User Preference Representation Based on Psychometric Models

Biyun Hu[1], Zhoujun Li[1], Wenhan Chao[1], Xia Hu[2] and Jun Wang[1]

[1]School of Computer Science and Engineering, Beihang University of China
XueYuan Road No.37, HaiDian District, Beijing 100191, China
[2] Ira A. Fulton School of Engineering, Arizona State University, USA
Tempe, AZ 85281 USA

byhu8210@yahoo.com.cn, lizj@buaa.edu.cn, cwh2k@163.com, huxia001@gmail.com,
wangjun0706149@cse.buaa.edu.cn

Abstract

Neighbourhood-based collaborative filtering is one of the most popular recommendation techniques, and has been applied successfully in various fields. User ratings are often used by neighbourhood-based collaborative filtering to compute the similarity between two users or items, but, user ratings may not always be representatives of their true preferences, resulting in unreliable similarity information and poor recommendation. To solve these problems, this paper proposes to use latent preferences for neighbourhood-based collaborative filtering instead of user ratings. Latent preferences are based on user latent interest estimated from ratings through a psychometric model. Experimental results show that latent preferences can improve the recommendation accuracy and coverage while lessening the prediction time of neighbourhood-based collaborative filtering by finding out reliable and effective neighbours; and latent preferences are better than user ratings for representing user preferences.

Keywords: User Rating, True Preference, Rating Residual, Latent Preference, Psychometric Model, Collaborative Filtering.

1 Introduction

Collaborative Filtering (CF) is a popular technique used to help recommendation system users find out the most valuable information based on their past preferences. These preferences can be explicitly obtained by recording the *ratings* that *users* have awarded on *items*, such as albums, movies, and books. CF algorithms can be mainly divided into three categories: model-based, neighbourhood-based and hybrid ones (Adomavicius and Tuzhilin 2005). Model-based approaches first learn a model from history dataset. The model is then used for recommending. A lot of machine learning algorithms and statistical techniques have been used to learn the model, such as probabilistic latent semantic analysis (Hofmann 2003), latent dirichlet allocation (Marlin 2003), matrix factorization (Ma, Yang, Lyu, and King 2008, Salakhutdinov and Mnih 2008), and clustering (Xue, Lin, Yang, Xi, Zeng, Yu, and Chen 2005). Although many model-based algorithms have been proposed, it seems that in real applications, neighbourhood-based algorithms are more popular used (Koren 2008), such as Amazon (Linden, Smith and York 2003) and TiVo (Ali and Van Stam 2004). These algorithms look into the similarity between users or items, and then use these relationships to make recommendations (Konstan, Miller, Maltz, Herlocker, Gordon, and Riedl 1997, Sarwar, Karypis, Konstan, and Riedl 2001, Linden, Smith and York 2003). However, user ratings may be deviated from true preferences by reasons such as wrong usage of a rating scale or type errors, resulting in unreliable similarity information and further causing poor prediction.

To overcome these drawbacks, the paper proposes to substitute latent preferences for user ratings to make recommendations. Latent preferences are computed based on user latent interest, which is estimated from user ratings through a psychometric model.

The rest of the paper is organized as follows: Section 2 provides a brief review of neighbourhood-based collaborative filtering. Section 3 analyses how user ratings may cause poor recommendation. The proposed preference representation is presented in Section 4. Experimental results are reported in Section 5 and discussed is Section 6. Finally, we conclude the paper and give future work.

2 Neighbourhood-based Collaborative Filtering

Neighbourhood-based CF algorithms can be further divided into two categories: user-based CF and item-based CF. The two often contain the following three steps:

Similarity weighting: For user-based CF, the similarity between two users is often computed based on the items co-rated by the two (*co-rated items*), and Pearson correlation coefficient is widely used. For item-based CF, the similarity between two items is usually evaluated based on the users who have co-rated the two (*co-rate users*), and adjusted cosine similarity is found best to compute the similarity (compared with cosine similarity and Pearson correlation coefficient (Sarwar, Karypis, Konstan, and Riedl 2001)).

Neighbour selection: This step requires that a number of neighbours of the *active user* (for user-based CF) or the *target item* (for item-based CF) be selected (the active user is the user whom the recommendations are for, and the target item is the unrated item for which a rating need to be predicted). These selected neighbours have the highest similarity weights. Noteworthy, not all neighbours chosen

Copyright (c)2011, Australian Computer Society, Inc. This paper appeared at the 22nd Australasian Database Conference (ADC 2011), Perth, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 115. H. T. Shen and Y. Zhang, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

are *effective neighbours* (they are the neighbours actually used in the following prediction step). For example, in Formula (1), user neighbour u_n is effective only when his/her rating for target item i $r_{u_n,i}$ is not missing.

Prediction: Predictions are often given as the weighted combination of neighbour ratings. For example, for user-based CF, the prediction is usually computed as Formula (1), where $p_{a,i}$ is the predicted rating for the active user a on the target item i, $\overline{r_a}$ active user a's average rating; $r_{u_n,i}$ the rating awarded to item i by active user a's neighbour u_n; $sim(a,u_n)$ the similarity between active user a and his/her neighbour u_n; and k the number of neighbours.

$$p_{a,i} = \overline{r_a} + \frac{\sum_{n=1}^{k} sim(a,u_n)(r_{u_n,i} - \overline{r_{u_n}})}{\sum_{n=1}^{k} |sim(a,u_n)|} \qquad (1)$$

3 User Rating and Rating Residual

User ratings are often used by neighbourhood-based CF, but those ratings may not always be representatives of user true preferences. Users may award random ratings to the items they don't care about, they may make type errors, and they may wrongly apply the rating scale used by a system. All these and other possible disturbances may deviate user ratings from user true preferences, causing *rating residual* (the difference between user ratings and their true preferences).

3.1 Assumptions

Ratings with residual can influence the neighbourhood-based CF. For ease of analysis, the following assumptions are first made. These assumptions are validated in the experiments in subsection 6.1.

1. Assumption 1. Two users who have co-rated more items tend to be more similar.
2. Assumption 2. It is likely that two items co-rated by more users are more similar.

Based on the assumptions, the following inferences can be drawn.

1. Inference 1. Less similar users are prone to co-rate fewer items.
2. Inference 2. Probably, less similar items are co-rated by fewer users.

3.2 Effects of Ratings with Residual

3.2.1 Effects on Similarity Weighting

Ratings with residual have two following negative effects on similarity weighting:

1. Negative effect 1. Ratings with residual can make less similar users/items become more similar.
2. Negative effect 2. Ratings with residual can make more similar users/items become less similar.

For example, as Table 1 shows, for a rating scale of 1-5, user u_1 is an ideal scorer and her rating r_{u_1} represents her true preference t_{u_1}, while u_2 is a more severe rater whose ratings are all chosen from the wrong rating scale 1-3. If u_2 has used the rating scale 1-5 correctly, then his true preferences may be the ratings given in the fourth row of Table 1. When using true preferences, the two users are more similar (the Pearson coefficient is 0.4), but when ratings are utilized, the two users are less similar (the Pearson coefficient is 0.2), that is, because of ratings with residual, more similar users u_1 and u_2 have become less similar. Similarly, user u_3's true preferences are given in the last row of Table 1, but when u_3 is rating, u_3 has a rating residual of 1 or -1, then u_3's observed ratings are presented in the fifth row of Table 1. When using true preferences, u_1 and u_3 are less similar (the Pearson coefficient is -0.1) , but when observed ratings are used, the two become more similar (the coefficient is 0.1).

3.2.2 Effects on Neighbour Selection

In the neighbour selection step, the two negative effects on similarity weighting work together to promote neighbourhood-based CF algorithms to use *unreliable neighbours* (actually less similar users/items). Considering the Inference 1 or 2, chances that these unreliable neighbours are not effective. For example: as Table 2 shows, because of ratings with residual, less similar user u_2 is selected as a neighbour of user u_1 , and now user-based CF algorithm needs to predict the rating that user u_1 will award to item i_4. Because u_1 and u_2 are less similar, according to Inference 1, it is likely that u_2 hasn't rated i_4 too, that is, u_2 is an invalid neighbour.

	i_1	i_2	i_3	i_4	i_5
r_{u_1}/t_{u_1}	1	5	3	4	2
r_{u_2}	2	3	1	2	3
t_{u_2}	3	5	2	3	4
r_{u_3}	1	2	4	2	3
t_{u_3}	2	3	5	1	4

Table 1: User *u*'s rating (*r*) and true preference (*t*) for item *i*.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	null	*	*	?	*	null	*
u_2	*	*	null	**null**	null	*	null

Table 2: u_2 is an unreliable neighbour of u_1, chances that u_2 is also an invalid neighbour. * is a rating, and *null* denotes a missing rating.

3.2.3 Effects on Prediction

Unreliable neighbours used would result in poor recommendation accuracy. Ineffective neighbours used would cause low recommendation coverage, and force neighbourhood-based CF algorithms to choose more neighbours for predicting, leading to increased *prediction time* (the computation time of Equation (1)).

In summary, the analysis above is illustrated in Figure 1.

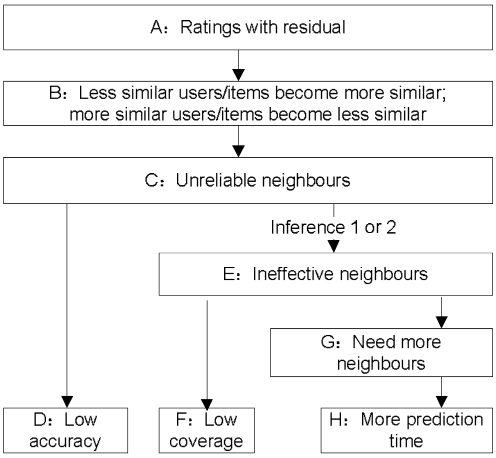

Figure 1: The negative effects of ratings with residual on neighbourhood-based collaborative filtering.

4 Latent Preference Representation Based on Psychometric Models

The analysis in previous section shows that user ratings are prone to have rating residual, and ratings with residual have negative effects on recommendation accuracy, coverage and prediction time. Naturally, we want to find a better representation of user preferences. We propose to substitute latent preferences for user ratings. This section first introduces psychometric models and their application in recommendation systems, and then parameter estimation methods for these models are sketched. Finally, latent preference is defined.

4.1 Psychometric Models

In psychometrics, latent trait models also known as item response theory models, are a series of mathematical models applied to data from tests or questionnaires for measuring latent traits, such as abilities, interests or attitudes. For ease of understanding, first consider the Rasch model (Rasch 1960), which can be expressed as Formula (2) or (3),

$$\log(p(r_{u,i}=1)/p(r_{u,i}=0))=q_u-b_i \tag{2}$$

$$p(r_{u,i}=1)=\frac{e^{q_u-b_i}}{1+e^{q_u-b_i}} \tag{3}$$

Rasch model was originally used in educational tests. In this typical application, $p(r_{u,i}=1)$ is the probability that student *u* will succeed on question *i*; q_u parameters are a linear continuous measure of student *ability*; and b_i parameters are a linear continuous measure of question *difficulty*. Bigger q_u values identify more able students, and bigger b_i values identify more difficult items. When a student *u*'s ability q_u is equal to the difficulty b_i of a question *i*, the student has a probability of 0.5 to answer the question correctly, considering the error in the response process. The more the student's ability is larger than the difficulty of the question, the more probable that he/she will succeed on the question.

Psychologists have extended the Rasch model because it can only handle binary scores (e.g. right or wrong, usually coded 1 or 0). A typical extended model, rating scale model (Andrich 1978), can be expressed as Formula (4) or (5),

$$\log(p(r_{u,i}=x)/p(r_{u,i}=x-1))=q_u-b_i-t_x \tag{4}$$

$$p(r_{u,i}=x)=\frac{e^{k_x+x(q_u-b_i)}}{1+\sum_{k=1}^{m}e^{k_k+k(q_u-b_i)}} \tag{5}$$

where *x* is a rating taken from successive rating categories set $\{0, 1, 2, \ldots, m\}$; $p(r_{u,i}=x)$ is the probability of observing rating *x* for person *u* encountering item *i*; t_x the ordered thresholds denoting the difficulty of being observed in rating *x* relative to rating *x*-1; and $k_0=0$, $k_x=-\sum_{k=1}^{x}t_k$, *x* = 1, 2, …, *m*-1, $k_m=0$ the category coefficients expressed in terms of the ordered thresholds t_1, t_2, …, t_m.

Since proposed, these psychometric models have been applied successfully in the analysis of educational tests, attitude surveys and other rated assessments.

4.2 Psychometric Models and Collaborative Filtering

In previous work (Hu, Li, and Wang 2010), psychometric models have been used successfully to solve the sparsity problem of traditional neighbourhood-based CF algorithms. This paper differentiates from the previous work in that it focuses on presenting a better representation of user preferences, and further discusses the benefits of the representation for existing CF algorithms. While better CF algorithms are needed, the quality of user preferences is also important and needs to be researched, because only when accurate user preference information can be obtained, can CF algorithms make precise recommendations.

In recommendation system application, the parameters in Rasch model have a different meaning and reading. According to the correspondence made by Battisti, Nicolini, and Salini (2005), who apply Rasch model to measure service quality, we have made a similar correspondence as shown in Figure 2. The factor related to the students that in educational test was the ability (q_u) becomes now the *interest*. The factor related to the questions that was the difficulty (b_i), in recommendation system application becomes the *agreeability*. Bigger q_u values identify more interested users. Noteworthy, *larger* b_i values identify *less* agreeable items. Take movie recommendation as an example, intuitively, it is probable that only people who are very interested in movie will show positive response for a film with little agreeability, on the contrary, it is unlikely that people who are not interested in movie will like an agreeable film.

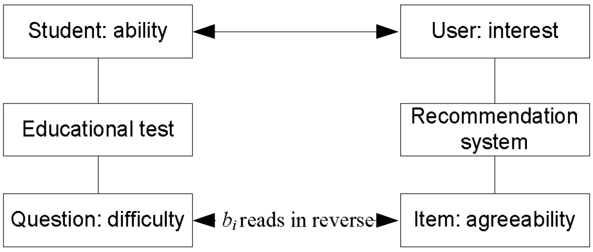

Figure 2: Correspondence between educational test and recommendation system application of Rasch model.

4.3 Parameter Estimation

Basic techniques for estimating these psychometric models include joint maximum likelihood, conditional maximum likelihood, marginal maximum likelihood, and Bayesian estimation with Markov chain Monte Carlo (Johnson 2007). In this paper, we have used the Winsteps Rasch measurement computer program for parameter estimation (Linacre 2007). In Winsteps, initially, the user interest q_u, item agreeability b_i, and threshold t_x are all estimated to be 0, and then the PROX (normal approximation) estimation algorithm is used for the first phase of estimation. This produces revised estimates derived by Linacre (1995):

$$q_u = m_u + \sqrt{1 + s_u^2 / 2.9}\,\log(R_u /(N_u - R_u)) \qquad (6)$$

where q_u is the revised interest estimate for user u; m_u the mean agreeability of the items rated by user u; s_u the standard variance of those item agreeability; R_u the total rating of user *u* (sum across all items rated by user *u*); and N_u the maximum possible total rating on those same items (the maximum rating category *m* * the number of items rated by user *u*). Similarly, for the item agreeability parameters,

$$b_i = m_i - \sqrt{1 + s_i^2 / 2.9}\,\log(R_i /(N_i - R_i)) \qquad (7)$$

where b_i is the revised agreeability estimate for item *i*; m_i the mean interest of the users who have rated item *i*; s_i the standard variance of those user interest; R_i the total rating of item *i* (sum across all users who have rated item *i*); and N_i the maximum possible total rating by those same users (the maximum rating category *m* * the number of users who have rated item *i*). Winsteps iterates on the user ratings and updates these PROX estimates until the increment of user interest or item agreeability is small or maximum PROX iterations are reached. Initial estimates of the threshold between rating category *x* and *x*-1 are:

$$t_x = \log(N_{x-1} / N_x) \qquad (8)$$

where N_{x-1} is the number of rating *x*-1 in the data. Winsteps takes the PROX estimates and uses JMLE (Joint Maximum Likelihood Estimation) for the second phrase of estimation. First, the expected total ratings for users and items are computed and compared with those observed raw total ratings, and then estimates are revised. For example, if a user's expected total rating is less than that user's observed raw total rating, then the ability estimate raised. Concrete estimation equations for JMLE are derived by Wright and Masters (1982).

4.4 Latent Preference

Just like students' scores are decided by their ability (but may be distorted by reasons such as raters with different severity), user preferences are decided by the latent interest of users, therefore, obtaining the user latent interest is the key to a better representation of user preferences. In this paper, we first infer user latent interest through psychometric models, and then compute user ***u***'s preference for item ***i*** based on user ***u***'s latent interest, we name the new preference information $lp_{u,i}$ as *latent preference*, and define it as Formula (9). Compared with user ratings $r_{u,i}$, latent preferences are decided by latent interest and free from rating residual, thus, latent preferences may be better to represent user preferences.

$$lp_{u,i} = \sum\nolimits_{x=1}^{m} x(p(r_{u,i} = x)) \qquad (9)$$

5 Experiments

5.1 MovieLens Dataset

The MovieLens dataset provided by the GroupLens Research Project[1] is used in the experiments. The dataset contains 100,000 ratings of approximately 1,682 movies made by 943 users. Ratings are discrete values from 1 to 5 (a rating scale of 1-5). Each user has at least 20 ratings. The sparsity level of the dataset is 0.9369. As the paper (Sarwar, Karypis, Konstan, and Riedl 2001) does, 80% of the dataset was randomly selected into a training set and the remaining into a test set.

5.2 Metrics

The following two recommendation quality metrics are reported in this paper.

Mean Absolute Error (MAE). It corresponds to the average absolute deviation of predictions to the actual ratings in the test set, as shown in Equation (10), where $p_{u,i}$ is the predicted rating for user ***u*** on item ***i***; and $r_{u,i}$ the tested rating. A smaller MAE value indicates a better accuracy. MAE is one of the most often used metrics, because most research has focused on improving the accuracy of recommendations (Herlocker, Konstan, Terveen, and Riedl 2004).

$$MAE = avg\,|\,p_{u,i} - r_{u,i}\,| \qquad (10)$$

Coverage. Recommendation coverage is less investigated than accuracy; however, it is an important metric, because systems with lower coverage may be less valuable to users (Herlocker, Konstan, Terveen, and Riedl 2004). As Equation (11) shows, the coverage is the ratio of predicted ratings to all the ratings in the test set.

$$\mathrm{cov}\,erge = the\_number\_of\_p_{u,i} / N \qquad (11)$$

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

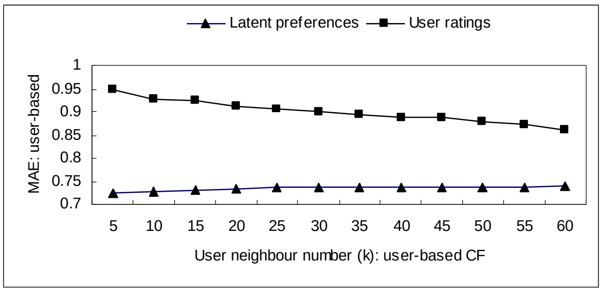

Figure 3.a: Using latent preferences and user ratings respectively, the recommendation *accuracy* of the *user-based* collaborative filtering algorithm with different neighbour numbers.

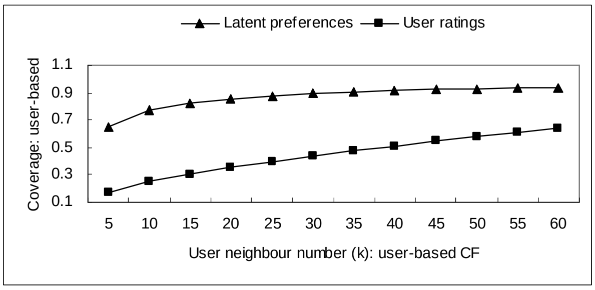

Figure 3.b: Using latent preferences and user ratings respectively, the recommendation *coverage* of the *user-based* collaborative filtering algorithm with different neighbour numbers.

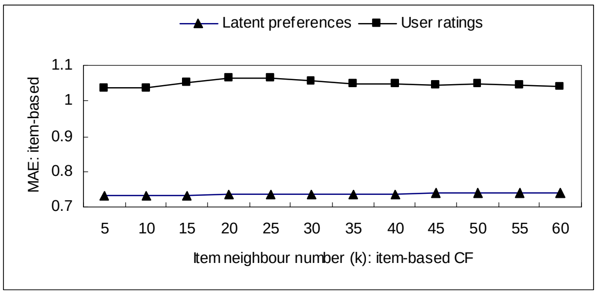

Figure 4.a: Using latent preferences and user ratings respectively, the recommendation *accuracy* of the *item-based* collaborative filtering algorithm with different neighbour numbers.

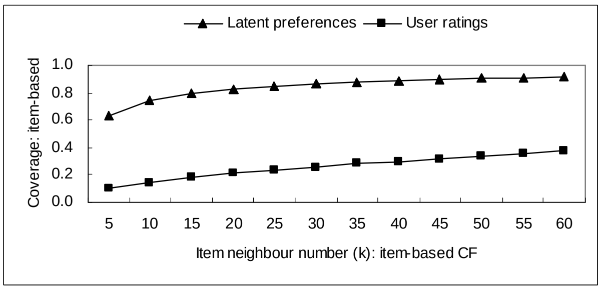

Figure 4.b: Using latent preferences and user ratings respectively, the recommendation *coverage* of the *item-based* collaborative filtering algorithm with different neighbour numbers.

5.3 Procedures

First, user latent interest was estimated from the training set through the rating scale model (ref. subsection 4.3). Next, for each rating in the training set, corresponding latent preference was computed (ref. subsection 4.4). Then, based on the new training set formed by latent preferences, the user-based CF and item-based CF algorithms were used respectively to make predictions for the test ratings. Finally, the prediction results were compared with that obtained using the original training set composed of user ratings.

5.4 Results

5.4.1 Recommendation Accuracy and Coverage

Using latent preferences and user ratings respectively, the recommendation accuracy and coverage of the user-based CF algorithm are reported in Figure 3.a and Figure 3.b, from which we can see that, latent preferences can improve the recommendation accuracy and coverage of the user-based CF algorithm. The accuracy ascends by 23.4% when k is set 5 (MAE decreases from 0.947 to 0.725); and the coverage increases by 209% when k set 10 (coverage increases from 0.25 to 0.772).

Using latent preferences and user ratings respectively, the recommendation accuracy and coverage of the item-based CF algorithm are reported in Figure 4.a and Figure 4.b, which show that, latent preferences can improve the recommendation accuracy and coverage of the item-based CF algorithm. The accuracy increases by 31% when k is set 20 (MAE decreases from 1.064 to 0.734); and the coverage ascends by 335% when k set 15 (coverage increases from 0.184 to 0.8).

5.4.2 Neighbour Number and Prediction Time

Neighbour number and accuracy: As Figures 3.a and 4.a show, compared with using user ratings, when latent preferences are employed, the two neighbourhood-based CF algorithms can get much better recommendation accuracy with only 5 neighbours.

Neighbour number and coverage: As can be seen from Figure 3.b, using latent preferences and 5 neighbours, the recommendation coverage of the user-based CF algorithm (0.646) is even higher than that obtained using 60 neighbours and user ratings (0.639). As Figure 4.b shows, using latent preferences and 5 neighbours, the item-based CF algorithm receives significantly better recommendation coverage (0.628) than that got using 60 neighbours and user ratings (0.38).

These results above show that latent preferences enable neighbourhood-based CF algorithms to get a better recommendation quality with fewer neighbours, so latent preferences can reduce the prediction time of these algorithms.

5.4.3 The Change Trend of Accuracy

As Figure 3.a and Figure 4.a show, when using latent preferences, the recommendation accuracy of the user-based and item-based CF algorithms drops slightly with the increasing of the neighbour number *k*; while using user ratings, the recommendation accuracy ascends

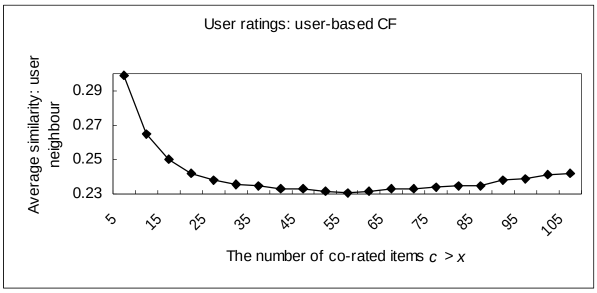

Figure 5.a: Using *user ratings*, when *c* > 55, the more items two users have co-rated, the more similar the two users tend to be.

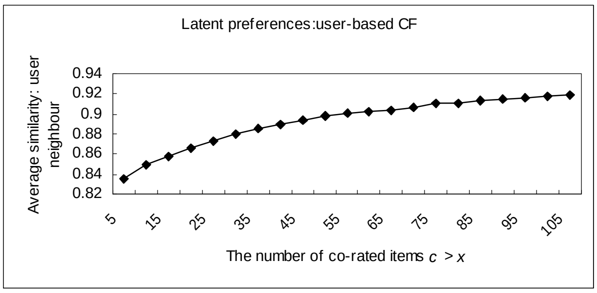

Figure 6.a: Using *latent preferences*, the more items two users have co-rated, the more similar the two users tend to be.

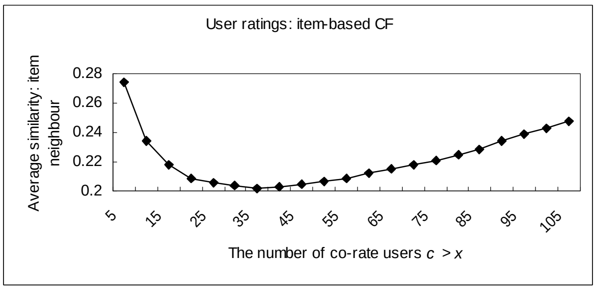

Figure 5.b: Using *user ratings*, when *c* > 35, it is likely that two items co-rated by more users are more similar.

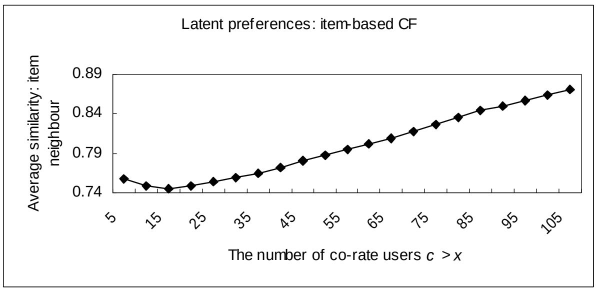

Figure 6.b: Using *latent preferences*, when *c* > 15, it is likely that two items co-rated by more users are more similar.

	Accuracy Increases	Coverage Increases	Time reduced	*k* increases
User -based	23.4%	209%	Better quality with only 5 neighbours	Accuracy slightly decreases
Item -based	31%	335%		

Table 3: The benefits of using latent preferences for neighbourhood-based collaborative filtering, and the influence of the neighbour number *k*.

(user-based) or nearly ascends (item-based) with *k* increases.

In summary, the experimental results reported above are summarized in Table 3.

6 Discussion of Experiments

6.1 Neighbour Reliability and Recommendation Accuracy

Recommendation accuracy is mainly related to the reliability of neighbours (whether those neighbours used are *really* similar); therefore, we can speculate that latent preferences enable neighbourhood-based CF algorithms to find out more reliable neighbours. To validate this, the average similarity values for neighbours based on different numbers of co-rated items for user-based CF or co-rate users for item-based CF ($c > x$, x was set 5, 10, 15,…, 105 respectively) are computed and reported in Figures 5.a, 5.b, 6.a, and 6.b. Those neighbours with similarity values smaller than 0 are omitted, because usually these neighbours are not used (rank behind in a neighbour list). These figures show the followings:

Assumptions verification: Using user ratings (Figures 5.a and 5.b), more reliable similarity values based on $c > 55$ for user-based CF and $c > 35$ for item-based CF confirm the Assumptions 1 and 2 respectively. The more items two users have co-rated, the more similar the two users tend to be (Assumption 1). It is likely that two items co-rated by more users are more similar (Assumption 2).

Latent preferences vs. user ratings: When using latent preferences (Figures 6.a and 6.b), the average similarity values computed by user-based CF conform to Assumption 1 exactly, and those by item-based CF almost comply with Assumption (2). For item-based CF, when $c \leq 15$ (Figure 6.b), it is likely that those unreliable similarity values arise from the data sparsity problem, because for one reason, data sparsity can also cause unreliable similarity information (Bobadilla and Serradilla 2009), and for another, the data set used in the experiments is more sparse for item-based CF than for user-based CF. There are at least 20 ratings for each user, but there is no such a restriction for each item. When using user ratings, the problems are the followings. *Problem 1:* Less similar users/items have become more similar. For example, for user-based CF, when *x* *decreases* from 55 to 5, using latent preferences (Figure 6.a), those related users become less and less similar; but when user ratings are used ((Figure 5.a), these users become more and more similar. *Problem 2:* More similar users/items have become less similar. For example, for item-based CF, when *x* increases from 15 to 35, using latent preferences (Figure 6.b), those related items become more and more similar; but when user ratings are used ((Figure 5.b), these items become less and less similar. The two problems are exactly the negative effects of ratings with residual on similarity weighting analysed in subsection 3.2.1 (Figure 1: A->B). These problems will promote neighbourhood-based CF algorithms to choose unreliable neighbours (Figure 1: B->C). For example, for user-based CF, using user ratings (Figure 5.a), when *x* = 105, although those related neighbours are more reliable, their average similarity value is smaller than that of those less reliable neighbours with 15 or fewer co-rated items (*c* ≤ 15), so these less reliable neighbours will first be chosen from by user-based CF, causing low recommendation accuracy (Figure 1: C->D). From the analysis above, we conclude that, compared with user ratings, latent preferences can make the relationships between two users/items become more clear and reliable.

Latent preferences enable more reliable neighbours. This can also be drawn from the change trend of the recommendation accuracy. When using latent preferences, the recommendation accuracy drops slightly with the increasing of the neighbour number (Figures 3.a and 4.a). This is an intuitive result. First, as the neighbour number increases, more less similar neighbours will be used, so the recommendation accuracy drops. Second, when reliable neighbours are used, the number of neighbours will not make much difference in the recommendation accuracy. For example, user u_1 has two neighbours u_2 and u_3, and the two neighbours all like item i_1. No matter one neighbour or two neighbours are used, the prediction result of user-based CF for user u_1 on item i_1 is the same, that is, user u_1 will like item i_1. When using user ratings, the recommendation accuracy increases or nearly increases as the neighbour number ascends (Figures 3.a and 4.a). This is counter-intuitive. More less similar neighbours are used, but the recommendation accuracy increases. We think that this phenomenon happens because that, as the neighbour number increases, more reliable neighbours will balance the negative effects of less reliable neighbours. For example, for item-based CF (Figure 5.b), first, those less reliable neighbours with average similarity value 0.274 (*x* = 5) will be used, as the neighbour number increases, more reliable neighbours with average similarity value 0.248 (*x* = 105) will be chosen. In this process, the negative effects of those less reliable neighbours may be balanced.

From the analysis above, we conclude that: latent preferences are better representatives of user preferences than user ratings, and they enable neighbourhood-based collaborative filtering algorithms to find out more reliable neighbours, thus can improve the recommendation accuracy of these algorithms.

6.2 Neighbour Effectiveness and Coverage and Prediction Time

In previous subsection, it is concluded that using latent preferences, more reliable neighbours can be found. Chances that these more reliable neighbours are more effective, thus they can improve the recommendation coverage and lessen the prediction time (it can be analysed similarly as done in subsections 3.2.2 and 3.2.3). For further validation, the average effective ratios of those neighbours for predicting all the test ratings are computed and reported in Table 4. For predicting a specific test rating, the effective ratio is computed as: (*the number of effective neighbours* / *k*). The results change only a little when we vary the neighbour number *k* from 5 to 30, so only the results with *k* set 5 are reported. As can be seen from Table 4, the average effective ratios of more reliable neighbours found by using latent preferences are much higher than those of less reliable neighbours found by using user ratings. When using latent preferences, for the first 5 neighbours used, averagely, one neighbour is effective for user-based or item-based CF algorithm; but when user ratings are used, the first 5 neighbours are nearly invalid (this shows the negative effects ratings with residual on neighbour selection (Figure 1: C->E), and this will further cause low recommendation coverage and long prediction time (Figure 1: A->B->C->E->F and A->B->C->E->G->H)).

Latent preferences		**User ratings**	
User-based CF	Item-based CF	User-based CF	Item-based CF
0.25	**0.24**	0.04	0.02

Table 4: The *average effective ratio*s of neighbours found by using latent preferences and user ratings when the neighbour number *k* is set 5.

From the analysis above, we conclude that: by finding out more effective neighbours, latent preferences can improve the recommendation coverage of neighbourhood-based collaborative filtering algorithms. This can further reduce the prediction time of these algorithms because fewer neighbours are needed.

7 Conclusions

The contributions of the paper include the followings. First, a theoretical analysis of the negative effects of using user ratings on the neighbourhood-based collaborative filtering is presented. Second, a new preference representation method, latent preference, is proposed. Third, experimental results have shown that latent preferences can improve the recommendation accuracy and coverage while lessening the prediction time of neighbourhood-based collaborative filtering algorithms by finding out reliable and effective neighbours. Fourth, experimental results have manifested the negative effects of using user ratings presented in the theoretical analysis. In conclusion, theoretical and experimental analysis has shown that latent preferences are better representatives of user preferences than user ratings.

In future work, we will further investigate whether latent preferences can improve the recommendation quality of model-based collaborative filtering algorithms.

Acknowledgements

Authors acknowledge Shyong Lam and Jon Herlocker for cleaning up and generating the MovieLens dataset, and John Michael Linacre for providing Ministep student version of Winsteps.

8 References

Adomavicius, G. and Tuzhilin, A. (2005): Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowledge and Data Engineering*, **17**(6):734-749.

Ali K. and Van Stam W. (2004): TiVo: Making Show Recommendations Using a Distributed Collaborative

Filtering Architecture. *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, 394–401, ACM Press.

Andrich, D. (1978): A rating formulation for ordered response categories. *Psychometrikia*, **43**: 561-573.

Battisti, F.D., Nicolini, G. and Salini, S. (2005): The Rasch model to measure the service quality. *The Journal of Services Marketing*, **3**: 58-80.

Bobadilla, J. and Serradilla, F. (2009): The effect of sparsity on collaborative filtering metrics. *Proc. Twentieth Australasian Database Conference*, Wellington, New Zealand, CRPIT **92**:9-17.

Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. (2004): Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, **22**: 5-53.

Hofmann, T. (2003): Collaborative filtering via Gaussian probabilistic latent semantic analysis. *Proc. 26th Annual International ACM* SIGIR *Conference*, New York, USA, 259-266, ACM Press.

Hu, B., Li, Z., and Wang, J. (2010): User’s latent interest-based collaborative filtering: *Proc. 32nd European Conference on Information Retrieval*, Milton Keynes, UK, LNCS **5993**: 619-622, Springer-Verlag.

Johnson, M.S. (2007): Marginal maximum likelihood estimation of item response models in R. *Journal of Statistical Software*, **20**(10): 1-24.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., and Riedl, J. (1997): GroupLens: applying collaborative filtering to usenet news. *Communications of the ACM*, **40**(3):77-87.

Koren, Y. (2008): Factorization meets the neighborhood: a multifaceted collaborative filtering model, *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, 426-434, ACM Press.

Linacre, J. M. (2007): WINSTEPS Rasch measurement computer program. Chicago: Winsteps.com.

Linacre, J.M. (1995): PROX for polytomous data. *Rasch Measurement Transactions*, **8**(4): 400

Linden, G., Smith, B. and York, J. (2003): Amazon.com recommendations: item-to-item collaborative filtering, *IEEE Internet Computing*, **7**:76-80.

Ma, H., Yang, H., Lyu, M. R., and King, I. (2008): SoRec: social recommendation using probabilistic matrix factorization. *Proc. 17th ACM conference on Information and knowledge management*, Napa Valley, California, 931-940, ACM Press.

Marlin, B. (2003): Modeling user rating profiles for collaborative filtering. *Proc. 17th Annual Conference on Neural Information Processing Systems*, British Columbia, Canada, 627-634, MIT Press.

Rasch, G. (1960): *Probabilistic models for some intelligence and attainment tests*. Nielsen & Lydiche, Copenhagen.

Salakhutdinov, R. and Mnih, A. (2008): Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. *Proc. 25th International Conference on Machine Learning*, Helsinki, Finland, 880-887, Omnipress.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001): Item-Based collaborative filtering recommendation algorithms. *Proc. 10th International World Wide Web Conference*, Hong Kong, China, 285-295, ACM Press.

Wright, B.D. and Masters, G.N. (1982): *Rating scale analysis*. Chicago: MESA Press.

Xue, G.R. , Lin, C., Yang, Q., Xi, W., Zeng, H.J. , Yu Y., and Chen Z. (2005): Scalable collaborative filtering using cluster-based smoothing. *Proc. 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, 114-121, ACM Press.