TELKOMNIKA, Vol. 11, No. 7, July 2013, pp. 3534 ~ 3540
e-ISSN: 2087-278X

Effect of User's Judging Power on the Recommendation Performance

Li-Yu Mao, Yuan Guan, Ming-Sheng Shang*, Shi-Min Cai
Web Sciences Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China
*Corresponding author, e-mail: shang.mingsheng@gmail.com

Abstract

In most B2C E-commerce websites, recommender systems make recommendations for each individual user based on his/her historical rating behaviors. Previous literatures focus on the overall performance of recommender system, while the performance of individual level receives little attention. In this paper, we discover that recommendation algorithms perform better on users who have strong judging power, and vice versa. We test our conclusion on three benchmark data sets, namely MovieLens, Netflix, and Amazon, which further provide evidence of the validity of our finding. Moreover, our finding may provide some guidance for designing recommendation algorithms more efficiently by concerning users' different judging power.

Keywords: *Recommender system, user's judging power, collaborative filtering, RMSE*

1. Introduction

The rapid development of Internet technology makes us encounter lots of information, e.g., tens of thousands of movies in Netflix, millions of books in Amazon, over one billion of web pages collected by delicious.com and so on. How to find the interesting part among them is a big challenge. The traditional search engine can only present all users with the same results, while it cannot provide personalized services concerning different users' interests and hobbies [1]. With such addressed issue, the personalized recommendation service springs up, and has been investigated extensively [2] [3] [4]. In most B2C e-commerce websites, recommender systems recommend objects for users based on their past online behaviors (e.g. click, browse, purchase) and users are no longer passive browsers but active participants.

A variety of personalized recommendation algorithms have been proposed by researchers, including collaborative filtering methods [5] [6], content-based methods [7] and hybrid ones [8]. However, previous work mainly focused on the overall performances of recommendation algorithms, while paid little attention on the recommendation performance of individual level. But in the real cases, there are various kinds of users, and they may rate objects in different ways. For example, the user may be someone who does not taken seriously about voting, or he/she has no experience in the related field and gives some irrational ratings. What's worse, some malicious spammers give biased ratings intentionally. We suppose that users have different judging power, which may have some certain impacts on the performances of recommendation algorithms.

There have been some ranking algorithms which can be used to distinguish users by their judging power [9] [10] [11]. For example, in Refs [9] [10], an iterative refinement (IR) algorithm is proposed. In [11], the iterative refinement algorithm is revised by De Kerchove and Van Dooren, which assign trust to each individual rating. In this paper, we propose an algorithm based on YZLM (Yu-Zhang-Laureti-Moret, see in Ref [9]) to measure users' judging power. The classic user-based collaborative filtering method (CF) is used to test the recommendation performance on different users with different judging power. We first divide all users into different groups by their judging power. Then we get the average intro-group recommendation performance of different groups. Through extensive experiments on three benchmark data sets, we find that CF performs better on users who have stronger judging power, vice versa. In other words, it shows that the accuracy performance on each user is positively correlated with his/her

Received January 28, 2013; Revised March 13, 2013; Accepted March 30, 2013

judging power. Moreover, our finding may provide some guidance for designing more efficient recommendation algorithms concerning each user's judging power.

This paper tackles the important issue of how to measure the user's judging power, and find that user's judging power indeed have a positive correlation impact on the recommendation performance. The paper is organized as follows. Section 1 is the introduction part. The recommendation algorithm and modified YZLM algorithm are introduced in section 2. Section 3 is about the experimental data sets and the results we get. The last part is our conclusions.

2. Algorithms and Metrics

2.1. User-Based CF Algorithm

The basic idea of CF algorithm can be divided into two steps: (1) Calculate the similarities between the target user and his/her neighbors through their history behaviors; (2) Predict the target user's preference for an unobserved object.

(1) The degree of similarity between user u and user v is measured by formula (1) through cosine similarity. Here, $r_{u,i}$ is the existing rating of user u to object i, and I_u denotes the set of objects rated by user u.

$$sim(u,v) = \frac{\sum_{i \in (I_u \cap I_v)} r_{u,i} * r_{v,i}}{\sqrt{\sum_{i \in I_u} r_{u,i}^2 * \sum_{i \in I_v} r_{v,i}^2}} \quad (1)$$

(2) The predicted value of the target user u to the object i is calculated by

$$p_{u,i} = \bar{r}_{u.} + \frac{\sum_{v \in N_u} sim(u,v) * (r_{v,i} - \bar{r}_{v.})}{\sum_{v \in N_u} sim(u,v)} \quad (2)$$

where the collection N_u is the set of users who also rated the object i. $\bar{r}_{v.}$ is the average rating given by user v.

2.2. Modified YZML Algorithm

Most websites such as Amazon, MovieLens and Netflix usually use the arithmetical average of the object's ratings as the estimation of its quality. However, it does not consider the differences of users' judging power. YZLM algorithm makes a distinction between users with different profiles by their judging power, which is proportional to users' weights. Users' judging power is then used by the weighted arithmetic average to estimate for the object's quality. In this way, we can get a more accurate estimation for the object's quality.

Suppose N users and M objects in a rating system. Each user has his/her own judging power (denoted by W_u for rater u, larger W_u corresponds to stronger judging power) and each object has an intrinsic quality (indicated by Q_j for object j). We assume that both the judging power and intrinsic quality are latent. $\sigma^2_{,u}$ represents the deviation of the rating vector of user u from the object's quality vector, and it has an inverse correlation with W_u. The Q_j and $\sigma^2_{,u}$ will be estimated by q_j and V_u. In YZLM algorithm, the object's quality is estimated by the weighted arithmetic average, where the weights is proportional to users' judging power and the users' judging power are updated by the estimated objects' quality. By iterative refinement, we can obtain the q_j and V_u as close as possible to the hidden values Q_j and $\sigma^2_{,u}$ after convergence of the algorithm.

The original implementation of YZLM algorithm considers only the case when all users have rated all objects, while it cannot be generalized to handle sparse data. In order to process sparse data, we use A, an N×M adjacency matrix, to record the spare data. If rater u rate object j, A_{uj}=1, otherwise A_{uj}=0[12].

Each user u is assigned with a weight value w_u, which is initially set as 1/N. $r_{u,j}$ is the rating user u rates to object j. The quality of object j is estimated by the weighted arithmetic average.

$$q_j = \sum_{u=1}^{N} A_{uj} w_u r_{u,j} \tag{3}$$

The rating variance of user u is computed as follows

$$\sigma_u^2 \approx V_u = \frac{1}{\sum_{i=1}^{M} A_{u,i}} \sum_{j=1}^{M} A_{u,j} (r_{u,j} - q_j)^2 \tag{4}$$

It should be noted that the $\sigma^2_{,u}$ sometimes may equal to 0. Thus, we constrain the value of $\sigma^2_{,u}$ to be not less than a certain small value ε>0 to prevent user weights from diverging (In our simulations, we use $\varepsilon=10^{-8}$). The updated normalized weight of user u is then given by

$$w_u = \frac{V_u^{-\beta}}{\sum_{v=1}^{N} V_v^{-\beta}} \tag{5}$$

where β≥0. It is very obvious that β=0 corresponds to the simple arithmetic average. The higher β will bring the greater penalization to users with larger deviation V_u. Yu et al. [9] noted that the case β=1/2 provides better numerical stability of the algorithm as well as translational and scale invariance, while in Ref [13], the case β=1 is the optimal from the point of view of mathematical statistics. Herein, we use β=1 because it yields superior performance, and the case β=1/2 does not alter the fundamental character of final result.

The algorithm is initialized by setting the user weights as $w_u=1/N$ for all users, then iterates repeatedly over the equations (3, 4, 5) until the change of the estimated quality vector between two adjacent iteration steps is less than a certain threshold value Δ.

$$|q - q'| = \sqrt{\frac{1}{|\{k \mid q_k' \neq 0\}|} \sum_{j \in \{k \mid q_k' \neq 0\}} (q_j - q_j')^2} \tag{6}$$

Note that the algorithm may fail to converge if the value of threshold Δ is set to be too small. Conversely, too large value of threshold may disrupt the iterative process [9]. Therefore it's better to take a few trials to choose an appropriate value, and the value is set as $\Delta=10^{-6}$ in our simulations [14].

2.3. Performance Metrics

The accuracy metric is often used to measure the performances of different recommendation algorithms [15]. The mean absolute error (MAE) is a widely used accuracy metric that computes the mean absolute deviation of two sequences. The MAE of user level, MAE_u, is calculated as follows:

$$MAE_u = \frac{\sum_{i \in T_u} |p_{u,i} - r_{u,i}|}{|T_u|} \tag{7}$$

where $p_{u,i}$ is the predicted rating generated by the algorithm of CF. $r_{u,i}$ is the actual rating user u gives to object i in the probe set, and $|T_u|$ is the number of ratings of user u in the probe set.

The unevenness of the weights assigned to individual user can be measured by the inverse participation ratio (IPR). Given the user normalized weights w_u, IPR can be computed as

$$IPR = \left(\sum_{u \in U} w_u^2\right)^{-1} \tag{8}$$

The IPR is reciprocal to another well-known measure, the Herfindahl-Hirschman concentration index (HHI). The IPR measures the effective number of users with respect to their weights w_u. When all weights are equal, w_u=1/N, then IPR=N. By contrast, when all weights but one are zero, IPR=1.

3. Experimental Results

3.1. Datasets

Three benchmark data sets are used to test the algorithms' performances:

(1) MovieLens (http://www.movielens.org) is a movie recommendation website, which uses users' ratings to generate personalized recommendations. The data we used is downloaded from http://www.grouplens.org/node/73.

(2) Netflix (http://www.netix.com) provides the world's largest online video rental service, offering more than 6.7 million subscribers access to 85,000 DVD and a growing library of over 4,000 full-length movies and television episodes that are available for instant watching on their PCs. The data we used is a random sample that consists of 3000 users who have rated at least 20 movies and 2779 movies having been rated by at least one user.

(3) Amazon (http://www.amazon.com) is a multinational e-commerce company. The original data were collected from 28 July 2005 to 27 September 2005, and the data we used is a random sample.

The basic statistics of three benchmark data sets are shown in Table 1, in which we can find that they have different sizes and different sparsity. In order to comprehensively test the recommendation performance, the data are randomly divided into two parts: the 80% training set (E) and the 20% probe set (T). The information of training set is treated as known information, while no information of probe set is allowed to use for prediction.

Table 1. Basic statistics of the tested data sets

Data set	Users	Objects	Ratings	Sparsity
MovieLens	943	1683	100000	6.30×10^{-2}
Netflix	3000	2779	197248	2.37×10^{-2}
Amazon	3604	4000	134679	9.34×10^{-3}

Furthermore, the distributions of ratings are show in Figure 1. It is interesting that the data of MovieLens and Netflix share the similar pattern, and differ from the data of Amazon. The main reason may be that the MovieLens and Netflix only include the media objects, and it makes the user easily compare the qualities of different objects. But the Amazon's data is highly sparse and Amazon's users only buy/rate what they actually like.

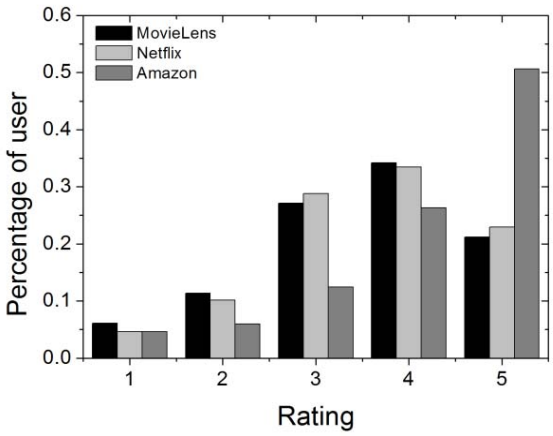

Figure 1. Distributions of ratings in three data sets

3.2. Results & Discussions

To investigate the effect of users' judging power on recommendation performance, we need to compute their judging power by modified YZML algorithm. Figure 2 shows the histograms of σ_u's distributions for the three dataset, in which three subplots represent the results of MovieLens, Netflix and Amazon, respectively. In three datasets, the center peaks of histograms are around 1 or 0.75, which suggests that most users' ratings differ from the objects' qualities.

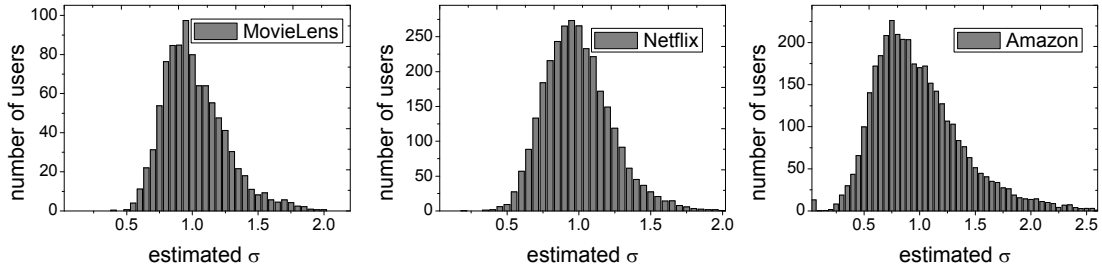

Figure 2. Histograms of σ_u's distributions by modified YZLM with β=1

The value of IPR closely relates to the histograms of σ_u's distributions according to equation (8). The IPR values dependent on β are summarized in Table 2 for all the three data sets. The case β=0 makes the homogeneous weight of users, which leads to the values of IPR euqal to the total number of users. As β increases, IPR value gradually decreases, indicating that YZLM algorithm can distinguish between different users. Users of poor judging power gradually lose the right to speak. We can find that the IPR value of sparser dataset declines faster, especially in the Amazon. IPR value drops to 13 when β=1. Figure 2 shows that the number of users in the first bin of Amazon (i.e., with σ_i close to zero) is 13. This value is equal to the value of IPR. These "ideal users" with small estimated $\sigma_u \approx \varepsilon$ have very large $w_u \approx 1/\varepsilon$(a small $\varepsilon=10^{-8}$ is chosen as a lower bound to avoid the divergence of user weights). This does not mean that the effective number of users is 13 and ratings of the other users are neglected. Other users count for all objects that have not been rated by the "ideal users". Besides, these "ideal users" correspond to users with a few ratings (near 8 for β=1). In an extreme case, if a user only rates an object and this object is only rated by him, his estimated $\sigma_u=\varepsilon$.

Table 2. IPRs for three data sets with different β

Dataset	β=0	β=0.5	β=1	β=1.5	β=2	β=2.5	β=3
MovieLens	943	895	762	557	209	2	3
Netflix	3000	2833	2276	373	5	5	7
Amazon	3604	1242	13	127	140	137	136

With the increase of β, "ideal users" gradually appeared in Movielens and Netflix. This is because YZLM algorithm is a process of iterative refinement. When β is large enough, these users who in the first iteration step with rather small estimated σ_u are given large weight in the second iteration step. Then, these users' ratings have the right to speak to the estimated quality values and further lower their estimated σ_u. By repeating these iterations, σ_u became smaller and smaller. Finally, there may appear some "ideal users" with estimated $\sigma_u \approx \varepsilon$.

We equally divide all users into ten groups according to their judging power by descending order. Each group consists of about 10% of the total number of users. We obtain all users' MAE_u and then count average the MAE_u of ten groups. Figure 3 is a comparison of average MAE_u of each group and the average MAE_u of all users. The horizontal axis indicates the average judging power (denoted by group's average σ_u, smaller σ_u corresponds to stronger judging power) of the user groups, and the vertical axis denotes the average MAE_u of users. The circle line and solid line represent the intra-group average MAE_u and the average MAE_u of

all users, respectively. The subplots show the results of MovieLens, Netflix and Amazon, respectively. In Figure 3, from left to right, we can find that the average judging power of groups is getting worse and the average MAE_u of groups emerge in ascending trend. Overall, there is a positively correlated relationship between CF's MAE and group's judging power, i.e., the recommendation performance is relatively good in the group with better judging power , and these users with poor judging power have a great impact on the performance of the CF algorithm.

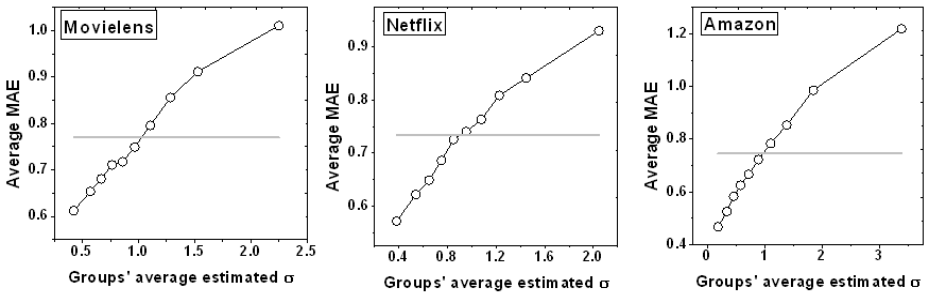

Figure 3. Average MAEs of groups and average MAEs of all users for three datasets

Recommender system can't meet all users' demands if it treats all users equally. The group's MAE_u is smaller if it consists of users with better judging power, which means that the predicted ratings generated by recommendation algorithm are closer to the actual rating score. But the MAE_u of groups with lower judging power are relatively higher, which means that the predicted ratings generated by the CF are quite different from the actual scores. These users with high MAE_u are suspect to raise the average MAE_u of all users (MAE of system level). Since CF cannot recommend accurately for users with poor judging power, we can utilize other recommendation algorithms which are more in line with the users' behaviors to make these users' MAE_u decrease. By this way, the average MAE_u of all users will be improved. That may be our subsequent research content.

4. Conclusion

In this paper, we firstly propose a natural extension of the YZLM algorithm to get users' judging power. Then we study the Intra-group performances of CF algorithm in ten groups with different judging power. Through experiments on three benchmark datasets, it shows that there is a positively correlated relationship between users' judging power and the recommendation performance. That's to say, users with strong judging power are more likely to get better recommendation performance, while the users who have poor judging power can only get bad recommendation results. Besides, the CF algorithm predicts the target user's preference based on preferences of his/her neighbors. Users with better judging power accords with mainstream preferences, so the recommendation performance is relatively better. On the contrary, for users with poor judging power, their preferences are more personalized and hard to be handled. Moreover, since CF cannot satisfy preferences for users with poor judging power, we can take other algorithm to cover CF's shortage. This may be our further studies.

References

[1] Brin S and Page L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst*. 1998; 30: 107-117.
[2] Resnick P and Varian H R. Recommender systems. *Commun. ACM*. 1997; 40(3): 56-58.
[3] Abeer Mohamed El-korany and Salma Mokhtar Khatab. Ontologybased Social Recommender System. *IAES International Journal of Artificial Intelligence (IJ-AI).* 2012; 1(3): 127-138.
[4] Muhammad Waseem Chughtai, Ali Bin Selamat and Imran Ghani. Goal-based hybrid filtering for user-to-user Personalized Recommendation. *International Journal of Electrical and Computer Engineering (IJECE)*. 2013; 3(3).

[5] Resnick P, Iacovou N, Suchak M, Bergstrom P and Riedl J. GroupLens: An open architecture for collaborative filtering of netnews. *In ACM CSCW*. 1994; 175-186.
[6] Linden G, Smith B, and York J. Amazon.com Recommendations Item-to-Item Collaborative Filtering. *IEEE Internet Computing*. 2003; 7(1): 76-80.
[7] Pazzani MJ and Billsus D. *Content-based recommendation systems*. The Adaptive Web. Berlin. Heidelberg. 2007; 4321: 325-341.
[8] R Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*. 2002; 12(4): 331-370.
[9] Yu YK, Zhang YC, Laureti P, and Moret L. Decoding information from noisy, redundant, and intentionally distorted sources. *Physica A*. 2006; 371: 732-744.
[10] Laureti P, Moret L, Zhang YC, and Yu YK. Information filtering via iterative refinement. *EPL*. 2006; 75(6): 1006-1012.
[11] De Kerchove C, Van Dooren P. Iterative filtering for a dynamical reputation system. *arXiv 0711. 3964*, 2007.
[12] De Kerchove C and Van Dooren P. Reputation systems and nonnegativity. *LNCIS*. 2009; 389: 3-16.
[13] Hoel PG. Introduction to Mathematical Statistics. Wiley, New York, 1984.
[14] Medo C M, Wakeling J, Mirylenka RK, et al. *Model of reviewers' behavior in peer reviews*. European Community.Report number: 213360. 2010.
[15] Vozalis E and Margaritis KG. *Analysis of recommender system algorithms*. In Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA-2003), Athens, Greece. 2003: 1-14.