Improving Memory-based Collaborative Filtering Using A Factor-based Approach

Zhenxue Zhang, Dongsong Zhang and Zhiling Guo

Department of Information Systems
University of Maryland, Baltimore County
Baltimore, MD 21250
{zzhang3, zhangd, zguo}@umbc.edu

Abstract

Collaborative Filtering (CF) systems generate recommendations for a user by aggregating item ratings of other like-minded users. The memory-based approach is a common technique used in CF. This approach first uses statistical methods such as Pearson's Correlation Coefficient to measure user similarities based on their previous ratings on different items. Users will then be grouped into different neighborhood depending on the calculated similarities. Finally, the system will generate predictions on how a user would rate a specific item by aggregating ratings on the item cast by the identified neighbors of his/her. However, current memory-based CF method only measures user similarities by simply looking at their rating trends while ignoring other aspects of overall rating patterns. To address this limitation, we propose a novel factor-based approach by incorporating user rating average, user rating variance, and number of overlapping ratings into the measurement of user similarity. The proposed method was empirically evaluated against the traditional memory-based CF method and other existing approaches including case amplification, significance weighting, and z-score using the MovieLens dataset. The results showed that the prediction accuracy of the proposed factor-based approach was significantly higher than existing approaches.

Introduction

Collaborative Filtering (CF) research was initiated a decade ago by three articles (Hill et al. 1995; Resnick et al. 1994; Shardanand & Maes 1995) and has drawn a lot of attention from both academics and industry since then. Nowadays, CF has been widely adopted in e-commerce, such as Amazon.com and Netflix.com, as an essential part of their business models to improve cross-selling and enhance customer loyalty (Sarwar et al. 2000). In October 2006, Netflix started a world-wide, multi-year contest of CF-based recommender systems. Anyone who can develop a system that improves the company's current recommender system by at least 10% will win a one million dollar prize (O'Brien 2006).

Collaborative filtering automates the word-of-mouth recommendation process, in which people share their preferences on items among friends to help each other find preferable ones. The underlying assumption of CF is that people who share similar preference on different items in the past tend to have similar preference on other items again in the future.

In CF systems, a user's preference is represented by his/her ratings on different items and those ratings are used to measure the similarity of different users' preferences. Therefore, in order to predict how a user (i.e., the active user) would rate a specific item (i.e., the target item), for which he/she has not rated, a CF system will first identify a group of users (called neighborhood) who are similar to the active user in terms of their preferences in the past and have already rated that item. Ratings on the target item from users in the neighborhood will then be used to generate the prediction for the active user. We use the following example to illustrate how a typical CF-based recommender system works.

Movies / Users	Titanic	Star Wars	Shrek	Harry Potter	Minority Report
Jack	4	5	5	1	5
Heather	5		4	5	2
Kyle	3	5	5	2	
David	5	4		4	4
Linda	5	1		5	1

Table 1. A User-Item Matrix in a Movie Recommender System

Table 1 shows a simplified user-item matrix that a CF-based recommender system maintains. Each row in the matrix represents a user, and each column stands for an item (a movie in this case). The value stored in each cell represents a specific user's rating on an item using a 1~5 scale. A rating of 5 means that the user considers a movie as one of his/her favorites, while a rating of 1 means that the user does not like that movie at all. Those empty cells indicate that certain users have not watched or rated a movie yet. In order to predict how Kyle (i.e., the active user) may rate the movie "Minority Report" (i.e., the target item), a recommender system compares four ratings on different movies that Kyle has already given with those on the same movies given by others. Obviously Kyle and Jack have very close ratings on the movies that both have rated,

implying that they have similar preferences on movies. Because Jack has rated "Minority Report" very high, the system infers that Kyle may also likely rate high for "Minority Report". As a result, this movie should be recommended to Kyle. Note that in reality an active user tends to have similar preferences to not just one but rather a group of users, who are usually referred to as the neighbors of the active user.

In this paper, we discuss problems with the current memory-based CF methods on how they measure user similarities on item preferences. A novel approach is proposed to address those problems and improve the performance of CF systems by incorporating several new factors to refine user weighting. We used the well-known MovieLens dataset to test the performance of the proposed technique against existing ones. Result shows that the factor-based approach significantly outperformed current methods.

Background

In general, there are two major approaches to collaborative filtering, namely memory-based CF and model-based CF (Breese et al. 1998). Memory-based CF systems utilize the original, entire user-item rating matrix to generate every prediction (Resnick et al. 1994), while model-based CF methods recommend items by first developing a descriptive model of user ratings based on a user-item matrix via different machine learning approaches such as Bayesian network and clustering. The generated model is then used for future prediction about user preferences (Breese et al. 1998). Although model-based CF methods overcome some shortcomings of memory-based counterparts, such as low scalability and high online computation overhead, some studies show that they are generally inferior to memory-based ones in terms of prediction accuracy (Breese et al. 1998; Calderon-Benavides et al. 2004; Herlocker et al. 1999). This study focuses on improving the accuracy of memory-based CF.

Memory-based Collaborative Filtering

The prediction process of memory-based CF usually involves three steps: user similarity measurement, neighborhood selection, and prediction generation.

User Similarity Measurement This is the first and most important step of the method. The goal of this step is to measure the similarity weight $w_{x,y}$ between a user pair x and y, based on their ratings on common items. Different similarity measures have been proposed in the CF literature (Breese et al. 1998; Herlocker et al. 1999; Shardanand & Maes 1995). Among them, Pearson's Correlation Coefficient (PCC) (Resnick et al. 1994) and Cosine Vector Similarity (CVS) (Breese et al. 1998) are the two most commonly used. Previous studies have shown that PCC performs better than CVS (Breese et al. 1998; Herlocker et al. 1999). Pearson's correlation coefficient is usually calculated in a general form as follows:

$$w_{x,y} = \frac{\sum_{i \in I_{x,y}} (r_{x,i} - \bar{r}_x^{'})(r_{y,i} - \bar{r}_y^{'})}{\sqrt{\sum_{i \in I_{x,y}} (r_{x,i} - \bar{r}_x^{'})^2 \sum_{i \in I_{x,y}} (r_{y,i} - \bar{r}_y^{'})^2}}, \quad (1)$$

where $r_{x,i}$ denotes the rating cast by user x on item i and $I_{x,y}$ stands for the set of items that both x and y have rated. $\bar{r}_x^{'}$, the mean rating given by user x on items that both users have rated, is define as:

$$\bar{r}_x^{'} = \frac{1}{|I_{x,y}|} \sum_{i \in I_{x,y}} r_{x,i}, \quad (2)$$

where $|I_{x,y}|$ is the number of items rated by both user x and y. By definition, the similarity weight $w_{x,y} \in [-1,1]$ measures linear dependencies between two users' preferences. A value of 1 indicates ratings of user x correlate perfectly with those of user y, while a value of -1 corresponds to a perfect negative correlation between two users' ratings. Furthermore, a value of $w_{x,y}$ equal to zero implies no linear relationship between the ratings of the two.

Cosine vector similarity measures the similarity weight $w_{x,y}$ by computing the cosine of the angle θ between N-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, which represent the rating profiles of users x and y, respectively. N is equal to the number of items for which both users have cast votes (i.e., $|I_{x,y}|$). A general cosine vector similarity between two rating vectors is defined as follows:

$$w_{x,y} = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \times \|\boldsymbol{y}\|} = \frac{\sum_{i \in I_{x,y}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_{x,y}} r_{x,i}^2} \sqrt{\sum_{i \in I_{x,y}} r_{y,i}^2}}, \quad (3)$$

where $\boldsymbol{x} \cdot \boldsymbol{y}$ denotes the dot product between rating vectors of users x and y. $\|\boldsymbol{x}\|$ and $\|\boldsymbol{y}\|$ denote the norm (or length) of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

By comparing (1) and (3), it is easily seen that the only difference between PCC and CVS is that PCC works with centered ratings, that is, ratings that have been shifted by the sample mean so as to have an average of zero. However, both PCC and CVS, when they are used to measure user similarity in CF, are examining the same aspect of user ratings, namely the overall linear dependency in terms of how two users rate items.

Once user similarity weights are calculated, a subset of similar users (i.e., the neighborhood) is then selected for generating final prediction for the active user. Commonly used methods for neighborhood selection are similarity thresholding and best-N-neighbors.

Neighborhood Selection The similarity thresholding approach uses a certain threshold value L to filter out users whose similarity with the active user is lower than L

(Shardanand & Maes 1995). The best-N-neighbors method considers the top N users that are most similar to the active user for generating predictions (Herlocker et al. 1999). The values L and N is determined by balancing the tradeoff between the accuracy of predictions and the coverage of a system, which is defined as the percentage of missing ratings that can be predicted by the system. Existing approaches pick an optimal value of L and N based on empirical test on specific datasets (Shardanand & Maes 1995; Herlocker et al. 1999).

Prediction Generation Finally, a prediction of preference $p_{a,i}$ for the active user a on an item i can be computed by aggregating ratings on i given by all users in a's neighborhood and their similarity weights $w_{a,x}$ calculated in the first step. The most widely used form of aggregation function is shown as follows (Resnick et al. 1994; Breese et al. 1998):

$$p_{a,i} = \bar{r}_a + \frac{\sum_{x \in N} w_{a,x} \times (r_{x,i} - \bar{r}_x)}{\sum_{x \in N} |w_{a,x}|}, \quad (4)$$

where $\bar{r}_a$ and $\bar{r}_x$ denote the mean ratings for all rated items cast by the active user a and user x respectively. N stands for the set of users in the neighborhood of the active user identified in the second step. The generally accepted justification for the form of aggregation function is that it accommodates different users' rating habits (e.g., some generous users tend to rate all items relatively high, while other picky users tend to do just the opposite). Therefore, only the rating differences rather than the real value of the ratings from neighborhood users are taken into account in the prediction. The resulting prediction is calculated as the active user's average score adjusted by the weighted rating differences of the neighborhood users.

Problems with Current Memory-based CF Method

Items / Users	*A*	*B*	*C*	*D*	*E*
a	2	4	3	2	?
b	2	4	3	2	4
x	3	5	4	3	5
y	1	5	3	1	5
z		4	3		3

Table 2. A User-Item Matrix Illustrating Different Factors

The analysis on user similarity measurement shows that PCC measures the degree to which two user ratings are linearly dependent. A positive PCC value indicates that whenever one user rates an item with a high score, so does the other. However, the PCC score does not necessarily reflect the true difference between two users' preferences. This method only measures one kind of user rating patterns while ignores other aspects and, in turn, may fail to measure the true differences in user preferences. The following example illustrates some problems with current memory-based CF method that is based solely on PCC similarity measure.

In Table 2, there are five users who have rated either all or some of the five items listed. The active user a has not rated the item E (target item). Using traditional PCC calculation (Equation 1), the weight $w_{a,b} = w_{a,z}$, which means that users b and z have the same amount of weights (or influence) when generating predictions for the active user on item E using Equation (4). They are treated as equally close to the active user a. However, if we further examine the ratings in Table 2, we can see that b and z have different numbers of overlapping ratings (i.e., number of items rated by both users) with the active user a. User b shares 4 overlapping ratings with user a while user z only shares 2. Intuitively, given the same level of correlation, the more a user shares overlapping ratings with the active user, the more reliable the user's rating can be used for prediction, and accordingly the higher the weight he/she should contribute to the prediction function (4). Unfortunately, current PCC-based CF system cannot differentiate users b and z.

Clearly, the effectiveness of a memory-based CF system can be improved if we use refined metrics to further differentiate users and evaluate user similarity revealed in their rating patterns. More importantly, greater improvement can be achieved if the method is more effective in differentiating users who have high weighting values because those users play a significant role in the final prediction. In the following section, we briefly review some existing schemes to improve the similarity weight. We then discuss their limitations and propose our new method.

Existing Approaches to Improving Similarity Weighting Schemes

There have been several approaches proposed to improving traditional similarity weighting schemes, such as case amplification, significance weighting, and z-Score.

Case Amplification Case amplification is a method of rescaling the original PCC weight by a nonlinear transformation. It is designed to reward high weights that are close to 1, or equivalently, punish low weights that are close to 0 (Breese et al. 1998). In particular, weights are transformed using the following function:

$$w'_{a,x} = \begin{cases} w_{a,x}^{\rho} & if\ w_{a,x} \geq 0 \\ -(-w_{a,x})^{\rho} & if\ w_{a,x} < 0 \end{cases}, \quad (5)$$

where ρ is a value larger than 1. The transformed weight $w'_{a,x}$ will then be used to replace $w_{a,x}$ in Equation (4). Previous studies have found that there is no significant effect of case amplification on system performance measured by Mean Absolute Error (MAE) (Breese et al. 1998). The problem with the case amplification approach is that it only focuses on the values of weights without

differentiating those weights with strong evidence from those without. For example, case amplification will yield the same transformed value $w'_{a,b} = w'_{a,z}$ in the previous example since their original value is the same. Additionally, the choice of ρ is rather arbitrary, which may hurt the generalizability of this method.

Significance Weighting Significance weighting is designed to devalue those similarity weights that are based on a very small number of overlapping ratings (Herlocker et al. 1999). The adjusted weight is calculated as follows:

$$w'_{a,x} = \begin{cases} w_{a,x} & if \ \left|I_{a,x}\right| \geq 50 \\ \frac{\left|I_{a,x}\right|}{50} w_{a,x} & if \ \left|I_{a,x}\right| < 50 \end{cases}, \qquad (6)$$

where $\left|I_{a,x}\right|$ is the number of items rated by both user x and the active user. Therefore, weights generated based on less than 50 overlapping ratings are devalued. It is found that significance weighting improves accuracy of the system by a large amount (Herlocker et al. 1999). Although this method partially solves the problem found in case amplification, its effectiveness is limited by the inability to differentiate those weights with large number of overlapping ratings.

Z-Score In order to account for the differences in users' rating distribution, z-scores of user ratings are used in place of rating differences in the prediction generation function (Herlocker et al. 1999). Formula (4) is transformed into the following form:

$$p_{a,i} = \overline{r}_a + \sigma_a \times \frac{\sum_{x \in N} w_{a,x} \times \frac{(r_{x,i} - \overline{r}_x)}{\sigma_x}}{\sum_{x \in N} \left|w_{a,x}\right|}, \qquad (7)$$

where σ_a and σ_x are standard deviations of the rating samples for the active user a and user x, respectively. No improvement on performance was found using z-score approach over original CF method in the previous study (Herlocker et al. 1999). However, in our opinion, normalizing the rating in the prediction function is a valid step to help improve the prediction accuracy since it explicitly takes into account each neighborhood user's rating variance, which is another useful measurement for user similarity in the rating pattern.

By introducing the existing approaches, we can see that major efforts in prior work are made towards either revising the prediction components or adjusting the weights in the original prediction Equation (4). Previous studies show positive support that further differentiating neighborhood users is an effective way to improve prediction accuracy.

A Factor-based Approach

In this section, we propose a novel approach to enhance the accuracy of current memory-based CF techniques. The proposed method incorporates several factors to strengthen the role of individual users who are genuinely similar to the active user on item preferences while reducing the weight of users who are different from the active user in terms of rating patterns.

A New Factor Treatment

Our general goal is to address the current limitations in similarity weight calculation in memory-based CF approaches by incorporating a few new factors using certain treatment. Since users who are identified as close neighbors of the active user should be more influential in the prediction, the treatment should be more potent in differentiating $w_{a,x}$ when its value is close to 1 than when it is close to 0. Therefore, our emphasis is to further differentiate those users whose PCC weights are close to 1 and reduce the weights of those much dissimilar neighborhood users to weaken their roles in the prediction. In this study, we chose and tested the exponential treatment.

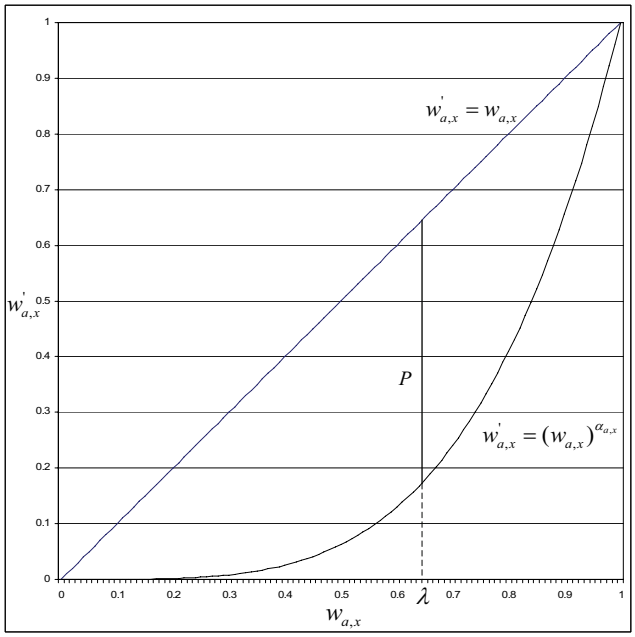

Figure 1. Exponential Treatment

Exponential Treatment The treatment takes weight transformation using a power function, as illustrated in Figure 1. The transformed weight $w'_{a,x}$ is determined as follows:

$$w'_{a,x} = \begin{cases} (\lambda \times w_{a,x})^{n \times \alpha_{a,x}} & if \ w_{a,x} \geq 0 \\ -(-\lambda \times w_{a,x})^{n \times \alpha_{a,x}} & if \ w_{a,x} < 0 \end{cases}, \qquad (8)$$

where $w_{a,x} \in [-1,1]$, $0 < \lambda \le 1$, $n > 0$, $\alpha_{a,x} \ge 1$.

Intuitively, the scaling exponent in the power function should not be too large or too small. If the scaling exponent is close to 1, the power function is close to a straight line. The difference between the transformed value and the original value is too small and will not serve our differentiation purpose. The higher the scaling exponent, the more the curve is tilted towards the point (1, 0), i.e., the bottom right corner in Figure 1. Consequently, if the scaling exponent is too large, the power function loses its differentiating effectiveness because the curve is too skew such that the functional transformation can only be effective to differentiate a small range of weights that are close to 1 but mistakenly treat all other eights equally useless in the prediction generation. Therefore, there is subtle tradeoff between extracting more information from genuine close users and discarding useful information in the prediction. Design of the power function requires careful selection of its parameters λ and *n*.

Here our treatment takes three logical steps. First, the value of $\alpha_{a,x}$ is used to identify close users who share similar patterns other than those captured by $w_{a,x}$. It will be shown shortly in the next section that the higher the $\alpha_{a,x}$, the higher the distance (dissimilarity) between a user pair. For a given value of $w_{a,x}$, the power function will transform it into a lower new weight if it has a higher $\alpha_{a,x}$. This is desirable. Nevertheless, as we will show later, the value of $\alpha_{a,x}$ varies differently across users and may not fall in the most effective range to set close neighbors apart from those dissimilar ones. Therefore, as the second step, a factor $n \in R^+$ is used to adjust the value of $\alpha_{a,x}$ to yield the best test result for the treatment. The value n is picked empirically in this study, as will be discussed in the evaluation section. Furthermore, since our interest is to further distinguish users who have $w_{a,x}$ close to 1, we take the third step to make the weight reduction even more effective. Applying a fractional factor λ to the original weight $w_{a,x}$ can effectively amplify the transformed functional difference while preserving the property of the identified closeness captured by $\alpha_{a,x}$. After the three-step mapping, we are able to significantly reduce the weights of those neighborhood users who have high PCC weights but are not truly similar to the active user.

In order to find the optimal λ, we first define potency of the exponentiation. The potency P of the exponentiation, as shown in Figure 1, is defined as the vertical distance between $w'_{a,x} = w_{a,x}$ and $w'_{a,x} = (w_{a,x})^{\alpha_{a,x}}$, which can be expressed as follows:

$$P(w_{a,x}) = w_{a,x} - (w_{a,x})^{\alpha_{a,x}}. \quad (9)$$

Since Equation (9) is a concave function within our defined region for parameter values, P must have a maximum value that can be found by taking the first derivative with respect to $w_{a,x}$:

$$\lambda \equiv w^*_{a,x} = \left(\frac{1}{\alpha_{a,x}}\right)^{\frac{1}{(\alpha_{a,x}-1)}}. \quad (10)$$

Because our focus is on users whose $w_{a,x}$'s are close to 1, we can define λ as the optimal value in Equation (10). The transformation function (8) will effectively bring those $w_{a,x}$'s that are close to 1 to the neighborhood of $w^*_{a,x}$, and therefore, obtain the maximal differentiation power.

Since $\alpha_{a,x}$ may take different values for different user pairs, the question is what value of $\alpha_{a,x}$ should be used in the calculation of the optimal λ. We pick $\alpha_{a,x}$ using an empirical approach, which will be described in detail in the evaluation section. Note that the case amplification method discussed in the previous section takes advantage of the nature of exponentiation too. However, we go one step further by distinguishing close neighbors from others using additional cues.

Finally, we used the adjusted weight $w'_{a,x}$ to replace the original weight $w_{a,x}$ in Equation (4) for prediction generation purpose.

Identify Factors

As we suggested before, PCC can only detect the difference in the overall trend of how users rate items. There are other features of user rating patterns that can be used to measure the true similarity of users' preferences, such as rating average, rating variance, and number of overlapping ratings.

Rating Average As mentioned earlier, users may have different mindsets when rating items. Rating average can be used to distinguish those users who tend to rate items in the same trend as the active user but probably using a different rating scale. For example, ratings of users *a* and *x* shown in Table 2 share the same trend and their PCC weight is equal to 1, but their rating averages are indeed different.

As discussed in the previous section, the rating average has somewhat already been taken into consideration in current memory-based CF approach. However, as shown in Equation (4), the current approach only considers the effect of the difference in the weighting components without affecting the determination of weight $w_{a,x}$ in the prediction. We believe the influence of users who have different rating averages in comparison to the active user should be weakened by lowering their weights during prediction generation. In this study, rating average factor $\alpha^{\mu}_{a,x}$ is used and it is defined as follows:

$$\alpha^{\mu}_{a,x} = \begin{cases} \mu_x / \mu_a & if\ \mu_a \le \mu_x \\ \mu_a / \mu_x & if\ \mu_a > \mu_x \end{cases}, \quad (11)$$

where μ_a and μ_x are the rating averages of the active user and user *x*, respectively.

Rating Variance Users may also have different ways to express their preferences. Some users like to rate items using extreme ratings (i.e., 1 or 5), while others may hesitate to do that. For example, users *a* and *y* in Table 2 rate items following the same trend and even have the same rating averages, but their rating variances are different.

The z-score approach discussed earlier is designed to take into consideration the different user rating variances (Herlocker et al. 1999). Because of the disappointing results, researchers of that study concluded that rating variance does not help in the prediction generation. However, in our opinion, rating variance can still be helpful to distinguish users who are truly similar to the active user on item preference from those who are not. As a matter of fact, different from Herlocker et al. (Herlocker et al. 1999), we find z-score performs better than the original CF method. Nevertheless, z-score approach focuses on the prediction components but fails to directly decrease weights of those neighbors who do not share similar rating variance with the active user. In order to overcome this shortcoming, we used rating variance $\alpha^{v}_{a,x}$, which is defined as follows:

$$\alpha^{v}_{a,x} = \begin{cases} \text{var}_x / \text{var}_a & \text{if } \text{var}_a \le \text{var}_x \\ \text{var}_a / \text{var}_x & \text{if } \text{var}_a > \text{var}_x \end{cases}, \qquad (12)$$

where var_a and var_x are the rating variance of the active user and user *x*, respectively. Additionally, the rating variance factor, after being taken into account in our approach, is expected to be complimentary to the z-score approach when they are combined.

Number of Overlapping Ratings As mentioned in the previous section, when users share very few commonly rated items, weights generated using PCC will become very unreliable. For example, in Table 2, users *a* and *z* cast exactly the same ratings on 2 out of 5 items. However, user *z*'s rating should not be used with full confidence on prediction generation for user *a*, when comparing to users (e.g., *b*) who rate exactly the same as user *a* on more items.

Significance weighting method (Herlocker et al. 1999) only punishes users with fewer than 50 overlapping ratings with the active user and ignores the differences among others. A more general approach should take full consideration of the impact of the factor. Here, the rating overlap factor $\alpha^{o}_{a,x}$ for exponential treatment is defined as follows:

$$\alpha^{o}_{a,x} = \frac{MIN(|I_a|, |I_x|)}{|I_{a,x}|}, \qquad (13)$$

where I_a and I_x stand for the set of all items rated by the active user *a* and user *x*, respectively. $MIN(|I_a|, |I_x|)$, the smaller value of the total numbers of items rated by *a* and *x*, is used because it is the maximum possible number of commonly rated items by both users.

Evaluation

We evaluated the proposed factor-based CF approach by comparing its predictive performance with four existing approaches, namely traditional memory-based CF (baseline), case amplification, significance weighting, and z-score. We adopted a commonly used task for evaluation, where individual items are presented one at a time to the user along with a prediction generated by the system (Shardanand & Maes 1995). As to the evaluation metric, Mean Absolute Error (MAE) was used in the study. MAE has been used to measure predictive accuracy of a recommender system and is defined as the average absolute deviation of the predictions generated by a system (i.e., $p_{x,i}$) on how users would rate different items to the actual ratings (i.e., $r_{x,i}$) on those items cast by users (Breese et al. 1998; Herlocker et al. 2004). If $S = p_{x,i} - r_{x,i}$ is the error of each individual prediction, MAE can be calculated using the following formula:

$$|\bar{S}| = \frac{\sum_{i \in I_t} |p_{x,i} - r_{x,i}|}{|I_t|}, \qquad (14)$$

where I_t is the set of all items that are selected for evaluation, and $|I_t|$ the number of items in the set.

We developed a prototype memory-based CF system using the MovieLens dataset, which was made publicly available by the GroupLens research group at University of Minnesota. The dataset contains about 1 million ratings cast by 6,040 users on 3,706 movies. Ratings range from 1 to 5, with 1 indicating the least favorable and 5 indicating the most favorable. Each user rated at least 20 movies and all the ratings were submitted between April 2000 and March 2003. This dataset has been widely used in collaborative filtering research (e.g., (Herlocker et al. 1999; Calderon-Benavides et al. 2004)).

In this study, we used the entire dataset to evaluate the effectiveness of all three factors with our treatment. The evaluation was carried out in a number of rounds. In each round, like previous studies (Herlocker et al. 1999), we randomly selected 10% of all users from the dataset. Each selected user (different from round to round) was treated as the active user. We then randomly selected a rating of that user and 'hid' it from the system as the target item—the one going to be predicted by the system using the rest of the data. This method is also referred to as the "All-but-one" technique (Breese et al. 1998; Herlocker et al. 2004). An MAE value was computed for each round. We repeated such process for 100 rounds for each method tested. The same test process and user-item combination are used to measure the performance of CF methods enhanced by each of the three factors with the proposed exponential

treatment as well as that of four existing methods without such enhancement.

Although the choice of neighborhood selection method will affect the performance of the system, we believe an effective weight adjustment mechanism will be sufficient since a weight equal to almost zero will only have negligible effect on the prediction generation. Therefore, no neighborhood selection method was used in the experiment. In another word, every user was considered in the final prediction generation process as long as he/she had rated the target item.

Methods \ Results		MAE	Improvement over baseline	P-value
Baseline		0.7240		
Case Amplification		0.7232	0.117%	0.00**
Z-Score		0.7204	0.495%	0.00**
Sig. Weighting		0.7171	0.956%	0.00**
Rating Average	w/o Z	0.7216	0.343%	0.00**
	w/ Z	0.7177	0.876%	0.00**
Rating Variance	w/o Z	0.7211	0.407%	0.00**
	w/ Z	0.7178	0.859%	0.00**
Overlapping Ratings No.	w/o Z	0.7120	1.671%	0.00**
	w/ Z	**0.7029**	**2.924%**	**0.00****

** P<0.01

Table 3. Summary of MAE Results

We used the paired t-test to compare the mean difference in the final MAE results obtained by using different methods. A summary of the results is listed in Table 3.

Our finding on significance weighting shows that it outperforms baseline method, which confirms previous finding. However, our results show some differences from other studies. Case amplification (when $\rho = 1.5$) actually performed better than the baseline algorithm by a small margin. z-score also yielded improvement against the baseline method. The positive result from the z-score approach also motivated us to test effectiveness of the method by combining z-score with each of the three factors we proposed. Results showed that all combinations outperformed both standalone z-score approach and their pure factor-based counterparts.

In order to pick an optimal *n* value, different candidate values were tested. The result reveals that for the rating average factor, which has a relatively small value range ($\alpha^{\mu}_{a,x} \in [1,5]$), an *n* value that is larger than 1 (e.g., $n = 1.4$) can improve the performance. On the other hand, rating variance factor ($\alpha^{v}_{a,x} \in [1,117]$) and number of overlapping ratings factor ($\alpha^{o}_{a,x} \in [1,304]$) require *n* values that are less than 1 (e.g., $n = 0.9$ and 0.6, respectively) in order to achieve better results. As noted earlier, the most effective *n* should be the one that effectively adjusts the factor $\alpha_{a,x}$ so that, on average, the scaling exponent in the power function has a shape not too flat (when the scaling exponent is close to 1) and not too tilted towards the corner (when the scaling exponent is too large).

During the evaluation, we found that we achieved better results when λ was not 1. This is because when λ equals to 1, $\alpha_{a,x}$ will lose its effect on those weights $w_{a,x}$ that are equal to 1. However, users with their weights $w_{a,x}$ associated with the active user equal to 1 have the largest influence on the final prediction result. Therefore, a λ less than 1 can eliminate the problem.

Although system performance depends on the choice of λ, we observed that the performance variations are relatively stable as the value of λ changes in its neighborhood. Theoretical analysis in the previous section did suggest a way to find an optimal value of λ, upon effectively identifying a representative $\alpha_{a,x}$. We use the following empirical method to choose the representative $\alpha_{a,x}$ and determine the corresponding optimal value for λ based on the one-to-one correspondence relationship in Equation (10).

First, we recorded all the values of $\alpha_{a,x}$ for each pair of users. It turned out that the distributions of $\alpha_{a,x}$ among user pairs for different factors share similar patterns. For all three factors, more than 90% of the user pairs have $\alpha_{a,x}$ values lower than one tenth of its maximum value listed earlier. What value of $\alpha_{a,x}$ will be representative is not clear-cut. For each factor, we picked a series of $\alpha_{a,x}$ spread out in its whole range to calculate λ using Equation (10). In our experiment, those optimal λ's all appeared when $\alpha_{a,x}$ values were within the range of 1 to one tenth of the maximum $\alpha_{a,x}$ value, where many similar values of $\alpha_{a,x}$ clustered together. This is because, as we believe, those optimal λ values represent the majority of the population for different factors in the system. We find that the system performance variation is not too sensitive to the exact value of $\alpha_{a,x}$ or, accordingly, the choice of λ as long as $\alpha_{a,x}$ is within the high density region. Therefore, as our empirical recommendation, a representative $\alpha_{a,x}$ could be chosen in the region with the highest density. For each factor, we recorded the results with the highest accuracy in Table 3.

Among the proposed factors, rating average had the smallest improvement margin, while overlap rating factor combined with z-score approach resulted in the largest improvement compared with the baseline.

Discussion

Similarity measurement is arguably the most important part of collaborative filtering systems. In this study, we aimed to develop methods that measure user preference similarity in CF based on different aspects of user rating patterns. The result shows that the proposed factor-based approach significantly outperformed existing memory-based CF methods. The exponential treatment is also proved to be an effective approach to integrate additional

factors into user similarity measurement. This is because the exponential treatment is able to set apart users that are different from the active user on their rating patterns using different factors. More importantly, our proposed exponential treatment is more effective than traditional methods in differentiating users whose weights are close to 1 and identifying genuinely close users who play critical roles in final prediction generation.

Among the three proposed factors, the one based on the number of overlapping ratings performed the best and it outperformed significance weighting approach, which focuses on the same aspect of user rating pattern, both with or without z-score combined. Rating variance based factor came in second, followed by the factor based on rating average. It is not surprising that the number of overlapping ratings factor performed the best, because with few overlapped ratings, similarity measures computed by PCC can be very unreliable and biased. The fact that the number of overlapping ratings factor has the widest range of values (from 1 to 304) shows that it has higher power in distinguishing users than the other two factors. The result also shows that the case amplification method can only slightly improve the performance of CF systems without taking into account other similarity features from users. Furthermore, different from the arguments in (Herlocker et al. 1999), we find that the variance of user ratings indeed has effect on prediction generation and can be used to improve predictive accuracy of the system. Both z-score approach and rating variance based factor treatment performed better than the original CF method. Additionally, these two approaches were also found to be complementary to each other in improving prediction accuracy.

Although our proposed approach demonstrates superior results than existing methods, there are several extensions that can be pursued in the future. A natural next step is to design methods for combining the three proposed factors. Our initial tests using a naïve integration technique yielded good performance improvement. However, we believe by using more advanced ways to combine factors the system has the potential to improve accuracy even further.

Secondly, we plan to use different datasets, tasks, and metrics to further investigate the performance of the system. The MovieLens dataset, one of the most popular dataset used in CF research, was used in this study. We plan to confirm the results using other datasets that are available to the public, such as EachMovie and Book-Crossing dataset. In order to focus on the accuracy of the system, this study employed a very simple task. We can adopt other tasks, like "find all good items" or "recommend sequence" (Herlocker et al. 2004) to evaluate the factor-based approach. Similarly, we expect other metrics, such as ROC curves, can offer us some different insights as to the effectiveness of the method.

Sparsity of the user-item rating matrix is an inherent problem in CF research and has huge effect on the performance of a CF system. Therefore, we plan to further evaluate the factor-based approach using rating matrices at different sparsity levels. We will test how sensitive the system performance is to different user-item matrices with various sparsity levels.

References

Breese, J. S., Heckerman, D. and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Fourteenth Conference on Uncertainty in Artificial Intelligence* Cooper, G. F. and Moral, S., Eds, pp 43-52. University of Wisconsin Business School, Madison, Wisconsin, USA: Morgan Kaufmann.

Calderon-Benavides, M. L., Gonzalez-Caro, C. N., Perez-Alcazar, J. d. J., Garcia-Diaz, J. C. and Delgado, J. 2004. A comparison of several predictive algorithms for collaborative filtering on multi-valued ratings. In *Proceedings of the 2004 ACM symposium on Applied computing*, pp 1033-1039. Nicosia, Cyprus: ACM Press.

Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international conference on Research and development in information retrieval*, pp 230-237. Berkeley, California, United States: ACM Press.

Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22 (1): 5-53.

Hill, W., Stead, L., Rosenstein, M. and Furnas, G. 1995. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp 194-201. Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co.

O'Brien, J. M. 2006. You're soooooooooo predictable. *Fortune*, 154 (11): p 224.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp 175-186. Chapel Hill, North Carolina, United States: ACM Press.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. 2000. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pp 158-167. Minneapolis, Minnesota, United States: ACM Press.

Shardanand, U. and Maes, P. 1995. Social information filtering: Algorithms for automating "Word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp 210-217. Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co.