#### Alternative Features for Recommendation.

In addition to alternative weighting schemes, there has been work on considering other factors to improve user satisfaction with recommender systems and to understand their preferences. Bonhard et al. [3] found that showing information about profile similarity between users (i.e. similarity on demographic data and film genre preferences and interests) increased user trust in the recommendations. They argued that incorporating social network profile information could improve recommendations. O'Donovan et al [22] also found that exposing some of the underlying details to the user could help improve their trust in the system. While our work uses alternative similarity measures based in social trust rather than using social details directly, this previous research

## EXPERIMENT

Our previous work demonstrated three factors beyond overall similarity that impacted recommended trust ratings: agreement on extremes, the single maximum difference, and the user's propensity to trust. Earlier research showed that using agreement on extremes in collaborative filtering did not perform better or worse than a simple Pearson correlation method; the results were statistically insignificant. Thus, in this experiment, we focus how well the other factors - maximum single difference and the user's difference from the average - can be used to make movie recommendations directly.

#### Data Sets

We used two main data sets for this experiment: FilmTrust and MovieLens. The FilmTrust data set [9] originates from the website of the same name, a social networking site where users rate and review movies. It contains approximately 27,100 ratings of 1,851 unique

movies made by 1,155 users. The MovieLens 1M data set [11] contains 1,000,209 ratings for approximately 3,900 movies by 6,040 users.

The distribution of ratings overall was remarkably similar between the two data sets. In FilmTrust, the scale of ratings (0.5 - 4) had eight possible values. In total, 28.6% of all ratings were extreme (a rating of either 0.5 or 4), with low extremes accounting for 2.5% and high extremes for 26.1%. MovieLens uses a different scale of 1-5 with no half values. Even with a different set of options, this data had a similar proportion of extreme ratings - 28.4%, with extreme low ratings making up 5.8% and extreme high ratings 22.6%.

### Similarity Measures

We compute the single largest difference on the item rated by both users for which the difference between their ratings is largest. However, since this measure increases as agreement decreases, we inverted it by taking 1 minus the ratio of the maximum difference between the users in question to the maximum possible difference in the system.

In our other work, the user's propensity to trust was an important factor. Just as some individuals are more inclined to trust than others, some users may be more inclined to give high movie ratings than others. In our previous work we adjusted the predicted trust value up or down based on the difference between the user's average trust rating and the overall average trust rating in the system. In these experiments, we will test the effectiveness of adjusting the predicted movie rating up or down based on the difference between the user's average movie rating and the overall average movie rating in the system.

This normalization step is not new to collaborative filtering; it was first discussed over 10 years ago in some of the first work on user-user collaborative filtering [11]. It conveniently follows from our work on trust, and we incorporate it into this analysis, but identifying this factor is not a new contribution of this work.

The following are details of the features used to compute recommended ratings.

- Overall correlation - Using the Pearson correlation coefficient, as in [11]
- Single largest difference - From the set of movies rated by $u_i$ and $u_j$, we found the largest difference in ratings on a single movie. To use this as a weight, we inverted the value by taking the maximum possible difference for the data set (4 in MovieLens and 3.5 in FilmTrust) minus the maximum difference between the users. The distribution of the maximum differences are shown in Table IV.
- User's Rating Statistic - This serves in place of the measure of a user's propensity to trust from the earlier experiment. We compute this statistic as the difference of the user's average movie rating and the overall average movie rating in the system.

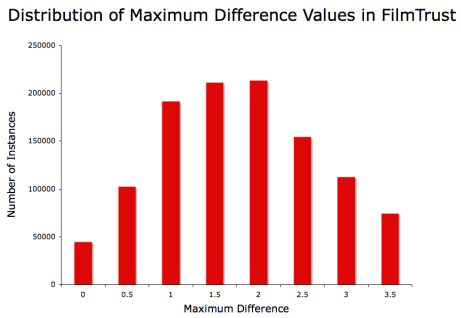


**Figure 1. Maximum Difference distributions in FilmTrust**

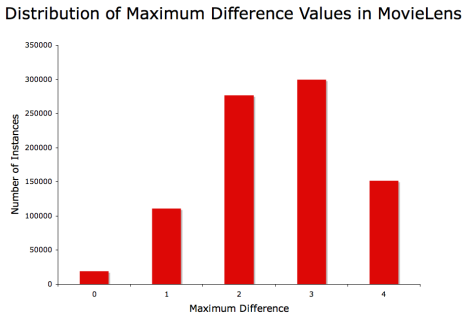


**Figure 2. Maximum Difference distributions in MovieLens**

### Computing Recommended Ratings

Each of these measures were used to generate ratings for movies. For a movie $m$ and user $u_i$ the predicted rating ($\hat{r}_{u_i,m}$) is computed as a weighted average of all ratings of movie $m$. Let $U_m$ be the set of all users who rated movie $m$, where $r_{u,m}$ is the rating a user $u$ has for movie $m$ and $w_{u_i,u}$ is the weight computed from user $u_i$ to that user. Then $\hat{r}_{u_i,m}$ is given by the following formula:

$$\hat{r}_{u_i,m} = \frac{\sum_{\forall u \in U_m} r_{u,m} * w_{u_i,u}}{\sum_{\forall u \in U_m} w_{u_i,u}}$$

This is a generalization of the method used in [11]. Their work uses the Pearson correlation as the weight. In this work, we experiment by using each of the similarity measures given above as weights. The estimation that uses the Pearson Correlation as a weight corresponds to traditional Collaborative Filtering.

There are many advances that have improved upon this original idea. However, our intention in this paper is to show generally how trust-inspired similarity measures compare to correlation-based measures rather than in-

troducing a new optimized recommendation algorithm. Thus, we use the Pearson correlation coefficient to represent this class of algorithms. In future work, similar optimization can be applied to create algorithms using these new similarity measures.

In addition to these methods, we also compared our results to a model-based recommender using a Singular Value Decomposition (SVD) recommendation engine. We based this on the perl Math::Preference::SVD module, developed for the Netflix Prize [2].

## RESULTS

In our earlier research [1], we found that the single largest difference between users significantly impacted the trust they had for one another. This was a surprising result then, and the only reason we discovered it was because subjects in our experiments pointed out that they would scan the data to find the largest difference and would use it to inform their trust decisions. However, that one number intuitively reveals very little about the more complex relationship in users' opinions. As such, we did not expect the Maximum Difference (MD) measure to perform well in this analysis.

Accuracy was determined by comparing the user $u_i$'s rating of movie $m$ with the predicted rating. We present the results both as mean absolute error (MAE) and root mean squared error (RMSE). Results are shown in Tables 3 and 4.

Within each dataset, we compared the results using the Pearson correlation, Maximum Difference, Maximum Difference plus User Difference, and SVD methods. An ANOVA analyzing the MAE values showed significant differences within both datasets; for FilmTrust ($F = 3777.71$, $p<0.001$) and for MovieLens ($F = 6059.38$, $p<0.001$). This was followed up by pairwise Student's t-tests.

To our surprise, the recommendations made using the MD as a weight outperformed the Pearson Correlation in both data sets ($p < 0.01$).

Normalizing the recommendation by adding the difference between the user's average rating and the population's average rating also offered significant benefits. Additional t-tests showed that adjusting by the User's Difference (UD) offered large and significant improvements in both data sets ($p < 0.01$). The MD+UD method was significantly more accurate than the Pearson Correlation method and the original MD method.

The model-based approach had mixed results. For the FilmTrust data set, MD + UD produced significantly more accurate results than SVD ($p < 0.01$). However, for MovieLens, SVD performed significantly better than all other methods. While this does not indicate that our methods are superior, it also does not prove that they cannot outperform a model-based approach. Further research with more optimized recommendation algorithms using our measures should be conducted to determine this one way or another.

[2]http://www.timelydevelopment.com/demos/NetflixPrize.aspx

**Table 3. MAE and RMSE results from different methods of computing recommendations on the FilmTrust dataset.**

| Method | FilmTrust Overall | |
|---|---|---|
| | MAE | RMSE |
| Pearson Correlation | 0.607 | 0.793 |
| Maximum Difference | 0.515 | 0.691 |
| Maximum Difference + UD | 0.490 | 0.643 |
| SVD | 0.644 | 0.905 |

**Table 4. MAE and RMSE results from different methods of computing recommendations on the MovieLens dataset.**

| Method | MovieLens Overall | |
|---|---|---|
| | MAE | RMSE |
| Pearson Correlation | 0.775 | 0.971 |
| Maximum Difference | 0.757 | 0.951 |
| Maximum Difference + UD | 0.727 | 0.928 |
| SVD | 0.674 | 0.823 |

## WHY IS THE MAXIMUM DIFFERENCE SUCCESSFUL?

Although the results are consistently good, it is not obvious why such a simple statistic as the single largest difference in users' ratings would lead to a significant improvement in recommendation accuracy. In the context of trust, the reason for this factor makes sense; users feel that the worst disagreement they have with a person - even if it happens once - may predict future disagreements of the same magnitude. Considering this in a social sense, a single betrayal can have a large impact on trust even if there is an extensive history of agreement and good behavior. In recommendations, though, this insight does not directly translate. While a single large difference may predict future large differences, it is not obviously that the magnitude of that difference should be predictive for all items.

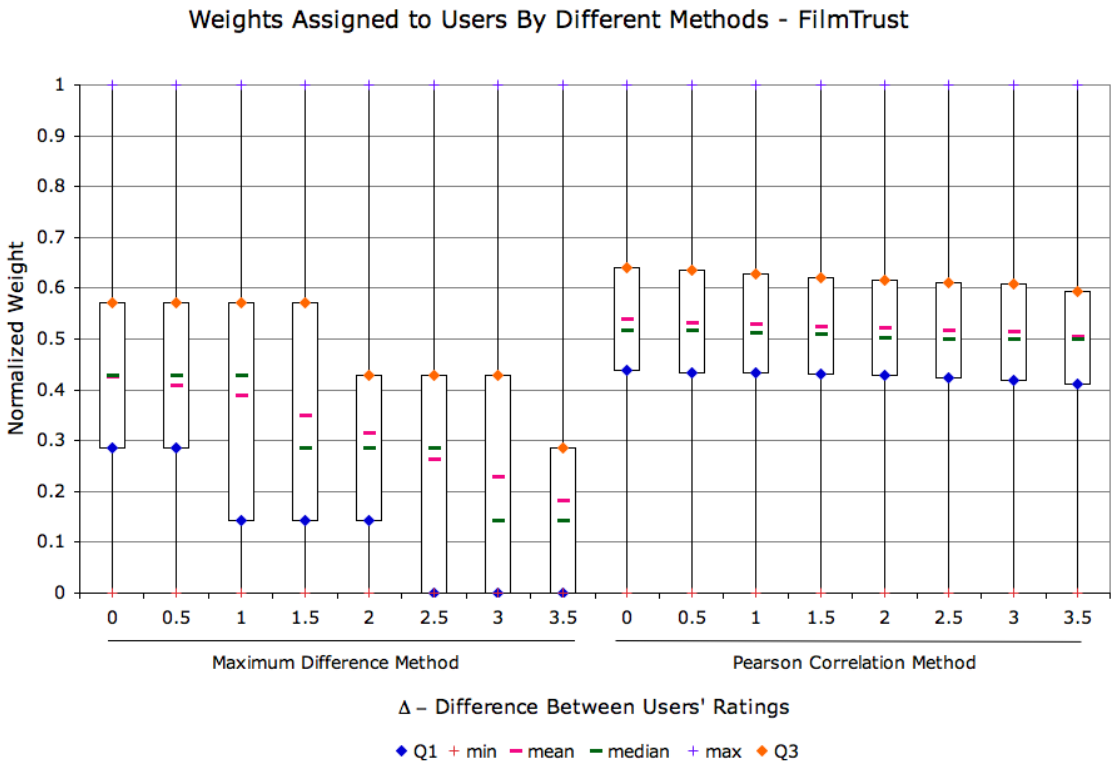


**Figure 3. Box Plots of the weights assigned to users with a given Δ on movies in FilmTrust. Note that the weight for users with larger Δ drops off much more sharply with the MD method than with the Pearson Correlation.**

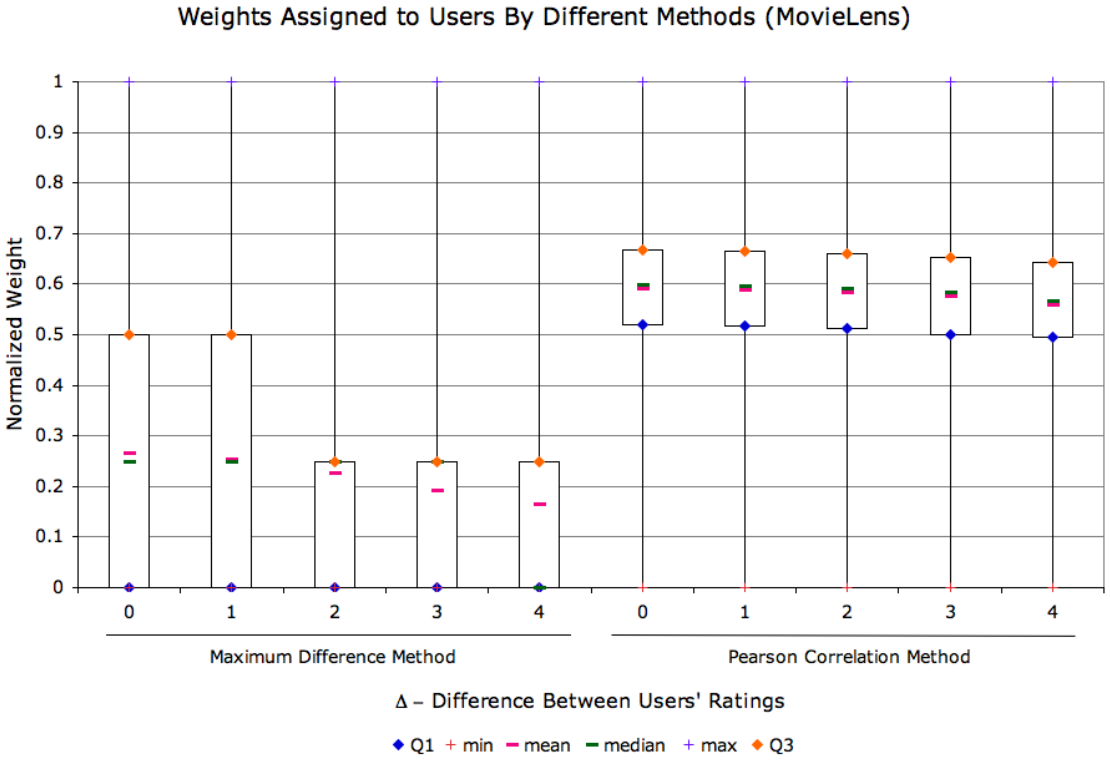


**Figure 4. Box Plots of the weights assigned to users with a given Δ on movies in MovieLens. Note that the weight for users with larger Δ drops off much more sharply with the MD method than with the Pearson Correlation.**

To gain further insights, we conducted an additional analysis that compared the weight given to people in computing the recommendations, broken down by the difference between their rating and the user's rating on a given movie. We used the same training and test sets as described above. For each user-movie rating pair in the test set $(u, m)$, we looked at all the users in the training set who had rated $m$, computed the difference between their rating and user $u$'s rating (call this $\Delta$), and the looked at the MD and Pearson correlation coefficient between $u$ and the user computed on the training set. We then grouped the observations by $\Delta$ to see how the weights corresponded with the rating difference. Ideally, as $\Delta$ increases, the weight given to the users should decrease.

Figures 3 and 4 show the results for FilmTrust and MovieLens respectively. They also reveal why the maximum difference may be outperforming the Person correlation. While the average weight decreases as $\Delta$ increases for both methods, the drop-off is much sharper with the MD statistic. Thus, much more weight is given to ratings that are close to $u$ and the predictions are more accurate.

An argument against this evidence may be that the MD appears to be predictive because it frequently matches the actual difference in ratings ($\Delta$). In fact, the maximum difference between two users matches their difference for a given movie only 19.4% of the time in FilmTrust and 23.2% in MovieLens. Note the rate is higher in MovieLens because MovieLens has only five possible ratings while FilmTrust has eight. To further explore this, we removed all user-user-movie items from the dataset where the MD= $\Delta$ and re-ran the analysis from above. As shown in figures 5 and 6, a similar pattern of weighting occurs. Both methods decrease weight to the user as $\Delta$ increases, with the exception of the largest possible difference, where the weight increases. This is due to the fact that all users with that largest maximum difference are eliminated, leaving only users with smaller differences and higher weights. In both data sets, the decrease in weight is much sharper for MD than for the Pearson Correlation.

Because the MD effectively lets us give more weight to people whose ratings are always close to the user's (indicated by a low MD), as a whole the statistic leads to more accurate recommendations.

## DISCUSSION

We found very three main results in these experiments.

- The Maximum Difference method produces significantly more accurate results than the Pearson Correlation method in both data sets.

- The Maximum Difference methods outperform the model-based SVD recommendation on one of the two data sets.

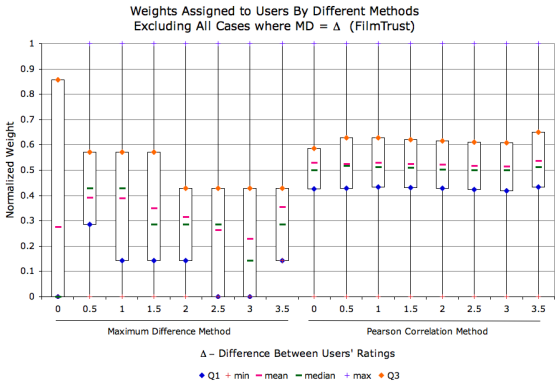


**Figure 5. Box Plots of the weights assigned to users with a given $\Delta$ on movies in FilmTrust excluding cases where MD = $\Delta$. Note the pattern is the same as the general case - MD decreases weights at a greater rate than the Pearson correlation method.**

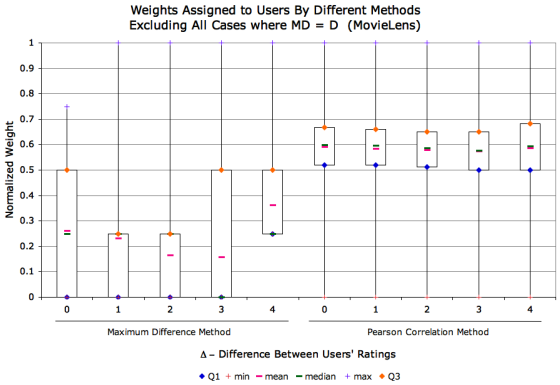


**Figure 6. Box Plots of the weights assigned to users with a given $\Delta$ on movies in MovieLens excluding cases where MD = $\Delta$. Note the pattern is the same as the general case - MD decreases weights at a greater rate than the Pearson correlation method.**

- Adjusting the predicted value by adding the difference between the user's average rating and the overall average rating significantly improves the results on both data sets.

The superior performance of the Maximum Difference statistic is surprising. Even though this echoes results we found in our earlier studies that linked MD with trust between users, we were not expecting to see the strong performance of the method here. Intuitively, it is unclear why simply identifying the biggest difference between two users should capture any meaningful information about the ability of one person to accurately recommend movies to the other. Essentially, the MD represents the pessimistic view; people are given credit based on their biggest "mistake" from the user's opinion. By doing this, users who are consistently closer receive more weight.

The fact that this measure works well in recommender systems has several implications. First, it makes it very

easy to recompute weights as information changes in the system. To update a weight between a pair of users simply requires checking to see if the difference on the newly rated movie is larger than the existing maximum difference. With no computationally difficult re-computation needed, systems can immediately use new information.

Secondly, it offers a straightforward step for improving existing user-based collaborative filtering algorithms. While some systems benefit from highly optimized model-based collaborative filtering algorithms, other websites use recommender systems to offer simple personalization. The maximum difference can be easily integrated as a replacement weight to improve these systems.

## CONCLUSION AND FUTURE WORK

In this study, we examined the accuracy of a simple similarity measure - the single largest difference between users - to generate recommendations in a use-based collaborative filtering system. Our results indicated that this statistic, derived from experiments on the relationship between trust and similarity, significantly outperform the more traditional Pearson correlation similarity measure and, in some cases, a model-based approach.

While, our goal in this research has been to look at some simple similarity measures for recommendations. We have not transitioned these measures into the more finely tuned recommendation engines. A next step will investigate what is necessary to make this transition. That will include straightforward algorithm development as well as issues of scaling and coverage. It may also be possible to apply some of these insights to model-based collaborative filtering, and this too is a topic of future work.

## REFERENCES


1. ANONYMIZED. anonymized. *to appear in Transactions on the Web* (2009).
2. BASU, C., HIRSH, H., AND COHEN, W. Recommendation as classification: using social and content-based information in recommendation. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence* (Menlo Park, CA, USA, 1998), American Association for Artificial Intelligence, pp. 714–720.
3. BONHARD, P., HARRIES, C., MCCARTHY, J., AND SASSE, M. A. Accounting for taste: using profile similarity to improve recommender systems. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems* (New York, NY, USA, 2006), ACM, pp. 1057–1066.
4. BREESE, J. S., HECKERMAN, D., AND KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence* (1998), pp. 43–52.
5. BREESE, J. S., HECKERMAN, D., AND KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Uncertainty in Artificial Intelligence. Proceedings of the Fourteenth Conference (1998)* (1998), Morgan Kaufmann, pp. 43–52.
6. CHIEN, Y.-H., AND GEORGE, E. A bayesian model for collaborative filtering. In *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics* (1999).
7. GARDEN, M., AND DUDEK, G. Semantic feedback for hybrid recommendations in recommendz. In *EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 754–759.
8. GEORGE, T., AND MERUGU, S. A scalable collaborative filtering framework based on co-clustering. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 625–628.
9. GOLBECK, J. Generating Predictive Movie Recommendations from Trust in Social Networks. *Proceedings of The Fourth International Conference on Trust Management* (2006).
10. GOLDBERG, D., NICHOLS, D., OKI, B. M., AND TERRY, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM 35*, 12 (1992), 61–70.
11. HERLOCKER, J. L., KONSTAN, J. A., BORCHERS, A., AND RIEDL, J. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1999), ACM, pp. 230–237.
12. HERLOCKER, J. L., KONSTAN, J. A., AND RIEDL, J. Explaining collaborative filtering recommendations. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work* (New York, NY, USA, 2000), ACM, pp. 241–250.
13. HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G., AND RIEDL, J. T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. 22*, 1 (2004), 5–53.
14. HOFMANN, T. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst. 22*, 1 (2004), 89–115.

15. Huang, Z., Chen, H., and Zeng, D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst. 22*, 1 (2004), 116–142.

16. Jin, R., Chai, J. Y., and Si, L. An automatic weighting scheme for collaborative filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2004), ACM, pp. 337–344.

17. Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. Grouplens: applying collaborative filtering to usenet news. *Commun. ACM 40*, 3 (1997), 77–87.

18. Ma, H., King, I., and Lyu, M. R. Effective missing data prediction for collaborative filtering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2007), ACM, pp. 39–46.

19. Massa, P., and Bhattacharjee, B. Using trust in recommender systems: an experimental analysis. In *Proceedings of the 2nd International Conference on Trust Management* (Oxford, UK, March 2004), C. Jensen, S. Poslad, and T. Dimitrakos, Eds., vol. 2995 of *LNCS*, Springer-Verlag.

20. Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., and Riedl, J. Movielens unplugged: experiences with an occasionally connected recommender system. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces* (New York, NY, USA, 2003), ACM, pp. 263–266.

21. O'Donovan, J., and Smyth, B. Trust in recommender systems. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces* (New York, NY, USA, 2005), ACM, pp. 167–174.

22. O'Donovan, J., Smyth, B., Gretarsson, B., Bostandjiev, S., and Höllerer, T. Peerchooser: visual interactive recommendation. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), ACM, pp. 1085–1088.

23. p. Perny, and Zucker, J. D. Preference-based search and machine learning for collaborative filtering: the "film-conseil" recommender system. *Information, Interaction , Intelligence 1*, 1 (2001), 9–48.

24. Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web* (New York, NY, USA, 2001), ACM, pp. 285–295.

25. Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Application of dimensionality reduction in recommender systems–a case study. In *ACM WebKDD Workshop* (2000).

26. Srebro, N., and Jaakkola, T. Weighted low rank approximation. In *Proceedings of the 20th International Conference on Machine Learning.* (2003).

27. Ungar, L., and Foster, D. Clustering methods for collaborative filtering. In *Proceedings of the Workshop on Recommendation Systems* (1998), AAAI Press, Menlo Park California.

28. Wang, J., de Vries, A. P., and Reinders, M. J. T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2006), ACM, pp. 501–508.

29. Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., and Chen, Z. Scalable collaborative filtering using cluster-based smoothing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2005), ACM, pp. 114–121.

30. Yu, K., Wen, Z., Ester, M., and Xu, X. Feature weighting and instance selection for collaborative filtering. In *DEXA '01: Proceedings of the 12th International Workshop on Database and Expert Systems Applications* (Washington, DC, USA, 2001), IEEE Computer Society, p. 285.

31. Ziegler, C.-N. *Towards Decentralized Recommender Systems*. PhD thesis, Albert-Ludwigs-Universität Freiburg, Freiburg i.Br., Germany, June 2005.

32. Ziegler, C.-N., and Golbeck, J. Investigating Correlations of Trust and Interest Similarity. *Decision Support Services* (2006).

33. Ziegler, C.-N., and Lausen, G. Analyzing correlation between trust and user similarity in online communities. In *Proceedings of the 2nd International Conference on Trust Management* (Oxford, UK, March 2004), C. Jensen, S. Poslad, and T. Dimitrakos, Eds., vol. 2995 of *LNCS*, Springer-Verlag, pp. 251–265.

34. Ziegler, C.-N., and Lausen, G. Spreading activation models for trust propagation. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service* (Taipei, Taiwan, March 2004), IEEE Computer Society Press.