Rating Personalization Improves Accuracy: A Proportion-Based Baseline Estimate Model for Collaborative Recommendation

Zhenhua Tan(✉), Liangliang He, Hong Li, and Xingwei Wang

Software College of Northeastern University, Shenyang 110819, China
{tanzh, wangxw}@mail.neu.edu.cn,
{1501666, 1501667}@stu.neu.cn

Abstract. Baseline estimate is an important latent factor for recommendations. The current baseline estimate model is widely used by characterizing both items and users. However, it doesn't consider different users' rating criterions and results in predictions may be out of recommendation's rating range. In this paper, we propose a novel baseline estimate model to improve the current performance, named PBEModel (Proportion-based Baseline Estimate Model), which uses rating proportions to compute the rating personalization. The PBEModel is modeled as a piecewise function according to different rating personalization. In order to verify this new baseline estimate, we apply it into SVD++, and propose a novel SVD++ model named PBESVD++. Experiments based on six real datasets show that the proposed PBEModel is rational and more accurate than current baseline estimate model, and the PBESVD++ has relatively higher prediction accuracy than SVD++.

Keywords: Recommender system · Latent factor model · Baseline estimate model · PBEModel · PBESVD++

1 Introduction

Recommender systems, which are important information filtering mechanisms to predict item rating or user preference, have been applied in variety internet-based systems, such as videos, music, news, and some social networks. Recommender systems learn user preference pattern to items from user-item transactions based on Collaborative Filtering (CF) [1], to predict users' possible interesting items from unknowns.

In order to establish recommendations, CF needs to compare items against users, which are quite different two objects. Usually, the neighborhood approach [2] and latent factor models [3] are two main disciplines to do the comparison in CF. In terms of latent factor models, there are several approaches, such as Singular Value Decomposition (SVD) model [4], pLSA [5], Latent Dirichlet Allocation [6] and SVD++ [7], to uncover latent features that explain observed ratings by historical feedbacks. In these models, the SVD++ model proposed by Yehuda Koren [7] has relatively high recommendation accuracy, and many researchers proposed related models based on it, such as TimeSVD++ [8], SocialSVD++ [9], and TrustSVD [10].

S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 104–114, 2017.
DOI: 10.1007/978-3-319-59288-6_10

To improve the prediction accuracy, most of researchers focus on the key challenges underlying the recommender systems and collaborative filtering are data sparseness and personalized data sparseness, such as [7–10]. Some researchers focused on properties of recommendations to propose more accurate prediction method, such as integrated QoS prediction approach HDOP [11], prediction method of unknown QoS properties [12], and reputation measurement based on malicious rating detection [13]. In this paper, we focus on the improvement of baseline estimate model.

Observed from current models, such as [7–10], we can find that almost all recommender models have a same baseline estimate model, as:

$$\hat{r}_{ui} = b_u + b_i + \mu \tag{1}$$

Where $\hat{r}_{ui}$ denotes the baseline prediction for unknown item i by user u; μ denotes the overall average rating; b_u and b_i denote the observed deviations of user u and item i from the average respectively.

By analyzed the baseline estimate model in the above Eq. (1), we find the prediction of $\hat{r}_{ui}$ for unknown items may be out of recommender range. The main reason is that the current baseline estimate model doesn't consider different users' rating criterions.

Thus, we propose a proportion-based baseline estimate model, named PBEModel (Proportion-based Baseline Estimate Model). The improved PBEModel consider users' rating personalization via a proportional baseline estimate method. To our knowledge, it is the first time to improve the baseline estimate model based on proportion concept. In order to prove the rightness, we also proposed a PBESVD++ model to improve the prediction accuracy of SVD++ based on PBEModel.

This paper has the following innovative features:

(1) We uncover the shortcoming and reason of the current baseline estimate model, and propose a novel Proportion-based Baseline Estimate Model (PBEModel) to get better prediction ability.
(2) We propose an improved SVD++ model based on PBEModel, named PBESVD++, and get more prediction accuracy based on different recommendation datasets.
(3) The proposed method in this paper may bring a novel approach to improve prediction accuracy of recommender system.

In the remainder of this paper, we introduce the related work in the next section. PBEModel is proposed in Sect. 3, and PBESVD++ is proposed in Sect. 4. The experiments and analysis follow in Sect. 5, with conclusion afterwards in the last section.

2 Related Work About SVD++

Collaborative filtering (CF) methods are usually used in recommender systems, based on collecting and analyzing a large amount of users' behaviors and predicting what users will like based on their similarity to other users. One of the primary areas of CF is latent factor model which try to explain the ratings by characterizing both items and users.

As one of the main methods of latent factor models, SVD is a commonly used matrix decomposition technique. SVD++ is an improved model based on SVD and had more remarkable improvement of recommendation accuracy than SVD model. The detailed improvement is that a free user-factor vector p_u is complemented by $|I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right)$, and a user u is modeled as $\left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right)\right)$. The equation of SVD++ is as:

$$\hat{r}_{ui} = b_u + b_i + \mu + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right)\right) \tag{2}$$

where $\hat{r}_{ui}$ denotes the prediction of unknown item i by user u; μ is the overall average rating; b_u and b_i represent the observed deviations of user u and item i, respectively, from μ; I_u represents the set of items rating by user u; p_u and q_i^T represent a d-dimensional latent feature vector of user u and item i, respectively; y_j represents the implicit influence of items rated by user u in the past on the ratings of unknown items in the future. Kumar et al. [9] merged the user's social factors into the SVD++ and proposed SocialSVD++ model to get better prediction performance with equation as:

$$\hat{r}_{ui} = b_u + b_i + \mu + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right) + |p_i|^{-\frac{1}{2}} \cdot \left(\sum_{i \in p_i} x_i\right)\right) \tag{3}$$

where p_i denotes the set of implicit feedback (the set of item that are rated by most users) in SocialSVD++. Guo et al. [10] took social trust information into account in the SVD++ and proposed TrustSVD model, which merged trust factor and get better accuracy, the equation is as:

$$\hat{r}_{ui} = b_u + b_i + \mu + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right) + |T_u|^{-\frac{1}{2}} \cdot \left(\sum_{v \in T_u} w_v\right)\right) \tag{4}$$

where w_v denotes the user-specific latent feature vector of users (trustees) trusted by user u. Yehuda Koren [8] proposed a Time Changing Baseline Predictors to improve the baseline estimate model by allowed the baseline ratings to be changed over time as $\hat{b}_{ui} = b_u(t_{ui}) + b_i(t_{ui}) + \mu$, where $b_u(\cdot)$ and $b_i(\cdot)$ are real valued functions that change over time; t_{ui} denotes time of rating in term of individual days. Finally, they proposed TimeSVD++ based on their SVD++.

All of these models have a relatively same baseline estimate model as Eq. (1). As it is known for many recommender systems, the baseline estimate model is a key factor. And, these similar considerations were mentioned in the Pearson method, which considers the differences of user ratings [14], and this concept also inspires us to improve the current baseline estimate.

3 Proposed PBEModel

As we can see from Sect. 2, related latent factor models for recommender systems are based on the current baseline estimate model described in Eq. (1) as $\hat{r}_{ui} = b_u + b_i + \mu$, where b_u represents the deviation of user u from the overall average rating, that is the difference between the average rating $\bar{r}_u$ of user u and the overall average rating $\bar{r}$. And b_i represents the deviation of item i from the overall average rating that is the difference between the average rating $\bar{r}_i$ of item i and the overall average rating $\bar{r}$. So, $b_u = \bar{r}_u - \bar{r}$ and $b_i = \bar{r}_i - \bar{r}$, where $\bar{r} = \mu$. Thus,

$$\hat{r}_{ui} = \bar{r}_u + \bar{r}_i - \bar{r} \tag{5}$$

Let $\Delta r = \bar{r}_u - \bar{r}$, then

$$\hat{r}_{ui} = \bar{r}_i + \Delta r \tag{6}$$

where Δr denotes the difference between the average rating of user u and the overall average rating ($\bar{r}$), which reflects the difference between the users rating habits and the public habits.

Obviously, Eq. (1) is equal to Eqs. (5) and (6). Based on the equivalent equations, we can easily find the current baseline estimate model does not consider different users' rating criterions. In real recommender systems, users have different rating preferences. Some are used to relatively higher ratings for items than others, while some have very strict rating criterion for a same item. This problem result in that predictions based on current baseline estimate model would be out of rating range. Let's take some examples, and assume the rating range is [1, 5].

Example (1): for user Alice.
Assume $\bar{r}_u = 4.2, \bar{r}_i = 4, \bar{r}_i = 3.1$, then $\hat{r}_{ui} = 4.2 + 4 - 3.1 = 5.1 > 5$.

Example (2): for user Bob.
Assume $\bar{r}_u = 4.2, \bar{r}_i = 4, \bar{r}_i = 4.2$, then $\hat{r}_{ui} = 4.2 + 4 - 4.2 = 4 = \bar{r}_i$.

Example (3): for user Carl.
Assume $\bar{r}_u = 1.5, = 2, = 4.2$, then $\hat{r}_{ui} = 1.5 + 2 - 4.2 = (-1.7) < 0$.

In Example (1), Alice's average rating $\bar{r}_u > \bar{r}$ and $\Delta \mathrm{r} > 0$, which means Alice has relatively slack rating criterion than most of other users, while Carl has stricter rating criterion in Example (3) where $\bar{r}_u < \bar{r}$ and $\Delta \mathrm{r} < 0$. In Example (2), Bob has a relatively average rating criterion with $\bar{r}_u = \bar{r}$ and $\Delta \mathrm{r} = 0$, and his prediction $\hat{r}_{ui}$ is equal to $\bar{r}_i$. Obviously, the predictions in Examples (1) and (3) are unreasonable because of they out of range [1, 5], and it proves that the baseline estimate model needs improvement. We consider the personalization of users' rating criterions to improve the baseline estimate model in this paper.

Definition 1. Rating Strict Level represents a user's personalization during rating items. Let **L1** denote user's relatively average criterion to make a rating between given range; let **L2** denote relatively strict criterion that means users in this level will make a relatively lower rating than others. And let **L3** denote relatively slack criterion and users in this level will make relatively higher ratings for items. Usually, a L1 user's average rating $\bar{r}_u = \bar{r}$, while a L2 user's $\bar{r}_u < \bar{r}$ and a L3 user's $\bar{r}_u > \bar{r}$. Figure 1 shows these users in different personalized rating.

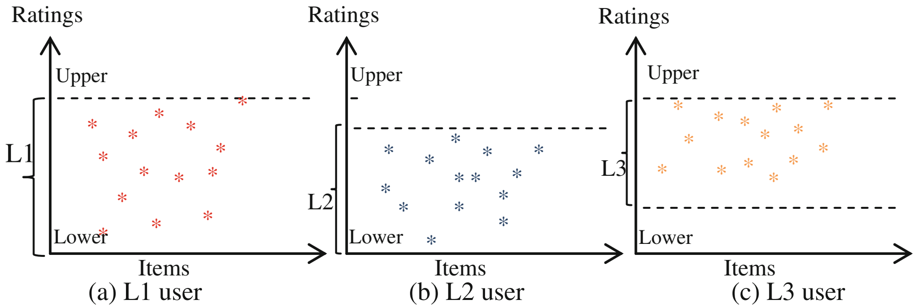

Fig. 1. Personalized rating levels

For users of L1, just like Bob in Example (2), their ratings ranged between the given data area [m, n] of recommender systems. But users of L2, whose rating criterion is stricter, usually have a personal upper limit for ratings, which is less than the given range upper limit n. For example, in a recommender system with rating range [1, 5], a L2 user would think number 4 is the best rating in his mind, while a L1 user would think number 5 is the best level. In this paper, this difference between users' rating criterion is the personalization. Therefore, in real systems, these three kinds of users have different personalized rating ranges, just as shown in Fig. 2.

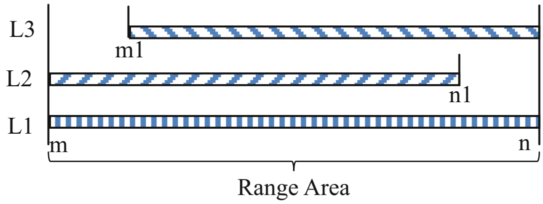

Fig. 2. Personalized compressed rating ranges

Ratings of L1 users ranges between given [m, n], and L2 between [m, n1] while L3 between [m1, n]. Based on the analysis above, we improve the baseline estimate model according to the three kinds of personalization. We compute users' rating personalization by rating proportion to improve the current baseline estimate model.

Firstly, assume the range area is [0, n].

(1) When $\bar{r}_u = \bar{r}$, according to Eq. (6), we can get

$$\hat{r}_{ui} = \bar{r}_i \tag{7}$$

(2) When $\bar{r}_u < \bar{r}$, then

$$(\hat{r}_{ui} - \bar{r}_u)/\bar{r}_u = (\bar{r}_i < \bar{r})/\bar{r} \tag{8}$$

(3) When $\bar{r}_u > \bar{r}$, then

$$(\hat{r}_{ui} - \bar{r}_u)/(n - \bar{r}_u) = (\bar{r}_i - \bar{r})/(n - \bar{r}) \tag{9}$$

Based on the above Eqs. (7)–(9), we can get the new baseline estimate equation for system with rating range [0, n], as

$$\hat{r}_{ui} = \begin{cases} (\bar{r}_u/\bar{r}) \cdot (\bar{r}_u/\bar{r}) + \bar{r}_u & when\ \bar{r}_u \le \bar{r} \\ ((n - \bar{r}_u)/(n - \bar{r})) \cdot (\bar{r}_i - \bar{r}) + \bar{r}_u & when\ \bar{r}_u > \bar{r} \end{cases} \tag{10}$$

Obviously, if $\bar{\mathrm{r}}_{\mathrm{u}} = \bar{\mathrm{r}}$, Eq. (10) is same with Eq. (5). Generally, if the rating range of recommender system is between [m, n], the baseline estimate equation could be:

$$\hat{r}_{ui} = \begin{cases} ((\bar{r}_u - m)/(\bar{r} - m)) \cdot (\bar{r}_i - \bar{r}) + \bar{r}_u & when\ \bar{r}_u \le \bar{r} \\ ((n - \bar{r}_u)/(n - \bar{r})) \cdot (\bar{r}_i - \bar{r}) + \bar{r}_u & when\ \bar{r}_u > \bar{r} \end{cases} \tag{11}$$

In Eq. (11), because $\bar{r}_u$, $\bar{r}_i$ and $\bar{r}$ belong to [m, n], it is easy to prove that $\hat{r}_{ui}$ also belong to [m, n]. Finally, the current baseline estimate model $\hat{r}_{ui} = b_u + b_i + \mu$ can be improved as:

$$\hat{r}_{ui} = \begin{cases} \frac{b_u + \mu - m}{\mu - m} \cdot b_i + b_u + \mu, & when\ b_u \le 0 \\ \frac{n - b_u - \mu}{n - \mu} \cdot b_i + b_u + \mu, & when\ b_u > 0 \end{cases} \tag{12}$$

Related elements, such as $\hat{r}_{ui}$, b_u, b_i and μ, have same meanings as Eq. (1), and rating area is [m, n] where n $\ge$ m $\ge$ 0. We named this novel baseline estimate model with Eq. (12) as **PBEModel**.

4 PBESVD++: An Improved SVD++ Based on PBEModel

In order to verify the recommendation performance of PBEModel, we apply it into SVD++, and proposed an improved SVD++ named PBESVD++. The SVD++ model is described as Eq. (2). Firstly, we replace the baseline estimate part in Eq. (2) with Eq. (12), and get the following new equation, as:

$$\hat{r}_{ui} = \begin{cases} \frac{b_u + \mu - m}{\mu - m} \cdot b_i + b_u + \mu + q_i^T \cdot \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right)\right) & when\ b_u \le 0 \\ \frac{n - b_u - \mu}{n - \mu} \cdot b_i + b_u + \mu + q_i^T \cdot \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right)\right) & when\ b_u > 0 \end{cases} \tag{13}$$

where I_u represents the set of items rating by user u; p_u and q_i^T represent a d-dimensional latent feature vector of user u and item i, respectively; y_j represents the implicit influence of items rated by user u in the past on the ratings of unknown items in the future. All of these parameters have same meanings as in SVD++. We named the new SVD++ model with Eq. (13) as **PBESVD++**.

Model Learning. Involved parameters in PBESVD++ could be learned by minimizing the regularized squared error function associated. The regularized squared error function is:

$$L = \frac{1}{2}\sum_{u}\sum_{i \in I_u}\left(\hat{r}_{ui} - r_{ui}\right)^2 + \frac{\lambda}{2}\left(\sum_{u} b_u^2 + \sum_{i} b_i^2 + \sum_{u} \|p_u\|_F^2 + \sum_{i} \|q_i\|_F^2 + \sum_{j} \|y_j\|_F^2\right) \quad (14)$$

where $\|\cdot\|$ denotes the Frobenius norm, and λ is to alleviate operation complexity and avoid over-fitting. We use same strategy proposed by Guo et al. [10] to do model learning. Therefore the new loss function can be obtained as:

$$L = \frac{1}{2}\sum_{u}\sum_{i \in I_u}\left(\hat{r}_{ui} - r_{ui}\right)^2 + \frac{\lambda}{2}\left(\sum_{u} |I_u|^{-\frac{1}{2}}\left(b_u^2 + \|p_u\|_F^2\right)\right) + \sum_{i} |U_i|^{-\frac{1}{2}}\left(b_i^2 + \|q_i\|_F^2\right) + \sum_{j} |U_j|^{-\frac{1}{2}}\|y_j\|_F^2 \quad (15)$$

Instead of least square solvers, we employ the gradient descent solver to faster solve this convex problem to obtain a local minimization of the objection function. We perform the following gradient descents on $b_u, b_i, p_u, q_i\, and\, y_j$ for all the users and items. In addition, the PBEModel is a piecewise function, so the gradient descent processes are divided into two parts in PBESVD++.

(1) When $\mathrm{b_u} \leq 0$,

$$\frac{\partial L}{\partial b_u} = \sum\nolimits_{i \in I_u}\left(\left(1 + \frac{b_i}{\mu - m}\right) \cdot e_{ui}\right) + \lambda \cdot |I_u|^{-\frac{1}{2}} \cdot b_u$$

$$\frac{\partial L}{\partial b_i} = \sum\nolimits_{u \in U_i}\left(\frac{b_u + \mu - m}{\mu - m} \cdot e_{ui}\right) + \lambda \cdot |U_i|^{-\frac{1}{2}} \cdot b_i$$

$$\frac{\partial L}{\partial p_u} = \sum\nolimits_{i \in I_u}\left(q_i \cdot e_{ui}\right) + \lambda \cdot |I_u|^{-\frac{1}{2}} \cdot p_u$$

$$\frac{\partial L}{\partial q_i} = \sum\nolimits_{u \in U_i}\left(\left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum\nolimits_{j \in I_u} y_j\right)\right) \cdot e_{ui}\right) + \lambda \cdot |U_i|^{-\frac{1}{2}} \cdot q_i$$

$$\forall_{j \in I_u}\left(\frac{\partial L}{\partial y_j} = \sum\nolimits_{i \in I_u}\left(e_{ui} \cdot \lambda \cdot |I_u|^{-\frac{1}{2}} \cdot q_i\right) + \lambda \cdot \left|U_j\right|^{-\frac{1}{2}} \cdot y_j\right)$$

(2) When $\mathrm{b_u} > 0$, we only need replace the first two steps,

$$\frac{\partial L}{\partial b_u} = \sum\nolimits_{i \in I_u}\left(\left(1 - \frac{b_i}{n - \mu}\right) \cdot e_{ui}\right) + \lambda \cdot |I_u|^{-\frac{1}{2}} \cdot b_u$$

$$\frac{\partial L}{\partial b_i} = \sum\nolimits_{u \in U_i} \left(\frac{n - b_u - \mu}{n - \mu} \cdot e_{ui} \right) + \lambda \cdot |U_i|^{-\frac{1}{2}} \cdot b_i$$

where $e_{ui} = \hat{r}_{ui} - r_{\mathrm{ui}}$, and represents the prediction error the predicted rating from the real rating for user u on item i.

Complexity Analysis. In terms of space complexity, it is obvious that the growth on PBESVD++ almost can be ignored compared with SVD++. Due to the consideration of PBESVD++ is more detailed, the time space complexity cannot be completely ignored. Nevertheless, the range area limit m and n are constants, so the magnitude of time complexity is not changed, just a linear growth. The time complexity of the PBESVD++ model is mainly from the loss function L (Eq. (15)) and its gradients. In terms of L, the time complexity is $O(d \cdot |R|)$, where d represents the matrix dimensionality, and $|R|$ represents the number of the observed ratings. And in terms of the gradients, the time complexities of $\frac{\partial L}{\partial b_u}$, $\frac{\partial L}{\partial b_i}$, $\frac{\partial L}{\partial p_u}$, and $\frac{\partial L}{\partial q_i}$ are $O(d \cdot |R|)$, and the time complexity of $\frac{\partial L}{\partial y_j}$ is $O(k \cdot d \cdot |R|)$, where k represents the average of the number a user marks or an item receives, respectively.

5 Experiments and Analysis

In order to verify the rightness and efficiency of proposed PBEModel and PBESVD++, we deploy the experiment platform based on LibRec [15]. Table 1 shows the details of experimental datasets.

Table 1. Details of datasets for experiments

Data sets	Users	Movies	Ratings	Density	Rating ranges
ml-latest-small	700	10000	100000	1.43%	[0.5,5]
FilmTrust	1508	2071	35497	1.14%	[0.5,4]
Flixster	53213	18197	409803	0.04%	[0.5,5]
Minifilm	55	334	1000	5.44%	[0.5,4]
Epinions	40163	139738	664824	0.05%	[1, 5]
Ciao	7375	99746	280391	0.03%	[1, 5]

We perform a series of experiments based on six datasets FilmTrust, Movielens Latest (ml-latest-small), Flixster, Minifilm, Ciao and Epinions. We test three important evaluation metrics in experiments, there are **MAE** (Mean Absolute Error), **RMSE** (Root Mean Square Error) and **RE** (Relative Error). Assumed $\tau = \{(u,i) | \exists_r((u,i,r) \in R)\}$, and N is the number of observed ratings.

There are $MAE = \frac{\sum\limits_{(u,i) \in \tau} |r_{ui} - \hat{r}_{ui}|}{N}$, $RMSE = \sqrt{\frac{\sum\limits_{(u,i) \in \tau} (r_{ui} - \hat{r}_{ui})^2}{N}}$, and $RE = \sqrt{\frac{\sum\limits_{(u,i) \in \tau} (r_{ui} - \hat{r}_{ui})^2}{\sum\limits_{(u,i) \in \tau} \left(r_{ui}^2\right)}}$.

Experiment (1). Baseline estimation performance of PBEModel.
In order to verify the rightness and efficiency of PBEModel, we do experiments on the six data sets respectively. Each data set is divided into two parts, the first part (80%) is used for training, the other one (20%) for testing. We compare performances of PBEModel with other three models. The first one is current baseline estimate model according to Eq. (1), we name it as **BE** in experiments; the second one is part of PBEModel with $b_u \leq 0$, we name it as **PBE_1**; the last one is another part of PBEModel with $b_u > 0$, and we name it as **PBE_2** during experiments. Table 2 shows the detailed results.

Table 2. Performance of PBEModel

Data Sets	Metrics	BE	PBEModel PBE_1	PBEModel PBE_2	PBEModel Both	Improved (vs BE)
ml-latest-small	RMSE	0.9262	0.9226	0.9136	0.9100*	1.75%
	RE	0.2525	0.2515	0.2490	0.2480*	1.75%
	MAE	0.7072	0.7051	0.6985	0.6964*	1.54%
Film-Trust	RMSE	0.8406	0.8367	0.8308	0.8262*	1.71%
	RE	0.2674	0.2662	0.2643	0.2628*	1.71%
	MAE	0.6389	0.6379	0.6300	0.6289*	1.56%
Minifilm	RMSE	0.9694	0.9571	0.9556	0.9430*	2.72%
	RE	0.3099	0.3059	0.3054	0.3014*	2.72%
	MAE	0.7307	0.7225	0.7221	0.7140*	2.29%
Filxster	RMSE	0.9080	0.9072	0.9029	0.9021*	0.65%
	RE	0.2406	0.2404	0.2392	0.2390*	0.65%
	MAE	0.6809	0.6799	0.6738	0.6729*	1.18%
Epinions	RMSE	1.1126	1.1062	1.1038	1.0972*	1.39%
	RE	0.2667	0.2652	0.2646	0.2630*	1.39%
	MAE	0.8423	0.8413	0.8234	0.8224*	2.37%
Ciao	RMSE	1.0542	1.0451	1.0375	1.0283*	2.46%
	RE	0.2506	0.2485	0.2467	0.2445*	2.46%
	MAE	0.7872	0.7827	0.7487	0.7442*	5.46%

Note: * is the best value.

We can see that the proposed baseline estimate model PBEModel gets wonderful higher performances of RMSE, MAE and RE, compared with current baseline estimate model. It also proves that the two parts of PBEModel are efficient, and has better baseline estimate accuracy. The improvement is up to 1.7% both in Movielens and FilmTrust, and up to 2.7% in Minifilm. Especially, the improvement exceeds 5.46% in Ciao, and 2.37% in Epinions.

Experiment (2). Performance of PBESVD++
We test the performance of PBESVD++ comparing with SVD++ on data sets Film-Trust, Flixster, and Minifilm. Each dataset is computed with d = 5 and d = 10. Table 3 discloses the data details.

Table 3. Performance of PBESVD++

Dataset	Metrics	SVD++	PBESVD++	Improved (vs SVD++)
FilmTrust (d = 5)	RMSE	0.8158	0.8117*	0.50%
	RE	0.2595	0.2582*	0.50%
	MAE	0.6250	0.6220*	0.47%
FilmTrust (d = 10)	RMSE	0.8119	0.8078*	0.51%
	RE	0.2583	0.2570*	0.51%
	MAE	0.6226	0.6196*	0.49%
Flixster (d = 5)	RMSE	0.9873	0.9762*	1.12%
	RE	0.2492	0.2464*	1.12%
	MAE	0.7406	0.7246*	2.16%
Flixster (d = 10)	RMSE	0.9857	0.9745*	1.14%
	RE	0.2488	0.2460*	1.14%
	MAE	0.7403	0.7238*	2.22%
Minifilm (d = 5)	RMSE	0.9485	0.9272*	2.25%
	RE	0.3032	0.2964*	2.25%
	MAE	0.7182	0.7062*	1.68%
Minifilm (d = 10)	RMSE	0.9450	0.9269*	1.92%
	RE	0.3021	0.2963*	1.92%
	MAE	0.7235	0.7051*	2.55%

Note: * is the best value.

As you can see, results show that PBESVD++ has relatively higher prediction accuracy than SVD++ during all the experimental datasets. Based on data set FilmTrust, the performance of RMSE, RE and MAE are improved 0.5% or so by PBESVD++. And the improved rate exceeds 1.12% during other experimental datasets. Especially in dataset Flixster and Minifilm, the performances are improved more than 2.22%.

6 Conclusion

In this paper, we analyze the current baseline estimate model, and find some interesting issues need to be improved. For example, the predictions may out of recommendation range. The main reason for this phenomenon is that the current baseline estimate doesn't consider different users' rating criterions. We call these criterions' differences as rating personalization. Thus, we proposed a novel baseline estimate model named PBEModel, which uses rating proportions to compute the rating personalization. In order to verify this new baseline estimate, we apply it into SVD++, and proposed a novel SVD++ model named PBESVD++. We disclose the mathematics modeling of PBEModel and PBESVD++ in this paper. We make a series of experiments with six real datasets, and results show that the PBEModel has more accuracy than current baseline estimate model, and the PBESVD++ have higher prediction performance than SVD++.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 61402097, No. 61572123 and No. 61502092; the National Science Foundation for Distinguished Young Scholars of China under Grant No. 61225012 and No. 71325002; the Natural Science Foundation of Liaoning Province of China under Grant No. 201602261; the Fundamental Research Funds for the Central Universities under Grant No. N151708005, and No. N151604001.

References

1. Goldberg, D., Nichols, D., Oki, B.M., et al.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12), 61–70 (1992)
2. Takács, G., Pilászy, I., Németh, B., et al.: Major components of the gravity recommendation system. ACM SIGKDD Explor. Newsl. **9**(2), 80–83 (2007)
3. Sarwar, B.M., Konstan, J.A., Borchers, A., et al.: Applying knowledge from KDD to recommender systems. Univ. Minnesota, Minneapolis **1**(612), 625–4002 (1999)
4. Deerwester, S., Dumais, S.T., Furnas, G.W., et al.: Indexing by latent semantic analysis. J. Am. Soc. Inform. Sci. **41**(6), 391 (1990)
5. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inform. Syst. (TOIS) **22**(1), 89–115 (2004)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
7. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
8. Koren, Y.: Collaborative filtering with temporal dynamics. Commun. ACM **53**(4), 89–97 (2010)
9. Kumar, R., Verma, B.K., Rastogi, S.S.: Social popularity based SVD++ recommender system. Int. J. Comput. Appl. **87**(14) (2014)
10. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI, pp. 123–129 (2015)
11. Wang, S., Ma, Y., Cheng, B., Yang, F.M.: Multi-dimensional QoS prediction for service recommendations. IEEE Trans. Serv. Comput., 12 (2016). doi:10.1109/TSC.2016.2584058
12. Ma, Y., Xin, X., Wang, S., Li, J., Sun, Q., Yang, F.: QoS evaluation for web service recommendation. China Commun. **12**(4), 151–160 (2015)
13. Wang, S., Zheng, Z., Wu, Z., Micheal, R.L., Yang, F.: Reputation measurement and malicious feedback rating prevention in web service recommendation system. IEEE Trans. Serv. Comput. **8**(5), 755–767 (2015)
14. Jannach, D., Zanker, M., Felfernig, A., et al.: Recommender Systems: an Introduction. Cambridge University Press (2010)
15. Guo, G.: LibRec: A Java Library for Rcommender Systems[EB/OL] (2016). http://www.librec.net